An Anti-DoS Duplicate Address Detection Model

GuangJia Song, JianHua Hu, Hui Wang,

***Abstract*—Duplicate address detection (DAD) is a necessary process before the host uses a new IP address to ensure its uniqueness. In the traditional DAD process, the detection target is public, thus making the detection process vulnerable to attacks, especially to denial-of-service attacks. A new detection method called Se-DAD is proposed in this paper to improve the security of DAD. In Se-DAD, the detected target is not disclosed to prevent the attacking node from forging a spoofing response. The hidden source MAC address also effectively prevents DoS attacks. Experiments show that Se-DAD is better than the previous detection methods considering address configuration failure rate, CPU, and memory overhead.**

***Index Terms*—duplicate address detection; address resolution; neighbor discovery; SEND; denial of service**

I. Introduction

In TCP/IP architecture, IP address is used as network address and host identifier [1]. The dual meaning of the IP address is crucial for the host. The IP protocol stipulates that a duplication must be detected before using an IP address; such process is mainly defined in address resolution protocol (ARP) [2] and neighbor discovery protocol (NDP) [3], [4]. Assuming the presence of three hosts in the local area network (LAN), namely A, B, and C, the general Duplicate address detection (DAD) process can be described as follows.

Step 1: If host A wants to use IP_X as its new address IP_X (in this paper, IP_X is always used to represent the destination address of DAD), then host A must conduct a broadcast and claim that it will use IP_X.

Step 2: Hosts B and C will check their address pools after receiving the broadcast. If IP_X exists, then host B (or C) must provide a reply.

Manuscript received May 20, 2021; revised March 11, 2022.

This work was supported by the National Social Science Foundation of China (Grant No.17BGL232), Basic public welfare research program of Zhejiang Province (Grant No. LGG22F020002), Shaoxing Educational Science Planning Project (Grant No. SGJ2022042)

Guangjia Song is a lecturer at the Department of engineering and technology, Jiyang College of Zhejiang A&F University, Shaoxing 311800, China and is the corresponding author, email: tysong@aliyun.com.

Jianhua Hu is an associate professor at the Department of engineering and technology, Jiyang College of Zhejiang A&F University, Shaoxing 311800, China, email: 651511632@qq.com.

Hui Wang is an engineer at the Department of Network Security, National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100000, China, email: wh@cert.org.cn.

Step 3: If host A receives a reply, then this phenomenon indicates that IP_X is a conflict. If no reply is received, then the IP_X is available.

The duplicate address detection (DAD) process faces many problems, particularly security. The protocol in the detection process assumes that all nodes are honest, but the actual situation indicates that malicious nodes are ubiquitous [5], [6]. Assuming that host C is malicious, step 1 of DAD indicates that all nodes know that host A will use IP_X because IP_X is public. Therefore, step 2 indicates that host C has two attack methods:

- Send a forged reply to generate a conflict.
- Conduct a DAD process whose destination is the same as the IP_X.

Both methods will cause failure in host address configuration according to the current protocols (ARP and NDP). If the attack is continuous, then host A cannot obtain an available address. Thus, a denial-of-service (DoS) attack is formed.

A Secure DAD model called Se-DAD is proposed in this paper. In Se-DAD, the destination of DAD and the MAC of the source host are not public. Thus, the attack node neither knows the destination address of DAD nor which node performs the DAD. The attack node cannot also perform targeted attacks. The rest of the paper is organized as follows. Section 2 introduces the main problems and research status of DAD. Section 3 presents the principle and working process of Se-DAD. Section 4 shows the experiment and analysis. Section 5 concludes this paper.

II. Related Works

A. Development of duplicate address detection

ARP request message and ARP reply message are used to complete the DAD, and the DAD and address resolution processes are similar. If host A wants to use IP_X as its new address, then it sends out an ARP request to perform an address resolution for IP_X. If a reply exists, then a conflict in the existing address is present. RFC5227 proposed a new method called address conflict detection (ACD) to avoid polluting the cache of other hosts [7]. ACD adds two new packets as follows: ARP probe and ARP announcement. ARP probe is similar to ARP request, but its “source IP” field is filled with “0.0.0.0” so as to reduce the cache pollution for other hosts.

The detection process in NDP mainly depends on neighbor solicitation (NS) and neighbor advertisement (NA) packets to complete. The format of the NDP message is shown in Fig. 1. “Target address” is used to store the destination of DAD. The

"Options" field is different depending on the "Type" field of the message, and it usually stores the MAC address. The "Type" of NS and NA is 135 and 136, respectively. The "Flags" field is valid only in NA. If the MAC addresses of hosts A and B are 00E0-FC00-0001 and 00E0-FC00-0002, respectively, then the IPv6 address of host B is 1::2:B. If host A also wants to use the 1::2:B, then host B will reply with an NA to show the address is conflict after host A broadcasts an NS. The details of NS and NA are shown in Fig. 2.

