Future Generation Computer Systems 72 (2017) 219–226

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

TrueID: A practical solution to enhance Internet accountability by assigning packets with creditable user identity code

Guangwu Hu[a,b], Wenlong Chen[c,*], Qi Li[b], Yong Jiang[b], Ke Xu[d]

[a] School of Computer Science, Shenzhen Institute of Information Technology, Shenzhen 518172, China
[b] Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[c] Information Engineering College, Capital Normal University, Beijing 100048, China
[d] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 3 November 2015
Received in revised form
31 August 2016
Accepted 7 September 2016
Available online 15 September 2016

Keywords:
Cyber-security
Internet accountability
IP address validation
PKI
SAVI

ABSTRACT

Despite the Internet has been rapidly developed in the past three decades, its intrinsic security mechanism, e.g., IP source address validation and user identification authentication, is still not well addressed. This results in numerous cyber security threats. In order to enhance the Internet accountability and deter potential cyber-attacks, in this paper, we propose TrueID, an IPv6 header extension scheme which can embed hash-based, creditable and undeniable user identity code inside IPv6 packets. We present the system architecture, header's format and viable implementation approaches with different credibility granularities. Meanwhile, to verify packet credibility and integrity, we design an Autonomous System (AS) level public-key distribution system which can disseminate user's public keys between allied ASes safely. Also, the prototype experiment has proved that our scheme possesses these features with desirable performance.

1. Introduction

Internet security is facing severe situations in recent years. According to the public report [1], the number of major cyber-attacks has reached to near a thousand in 2014 compared to a few hundreds in 2011. Moreover, some cyber-attacks are involved with political factors, arousing disputations among nations. Take two recent incidents as examples. In February 2013, New York Times cited a report and claimed that Chinese army had involved with many hacking activities against the US government and companies in the last few years [2]. And right in the next month, Korea Communications Commission (KCC) declared that the cyber-attacks happened in Korean major television stations and financial institution networks were from domestic areas rather than China [3]. From the technical aspect, the key challenge behind these accidents is that the Internet misbehaviors are short of solid evidence to link to the perpetrators' identities, since the current Internet cannot provide validation mechanism on IP source address and user identification. At the initial period of the Internet, this is not an issue due to network simplicity and the trustworthy relationships between users. Now that Internet has become a critical infrastructure of modern society and covered more than 2.4 billion netizens, the Internet accountability needs to be enhanced urgently.

Some may argue that a packet's IP source address can reveal its sender identity. Indeed, an IP source address owns dual-semantics which can reflect the host location and its identity simultaneously. But it is not enough to represent the user's identity. The reasons are at least the following:

(1) Packet's source IP address can be alternated. According to the MIT spoofer project [4], with 17.9%, 24.6% and 38.4% address space, IP prefix and Autonomous System (AS) can be spoofed in IPv4 networks by December 14, 2013. In other words, nearly 1/5 global IPv4 addresses could be manipulated by malicious perpetrators. And no indication shows any improvement in the IPv6-based next generation Internet;
(2) Some may believe that Transmission Control Protocol (TCP) based packet flows can ensure IP address credibility for both ends because of the triple handshake mechanism. However, this assertion is insufficient due to the attacks such as TCP-sequence prediction [5] and TCP session hijack [6];

[*] Corresponding author. Fax: +86 010 6260 3064.
E-mail addresses: hu.guangwu@sz.tsinghua.edu.cn (G. Hu), cwl@csnet1.cs.tsinghua.edu.cn (W. Chen), qi.li@sz.tsinghua.edu.cn (Q. Li), jiangy@sz.tsinghua.edu.cn (Y. Jiang), xuke@tsinghua.edu.cn (K. Xu).

http://dx.doi.org/10.1016/j.future.2016.09.005
0167-739X/© 2016 Published by Elsevier B.V.

(3) There are still a large proportion of services and protocols based on User Datagram Protocol (UDP), Internet Protocol (IP) and other connectionless protocols, e.g., Domain Name System (DNS), Open Shortest Path First (OSPF), and Internet Control Message Protocol (ICMP);
(4) Common practices in many networks, such as Traffic Engineering (TE), Network Address Translation (NAT)-based middle boxes, are deteriorating this situation severely.
(5) Even if the source IP address of a malicious packet is not spoofed, we are still unable to identify or verify the offender's identity, because there is no direct connection between an IP address and the user's identity. And in the current Internet architecture, we do not have such cooperation mechanism between different ASes which can help victims to nail down sender identity merely based on IP address; Moreover, the mapping relationship between ASes and their IP prefixes are changing slightly and irregularly [7].

In order to make every packet carry its sender's reliable and undeniable identity so that users can verify the credibility for their correspondents, in this paper, we propose TrueID, an IPv6 header extension scheme which can embed hash-based, creditable and undeniable user identity code inside IPv6 packets. The final goal of our scheme is to provide packet integrity checking, sender authentication and replay-attack prevention services, which can help to enhance the Internet accountability and deter potential cyber-attacks. Compared to the existing work, the merits or the main differences in our scheme are as follows: (1) we utilize the Software Defined Networking (SDN) based architecture and user's access devices instead of modifying routers and client's host-stacks to realize our purpose, which would lower the front-end investment and maximize the existing network assets; (2) In order to promote our scheme into inter-domain area, we develop a distributed AS-level Public Key Infrastructure (PKI) cooperation system, which can disseminate user's public keys between allied ASes to verify user's received packets' credibility within each other.

Although TrueID is much more the same with the Authentication Header (AH) header [8] in the IPsec architecture [9], significant differences still exist. (1) First of all, TrueID dedicates to enhance the Internet accountability by introducing user identity code and supporting third-party audit/verification, while AH targets to offer end-to-end authentication; (2) Secondly, AH or IPsec takes the symmetric encryption algorithm and introduces Internet Security Association and Key Management Protocol (ISAKMP) to negotiate encryption algorithm, shared keys and related security parameters in communicating both ends, so as to trade-off system performance and security strength. Besides, it is also hard to defy the man-in-the-middle attack. On the contrary, TrueID adopts the asymmetric encryption algorithms (e.g., RSA and DSA) and the SDN architecture instead of end hosts to generate signature. Thus, it has much safer and higher performance than AH; (3) Moreover, no matter in transport mode or tunnel mode, AH still has problems in the NAT scenario because its verification range includes the IP source address field. Oppositely, the TrueID header is independent to outside IP header so that it can cope with NAT, IPv4/IPv6 transition included situations.

