Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Anonymous-address-resolution model*

Guang-jia SONG†, Zhen-zhou JI
(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*)
†E-mail: tysong@aliyun.com

Received Nov. 3, 2015; Revision accepted May 11, 2016; Crosschecked Sept. 8, 2016

Abstract: Address-resolution protocol (ARP) is an important protocol of data link layers that aims to obtain the corresponding relationship between Internet Protocol (IP) and Media Access Control (MAC) addresses. Traditional ARPs (address-resolution and neighbor-discovery protocols) do not consider the existence of malicious nodes, which reveals destination addresses in the resolution process. Thus, these traditional protocols allow malicious nodes to easily carry out attacks, such as man-in-the-middle attack and denial-of-service attack. To overcome these weaknesses, we propose an anonymous-address-resolution (AS-AR) protocol. AS-AR does not publicize the destination address in the address-resolution process and hides the IP and MAC addresses of the source node. The malicious node cannot obtain the addresses of the destination and the node which initiates the address resolution; thus, it cannot attack. Analyses and experiments show that AS-AR has a higher security level than existing security methods, such as secure-neighbor discovery.

Key words: Network security, Address resolution, Neighbor discovery, Anonymous
http://dx.doi.org/10.1631/FITEE.1500382 **CLC number:** TP393.1

1 Introduction

The address-resolution process in Internet Protocol Version 4 (IPv4) is called neighbor discovery in IPv6—In this study, we believe that address resolution and neighbor discovery are the same. This process is important for network communication because message transmission cannot be conducted using only the IP address of the target; the Media Access Control (MAC) address should also be known (Fall and Stevens, 2011). Address-resolution protocols (ARPs) function mainly in obtaining the MAC by IP address. The consequences caused by an attacker on the address-resolution process are serious. For example, the man-in-the-middle attack intercepts and tampers with data, which may lead to network communication interruption and other serious consequences. Thus, address-resolution attack is a major threat to Local Area Network (LAN) security (Gouda and Huang, 2003; Rehman and Manickam, 2015).

Research on the security of ARPs has focused mainly on three aspects. The first aspect refers to the defense and detection technologies, i.e., the use of technical means to defend or detect attacks. This feature monitors and records all <IP, MAC> mappings of the hosts in the network for a long period. When the <IP, MAC> mapping in the address-resolution message that the host sends out is not consistent with the record, a spoofing attack exists (Nam *et al.*, 2010; Oh *et al.*, 2012; Kumar *et al.*, 2013). The traditional IP and MAC addresses binding and the division of the virtual LAN method can be classified into this category; however, this method belongs to passive defense and increases network complexity and maintenance cost. The second aspect is the protocol improvement technology, which changes mainly the protocol process or increases the number of protocol steps to enhance the security of ARPs (Bruschi *et al.*, 2003; Goyal and Tripathy, 2005; Issac

* Project supported by the National Natural Science Foundation of China (No. 61472100)
ORCID: Guang-jia SONG, http://orcid.org/0000-0001-6429-2426

and Mohammed, 2005). For example, Gouda and Huang (2003) modified a Dynamic Host Configuration Protocol (DHCP) server in LAN and expanded the DHCP to support the address-resolution process. However, this change caused a single point of failure problem and increased the network cost. The third aspect is the encrypted communication technology, such as the use of the asymmetric encryption technology to encrypt an address-resolution message to prevent IP address spoofing (Goyal and Tripathy, 2005).

Although RFC4861 suggests using IP security (IPsec) to protect the Neighbor-Discovery Protocol (NDP) message, the use of IPSec in neighbor discovery still faces several problems. The Internet Engineering Task Force (IETF) proposed SEcure-Neighbor Discovery (SEND) to enhance the security of NDP. SEND uses cryptographically generated address (CGA), digital signature, and time stamp to protect the NDP message (Arkko *et al.*, 2005). However, key management remains a problem in encryption communications; the coexistence of SEND and NDP may also cause routing problems (Hou *et al.*, 2012). Given its high complexity (Rafiee *et al.*, 2011), SEND remains to be in the experimental stage (Oh and Chae, 2007; Su *et al.*, 2010; AlSa'deh *et al.*, 2012). A logical problem in encrypted communication is that both parties must know the MAC address of each other before key change; acquiring the MAC address is the purpose of address resolution. If fraud occurs in the MAC-obtaining process, then the follow-up communication security is lost.

Source address validation architecture is a new security method, which filters packets based on their source addresses. This method prevents attacks directly from the source and provides convenience for source address tracking, traceability, and network diagnosis and management (Wu *et al.*, 2007; 2008). Source address validation implementation (SAVI) is a protocol and a security mechanism that needs the entire network support, and remains in its experimental stage. Source address validation in autonomous regions needs a router support, which must complete several centralized computing to affect network performance. Also, SAVI is difficult to deploy because equipment manufacturers implement the Simple Network Management Protocol (SNMP) in various ways (Li *et al.*, 2012; Xiao and Bi, 2013).

Security research on ARPs remains in the stage of experience. Most of the studies lack theoretical supports. Single point of failure, hardware cost, operation complexity, and other factors also restrict the development of ARPs. This study aims to resolve the address-resolution security issue by proposing a new secure ARP called the anonymous-address-resolution (AS-AR) protocol. This protocol does not reveal the destination address; the MAC and IP addresses of the source node are hidden in the address-resolution process. As a result, AS-AR has a higher security level than existing security methods.

2 Address-resolution spoofing

Taking ARP as an example (NDP is similar), each host in the LAN has an ARP cache table to store the IP and MAC addresses of other hosts. When host A wants to send a data packet to host B, A first checks whether the MAC of B is available in the cache; if not, host A carries out an ARP broadcast and asks host B to reply to its MAC. All hosts in LAN receive this broadcast; however, only host B gives an ARP reply containing its IP and MAC addresses. After receiving a reply, host A updates its cache and sends a data packet to host B according to the MAC address in the cache (Plummer, 1982).

Although ARP is simple and efficient, security risks remain. First, all hosts in the LAN are assumed to be credible in ARP; however, malicious hosts may exist because of virus or malicious programs. Second, ARP lacks a verification mechanism. For example, when host A needs to communicate with host B, host A does not check whether the reply is true if host A receives an ARP reply after the ARP broadcast. Host A updates its cache as long as the destination MAC address in the ARP reply is its own. This process provides convenience for LAN ARP spoofing. The process of a common attack is as follows:

When host A broadcasts an ARP request to resolve the MAC address of host B, host C sends a reply, pretending to be host B. After receiving the reply, host A does not know that it is a fake message and mistakenly updates its ARP cache table, regarding host C as host B, and sends the data packet to host C that should have been sent to host B.