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 NDP message

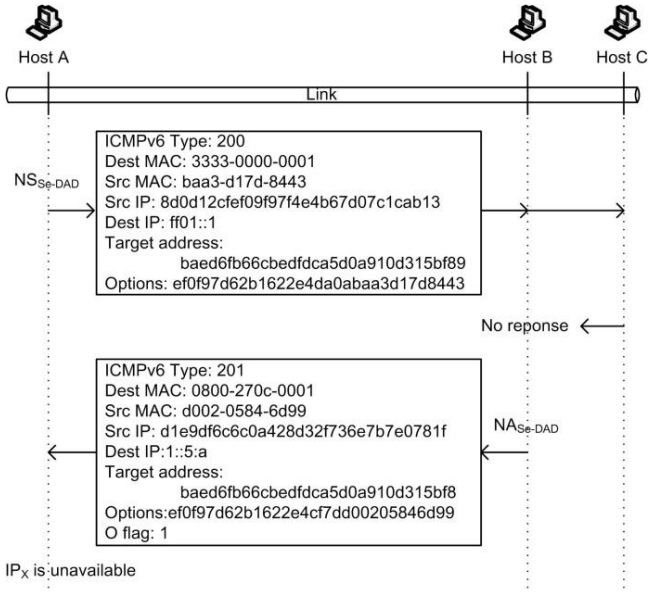

Fig. 2. NS and NA

B. *Existing problems*

The host usually has only one IP address in IPv4, and the address rarely changes. However, this situation gradually changes with the emergence of the wireless sensor network (WSN), mobile ad-hoc network (MANET), and the promotion and application of IPv6. On the one hand, the addresses that a host owned increased. On the other hand, the nodes are mobile in WSN, MANET, and mobile IPv6 (MIPv6). The nodes may leave an old network and enter a new one at any time. The network itself will also produce partitions or merge. All these incidents will elicit changes in the IP of nodes. Thus, the DAD process will become frequent, and the problems encountered by DAD will gradually emerge [8]. DAD is currently facing three aspects of the problem that must be solved urgently.

Time consuming

DAD process must be completed in 1–3 s according to the current protocols. Delays are intolerable for time-restricted applications, especially in MIPv6. Many improved methods are proposed to reduce time consumption. For example, the typical solution in MIPv6 is to use a new IP address first and then perform the DAD process or use proactive-DAD to achieve rapid handover [9]–[11].

Overhead

Energy is valuable in wireless environments, such as MANET and WSN. Effective routing algorithms can increase the lifetime of the network, such as [12]. However, the overhead of DAD cannot be ignored. The DAD process must be performed whenever a node obtains a new IP address or network partitioning or merging to avoid address conflict. An excessively large overhead of DAD will affect the survival of the network. The host usually adopts a special address configuration or uses a global address database to reduce the overhead and the number of DAD. The node can complete the DAD by comparing the address with the central database, which avoids the consumption of network resources by flooding [13], [14].

Security issues

Spoofing attack is the main threat of DAD. The detection address is public in DAD, and any malicious node can send a false reply to convince the host regarding the existence of an address conflict. A malicious node can also use attack tools, such as THC-IPv6, to achieve the aforementioned purpose [15]. Some researchers use centralized authority to increase security; however, this approach is unrealistic for many network environments [16]. IPSec is currently experiencing difficulties in the protection of the link layer protocols. The premise of the IPSec lies in the completed key exchange by the two sides, and IPSec protects the point-to-point communication, excluding broadcast communication. However, most of the DAD processes occur before point-to-point communication is established [17]. The IPSec mechanism has played a role between A and C. However, host C can still cheat due to the disclosure of the detection address.

Furthermore, considering the viewpoint of nodes, determining the legality of the mapping between IP and MAC given by the other is impossible. Intrusion detection is an effective means to improve network security but plays a small role in dealing with DAD security [18].

SEND is an innovative protocol; its feature is self-certified [19]. With DAD as an example, if host C shows that IP_X is involved in a conflict, then host A can require C to provide proof of IP_X ownership. However, SEND uses cryptographically generated address (CGA) as its address format; a feature of CGA is that the auxiliary parameters cannot be inferred from CGA itself [20]. Therefore, SEND can open the destination of DAD without fear of being cheated.

The algorithm of CGA is remarkably complex and needs a considerable amount of calculation. Literature [21] suggested using the entropy of the system state to generate Interface IDentifier (IID) to shorten the address generation time. Authentication can also be used to restrict node behavior and improve the overall security level of LAN, but this method requires a centralized server and high deployment cost [22].