The rest of this paper is organized as follows: Section 2 summarizes the related work. Section 3 elaborates our scheme, including the system architecture, the format of this extension header and three viable approaches for the scheme implementation. Section 4 evaluates the scheme and Section 5 clarifies some issues that readers might concern. At last, Section 6 concludes the paper.

2. Related work

We have derived our work from many related work that focuses on the subject of IP source address validation and user identification authentication.

2.1. IP source address validation

Reliable IP source address is the necessary condition for user identification (UID) verification in intra-domain areas. To realize this goal, the Source Address Validation Architecture (SAVA) scheme [10] provides a transparent network service to ensure every packet holds an authenticated source IP address. It consists of three levels, the Inter AS, the Intra AS and the first hop. During each level of this hierarchical architecture, it obtains different granularities of authenticity. In the first hop, the Source Address Validation Improvement (SAVI) proposal [11] was approved by Internet Engineering Task Force (IETF). Following the SAVI specification, a layer 2.5 switch (i.e., the SAVI switch) is set up in the user access subnet, which can filter spoofing packets by establishing the triangle relationship of the IP address, the MAC address and the uplink port for each access host. As to the binding relationship establishment and packet's anti-spoofing fulfillment, it is accomplished by the IP address assignment protocols (e.g., DHCPv6) sniffing and the Control Packet Snooping (CPS) protocol. Compared with the well-known solution of unicast Reverse Path Forwarding (uRPF) [12], SAVI is more accurate because its effect point is user's access switch rather than the access router; besides, SAVI does not need to consider the asymmetric routing issue which bothers uRPF a lot. Till now, various SAVI switches have already been implemented by lots of network equipment providers, for instance, Huawei and ZTE. The other line of IP sources address validation schemes like the Source Address Validity Enforcement (SAVE) protocol [13] and the Inter-Domain Packet Filters (IDPF) [14] are dedicated to filter spoofing packets according to the forwarding table or the Border Gateway Protocol (BGP) update messages. Additionally, some proposals establish packet filtering mechanisms based on the bloom filtering [15], the hop-count [16] and even the history IP filtering record [17], which are all confirmed to be inaccurate because there are possibilities of false positives or false negatives. Lastly, there are some other schemes from the angles of protocol/host-stack re-design to solve this problem. For instance, SPM [18], StackPi [19] and Base [20] achieve this purpose by utilizing 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). Nevertheless, these solutions cannot support users' identities identification or authentication for communication both-ends or third parties, even though they achieve anti IP source address spoofing effect with different granularities and different costs.

2.2. Host/user identity authentication

Due to the aforementioned reasons, it is hard to verify correspondents' identities solely relying on the IP source address. Thus, to allow packets to bring with their senders' identities, researchers have already made great efforts. In the early 1997, Global, Site, and End-system address elements (GSE) as the candidate IPv6 address architecture proposal [21] designed the IPv6 address in the form of $6 + 2 + 8$. That is, using the first 6 bytes "Routing Goop" to represent the routing prefix, the following 2 bytes to represent the subnet indicator and the last 8 bytes to indicate the identity or the interface of the end system. Even in our current IPv6 standard, under the Stateless Address Auto Configuration (SLAAC) address assignment mode, a host needs to transform its 48-bit MAC address into an IEEE EUI-64-bit address before combining with its access router announced 64-bit routing prefix to serve as its IP address [22]. In other words, the last 64 bits of this address are actually the host identity. But unfortunately, neither of them can guarantee the credibility of the IP source addresses because of the IP spoofing issue. Therefore, the host identity is unreliable.

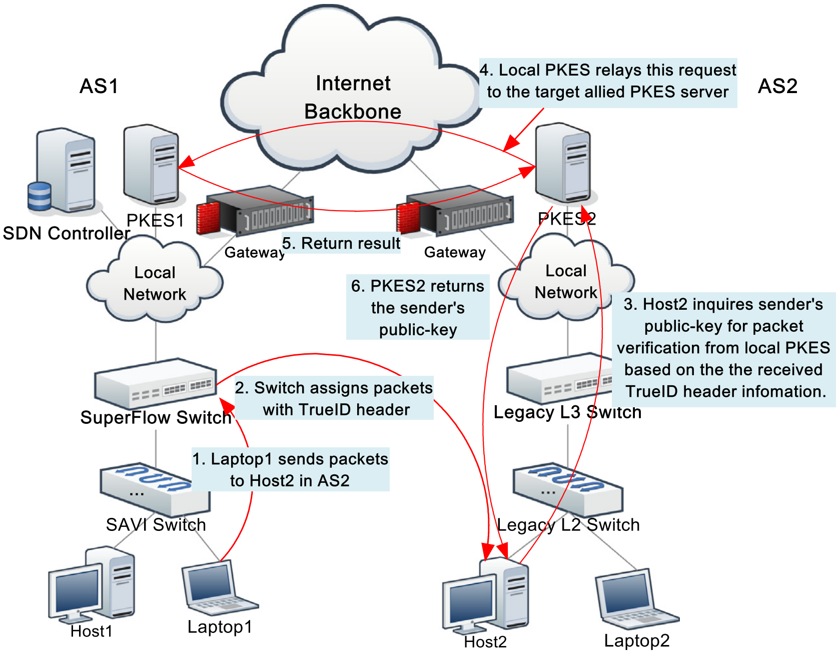

AS1
Internet Backbone
4. Local PKES relays this request to the target allied PKES server
AS2
SDN Controller
PKES1
PKES2
Gateway
5. Return result
Gateway
Local Network
6. PKES2 returns the sender's public-key
Local Network
3. Host2 inquires sender's public-key for packet verification from local PKES based on the the received TrueID header infomation.
SuperFlow Switch
2. Switch assigns packets with TrueID header
Legacy L3 Switch
1. Laptop1 sends packets to Host2 in AS2
SAVI Switch
Legacy L2 Switch
Host1
Laptop1
Host2
Laptop2

Fig. 1. The system architecture and packet verification procedures. We utilize the SAVI switch to keep the IP source address authenticity in the user access subnet and take the L3 SuperFlow switch or gateway device to accomplish the goal of embedding users' credible identities code into their packets. The PKES servers are responsible for providing public-key inquiry services.