In view of this attack, NDP is used. NDP does not directly trust a neighbor-discovery broadcast; i.e., after receiving a neighbor advertisement (NA) broadcast, the host does not directly update the cache entry state to 'reachable'. The address entry is updated to 'reachable' only after the test and confirmation of the target host, as follows:

1. After the neighbor solicitation (NS) broadcast, the host receives an NA with the 'S' bit (included in the 'RSO' field) set to 1.

2. An upper layer confirmation, such as a Transmission Control Protocol (TCP) message, is received from the other end (Narten *et al.*, 2007).

However, attacks still exist in NDP, such as the use of THC-IPv6 tools. Thus, the spoofing attack remains the main threat to ARP (NDP) (van Heuse, 2016).

3 Anonymous-address-resolution

3.1 Cryptography methods used in AS-AR

1. Hash function: The hash function used in AS-AR is the message digest algorithm 5 (MD5). Hash function $h : \{0,1\}^* \rightarrow \{0,1\}^n$ maps a set of bit strings with an arbitrary length ($\{0,1\}^*$) to a set of bit strings with a length of n ($\{0,1\}^n$) (Stinson, 2005). By definition, hash function h can map a message x with an arbitrary length to a shorter message y with a fixed length. Message x is generally called preimage; y is generally called the message digest. The commonly used hash functions include secure hash algorithm 1 (SHA-1) and MD5. The preimage, second preimage, and collision problems are used to measure the security of the hash function:

(1) Resistance to a preimage attack (one-way): for any given output y, finding an x that makes $h(x) = y$ is computationally infeasible.

(2) Resistance to a second preimage attack: for any given input x, finding an input x', which is unequal to x and makes $h(x') = h(x)$, is computationally infeasible.

(3) Resistance to a collision attack: finding two unequal inputs x and x' that make $h(x) = h(x')$ is computationally infeasible.

2. International date encryption algorithm (IDEA): IDEA is a symmetric block encryption algorithm. This algorithm overcomes the short key problem of the Data Encryption Standard (DES). The length of IDEA key is 128 bits.

3.2 Design goals of AS-AR

From a logical point of view, the reason host C can cheat host A is that host C knows that host A is looking for host B. If host C does not know the target that host A is looking for, then host C will have difficulty to cheat. Moreover, if host C does not know that host A is currently carrying out an address resolution, host C will not know the attack target. In the original ARPs, the destination address and the IP and MAC addresses of the source host are all open and broadcasted in the network, thereby making all hosts in LAN (including the attacker) know the destination address of the address resolution and who is performing the address resolution. These are the weaknesses of the original ARPs.

To overcome the weaknesses of ARPs, for AS-AR, in addition to achieving the function of address resolution, two additional goals are as follows:

1. AS-AR does not leak the destination of an address resolution.

2. AS-AR does not leak the IP and MAC addresses of the source host.

To achieve goals 1 and 2, the destination of address resolution, IP and MAC addresses of the source host need to be hidden in AS-AR while allowing the target host to interpret these pieces of information. First, we use the one-way characteristic of the hash function to hide IP_X (IP_X represents the destination address of address resolution). In AS-AR, only the hash value of IP_X is public. Moreover, only the host with a specific IP address is allowed to know the real destination of address resolution; other hosts know only the hash value of address IP_X.

Second, we know that when host A wants to communicate with host B, whose IP address is IP_X, only host A itself knows the communication requirements. Thus, IP_X can be regarded as a common secret between hosts A and B. IP_X can be used as a public key between hosts A and B, and we can use it to encrypt the IP and MAC addresses of the source host to achieve encryption communication.

Given that IP_X is not open in AS-AR and that the IP and MAC addresses of the source host are encrypted by IP_X, goals 1 and 2 are achieved.

3.3 Workflow of AS-AR

3.3.1 Message format of AS-AR

For easy understanding, NDP is used as the prototype for AS-AR. The NDP message is composed of three parts, namely, Ethernet header, IPv6 header, and ICMPv6 part (Fig. 1). Specifically, in NDP, NS does not have an 'RSO' field. The values of the 'Type' field for NS and NA are 135 and 136, respectively. The 'Target address' field usually contains the destination IP address of the address resolution. The 'Options' field has different functions based on the types of ICMPv6 messages, which usually stores the MAC address of the host in NDP.

Unlike NDP, AS-AR uses two new packets, $\mathrm{NS}_{\text{AS-AR}}$ and $\mathrm{NA}_{\text{AS-AR}}$, to complete the address resolution. The values of 'Type' fields for $\mathrm{NS}_{\text{AS-AR}}$ and $\mathrm{NA}_{\text{AS-AR}}$ are 200 and 201, respectively (200 and 201 are ICMPv6 retention values for experiment). The following functions are used in the protocol:

1. Left(x, n) intercepts n bits from the left side of string x and obtains an n-bit binary string.

2. $E_K(x)$ encrypts x using IDEA with key K.

3. $D_K(x)$ decrypts x using IDEA with key K.

4. $H(x)$ computes the hash value of x and the length of the result is 128 bits.

Ethernet header	Dest MAC Src MAC Type
IPv6 header	Src IP Dest IP Next header
ICMPv6	RSO (only for NA) Type Target address Options

Fig. 1 Format of the NDP message

3.3.2 Workflow of AS-AR

The workflow of AS-AR is shown in Fig. 2. In the following description, we use IP_A and MAC_A to represent the IP and MAC addresses of the source host A, respectively, and use IP_B and MAC_B to represent the IP and MAC addresses of the target host B, respectively. AS-AR includes three stages:

1. Resolution initiation stage: When host A wants to communicate with host B whose IP address is IP_X, host A carries out the following steps:

Step 1: Each field of $\mathrm{NS}_{\text{AS-AR}}$ (Table 1) is calculated according to IP_X.

Step 2: $\mathrm{NS}_{\text{AS-AR}}$ is broadcasted.

2. Responses stage: Other host (represented by host B) carries out the following steps:

Step 3: After $\mathrm{NS}_{\text{AS-AR}}$ is received, 'Target address', 'Src IP', and 'Options' fields are extracted.

Step 4: If the address pool is not empty, an IP address is extracted and denoted as IP_Y; step 5 is then conducted. If no more address remains in the address pool, then the process is completed.