If SAVI is deployed in the network, especially SEND-SAVI, then the attack of deception can be effectively prevented [23], [24]. In the SAVI environment, the switch can

bind an IP address to a switch Port. If the host sends a message that the source IP address is inconsistent with the binding information, then the switch will refuse to forward it [25]. However, binding information in SAVI is extracted from the DAD message by monitoring network traffic. Therefore, SAVI does not check the NS message during DAD.

III. An Anti-DoS Duplicate Detection Model

A. Se-DAD

A safe DAD process called Se-DAD is designed to solve the security problem of DAD, especially DoS attacks. In addition to achieving the basic function of DAD, the extra design goals of Se-DAD are as follows.

(1) Protocol does not require a third-party authority;

(2) Protocol can prevent forge response;

(3) Protocol can prevent DoS attacks.

The IP_X must be hidden in the DAD process to achieve goal (2). Meanwhile, the MAC address of the source host must also be hidden to achieve goal (3). Only host A knows the communication requirements when it wants to communicate with the target hos, which has a network address of IP_X. Thus, IP_X can be regarded as a secret between host A and the target host. IP_X can also be used as a public key to encrypt the address information to achieve goals (2) and (3).

The NDP is taken as the prototype of Se-DAD. Two new packet formats in Se-DAD are defined as follows: $NS_{Se\text{-}DAD}$ and $NA_{Se\text{-}DAD}$. Their Type fields separately use the retention values (200 and 201) of ICMPv6.

The following functions are used in Se-DAD:

$Right(x,n)$, intercept n bit from the right side of string x and return an n bit binary string;

$E_K(x)$, using K as the key to encrypt string x;

$D_K(x)$, using K as the key to de-encrypt string x;

$H(x)$, compute the hash value of x and return a 128-bit binary string.

Se-DAD is designed as follows.

(1) DAD initiation stage. Host A performs the following operation.

Step 1: If host A wants to use a new address IP_X, then host A broadcasts a DAD message $NS_{Se\text{-}DAD}$, and each field assignment is shown in Fig. 3.

Ethernet header	Dest MAC	3333-0000-0001
	Src MAC	$Rigth(E_{IP_X}(Src\_MAC),48))$
	Type	0x86dd
IPv6 header	Src IP	$E_{IP_X}(Src\_IP)$
	Dest IP	FF02::1
	Next header	0x3a
ICMPv6	RSO(only for NA)	Null
	Type	200
	Target address	$H(IP_X)$
	Options	$E_{IP_X}(Src\_MAC)$

Fig. 3. Fields Assignment of $NS_{Se\text{-}DAD}$

(2) Response stage. Other hosts (represent by host B) conduct the following.

Step 2: Receive NSSe-DAD and extract "Target address," "Src IP," and "Options" fields;

Step 3: If the address pool is not empty, then host B takes out an address note as IP_Y (go to Step 4). If no address is found in the address pool, then the process ends.

Step 4: Compute $h(IP_Y)$

If $h(IP_Y)$ ==Target address ($NS_{Se\text{-}DAD}$), then take IP_Y as the key, and decrypt the IP and MAC of host A:

$IP_A = D_{IP_Y}(Src\_IP(NS_{Se-DAD}))$,

$MAC_A = D_{IP_Y}(Src\_MAC(NS_{Se-DAD}))$.

Then, host B sends $NA_{Se\text{-}DAD}$ to reply to host A, and each field assignment is shown in Fig. 4.

Ethernet header	Dest MAC	MAC_A
	Src MAC	$Rigth(E_{IP_Y}(MAC_B),48))$
	Type	0x86dd
IPv6 header	Src IP	$E_{IP_Y}(IP_Y)$
	Dest IP	IP_A
	Next header	0x3a
ICMPv6	RSO(only for NA)	O=1
	Type	201
	Target address	$E_{IP_Y}(IP_Y)$
	Options	$E_{IP_Y}(MAC_B)$

Fig. 4. Fields Assignment of $NA_{Se\text{-}DAD}$

- If $H(IP_Y) \neq Target\ address(NS_{Se-DAD})$, then return to Step 3.

(3) Verification process. Host A operates as follows.

Step 5: Within a specified period (1–3 s), host A verifies all the $NA_{Se\text{-}DAD}$ received and tests whether the "Target address" field and IP_X satisfy the equation.

- If Target address ($NA_{Se\text{-}DAD}$) = $E_{IP_X}(IP_X)$, then IP_X is a conflict with the reply host.
- If Target address ($NA_{Se\text{-}DAD}$) $\neq E_{IP_X}(IP_X)$, then no conflict exists; thus, $NA_{Se\text{-}DAD}$ is discarded.