In order to verify our correspondents' identifications, many studies showed their merits mainly from angles such as self-certifying [23–25], IP address semantic separation [26,27] and other creative and revolutionary designs [28,29]. For example, Cryptographically Generated Addresses (CGA) [23] and Accountable Internet Protocol (AIP) [24] encrypt IP source address with asymmetric key cryptography so that shared keys between both ends can authenticate each other. But such designs need extra secure key agreement protocols since key generation and public-key distribution are accomplished by individual hosts without Certificate Authority (CA) support, which is non-suitable for large-scale networks. To address this drawback, TrueIP [25] takes IP source address as the public key and utilizes the Identity Based Cryptography (IBC) to produce the private key, thus correspondents can verify the authenticity of each other directly without public-key acquirements. However, it is uneasy to revoke IBC keys since all keys need to be regenerated if one private key is compromised. Moreover, the Host Identity Protocol (HIP) [26] sets up a new layer named Host Identity (HI) in the middle of IP and transportation layers. It achieves 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-similar system to resolve the mapping relationship between HI and IP. Moreover, although the Locator/ID Separation Protocol (LISP) included schemes [27] try to separate routing-prefix properties from the IP dual-semantics so as to relieve the routing explosion dilemma, they solved our case indirectly in some degree but shared the same drawbacks with HIP. In addition, David et al. propose an Accountable and Private Internet Protocol (APIP) [28], which tries to balance the accountability and privacy by splitting the source address into an accountability address and a return address. However, this IP source address replaced by accountability address design is quite similar with the NAT mechanism, which might cause single point failures. Lastly, the clean-slated idea of Role-Based Architecture (RBA) [29] tears the current stack-based TCP/IP architecture apart and creates a novel role-based flat model, which refreshes our minds but is hard to realize.

These studies give us a lot of inspirations, but none of them can satisfy our purpose directly. We will elaborate our solution in the next section.

3. The TrueID scheme

In this section, we depict the TrueID architecture and its components in detail. Our threat model is that any node in the networks can generate packets of any format and alternate any fields of them. But the network devices, such as router, switch and gateway, cannot be tampered with.

3.1. Architecture

The above facts make us believe that the authenticated IP source address is the prerequisite for intra-domain packet verification only, while packets bringing sender's creditable identity is a necessary condition for the same purpose in the inter-domain premises. Thus, we made a slight change on the legacy network architecture. We attempt to keep IP source address authenticity within domain through the SAVI solution. Furthermore, we try to utilize the layer 3 (L3) switch or gateway device to embed user's credible identities into their packets.

As illustrated in Fig. 1, in the domain of AS1, we deploy the SAVI switch in the user access subnet, which can filter spoofing packets and offer the credibility granularity of a single host. Then, SuperFlow switch [30] is deployed on the top of the L3 SAVI switch. This switch can not only support the SDN/Openflow [31] architecture to control flows with different granularities, but also bind user's private keys with other user related information together upon user's self-authentication. This design will enhance packet's reliability through embedded user digital signatures. Lastly, an SDN controller and a Public Key Exchange Server (PKES) are set up in this network as well. The former is to distribute flow control rule, while the latter stores local user's keys and responses public-key inquiry from local or allied domains.

Fig. 1 also shows basic packet verification procedures. On the outbound route, the SuperFlow switch follows the instructions

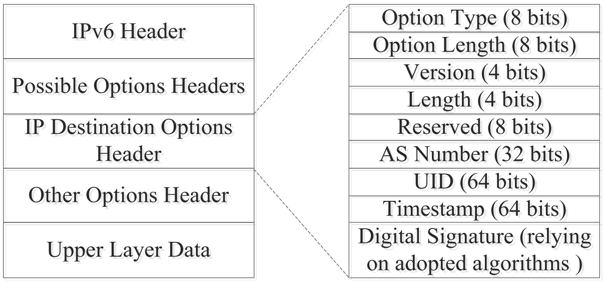


Fig. 2. The format of TrueID option header. The UID and AS Number indicate user's identity code and belonging AS separately. Timestamp is for preventing replay-attacks, and digital signature is designed for verification packet's credibility and integrity.

from centralized controller(s) to control user's flows and assign legitimate packets with our designed IPv6 optional header (we named it as TrueID header). In the phase of verification, the receiver inquires the sender's public key from its local PKES based on the TrueID header indicating sender's identity code. After that, local PKES directly return results (if the sender is local) or relay this request to the target PKES (if the sender comes from allied ASes). Eventually, the receiver retrieves the sender's public key, determine the sender authenticity and take further actions (e.g., report to local authority to yield flows that come from the unverified sender).

3.2. The format of TrueID header

We leverage on the IPv6 Destination Options Header (DOH) rather than the Authentication Header (AH) to carry sender-related data. The reason lies in the facts that DOH is light-weighted and intermediate nodes do not need to involve any computation or modification cost. Fig. 2 shows the TrueID header format, whose field design is derived from the following considerations:

(1) We assign the value "0001111" 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 consecutive 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" indicates the length of this option header can contain 64 bytes data at most;

(3) "Version" depicts the header version and "Reserved" is for reservation;

(4) Noticing that the Internet is governed and composed by ASes rather than nations, we put the sender's AS Number (ASN) into this header. Besides, tracking packet's original AS merely based on its IP source address is an inaccurate time-consuming job due to reasons such as IP address spoofing, multi-homing and inquiring cost;

(5) As for the UID field, we utilize 8 bytes to represent sender's digital identity code. In fact, 64-bit UID possesses enough space to contain all users and devices within a domain. In terms of UID code, we take the following function to generate

$$UID = H_{outside}(H_{inside}(user\ original\ code), user\text{-}password).$$

Here, H_{inside} and $H_{outside}$ indicate the normal hash (i.e., MD5) and the hash-encryption-combined (i.e., HMAC-SHA1) algorithms, respectively. Password is the key parameter of the HMAC function [32]. This double hash and encryption algorithm ensures adversaries cannot infer to user's original code. Even though the produced result is larger than the field's 64-bit space (i.e., HMAC-MD5:128 bits, HMAC-SHA1:160 bits), we can take the first 64 bits of this hashed value as the code. Moreover, we can rehash this result if duplicated UID code appears, even though this probability is quite low.