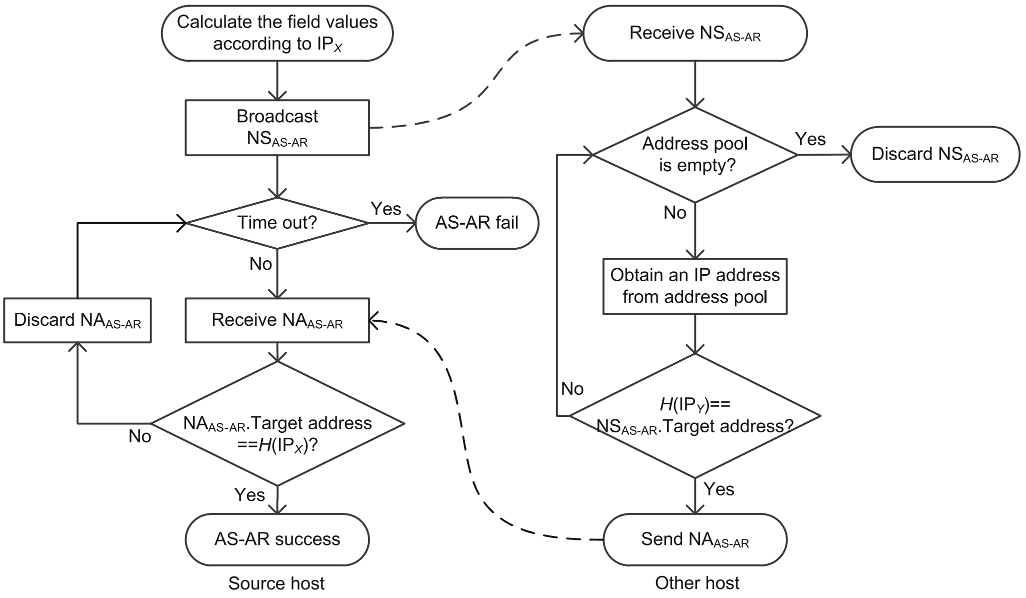

Calculate the field values according to IP_X
Broadcast $\mathrm{NS}_{\text{AS-AR}}$
Time out?
Yes
AS-AR fail
No
Discard $\mathrm{NA}_{\text{AS-AR}}$
Receive $\mathrm{NA}_{\text{AS-AR}}$
$\mathrm{NA}_{\text{AS-AR}}$.Target address $==H(\mathrm{IP}_X)$?
No
Yes
AS-AR success
Source host
Receive $\mathrm{NS}_{\text{AS-AR}}$
Address pool is empty?
Yes
Discard $\mathrm{NS}_{\text{AS-AR}}$
No
Obtain an IP address from address pool
No
$H(\mathrm{IP}_Y)==$ $\mathrm{NS}_{\text{AS-AR}}$.Target address?
Yes
Send $\mathrm{NA}_{\text{AS-AR}}$
Other host

Fig. 2 Flowchart of AS-AR

Table 1 Format of $\mathrm{NS}_{\text{AS-AR}}$

Part	Description	Value
Ethernet header	Dest MAC	3333-0000-0001
	Src MAC	Left($E_{\mathrm{IP}_X}(\mathrm{MAC}_A)$,48)
	Type	0x0806
IPv6 header	Src IP	$E_{\mathrm{IP}_X}(\mathrm{IP}_A)$
	Dest IP	ff02::1
	Next header	0x3a
ICMPv6	Type	200
	Target address	$H(\mathrm{IP}_X)$
	Options	$E_{\mathrm{IP}_X}(\mathrm{MAC}_A)$

Step 5: $H(\mathrm{IP}_Y)$ is calculated.

(1) If $H(\mathrm{IP}_Y)$=$\mathrm{NS}_{\text{AS-AR}}$.Target address, then IP_Y is used as the key to decrypt the IP and MAC addresses of host A, i.e.,

$$\mathrm{IP}_A = D_{\mathrm{IP}_Y}(\mathrm{NS}_{\text{AS-AR}}.\text{Src IP}),$$
$$\mathrm{MAC}_A = D_{\mathrm{IP}_Y}(\mathrm{NS}_{\text{AS-AR}}.\text{Options}),$$

IP_Y is used as the key to encrypt the IP and MAC addresses of itself, and $\mathrm{NA}_{\text{AS-AR}}$ (Table 2) is sent out to reply to host A.

Table 2 Format of $\mathrm{NA}_{\text{AS-AR}}$

Part	Description	Value
Ethernet header	Dest MAC	MAC_A
	Src MAC	Left($E_{\mathrm{IP}_Y}(\mathrm{MAC}_B)$,48)
	Type	0x0806
IPv6 header	Src IP	$E_{\mathrm{IP}_Y}(\mathrm{IP}_Y)$
	Dest IP	IP_A
	Next header	0x3a
ICMPv6	RSO	S=1
	Type	201
	Target address	$E_{\mathrm{IP}_Y}(\mathrm{IP}_Y)$
	Options	$E_{\mathrm{IP}_Y}(\mathrm{MAC}_B)$

(2) If $H(\mathrm{IP}_Y) \neq \mathrm{NS}_{\text{AS-AR}}$.Target address, then step 4 is repeated.

3. Verification stage: Host A carries out the following steps:

Step 6: Within the specified time (usually 3 s), host A verifies all received $\mathrm{NA}_{\text{AS-AR}}$ and checks whether the 'Target address' field is the same as $E_{\mathrm{IP}_X}(\mathrm{IP}_X)$:

(1) If $\mathrm{NA}_{\text{AS-AR}}$.Target address$=\mathrm{E}_{\mathrm{IP}_X}(\mathrm{IP}_X)$, then the address resolution is successful. $D_{\mathrm{IP}_X}(\mathrm{NA}_{\text{AS-AR}}.\text{Options})$ is the determined MAC address.

(2) If $\mathrm{NA}_{\text{AS-AR}}$.Target address $\neq \mathrm{E}_{\mathrm{IP}_X}(\mathrm{IP}_X)$, then $\mathrm{NA}_{\text{AS-AR}}$ is discarded.

Algorithm 1 shows the resolution initiation and verification stages that host A uses. Algorithm 2 shows the operations of host B in the response stage.