The Se-DAD workflow is shown in Fig. 5.

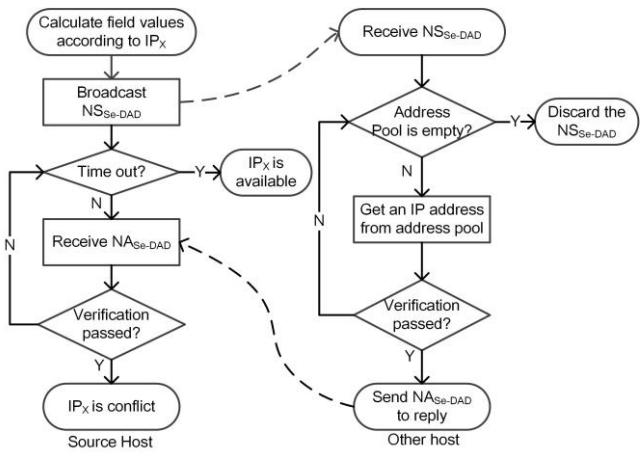

Fig. 5. Flowchart of Se-DAD

B. Example of Se-DAD

TABLE I
BASIC INFORMATION OF HOSTS

Host	IP	MAC	Hash(IP)
A	1::5:A	0800-270c-0001	1298591506654c57623885a782f31ed9
B	1::5:B	0800-270c-0002	baed6fb66cbedfdca5d0a910d315bf89
C	1::5:C	0800-270c-0003	e77b56dc72bf9e34eba732592a5a6dd0

An instance is used to show the workflow of Se-DAD, and the basic details of hosts A, B, and C are presented in Table I. Assuming that host A will use the new address $IP_X = 1::5:B$, then host A calculates its hash value as follows:

$$Target\ address(NS_{Se-DAD}) = h(1::5:B) = \text{baed6fb66cbedfdca5d0a910d315bf89}$$

Herein, the hash algorithm is MD5, which generates a message digest with 128 bits. Using IP_X as the key, host A calculates the "Src_IP," "Src_MAC," and other fields of NS_{Se-DAD}.

$$Src\_IP = E_{1::5:B}(1::5:A) = \text{8d0d12cfef09f97f4e4b67d07c1cab13}$$

$$Src\_MAC = Right(E_{1::5:B}(0800-270c-0001), 48) = baa3d17d8443$$

$$Dest\_MAC = 3333-0000-0001$$

$$Dest\_IP = FF02::1$$

$$Options = E_{1::5:B}(0800-270c-0001) = \text{ef0f97d62b1622e4da0abaa3d17d8443}.$$

Then, host A broadcasts NS_{Se-DAD}, and the details are shown in Fig. 6.

Hosts B and C will receive the NS_{Se-DAD}. Host C obtains 1::5:C from its address pool, and

$$\text{hash}(1::5:C) = \text{e77b56dc72bf9e34eba732592a5a6dd0}$$

is calculated.

$\text{hash}(1::5:C)$ is not equal to the Target address field of NS_{Se-DAD}, and no address is available in the address pool. Then, C discards the NS_{Se-DAD}. For host B, 1::5:B is obtained from its address pool, as calculated in the following:

$$\text{hash}(1::5:B) = \text{baed6fb66cbedfdca5d0a910d315bf89}.$$

Address "1::5:B" is found to be equal to the "Target address" field of NS_{Se-DAD}. Then, 1::5:B is used as the key to decrypt the IP and MAC of host A.

$$IP_A = D_{1::5:B}(\text{8d0d12cfef09f97f4e4b67d07c1cab13}) = 1::5:A$$

$$MAC_A = D_{1::5:B}(\text{ef0f97d62b1622e4da0abaa3d17d8443}) = 0800-270c-0001$$

.

Host B also uses 1::5:B as the key encrypt of its IP and MAC:

$$E_{1::5:B}(1::5:B) = \text{d1e9df6c6c0a428d32f736e7b7e0781f}$$

$$E_{1::5:B}(0800-270c-0002) = \text{ef0f97d62b1622e4cf7dd00205846d99}$$

Then, NA_{Se-DAD} is sent as a reply, and each field assignment is as follows:

$$Src\_IP = E_{1::5:B}(1::5:B) = \text{d1e9df6c6c0a428d32f736e7b7e0781f}$$

$$Src\_MAC = \text{Right}(E_{1::5:B}(0800-270c-0002), 48)$$

$$Dest\ IP = 1::5:A$$

$$Dest\ MAC = 0800-270c-0001$$

$$Target\ address = E_{1::5:B}(1::5:B) = \text{d1e9df6c6c0a428d32f736e7b7e0781f}$$

$$Options = E_{1::5:B}(0800-270c-0002) = \text{ef0f97d62b1622e4cf7dd00205846d99}.$$

Host A found the Target address field of $NA_{Se-DAD} = E_{IP_X}(IP_X)$ after receiving the NA_{Se-DAD} and then realized that 1::5:B is a conflict. The detail of the Se-DAD process is shown in Fig. 6.