Algorithm 1 shows an example by using the HMAC-MD5 method with C# code, which helps to understand this UID generation process. Suppose the user original identity code is "018900000000" and his password is "20093", the produced result will be a hash-based 128-bit long hexadecimal string "387892295045049a3f63beccc40771f1";

(6) In order to prevent the replay attacks, i.e., perpetrators could perform the SYN-flood attack by coining large amount of TCP-SYN packets with different sender identities and releasing to victims in a short time, we add the timestamp field into this header so that the receiver can directly discard those packets with outdated timestamp;

Algorithm 1: User Identification Code Generation

```
Input: data is the user original identity code, key is the password
Output: Md5-based hash user identification code
String GenerateUID(String data, String key,)
{
  HMAC myhmac = HMAC.Create();
  myhmac.Key = Encoding.Default.GetBytes(key);
  byte[] temp = myhmac.ComputeHash(Encoding.Default.GetBytes(data));
  StringBuilder sb = new StringBuilder();

  for (int i = 0; i < temp.Length; i++)
    sb.Append(temp[i].ToString("x2"));
  return sb.ToString();
}
```

(7) Lastly, to ensure that the UID code is undeniable, we enforce this header to bring sender' digital signature, which comes from the algorithm:

$$Signature(P) = H(K_s \| P_{header} \| H(P_{body})).$$

Here K_s stands for sender's private key. H is a digital signature function, e.g., RSA or DSA, P_{header} and P_{body} refer to the packet's header and body part, respectively. In Algorithm 2, we combine the DSA signature algorithm with the SHA-1 hash algorithm that could produce 160-bit long signature string. Certainly, adopting other different hash algorithms will generate different lengths of signatures which would affect the packet payload ratio. We will discuss this problem in Section 5.

Algorithm 2: Digital Signature Generation

```
Input: data is the binary content for generating signature, private-key is the private key of user
Output: string of digital signature
String GenSignature(byte[] data, string private-key)
{
  DSACryptoServiceProvider DSA = new DSACryptoServiceProvider();
  DSA.FromXmlString(private-key); //import the private key
  DSASignatureFormatter DSAFormatter = new DSASignatureFormatter(DSA);
  DSAFormatter.SetHashAlgorithm("SHA1"); //using SHA1 as hash algorithm
  return DSAFormatter.CreateSignature(data);
}
```

Correspondingly, we also present the digital signature verification process for the receiving end to verify packet authenticity, as illustrated in Algorithm 3.

Algorithm 3: Digital Signature Verification

```
Input: data is the binary content for verification, public-key is the public key of send user, and the SignedHashValue is the original digital signature.
Output: boolean value, true or false
Boolean GenerateKey(byte[] data, string public-key, string SignedHashValue)
{
  DSACryptoServiceProvider DSA = new DSACryptoServiceProvider();
  DSA.FromXmlString(public-key); //import the public key
  DSASignatureFormatter DSAFormatter = new DSASignatureFormatter(DSA);
  DSAFormatter.SetHashAlgorithm("SHA1"); //using SHA1 as hash algorithm
  return DSADeformatter.VerifySignature(data, SignedHashValue);;
}
```

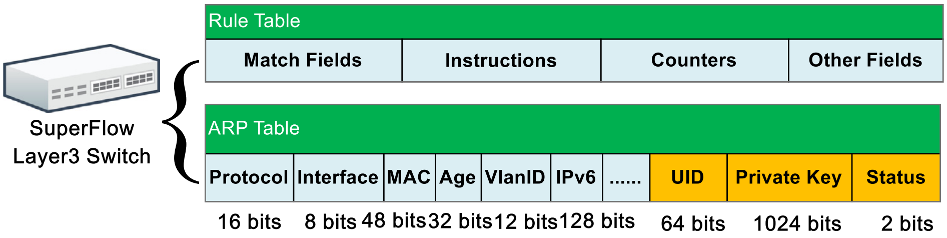


Fig. 3. The ARP table redesign in the SuperFlow switch. The fields UID, private key and status are new fields, which represent user's identity code, private key and the current status (e.g., authentication done, host alive, host timeout).

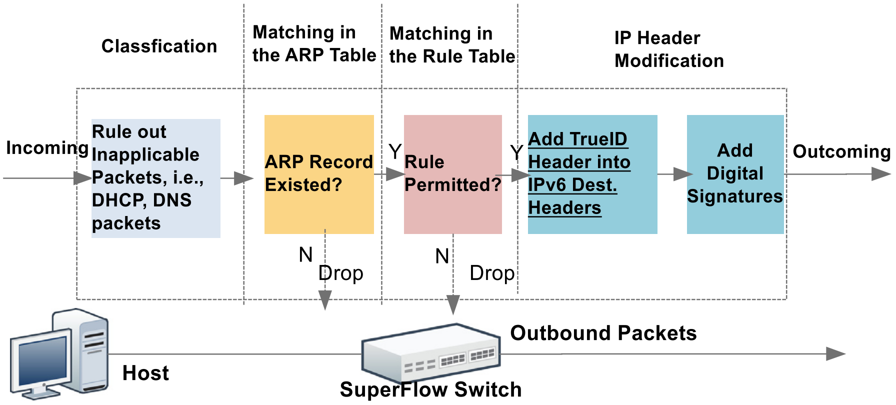


Fig. 4. The procedures in the SuperFlow switch to embed outbound packets with TrueID option header.