Algorithm 1 Sending $\mathrm{NS}_{\text{AS-AR}}$ and verifying $\mathrm{NA}_{\text{AS-AR}}$

```
Input: IPv6 address $\mathrm{IP}_X$
Output: true: AS-AR succeeds; false: AS-AR fails
Broadcast $\mathrm{NS}_{\text{AS-AR}}$
while AS-AR time-out $\neq$ true do
    Receive $\mathrm{NA}_{\text{AS-AR}}$
    if $\mathrm{NA}_{\text{AS-AR}}$.Target address $==E_{\mathrm{IP}_X}(\mathrm{IP}_X)$ then
        $\mathrm{MAC}_B = D_{\mathrm{IP}_X}(\mathrm{NA}_{\text{AS-AR}}.\text{Options})$
        return true
    else
        Discard $\mathrm{NA}_{\text{AS-AR}}$
    end if
end while
return false
```

Algorithm 2 Receiving and verifying $\mathrm{NS}_{\text{AS-AR}}$

```
Input: $\mathrm{NA}_{\text{AS-AR}}$
Output: true: send $\mathrm{NA}_{\text{AS-AR}}$ to reply; false: discard $\mathrm{NS}_{\text{AS-AR}}$
Receive $\mathrm{NS}_{\text{AS-AR}}$
while address pool is not empty do
    Take out an IP address as $\mathrm{IP}_Y$
    if $H(\mathrm{IP}_Y) == \mathrm{NS}_{\text{AS-AR}}$.Target address then
        Send out $\mathrm{NA}_{\text{AS-AR}}$
        return true
    end if
end while
Discard $\mathrm{NS}_{\text{AS-AR}}$
return false
```

3.4 Example of AS-AR

We illustrate AS-AR with an example. We suppose that three hosts exist in LAN, and their basic information is listed in Table 3.

Assuming that host A wants to resolve the MAC address of host B whose IP address is 1::5:b, host A first calculates the following:

$$H(1::5:\mathrm{b}) = \mathrm{baed6fb66cbedfdca5d0a910d315bf89}. \tag{1}$$

Host A then uses 1::5:b as the key to encrypt its

Table 3 Basic information of three hosts

Host	IP	MAC	Hash of IP
A	1::5:a	0800-270c-0001	1298591506654c57623885a782f31ed9
B	1::5:b	0800-270c-0002	baed6fb66cbedfdca5d0a910d315bf89
A	1::5:c	0800-270c-0003	e77b56dc72bf9e34eba732592a5a6dd0

own IP and MAC addresses as follows:

$$\begin{aligned}&\mathrm{NS}_{\text{AS-AR}}.\text{Src IP} = E_{1::5:\text{b}}(1::5:\text{a})\\ &= \text{8d0d12cfef09f97f4e4b67d07c1cab13}. \end{aligned} \tag{2}$$

$$\begin{aligned}&\mathrm{NS}_{\text{AS-AR}}.\text{Src MAC}\\ &= \text{Left}(E_{1::5:\text{b}}(\text{080-270c-0001}), 48)\\ &= \text{ef0f-97d6-2b16}. \end{aligned} \tag{3}$$

Host A fills the 'Dest MAC' field of $\mathrm{NS}_{\text{AS-AR}}$ with '3333-0000-0001' (MAC broadcast address), the 'Options' field of $\mathrm{NS}_{\text{AS-AR}}$ with $E_{1::5:\text{b}}$(080-270c-0001), and the 'Target address' field with H(1::5:b). Host A then broadcasts the $\mathrm{NS}_{\text{AS-AR}}$.

Both hosts B and C can receive this $\mathrm{NS}_{\text{AS-AR}}$. Host C takes out the address 1::5:c from its address pool, calculates

$$H(1::5:\text{c}) = \text{e77b56dc72bf9e34eba732592a5a6dd0}, \tag{4}$$

and finds that H(1::5:c) is not equal to $\mathrm{NS}_{\text{AS-AR}}$.Target address and no more address remains in the address pool. Thus, host C discards $\mathrm{NS}_{\text{AS-AR}}$. Host B takes out the address 1::5:b from its address pool, calculates

$$H(1::5:\text{b}) = \text{baed6fb66cbedfdca5d0a910d315bf89}, \tag{5}$$

and finds that H(1::5:b) is equal to $\mathrm{NS}_{\text{AS-AR}}$.Target address. Accordingly, host B uses 1::5:b as the key to decrypt the IP and MAC addresses of host A.

$$\begin{aligned}&\mathrm{IP}_A\\ &= D_{1::5:\text{b}}(\text{8d0d12cfef09f97f4e4b67d07c1cab13})\\ &= 1::5:\text{a}. \end{aligned} \tag{6}$$

$$\begin{aligned}&\mathrm{MAC}_A\\ &= D_{1::5:\text{b}}(\text{ef0f97d62b1622e4da0abaa3d17d8443})\\ &= \text{0800-270c-0001}. \end{aligned} \tag{7}$$

Using 1::5:b as the key, host B encrypts its own IP and MAC addresses and then sends $\mathrm{NA}_{\text{AS-AR}}$ as a reply to host A. The field assignment is as follows:

$$\begin{aligned}&\mathrm{NA}_{\text{AS-AR}}.\text{Src IP} = E_{1::5:\text{b}}(1::5:\text{b}),\\ &= \text{d1e9df6c6c0a428d32f736e7b7e0781f}. \end{aligned} \tag{8}$$

$$\begin{aligned}&\mathrm{NA}_{\text{AS-AR}}.\text{Src MAC}\\ &= \text{Left}(E_{1::5:\text{b}}(\text{0800-270c-0002}),48). \end{aligned} \tag{9}$$

$$\mathrm{NA}_{\text{AS-AR}}.\text{Dest IP} = \mathrm{IP}_A. \tag{10}$$

$$\mathrm{NA}_{\text{AS-AR}}.\text{Dest MAC} = \mathrm{MAC}_A. \tag{11}$$

$$\mathrm{NA}_{\text{AS-AR}}.\text{Target address} = E_{1::5:\text{b}}(1::5:\text{b}). \tag{12}$$

$$\mathrm{NA}_{\text{AS-AR}}.\text{Options} = E_{1::5:\text{b}}(\text{0800-270c-0002}). \tag{13}$$

After host A receives $\mathrm{NA}_{\text{AS-AR}}$, it needs to verify its 'Target address' field; if $\mathrm{NA}_{\text{AS-AR}}$ passes the verification, then host A uses 1::5:b as the key to decrypt $\mathrm{NA}_{\text{AS-AR}}$.Options field to acquire the MAC address of host B. The address resolution is thus success. The $\mathrm{NS}_{\text{AS-AR}}$ and $\mathrm{NA}_{\text{AS-AR}}$ used in this process are shown in Fig. 3.