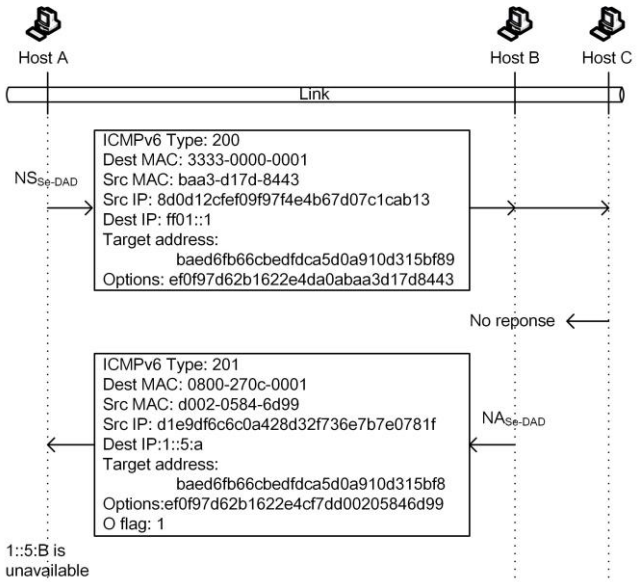

Fig. 6. An example of Se-DAD

IV. EXPERIMENTS AND ANALYSIS

A. Experiments

A series of comparative experiments are conducted to verify the effectiveness of Se-DAD. The testing platform specifications are as follows: Linux Ubuntu 18.04, Intel i9 10900x CPU, 32 GB Corsair DDR4 memory. Network topology generation software is Mininet, the programming language is Python 3.80, and switch software is Open vSwitch. The topology contains eight nodes, where node A is the main node, and periodically performs DAD. Node B is used to generate background traffic, and the distribution of traffic loads is shown in Fig. 7. Node C is a malicious node that attacks the DAD process of node A. Other nodes are participating nodes, and their behavior follows the IPv6 protocol (Table II shows the details of the network environment).

The experiment is divided into three scenarios to simulate the DAD in NDP, SEND, and Se-DAD. The experimental statistics include address configuration failure and packet loss rates as well as CPU and memory overhead. The experimental results are shown in Figs. 8–11.

Scenario 1: DAD in NDP. Host C responds by forging NA according to the target field of the NS.

Scenario 2: In Se-DAD, host C initiates NS, which is the same as host A, and sends it out to deceive host A that an address conflict exists.

Scenario 3: In SEND, host C sends several false responses and then requires host A to verify, thus consuming host resources.

If a DAD process P is performed *m* times in the presence of DoS attacks, and all *m* times have failed, then DoS is fully functional in P. If DAD is successfully performed *m* times, that is, the failure rate of P is 0, then P is immune to DoS attacks. Thus, the DAD failure rate (also called address configuration failure rate) can be used to measure P. The calculation of failure rates is shown in Formula (1).

$$\text{DAD failure rate} = 1 - \frac{\text{DAD success times}}{\text{Total DAD times}} \quad (1)$$