3.3. *Implementation*

We give three viable TrueID header's implementation methods varying from user access switch to domain gateway. Since these approaches possess different trade-offs between UID-code's credibility and deployment cost, network authorities can make their choices based on their own requirements.

3.3.1. *L2 SAVI switch*

Briefly, in the SAVI switch, there are two control data-tables, i.e., the Binding Status Table (BST) and the Filtering Table (FT). The former stores the binding relationship and the latter forms the banned list of spoofing hosts. Relying on them, the SAVI switch can authenticate IP source address for its access hosts and prevent hosts attached to the same link from spoofing each other. In order to let the SAVI switch fill TrueID header into targeted packets, we expand the BST table so as to make it contain all of its access hosts' UID codes and private keys. The next detail is similar with the L3 SuperFlow switch, which will be introduced in the next subsection.

3.3.2. *L3 SuperFlow switch*

Considering SAVI is an L2 switch, modifying IP headers has already violated layering principles in some degree. Consequently, we shift our focus onto user's L3 access switch. Coincidentally, OpenFlow serves as the de-facto model of the SDN architecture; it can achieve fine-grained flow control, such as packet forwarding, dropping, and packet header modification. Thus, we consider utilizing SDN-supported switches to realize our purposes, which was named as the SuperFlow switch [30]. We mainly insert UID, private key and status three new fields into its Address Resolution Protocol (ARP) table so as to make it bind user's specific information with its host's properties, which is shown in Fig. 3. This design cannot only take advantage of SDN's centralized control pattern, but also integrate procedures such as user authentication and private-key distribution. The UID and private key will be distributed to the SuperFlow switch when the user was identified.

In the meantime, the process for handling packets in the switch can be divided into four procedures, as shown in Fig. 4. Firstly, ruling out inapplicable packets, e.g., DNS, Dynamic Host Configuration Protocol (DHCP) packets. Secondly, judging sender's identity and status in case of offline host or unidentified users, which is called ARP-matching. Thirdly, the rule-matching phase is the same as the OpenFlow specification, which can forbid unpermitted packets to be forwarded. Lastly, the modification process will eventually fulfill our design.

3.3.3. *Gateway*

Although achieving the same goal in the domain gateway has the advantage of private key distribution exemption and centralized computing, the performance would be the major concern. Meanwhile, the range of IP source address spoofing will expand to the whole domain, and the credibility granularity in the UID code will be coarse-grained rather than single user granularity. Nevertheless, deploying anti-spoofing measures and enhancing gateway performance can mitigate these problems.

3.4. *Packet verification*

In order to verify packet's integrity and credibility, receivers need to recalculate the hash values with sender's public keys. Consequently, it raises another problem which is how to safely and reliably disseminate user's public keys. Probably, the simplest way is to set up a global unified PKI system which stores global netizens' public keys together. Then, the verifier can directly retrieve the public key from it. Obviously, this idea is unrealistic due to the management and control matters. The alternative way is to adopt the distributed manner, i.e., each AS establishes their PKI system and provides a public-key inquiry service. But the problem in such design is that the system work will need a DNS-like system to index these servers so as to provide Internet-wide services. Consequently, we consider establishing an AS-level PKI cooperate architecture, which can provide public-key inquiry services between allied CA servers. We name this decentralized

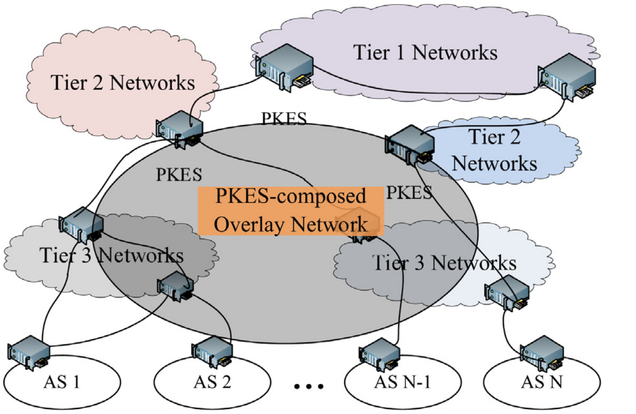


Fig. 5. The distributed PKI cooperate architecture between allied ASes. Multiple PKES allies assembles a overlay network.

CA server as Public Key Exchange Server (PKES), which is shown in Fig. 5. Meanwhile, to avoid PKES spoofing, correspondences in this PKES-composed overlay network need to be signed by their individual private keys.

When a host tries to retrieve a specific UID's public key, it can resort to its local PKES by sending the request with the packet sample. Then PKES will act as a proxy and relay this request to the ASN number indicated PKES if these two PKESs are allied. Finally, the local PKES will return and cache (within expiration time) the key for further requests. Theoretically, the average storage and computation cost in this PKES is only $1/n$ (suppose the number of PKES is n) of the traditional centralized scheme with one CA server.

Furthermore, we estimate the convergence time of this PKES-composed overlay network. Considering this network can be expressed by an undirected graph $G(V, E)$, in which V is the collection of PKES nodes and E is the virtual link between these PKES nodes. Thus, the convergence time T_i for node i can be denoted by Formula 1. It should be noted that the notations N and $|E|$ denote the node number and link number (or degree) of the whole network, respectively, while e_i indicates the link degree of node i. Consequently, the average convergence delay of the whole network can be presented by the average value of T_i. As to the PKES node deployment benefit, we will analyze it in Section 4.2.

$$T_i = \left(\frac{N * (N-1)}{2} - |E|\right) / e_i \quad \text{where } N = |V| \tag{1}$$

$$T_{avg} = 1/N * \sum_{i=1}^{N} T_i. \tag{2}$$

4. Evaluation

Our test-bed is similar to the topology shown in Fig. 1, except that we removed the controller and the SAVI switch. We evaluate our scheme in terms of desired performance properties. Specifically, we examine (1) SuperFlow switch's forwarding throughput and goodput (exclude IP and TrueID headers and only take upper layer payload into account); (2) The public-key retrieval delay in PKES and end-to-end transmission performance when the network applies our scheme; (3) The benefit ratio under the various proportions of AS allies' deployment.