4 Security analysis and experiment

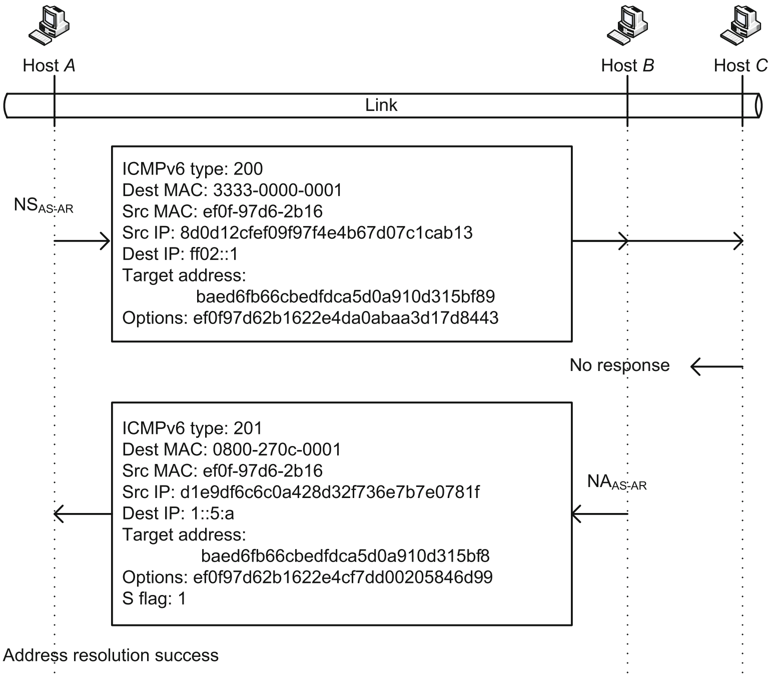
4.1 Security analysis

In the following analysis, we assume that the network environment is LAN and that nodes exchange information through switches. Assuming that host C is an attack node, host C has the following communication abilities:

1. C can receive broadcast and unicast.
2. C can change its protocol stack to send any NDP message.
3. C cannot monitor point-to-point communication; e.g., the message is forwarded from one port of the switch to another.

4.1.1 Security analysis of address information

Address information is important in the address-resolution process. Five pieces of address information are important, namely, the IP and MAC addresses of the source node (IP_A and MAC_A), the destination of address resolution (IP_X), and the IP



```
ICMPv6 type: 200
Dest MAC: 3333-0000-0001
Src MAC: ef0f-97d6-2b16
Src IP: 8d0d12cfef09f97f4e4b67d07c1cab13
Dest IP: ff02::1
Target address:
           baed6fb66cbedfdca5d0a910d315bf89
Options: ef0f97d62b1622e4da0abaa3d17d8443
```

```
ICMPv6 type: 201
Dest MAC: 0800-270c-0001
Src MAC: ef0f-97d6-2b16
Src IP: d1e9df6c6c0a428d32f736e7b7e0781f
Dest IP: 1::5:a
Target address:
            baed6fb66cbedfdca5d0a910d315bf8
Options: ef0f97d62b1622e4cf7dd00205846d99
S flag: 1
```

Fig. 3 Example of AS-AR

and MAC addresses of the destination node (IP$_B$ and MAC$_B$).

In traditional ARPs, given that the request is sent out in broadcast in the resolution initiation stage, IP$_A$, MAC$_A$, and IP$_X$ are all broadcasted in plaintext; hence, the attacker can obtain these three items. Host C can use such information to attack. In the response stage, the destination node replies with unicast. Consequently, the attacker cannot obtain IP$_B$ and MAC$_B$ by monitoring the reply message. Other network environments may be different, such as in a wireless network; thus, the attacker may obtain all the five items.

In AS-AR, given IP$_A$ and MAC$_A$ of host A, IP$_X$, IP$_B$, and MAC$_B$ of host B are all encrypted. Thus, none of the items can be obtained by the attacker.

4.1.2 Security analysis of the 'Target address' field

Birthday attack shows that for the message digest with a length of n, there will be a probability of 50% of causing a collision for the random plaintext selection operated $O(2^{n/2})$ times. For the iterated function with an MD structure, differential attack is an effective method of searching for collision. Wang *et al.* (2005) and Wang and Yu (2005) presented an effective attack method for a hash function with an MD structure. Collision for MD5 is found after cycling for more than 2^{39} times, and the computation time is 15 min or more. Therefore, from the collision perspective, the AS-AR using MD5 has some risks, in which an attacker can find a collision through a great amount of calculations. However, three situations make this attack infeasible:

1. Address resolution is generally completed within 3 s; an attack with limited computing power cannot find a collision within such a short period.

2. The attacker cannot use a collision address to decrypt the IP and MAC addresses of the source host; hence, it cannot send NA$_{\text{AS-AR}}$ to cheat.

3. Even if the attacker finds a collision in 3 s, the collision address is invalid for the source host, and it cannot pass the verification stage; the only way to attack is finding the preimage.

4.1.3 Attack mode analysis

1. Ordinary attack: Based on the destination address of address resolution, the attacker can forge a reply to cheat the victim host. However, in AS-AR, the destination address is closed; thus, the attacker needs to use a random address to carry out an attack. In theory, the probability of success is almost zero. Meanwhile, the attacker does not know the MAC address of the host who sent NS$_{\text{AS-AR}}$ and hence cannot send a packet to the victim host. Therefore,

an ordinary attack is infeasible in AS-AR.

2. Intensive attack: Based on the network prefix announced in router advertisement, the attacker generates a large number of random addresses and sends $\text{NA}_{\text{AS-AR}}$ to reply, to increase the success rate of attack.

We assume that only one network prefix exists in the network. Given that the length of the IPv6 address is 128 bits and that the length of the network prefix is 64 bits, the possible size of the address space is as high as 2^{64}. The length of NA is 90 bytes, and the address-resolution process is completed in 3 s. Taking bandwidth 1000 Mb/s as an example, the number of NDP messages that an attacker can send in 3 s is

$$n = \frac{1000 \times 2^{10} \times 2^{10} \times 3}{90 \times 2^3}. \tag{14}$$

Supposing that the probability that these addresses do not include IP_X is P, then

$$\begin{aligned} P &= \left(\frac{2^{64}-1}{2^{64}}\right)\left(\frac{2^{64}-2}{2^{64}}\right)\cdots\left(\frac{2^{64}-n+1}{2^{64}}\right) \\ &= \prod_{i=1}^{n-1}\left(1-\frac{i}{2^{64}}\right). \end{aligned} \tag{15}$$

Accordingly, the success probability of an attack is

$$1-\prod_{i=1}^{n-1}\left(1-\frac{i}{2^{64}}\right) \approx 1-\exp\left(-\frac{1}{2^{21}}\right). \tag{16}$$

Thus, the success rate of an intensive attack is almost zero.