TABLE II
NETWORK ENVIRONMENT

Network type	Bandwidths	Normal node	Attack node	Switch
LAN	100M	7	1	1

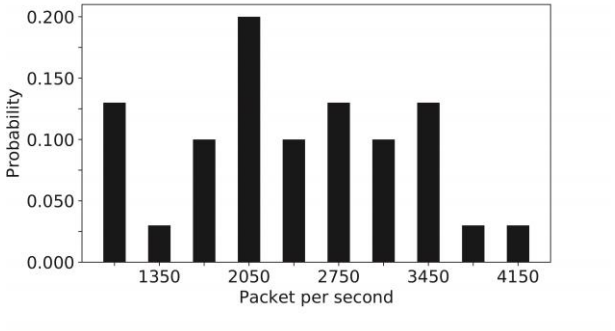

Fig. 7. Probability distribution of traffic

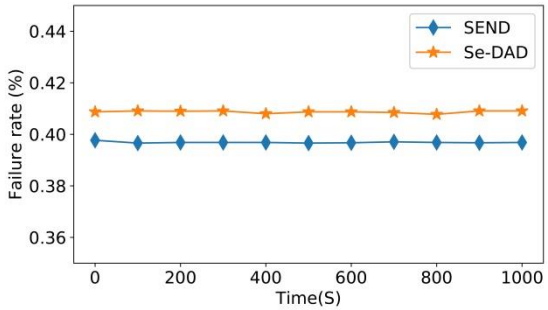

Fig. 8. Address collision rate

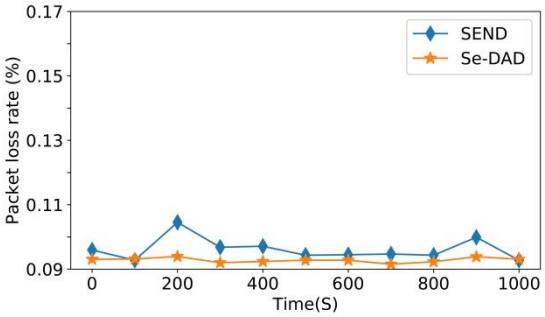

Fig. 9. Packet loss rate

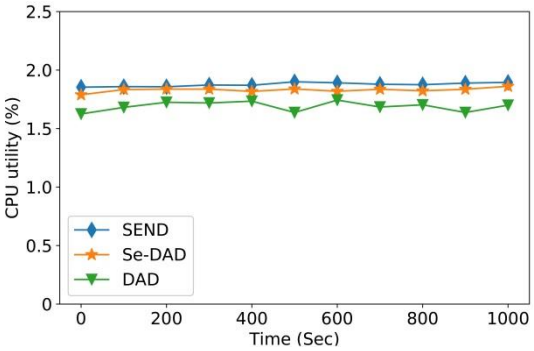

Fig. 10. CPU overhead comparison

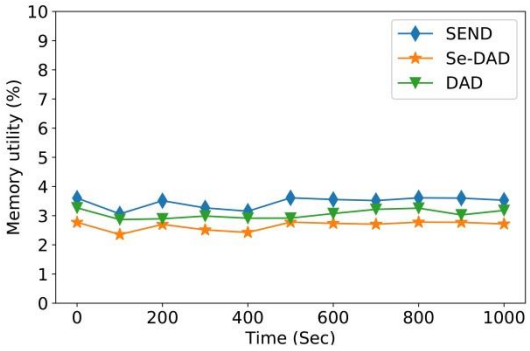

Fig. 11. Memory overhead comparison

Fig. 8 shows that the address configuration failure rates of SEND and Se-DAD are similar, but that of Se-DAD is slightly high. Attack nodes in SEND cannot provide the auxiliary parameters of CGA. Therefore, SEND cannot forge a reply that meets the CGA verification.

Fig. 9 shows that the attacker can send numerous forged DAD replies to consume the resources of the victim host, thus depleting its resources to process packets, which partly leads to missing replies of DAD. Therefore, the address configuration failure rate is lower than the background address conflict rate. The detection address and the MAC of the source host are encrypted in Se-DAD, and the forged replies of attackers cannot be forwarded to the source host. Thus, its DAD failure rate is proximal to the background conflict rate. The Se-DAD fail rate is higher than SEND. However, this finding does not indicate that SEND is effective in experiments because some real address conflicts occur, and SEND is not realized due to the packet loss.

Fig. 10 shows the CPU overhead in three scenarios. DAD in NDP does not require encryption or hash calculation; thus,

its CPU overhead is the lowest and is most vulnerable to DoS attacks. SEND needs numerous centralized calculations when generating CGA addresses, including hash, encryption, and digital signature; thus, the CPU overhead is the highest. Se-DAD only needs encryption calculation when constructing NS or NA messages, and the decryption operation is only required in special cases. Therefore, the CPU overhead is lower than SEND, between SEND and DAD.

Considering memory overhead, Fig. 11 shows that SEND will receive numerous attack messages because it exposes its MAC address and the target address field in DAD. Thus, storing and verifying these messages will consume a considerable amount of storage space. NDP also faces the same problem but does not require complex verification. Therefore, the memory overhead is slightly lower than SEND. The address information of the initiating host is hidden in Se-DAD. Thus, Se-DAD will not receive a large number of forged NA. Therefore, its memory overhead is the lowest.

B. Security analysis

Se-DAD mainly faces two threats: address conflict and DoS attack [26].