4.1. Basic performance

In this testbed, we first take two 1 Gbps commodity routers to replace gateway devices. Meanwhile, we utilize a commodity machine (Core i5-4590 3.3 GHz CPU, 8G DDR3 RAM, two 1 Gbps network cards, Ubuntu Linux kernel 3.2.0–3) to implement the OpenFlow switch, and two machines in the same model act as the PKES server.

First, we measure the SuperFlow switch's throughput and goodput with different size of packets and different traffics, the result is shown in Fig. 6(a). Compared with the baseline that we measured from a normal hardware 1 Gbps L3 switch with a ping method, we observe that the SuperFlow switch's maximum forwarding throughput is about 700 Mbps, and the goodput goes up sharply as the packet size or payload ratio increases. Meanwhile, from Fig. 6(b), we can also be aware that neither TCP nor UDP flows has obvious impacts on the switch throughput. As a whole, we admit that the switch performance is limited to some degree due to software implementation. In the future, we plan to use the NetFPGA [33] card to implement SuperFlow switch to improve its performance.

Furthermore, we test the public-key retrieval delay with different number of test packets. The result in Fig. 7a shows that the delay in the public-key inquiry procedure increases relatively

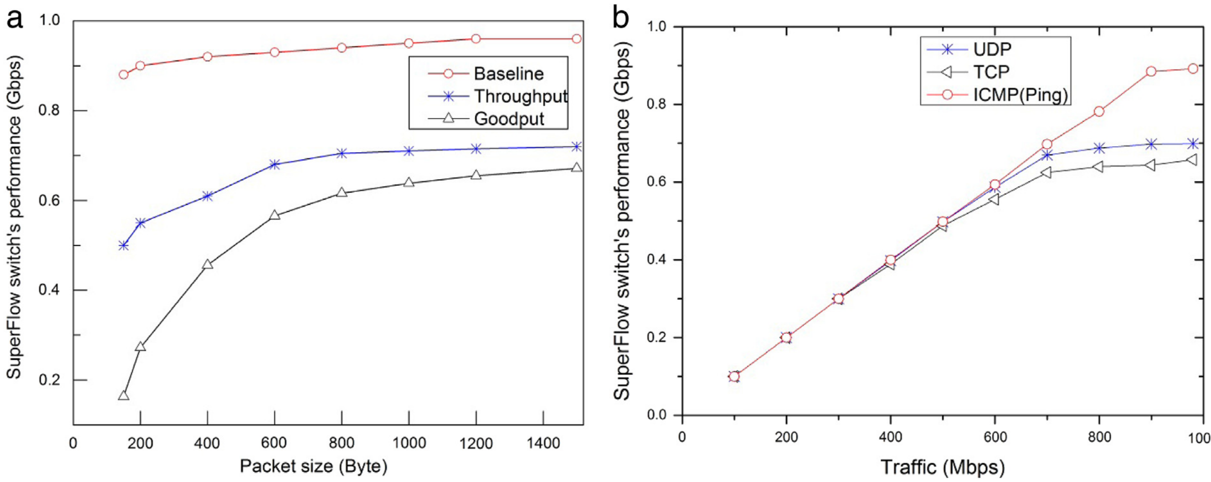


Fig. 6. The SuperFlow switch's performance with (a) different sizes of packets and (b) different traffics. The baseline data comes from a normal 1 Gbps commodity switch with ping method, while the throughput and goodput state the packet and the packet's payload forwarding capacity possessed by the SuperFlow switch.

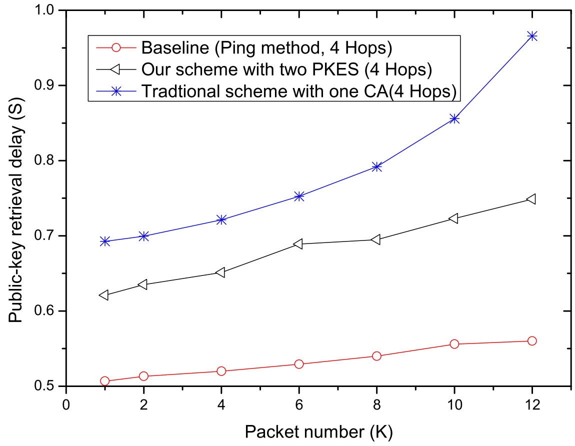


Fig. 7a. The public-key retrieval delay in PKES.

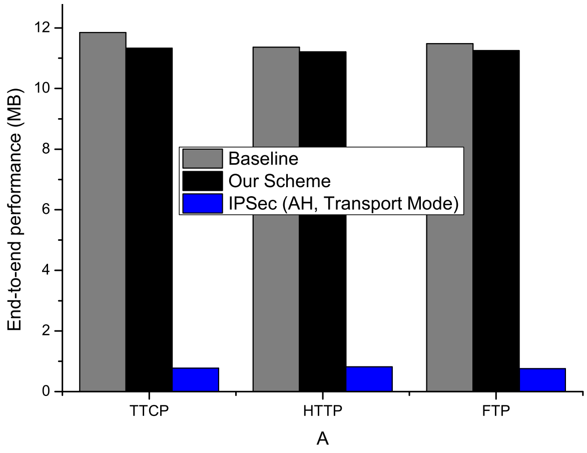


Fig. 7b. End-to-end transmission performance.

quickly than the baseline. But it is still faster than the traditional PKI scheme with one CA server. Lastly, in order to evaluate the end-to-end performance, we use two computers of the same mode but only equipped with one 100 Mbps network adaptor to act as two client hosts in two different ASes. Fig. 7b clearly indicates our scheme significantly outperforms the IPsec scheme and it is close to the baseline without applying any scheme.