4.1.4 Real collision occurrence

Assuming that host B has an address IP_Z, where $\text{IP}_Z \neq \text{IP}_X$ but $H(\text{IP}_Z) = H(\text{IP}_X\ )$, a real collision occurs. When host A performs address resolution, host B thinks it is the target that host A is looking for, uses IP_Z to decrypt IP_A and MAC_A, and sends $\text{NA}_{\text{AS-AR}}$ to reply. Since $\text{IP}_Z \neq \text{IP}_X$, the MAC address that host B decrypted is wrong, indicating that $\text{NA}_{\text{AS-AR}}$ was sent to a non-existent MAC address. Because the switch has the ability of address learning, it will broadcast this $\text{NA}_{\text{AS-AR}}$. All nodes (including the attacker) will receive this $\text{NA}_{\text{AS-AR}}$. However, in addition to the plaintext contained in the Dest MAC field, other address fields contain ciphertext. Host C accordingly cannot obtain enough address information from $\text{NA}_{\text{AS-AR}}$ to attack host A or B.

4.2 Comparative analysis

4.2.1 Comparison between AS-AR and IPSec

Internet Protocol Security (IPSec) is a tunnel encryption protocol proposed by IETF. It has the following three work modes:

1. Host to host (transmission mode).
2. Gateway to gateway (tunnel mode).
3. Host to gateway.

The transmission mode is used to guarantee the communication security between host and host. This mode uses mainly the authentication header (AH) to encrypt the IP header and encapsulating security payload (ESP) to encrypt the payload. However, similar to most of the encryption methods, a key exchange process is necessary before the encrypted communication between the hosts. IPSec uses the Internet key exchange (IKE) protocol to perform key exchange. IKE is a point-to-point and unicast key management protocol. The two sides establish point-to-point communication, which is a premise that IKE plays a role. Address resolution is the premise of establishing point-to-point communication; thus, IPSec cannot be used to protect address-resolution security currently.

4.2.2 Comparison with SEND

IETF proposed SEND to enhance the security of NDP. SEND uses a new address format, CGA, to allow a host to prove that it has a particular address. If a host sends NA to reply to NS (CGA format), then the host can request the reply node to provide original auxiliary parameters of CGA. A characteristic of CGA is that the original auxiliary parameters cannot be inferred from the CGA itself. Therefore, CGA prevents the malicious host to cheat.

From the algorithm perspective, the safety of SHA-1 is higher than that of MD5, and SEND performs hash calculation twice. Thus, CGA can be verified twice. Only when both of the two verifications succeed, is the response host trusted. In contrast, AS-AR performs the verification only once.

From the information-hiding perspective, the performance of AS-AR is better. Given that the address-resolution process is designed to obtain the MAC address of the target, the SEND protocol exposes the source node of the MAC address; thus, the attacker can know who to attack. The attacker can send invalid NA packets, thereby making the host decrypt and verify to form denial-of-service (DoS). In AS-AR, the MAC address of the source node is also hidden. Only a specific host knows who launched the NS; thus, AS-AR has higher security and can defend from the DoS. In SEND, the malicious node can identify the IP and MAC addresses of the source host. However, in AS-AR, the attacker cannot identify who is performing address resolution and knows only that a node has initiated the address resolution. Aside from the hash value of the destination IP address, other information is unknown.

4.3 Simulation

We conduct a simulation to test the security performance of AS-AR. The simulation software is the Optimized Performance Network Engineering Tool (OPNET), and the network environment is LAN, including one switch node, one attack node, and seven normal nodes. Normal nodes carry out address-resolution periodicity. Each node contains two processors, namely, Src1 and Src2. Src1 is used to perform background traffic. The distribution of the number of packets generated per second in background traffic (Fig. 4) is obtained based on a 30-day statistical data in a university firewall (Heilongjiang University of Chinese Medicine, China, from June 19, 2013 to July 18, 2013. The data acquisition software is Solarwinds Orion and the firewall model is Hillstone M6860). Src2 is used to generate address-resolution packets by using a uniform distribution with a mean of 2.75 packets per second. The number of packets generated per second in background traffic is 1000 times that in address-resolution traffic.

In the simulation, the number of address of each node is 2^{12} and the number of network prefixes of the LAN is 2^6. Each network prefix has a length of 96 bits and an address space of 2^{32} to enhance the contrast effect. The address-resolution processes of SEND and AS-AR are simulated in the presence of an attack node. The cache pollution rate (CPR) is used to evaluate the performance of the address-resolution algorithms. CPR can be obtained by

$$\mathrm{CPR}=\frac{\text{Number of error entries in cache}}{\text{Number of total entries in cache}}. \tag{17}$$

For example, if 10 entries exist in cache, and one of them is wrong, then CPR is 10%. In the simulation, the attack node is set to have a significant computing power; i.e., the attacker can calculate the collision according to the hash value. Details are as follows:

1. In SEND, the attack node can calculate the collision auxiliary parameters based on CGA.

2. In AS-AR, the attack node can calculate the collision address based on the ‘Target address’ field.

The simulation results for SEND and AS-AR are shown in Fig. 5. In the simulation, the cache is initially empty; thus, the pollution rate is close to zero. Address resolution is then started, and the destination address is randomly generated. In the CGA environment, the attack node (host C) can generate the collision auxiliary parameters according to CGA (lower 64 bits of the ‘Target address’ field) after receiving NS and then sends NA to reply. CGA verification is a collision test, not a preimage verification. As long as the auxiliary parameters can be verified by hash1 and hash2, CGA verification is successful. Thus, host C can pass the verification successfully and cheat on host A, thereby making the cache store of host A an incorrect entry.

In AS-AR, host C can compute the collision address IP_Y based on the ‘Target address’ field and then sends $\mathrm{NA}_{\text{AS-AR}}$ to reply. Given that the IP and MAC addresses of host A are encrypted, host C cannot use IP_Y to decrypt. Thus, $\mathrm{NA}_{\text{AS-AR}}$ cannot be transmitted to host A. Only when host C finds a collision address that is the same as IP_X (the collision address is the preimage), can host C carry out an effective attack. Therefore, in the presence of the attack node, the CPR of SEND is much larger than that of AS-AR.