Address conflict

Herein, assuming that the target address length is L, the LAN contains N nodes, and each node has M IPv6 addresses. Then, the total number of addresses in the LAN is N × M. Theoretically, the probability that these addresses are completely different is:

$$\begin{aligned} P_{Not\ conflict} &= (\frac{2^L-1}{2^L})\times(\frac{2^L-1}{2^L})\times\cdots\times(\frac{2^L-N\times M+1}{2^L}) \\ &= (1-\frac{1}{2^L})\times(1-\frac{2}{2^L})\times\cdots\times(1-\frac{N\times M-1}{2^L}) \\ &= \prod_{i=1}^{N\times M}(1-\frac{i}{2^L}) \end{aligned}$$

According to the birthday paradox, the probability of conflict in these addresses is:

$$P_{conflict} = 1-\prod_{i=1}^{N\times M}(1-\frac{i}{2^L})$$

The length of IPv6 is 128 bits, and the number of LAN nodes must not exceed 500. Considering the worst case, assume that the number of LAN nodes is 512, and each node has 1024 addresses. The network prefix is assumed to be 64 bits. Therefore, the IID length is 64; that is, L is 64, N is 512, and M is 1024. Then, the address collision probability is:

$$P_{conflict} = 1-\prod_{i=1}^{512\times1024}(1-\frac{i}{2^{64}})$$

$$P_{conflict} \approx 1-e^{-\frac{1}{2^{35}}}$$

where $P_{conflict}$ is a minimum. Therefore, the probability of address collision can be ignored in Se-DAD.

DoS attack

Attacker C does not know because the MAC address of A is encrypted. Assume that the bandwidth of the LAN is 10 Gbps: if C forges a large number of NA (the destination MAC field of these NA is random) and attempts to attack A, then the limit number of NA that C can send in 3 s is computed as shown below because the size of NA is 78 bytes.

$$C=\left\lfloor \frac{3\times10Gb}{78Byte} \right\rfloor = 1.43\times2^{25}$$

The probability that these NA can successfully reach A is:

$$P_{conflict} = 1-\prod_{i=1}^{1.43\times2^{25}}(1-\frac{i}{2^{48}})$$

$$P_{conflict} \approx 1-e^{-\frac{1}{2^{23}}} = 1.19\times10^{-7}$$

The attack of C hardly poses a threat to the DAD process of A due to such a low probability.

C. Comparison

Se-DAD is compared with several typical studies considering encryption technology, additional facilities, traffic monitoring, communication overhead, and database support. The comparison results are shown in Table III.

If both sides use encryption technology, then the protocol performance will be affected to a certain extent, which is observed in methods [19] and [29]. Reference [28] must include an additional server in the network and ensure that the server is always secure; however, an additional server increases the cost of deployment. In reference [29], the security server needs periodic broadcasting to collect <IP, MAC> mappings of all hosts in the LAN, which increases the communication overhead. The methods adopted in references [29], [28], and [27] must create a mirror port on the switch such that all network traffic can be monitored to realize message filtering. These methods must be supported by the switch and also need database support to record the address mappings. The deployment costs are remarkably high. Compared with these solutions, Se-DAD does not need to monitor all network traffic and add security servers or database support. Thus, Se-DAD is a lightweight security solution with low implementation costs.

V. Conclusion

As an important part of ARP and NDP, the security of the DAD process has not been sufficiently studied. In DAD, the destination address of detection and the MAC of the source host are all public. Thus, the attack node not only knows the detection address but also which host is performing the detection. This scenario is convenient for the attack node. The proposed method of Se-DAD overcomes the two abovementioned problems. The destination of detection and the MAC of the source host are closed in Se-DAD. Thus, an anonymous DAD is present, thereby effectively preventing DoS. Se-DAD also has some shortcomings.

TABLE III
COMPARISON OF DAD-H WITH OTHER SECURITY MECHANISMS

	Cryptography used	Performance degradation	Additional facility	Traffic monitor	Data Base support
Se-DAD	Yes	Low	Null	No	No
SEND[19]	Yes	Middle	Null	No	No
FCFS-SAVI[27]	No	Low	Switch/Router	Yes	Need
NDPmon[28]	No	Low	Secure server	Yes	Need
Rehman[29]	Yes	Middle	Secure server	Yes	Need

Se-DAD needs redeployment of the protocol stack because of the use of two new packet formats. Moreover, the communication and computation costs of Se-DAD are higher than that of the traditional DAD process because of the encryption overhead. However, the security of Se-DAD is higher than the traditional DAD process.

REFERENCES

[1] Fall, Kevin R., and W. Richard Stevens, "TCP/IP illustrated, volume 1: The protocols". Addison-Wesley, 2011.

[2] Plummer, David, "RFC0826: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48. bit Ethernet address for transmission on Ethernet hardware." (1982). http://www.rfc-editor.org/pdfrfc/rfc826.txt.pdf.

[3] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)." Standards Track, http://www.ietf.org/rfc/rfc4861. txt (2007). http://www.rfc-editor.org/pdfrfc/rfc4861.txt.pdf.

[4] Thomson, S., T. Narten, and T. Jinmei, "RFC 4862: Ipv6 stateless address autoconfiguration, Sept. 2007." Status: Draft Standard. http://www.rfc-editor.org/pdfrfc/rfc4862.txt.pdf.