4.2. PKES node deployment benefit analysis

Moreover, we estimate the deployment benefit with different PKES allies' deployment ratios based on the IPv6 topological data from CAIDA [34]. Given that our model is a unidirectional graph, the PKES overlay network can be denoted as $G = (V, E)$, where V is the set of PKES nodes, and E is the collection of links between these allied PKESs. Therefore, the benefit ratio of the deployed links can be expressed as follows:

$$\text{Benefit Ratio} = \frac{2\sum_{1}^{|E|} e_{xy}}{N * (N-1)} \quad \text{where } N = |V| \tag{3}$$

$$e_{xy} = 1 \quad \text{if } e_{xy} \in E \text{ or } e_{xt} \in E \text{ and } e_{ty} \in E$$

$$\text{Otherwise,} \quad e_{xy} = 0.$$

Based on the above formula, we divided the data into three groups with different sorting methods: the original, ascending and descending orders by the number of AS owned links. From Fig. 8, we can find out the method of deploying the ASes with larger

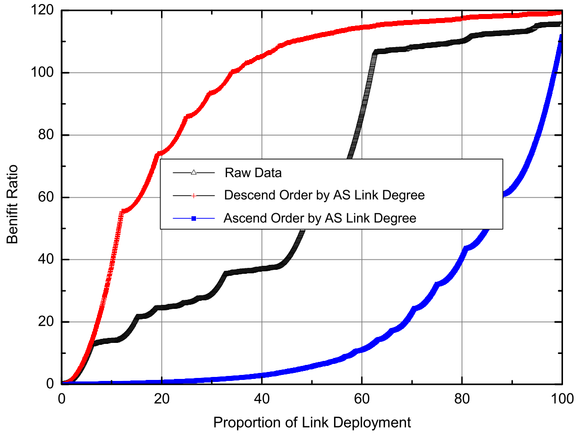


Fig. 8. The benefit ratios with three deployment methods.

degree/number of links first can achieve a good effect, i.e., merely 37% of link deployment under this method can trade for nearly 100% benefit ratio. In other words, deploying the links between major ASes, such as large Points Of Presences (POPs), which can get the effect of excellent public-key distribution.

5. Discussion

In this section, we discuss some publicly concerned issues from our point of view.

1. Privacy issue

In terms of privacy issue, netizens do not need to worry about it since the UID tag in TrueID header is just a hash-based code rather than user's real identity. The original UID code can be decided by each individual AS authority, and it will be formatted with the means of double hash and encryption. From the technical aspect, it was proved that this design is pretty secure which makes perpetrators unable to deduce to the original UID code merely based on the formatted one. On the contrary, the general public can benefit from this proposal since they can verify the suspicious packets and report their organizations to filter these unverifiable packets.

2. System security issue

By far, our scheme has kept the promise of protocol security, which can defy attacks of packet spoofing, identity illegal modification and packet replay attack. But the rest components of the whole system, such as SDN controller, SuperFlow switch, PKES server and etc., are important as well since they play key roles in the key generation/distribution, signature generation/verification, vital data storage and other procedures. Thereby, they may suffer from different attacks which impact their stability and feasibility. However, we can leverage extra counter measures (i.e., load-balancing, access control, anomaly detection) to mitigate these threats, but which is out of the scope of this paper.

3. PKI vs. IBS

Indeed, Identity-based Signature (IBS) is another excellent asymmetric key signature scheme. Compare to the PKI scheme, it takes arbitrary data, e.g., IP address, as its public key which effectively simplifies the processes such as public key distribution and escrow. However, since IBS adopts IBC as its encryption algorithms, the inevitable key revocation problem, as we mentioned in Section 2.2, becomes its biggest concern on the key management issue. Nevertheless, IBS can still be applied to our scheme.

4. Signature length and payload ratio issue

Technically, the length of digital signature equals to the key size of signatures adopted by algorithms, i.e., if we take an RSA 1024-bit private key to sign a packet, the produced signature will be 1024-bit long. This result might be a concern since it will affect the payload ratio. Fortunately, signature algorithms can apply to many message-digest algorithms, which means that any length of messages can be produced to a fixed length of result depending on the property of the adopted hash algorithms, e.g., RSA-MD5:128 bits, RSA-SHA-1:160 bits, RSA-SHA-256: 256 bits, and RSA-SHA-512: 512 bits. Thus, the proportion of the whole TrueID header in MTU (1500 bytes) would be 2.67%, 2.93%, 3.73%, and 5.87%, respectively. Thus we believe this payload ratio is affordable.

5. The MTU issue

Since TrueID needs to add extra header into packets, it will enlarge their size slightly. However, Maximum Transmission Unit (MTU) does not allow oversized packets to be transmitted. Our suggestion is to decrease the MTU value in the MTU announcement if necessary.

6. Conclusion

"On the Internet, nobody knows you are a dog", the classic proverb indirectly explains the main reason that cyber-attacks happened frequently, i.e., the current Internet is lack of accountability and perpetrators are irresponsible for their actions, since no evidence can link them to their misbehaviors. In order to enhance the Internet accountability, we present the scheme of TrueID, which can label Internet actions with solid proofs through embedding packets with sender's creditable and undeniable identity code. We depict the format of the TrueID option header and state the reason why we design header fields as such. We also provide feasible implementation approaches with different trade-offs between deployment costs and credibility granularity. Most importantly, in order to deploy our scheme in a wider area, we propose an AS-level PKI cooperate architecture which can disseminate public keys between allied ASes. Given the critical Internet insecurity situation nowadays, we hope that our scheme can shed light on enhancing Internet accountability and deterring possible cyber-attacks.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (No. 61402255, No. 61170292, No. 61373161), the National High Technology Research and Development Program of China (863 Program) (No. SS2015AA010203), Guangdong Natural Science Foundation (No. 2015A030310492), Fundamental Research Project of Shenzhen Municipality (No. JCYJ20160301152145171) and Shenzhen Key Laboratory of Software Defined Networking (No. ZDSYS20140509172959989).

References

[1] Hackmageddon, Cyber-attacks-timeline-master-indexes. http://hackmageddon.com/cyber-attacks-timeline-master-indexes.

[2] CNET. Chinese Army linked to hacks of U.S. companies, agencies. http://news.cnet.com/8301-1009_3-57569983-83/chinese-army-linked-to-hacks-of-u.s-companies-agencies.

[3] N.K. News, South Korea was Source of Wednesday's Cyber Attack. http://www.nknews.org/2013/03/south-korea-was-source-of-wednesdays-cyber-attack/.