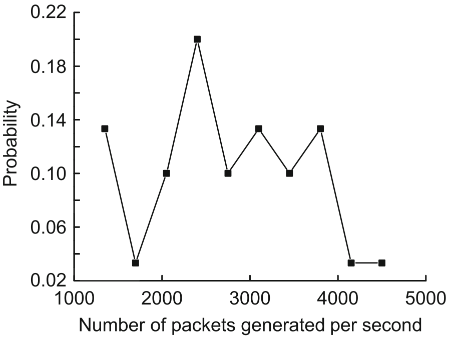


Fig. 4 Probability distribution of the number of packets generated per second in background traffic

4.4 Comparisons with other solutions

AS-AR is compared with five other typical methods in the aspects of cryptography used,

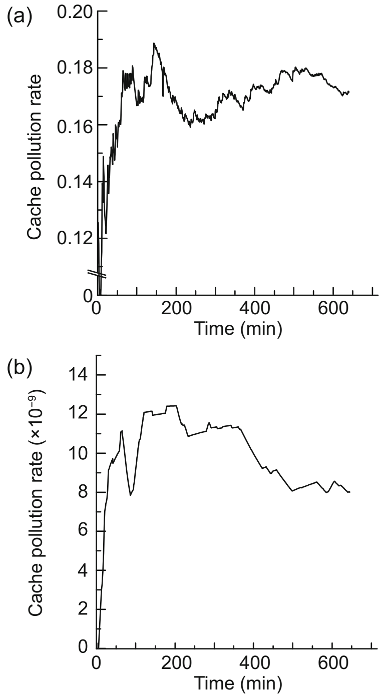


Fig. 5 Pollution rate comparison for SEND (a) and AS-AR (b)

third-party used, traffic monitor, performance degradation, and communication overhead (Table 4).

The methods used by Gouda, S-ARP, S-UARP, and Active DES all need an additional server in the network and must ensure that the server is always safe. However, this process adds to the deployment cost. In Active DES, the intrusion detection system (IDS) conducts port mirror on the switch to monitor all network traffic, but they require switch support.

The original ARPs need a broadcast process and a unicast process to complete the address-resolution process. For a LAN with n nodes, the communication overhead is $O(n+1)$, where n represents the overhead of a broadcast process, and 1 represents the overhead of a unicast process. Given that ES-ARP requires both the address-resolution request and the address-resolution response be transmitted in a broadcast manner, its communication overhead is $O(2n)$. Active DES monitors the address-resolution behavior of hosts; each time the node launches an address-resolution process, IDS has to conduct a similar address-resolution process to observe the event timing. Therefore, the communication overhead of Active DES is $O(2n+2)$. For Gouda and S-UARP, considering that a secure server was added and the secure server stored all IP and MAC addresses, common nodes could obtain the MAC address of other nodes by querying the secure server; thus, the broadcast process was avoided. The communication overhead of Gouda is accordingly $O(2)$. In S-UAR, after the node received the query results, it had to send an acknowledgement message to the server; hence, its communication overhead is $O(3)$. Both the resolution processes of AS-AR and S-ARP contain a broadcast and a unicast; the communication overheads are the same as those of the original ARPs, i.e., $O(n+1)$.

AS-AR, Gouda, S-ARP, and S-UARP all use cryptography methods, which will affect the performance of the protocol. We conduct a performance test of AS-AR, Gouda, S-ARP, and S-UARP (experimental platform: Windows 7 Service Pack 3, Intel i5 m520 CPU, 4 GB Corsair DDR3 memory). The test results are shown in Table 5.

In terms of time comparison, S-UARP and S-ARP need to encrypt or sign the entire message. Their time consumptions are more than 1 ms. Given that AS-AR and Gouda encrypted only part of the message, their time consumptions are less than 5 μs. Considering that no encryption method is used, the performance of ES-ARP is better than that of Gouda, S-ARP, and S-UARP. DES monitors only the address-resolution process, and it presents

Table 4 Comparisons between AS-AR and other solutions

Existing solution	Cryptography used	Third-party used	Traffic monitor	Performance degradation	Communication overhead
AS-AR	Yes	Yes	No	Middle	$O(n+1)$
ES-ARP	No	No	No	Low	$O(2n)$
S-ARP	Yes	Yes (authoritative key distributor)	No	High	$O(n+1)$
Active DES	No	Yes (intrusion detection system)	Yes	Very low	$O(2n+2)$
Gouda	Yes	Yes (secure server)	No	Low	$O(2)$
S-UARP	Yes	Yes (DHCP+secure server)	No	High	$O(3)$

ES-ARP: Ataullah and Chauhan (2012); S-ARP: Bruschi *et al.* (2003); Active DES: Barbhuiya *et al.* (2011); Gouda: Gouda and Huang (2003); S-UARP: Issac and Mohammed (2005)

Table 5 Performance comparisons between AS-AR and other solutions

Solution	Cryptography	Scope of encryption	Time consumption
AS-AR	MD5 and IDEA	Part of the message	3.0809 µs
Gouda	SHA-1	Part of the message	0.7968 µs
S-ARP	SHA-1 and DSA	Entire message	1.7913 ms
S-UARP	EES	Entire message	13.3374 ms

DSA: digital signature algorithm

almost no effect on the protocol performance. The performance degradation of these methods is shown in the fifth column of Table 4.

AS-AR includes MD5 calculation and IDEA calculation. The time complexities of MD5 and IDEA are $O(n^6)$ and $O(n^{6.9})$, respectively; therefore, the time complexity of AS-AR is $O(n^{6.9})$. Although using the encryption method will increase the running time of the protocol, the time consumption of the original address-resolution process is approximately 0.8 ms in the Ethernet environment (2.8 ms in the wireless environment). AS-AR increases only the running time of the protocol by less than 0.5%, which will insignificantly affect the performance of the protocol. AS-AR does not need to monitor network traffic or to add a security server; hence, the deployment cost is low. However, AS-AR also has some shortcomings. If the nodes in the LAN have a large address pool, it will affect the network performance.

5 Conclusions

The design principle of a security protocol is that an attacker cannot obtain more information than the protocol itself through breach of protocol. Address information is, undoubtedly, the most important information for address resolution. Existing ARPs are vulnerable to attack because an attacker can obtain considerable address information, such as the destination of the resolution and the IP and MAC addresses of the source node, thus allowing attackers to take initiatives. AS-AR uses an anonymous way to hide address information, that is, regarding the destination address as important information because it is the characteristic value of the target. AS-AR considers the destination address as the key to encrypt other address information. In an ideal case, the attacker cannot obtain any address information from the source and destination nodes, thereby ensuring the security of the resolution process. Comparative analysis shows that AS-AR is better than SEND and other existing security solutions in terms of information hiding. However, AS-AR also has some shortcomings. Deploying AS-AR in the existing network environment remains difficult, but it can be used in the next-generation network.