[5] Kempf, J., and Nordmark, E./Nikander, P, "IPv6 Neighbor Discovery (ND) Trust Models and Threats." Work in Progress (2004).

[6] T. Narten, and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6." Internet Society (2007).

[7] Cheshire, Stuart, "IPv4 Address conflict detection." (2008). https://tools.ietf.org/html/rfc5227.

[8] Ahmed, Msaas, R. Hassan, and N. E. Othman, "IPv6 Neighbor Discovery Protocol Specifications, Threats and Countermeasures: A Survey." IEEE Access PP(2017):1-1.

[9] Tseng, Chien-Chao, Wong Yung-Chang, Yen Li-Hsing, and Hsu Kai-Cheng, "Proactive DAD: A fast address-acquisition strategy for mobile ipv6 networks." Internet Computing, IEEE 10.6 (2006): 50-55.

[10] Soliman, Hesham, Ludovic Bellier, and Karim El Malki, "Hierarchical mobile IPv6 mobility management (HMIPv6)." (2005). https://tools.ietf.org/html/rfc4140.

[11] Moore, Nick, "RFC 4429: Optimistic duplicate address detection (DAD) for IPv6." (2006). http://tools.ietf.org/html/rfc4429.

[12] Vidhyapriya, R., and D. Vanathi, "Energy Efficient Adaptive Multipath Routing for Wireless Sensor Networks." IAENG International Journal of Computer Science, vol.34, no.1, pp56-64, 2007.

[13] T. R.Reshmi, and K. Murugan, "Filter-based address autoconfiguration protocol (FAACP) for duplicate address detection and recovery in MANETs." Computing 97.3 (2015): 309-331.

[14] Garc á Villalba, and Luis Javier, "Distributed dynamic host configuration protocol (D2HCP)." Sensors 11.4 (2011): 4438-4461.

[15] Heuse, and Marc, "THC IPv6 attack tool kit." http://www.aldeid.com/wiki/THC-IPv6-Attack-Toolkit.

[16] Garc á-Martínez, Alberto, and Marcelo Bagnulo. "An Integrated Approach to Prevent Address Spoofing in IPv6 Links." Communication letters, IEEE, 2012, 16(11)

[17] Seo, Karen, and Stephen Kent, "Security architecture for the internet protocol." (2005). http://tools.ietf.org/html/rfc4301

[18] Y. Maleh, and A. Ezzati, "Lightweight Intrusion Detection Scheme for Wireless Sensor Networks", IAENG International Journal of Computer Science, vol.42, no.4, pp347-354, 2015.

[19] Arkko, Jari, J. Kempf, B. Zill, and P. Nikander, "Secure neighbor discovery (SEND)". RFC 3971, March, 2005.

[20] Aura, Tuomas. "RFC 3972: Cryptographically generated addresses (CGA)." (2005).

[21] Reshmi, T. R., et al. "Light Weight Cryptographic Address Generation(LW-CGA) Using System State Entropy Gathering for IPv6 Based MANETs." China Communications (2017).

[22] Lu, Yiqin, Meng Wang, and Pengsen Huang, "An SDN-based authentication mechanism for securing neighbor discovery protocol in IPv6." Security and Communication Networks 2017 (2017).

[23] Netze, Heise. "A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience." Heise Zeitschriften Verlag (2008).

[24] J Bi，G Yao，J Halpern，and E Levyabegnoli, "SAVI for Mixed Address Assignment Methods Scenario." (2011). https://tools.ietf.org/html/draft-bi-savi-mix-04.

[25] J.Wu, J. Bi, X. Li, G. Ren., and X. Xu, "A source address validation architecture (SAVA) testbed and deployment experience". Available: http://tools.ietf.org/html/rfc5210.

[26] Wang, Xiaoyun, and Hongbo Yu, "How to break MD5 and other hash functions." Advances in Cryptology–EUROCRYPT 2005. Springer Berlin Heidelberg, 2005. 19-35.

[27] Bagnulo, M , and A. Garcia-Martinez, "SAVI: The IETF standard in address validation." IEEE Communications Magazine, vol.51, no.4, pp.66-73, (2013).

[28] Frederic Beck, Thibault Cholez, Olivier Festor and Isabelle Chrisment, "Monitoring the neighbor discovery protocol." 2007 International Multi-Conference on Computing in the Global Information Technology (ICCGI'07). IEEE, 2007.

[29] Rehman, S. U. , and S. Manickam, "Integrated Framework to Detect and Mitigate Denial of Service (DoS) Attacks on Duplicate Address Detection Process in IPv6 Link Local Communication." International Journal of Security and its Applications 9.11(2015):77-86.