[4] MIT. MIT spoofer project. http://spoofer.cmand.org/summary.php.

[5] Wikipedia. TCP sequence prediction attack. http://en.wikipedia.org/wiki/TCP_sequence_prediction_attack.

[6] Wikipedia. Session hijacking. http://en.wikipedia.org/wiki/Session_hijacking.

[7] CAIDA. Routeviews prefix to as mapping dataset. http://www.caida.org/data/routing/routeviews-prefix2as.xml.

[8] S. Kent, R. Atkinson, IP Authentication Header. RFC 2402, November 1998.

[9] S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, IETF RFC 2401,1998.

[10] Jianping Wu, Gang Ren, Xin Li, Source address validation: Architecture and protocol design, in: Proceedings of IEEE ICNP, 2007.

[11] J. Wu, et al. Source Address Validation Improvement Framework. RFC 7039, October 2013.

[12] Cisco. URPF. http://www.cisco.com/web/about/security/intelligence/unicast-rpf.html.

[13] J. Li, J. Mirkovic, M.Q. Wang, P. Reiher, L.X. Zhang, SAVE: Source address validity enforcement protocol, in: Proceedings of IEEE INFOCOM, 2002.

[14] Z.H. Duan, X. Yuan, J. Chandrashekar, Constructing inter-domain packet filters to control IP spoofing based on BGP updates. in: Proceedings of the IEEE INFOCOM, 2006.

[15] Chen Wei, Dit-Yan Yeung, Defending against TCP SYN flooding attacks under different types of IP spoofing, in: Proceedings of International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006.

[16] C. Jin, H.N. Wang, K.G. Shin, Hop-Count filtering: An effective defense against spoofed DDoS traffic, in: Proceedings of the 10th ACM Conf. on Computer and Communications Security. Washington, 2003.

[17] T. Peng, C. Leckie, K. Ramamohanarao, Protection from distributed denial of service attacks using history-based IP filtering, in: Proceedings of the IEEE Int'l Conf. on Communications. Anchorage, 2003.

[18] Y. Afek, A. Bremler-Barr, S. Schwarz, Improved BGP convergence via ghost flushing, IEEE J. Sel. Areas Commun. 22 (10) (2004) 1933–1948.

[19] A. Perrig, D. Song, A. Yaar, StackPi: A New Defense Mechanism against IP Spoofing and DDoS Attacks, Technical Report, CMU-CS-02-208, Carnegie Mellon University, 2003.

[20] H. Lee, A. Perrig, D. Smith, BASE: An incrementally deployable mechanism for viable IP spoofing prevention, in: Proceedings of ACM Symposium on Information, Computer and Communications Security, 2007.

[21] IETF. GSE - An Alternate Addressing Architecture for IPv6. http://tools.ietf.org/html/draft-ietf-ipngwg-gseaddr-00.

[22] R. Hinden, et al. IP version 6 addressing architecture. RFC 4291,2006.

[23] T. Aura, Cryptographically generated addresses, CGA, RFC 3972, 2005.

[24] D. Andersen, H. Balakrishnan, N. Feamster, et al. Accountable Internet Protocol, AIP, in: Proceedings of ACM SIGCOMM, 2008.

[25] C. Schridde, M. Smith, B. Freisleben, TrueIP: prevention of IP spoofing attacks using identity-based cryptography, in: Proceedings of ACM SIN, 2009.

[26] R. Moskowitz, P. Nikander, Host identity protocol, HIP architecture. RFC 4423, 2006.

[27] D. Farinacci, V. Fuller, D. Meyer, et al. Locator/ID separation protocol, LISP. RFC 6830, January 2013.

[28] David Naylor, et al. Balancing accountability and privacy in the Network, in: Proceedings of ACM SIGCOMM, 2014.

[29] Robert Braden, Ted Faber, Mark Handley, From protocol stack to protocol heap: role-based architecture, ACM SIGCOMM Comput. Commun. Rev. 33 (1) (2003) 17–22.

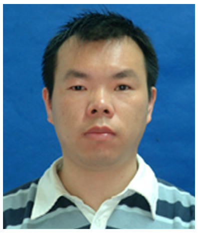
[30] Hu Guangwu, Ke Xu, Jianping Wu, SuperFlow: A controllable, manageable and scalable architecture for large-scale enterprise networks, in: Proceedings of IEEE HPCC, 2013.

[31] OpenFlow. http://www.openflow.org.

[32] H. Krawczyk, M. Bellare, R. Canetti, HMAC: Keyed-Hashing for Message Authentication. RFC 2104, 1997.

[33] NetFPGA. http://netfpga.org.

[34] CAIDA. The CAIDA UCSD IPv6 Topology Dataset. http://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml.



Guangwu Hu was born in 1980, he received the Ph.D. degree in computer science and technology from Tsinghua University. Now, he is a lecture of Shenzhen Institute of Information Technology. His research interests include software defined networking, Next-Generate Internet and Internet security.

Wenlong Chen (corresponding author) was born in 1976, he received the Ph.D. degree from University of Sci. and Tech. Beijing. He is now an associate professor in the College of Information Engineering of Capital Normal University. His research interests include network protocol, network architecture, high performance router, IPv4/IPv6 transition.

Qi Li received the B.Sc. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 2003 and 2012, respectively, where he is currently an associate professor with the Graduate School at Shenzhen. He was a researcher with the Swiss Federal Institute of Technology, Zurich, Switzerland, and a post-doctoral fellow with the Institute for Cyber Security, University of Texas at San Antonio, San Antonio, TX, USA. His research interests include system and network security, Internet, and largescale distributed systems.

Yong Jiang was born in 1975, he received the Ph.D. degree in computer science and technology from Tsinghua University. He is now the professor and Ph.D. supervisor of Graduate School at Shenzhen, Tsinghua University. His research interests include Next-Generate Internet and mobile Internet.

Ke Xu was born in 1976, he received the Ph.D. degree in computer science and technology from Tsinghua University. He is now the professor and Ph.D. supervisor of Tsinghua University. His research interests include computer network architecture, protocol engineering and Next-Generation Internet.