References

AlSa'deh, A., Rafiee, H., Meinel, C., 2012. Stopping time condition for practical IPv6 cryptographically generated addresses. 26th IEEE Int. Conf. on Information Networking, p.257-162.
http://dx.doi.org/10.1109/ICOIN.2012.6164388

Arkko, J., Kempf, J., Zill, B., *et al.*, 2005. SEcure Neighbor Discovery (SEND). Internet Engineering Task Force. Available from http://tools.IETF.org/html/rfc3971.

Ataullah, M., Chauhan, N., 2012. ES-ARP: an efficient and secure address resolution protocol. IEEE Students' Conf. on Electrical, Electronics & Computer Science, p.1-5.
http://dx.doi.org/10.1109/SCEECS.2012.6184794

Barbhuiya, F.A., Biswas, S., Nandi, S., 2011. An active DES based IDS for ARP spoofing. IEEE Int. Conf. on Systems, Man & Cybernetics, p.2743-2748.
http://dx.doi.org/10.1109/ICSMC.2011.6084088

Bruschi, D., Ornaghi, A., Rosti, E., 2003. S-ARP: a secure address resolution protocol. IEEE 19th Annual Computer Security Applications Conf., p.66-74.
http://dx.doi.org/10.1109/CSAC.2003.1254311

Fall, K.R., Stevens, W.R., 2011. TCP/IP Illustrated, Volume I: the Protocols. Addison-Wesley, London.

Garcia-Martine, A., Bagnulo, M., 2012. An integrated approach to prevent address spoofing in IPv6 links. *IEEE Commun. Lett.*, **16**(11):1900-1902.
http://dx.doi.org/10.1109/LCOMM.2012.100812.121517

Gouda, M.G., Huang, C.T., 2003. A secure address resolution protocol. *Comput. Netw.*, **41**(1):57-71.
http://dx.doi.org/10.1016/S1389-1286(02)00326-2

Goyal, V., Tripathy, R., 2005. An efficient solution to the ARP cache poisoning problem. *LNCS*, **3574**:40-51.
http://dx.doi.org/10.1007/11506157_4

Hou, Y., Wang, Z., Wang, Y., *et al.*, 2012. Routing attack in the ND and SEND mixed environment. 4th IEEE Int. Conf. on Multimedia Information Networking and Security, p.959-962.
http://dx.doi.org/10.1109/MINES.2012.196

Issac, B., Mohammed, L.A., 2005. Secure unicast address resolution protocol (S-UARP) by extending DHCP. 13th IEEE Int. Conf. on Networks, p.1-6.
http://dx.doi.org/10.1109/ICON.2005.1635503

Kumar, N., Bansal, G., Biswas, S., *et al.*, 2013. Host based IDS for NDP related attacks: NS and NA spoofing. Annual IEEE India Conf., p.1-6.
http://dx.doi.org/10.1109/INDCON.2013.6726054

Li, J., Wu, J., Xu, K., *et al.*, 2012. A hierarchical interdomain authenticated source address validation solution. *Chin. J. Comput.*, **35**(1):85-100 (in Chinese).
http://dx.doi.org/10.3724/SP.J.1016.2012.00085

Nam, S.Y., Kim, D., Kim, J., 2010. Enhanced ARP: preventing ARP poisoning-based man-in-the-middle attacks.

IEEE Commun. Lett., **14**(2):187-189. http://dx.doi.org/10.1109/LCOMM.2010.02.092108

Narten, T., Nordmark, E., Simpson, W., *et al.*, 2007. Neighbor Discovery for IP Version 6 (IPv6). Internet Engineering Task Force. Available from http://tools.IETF.org/html/rfc4861.

Oh, H., Chae, K., 2007. An efficient security management in IPv6 network via MCGA. 9th Int. Conf. on Advanced Communication Technology, p.1179-1181. http://dx.doi.org/10.1109/ICACT.2007.358569

Oh, M., Kim, Y.G., Hong, S., *et al.*, 2012. ASA: agent-based secure ARP cache management. *IET Commun.*, **6**(7): 685-693. http://dx.doi.org/10.1049/iet-com.2011.0566

Plummer, D.C., 1982. An Ethernet Address Resolution Protocol—or—Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware. Internet Engineering Task Force. Available from http://tools.IETF.org/html/rfc826.

Rafiee, H., AlSa'deh, A., Meinel, C., 2011. WinsSEND: Windows SEcure Neighbor Discovery. 4th Int. Conf. on Security of Information and Networks, p.243-246. http://dx.doi.org/10.1145/2070425.2070469

Rehman, S.U., Manickam, S., 2015. Integrated framework to detect and mitigate denial of service (DoS) attacks on duplicate address detection process in IPv6 link local communication. *Int. J. Secur. Appl.*, **9**(11):77-86. http://dx.doi.org/10.14257/ijsia.2015.9.11.08

Stinson, D.R., 2005. Cryptography: Theory and Practice. CRC Press.

Su, G., Wang, W., Gong, X., *et al.*, 2010. A quick CGA generation method. 2nd IEEE Int. Conf. on Future Computer and Communication, p.769-773. http://dx.doi.org/10.1109/ICFCC.2010.5497324

van Heuse, M., 2016. THC IPv6. Available from https://www.thc.org/thc-ipv6.

Wang, X., Yu, H., 2005. How to break MD5 and other hash functions. Int. Conf. on Theory & Applications of Cryptographic Techniques, p.19-35. http://dx.doi.org/10.1007/11426639_2

Wang, X., Lai, X., Feng, D., *et al.*, 2005. Cryptanalysis of the hash functions MD4 and RIPEMD. *LNCS*, **3494**:1-18. http://dx.doi.org/10.1007/11426639_1

Wu, J., Ren, G., Li, X., 2007. Source address validation: architecture and protocol design. IEEE Int. Conf. on Network Protocols, p.276-283. http://dx.doi.org/10.1109/ICNP.2007.4375858

Wu, J., Bi, J., Li, X., *et al.*, 2008. A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience. Internet Engineering Task Force. Available from https://datatracker.ietf.org/doc/rfc5210/?include_text=1.

Xiao, P., Bi, J., 2013. OpenFlow based intra-AS source address validation. *J. Chin. Comput. Syst.*, **34**(9):1999-2003 (in Chinese). http://dx.doi.org/10.3969/j.issn.1000-1220.2013.09.007