Improved Mechanism to Prevent Denial of Service Attack in IPv6 Duplicate Address Detection Process

Shafiq Ul Rehman, Selvakumar Manickam
National Advanced IPv6 Centre (NAv6)
University of Science Malaysia
Penang, Malaysia

***Abstract*—From the days of ARPANET, with slightly over two hundred connected hosts involving five organizations to a massive global, always-on network connecting hosts in the billions, the Internet has become as important as the need for electricity and water. Internet Protocol version 4 (IPv4) could not sustain the growth of the Internet. In ensuring the growth is not stunted, a new protocol, i.e. Internet Protocol version 6 (IPv6) was introduced that resolves the addressing issue IPv4 had. In addition, IPv6 was also laden with new features and capabilities. One of them being address auto-configuration. This feature allows hosts to self-configure without the need for additional services. Nevertheless, the design of IPv6 has led to several security shortcomings. Duplicate Address Detection (DAD) process required for auto-configuration is prone to Denial of Service (DoS) attack in which hosts are unable to configure themselves to join the network. Various mechanisms, SeND, SSAS, and the most recent being Trust-ND, have been introduced to address this issue. Although these mechanisms were able to circumvent DoS attack on DAD process, they have introduced various side effects, i.e. complexities and degradation of performance. This paper reviews the shortcomings of these mechanism and proposes a new mechanism, Secure-DAD, that addresses them. The performance comparison between Trust-ND and Secure-ND also showed that Secure-DAD is more promising with improvement in terms of processing time reduction of 45.1% compared to Trust-ND while preventing DoS attack in IPv6 DAD process.**

Keywords—Secure-DAD; Duplicate Address Detection; Denial of Service Attack; IPv6 Security; Address auto-configuration

I. Introduction

Address auto-configuration [1] is the main feature of IPv6 Internet protocol [2]. This mechanism allows IPv6 enabled devices to configure IP addresses automatically without the need for addition services providers such as; DHCPv6, thus provides flexibility in address configuration. However, self-generated IP address has to be unique in order to prevent the conflict of IP address among hosts in IPv6 network [1]. Although, it can be argued that IP conflict is extremely remote due to the immensity of the address space, this will not be the case in the coming years due to the growth in mobile device and new drivers such as; Internet of Things (IoT) [3, 4] and Cloud [4]. Therefore, there is a mechanism known as Duplicate Address Detection (DAD) process [1, 5] to verify the uniqueness of self-generated IP address. In IPv6 network every host must perform DAD process in order to configure a unique valid IP address.

In IPv6 network, for Neighbor Discovery [6, 7] IPv6 hosts use two types of ICMPv6 [8] messages also known as neighbor discovery messages i.e. Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages. Neighbor solicitation (NS) message is used to send a query to neighboring hosts on same link and in response to that query existing hosts use Neighbor Advertisement (NA) message. While performing DAD process, new hosts send NS message to verify whether the self-generated IP address is already obtained by any existing host on a same link. If any existing host has configured the same IP address then that host replies back with NA message that the self-generated IP address is already configured.

During standard DAD process IPv6 hosts are considered trustworthy. Therefore, IPv6 hosts rely on the information being exchanged on a same link. Thus, malicious host can exploit the DAD process by disrupting the communication during address verification between the hosts. Research [5, 9, 10] have shown that DAD process is vulnerable to denial of service (DoS) attacks. During DoS-on-DAD attack, a malicious host tries to prevent the victim host to configure a unique valid IP address by claiming the existence of self-generated IP address via sending fake NA messages in reply to its NS messages. Hence, victim host is unable to configure its unique IP address. Thus, victim host cannot communicate on a same link due to DAD process failure.

Considering this vulnerability with DAD process, some of the security mechanisms have been proposed such as; SeND [10], SSAS [11], and Trust-ND [12]. SeND mechanism was suggested to solve the security concerns of ND messages. However, this mechanism is not trivial due to its design which possess heavy computation and complexity issues during ND message processing [11, 12]. In order to address this issue, Simple Secure Addressing Scheme (SSAS) was proposed. This mechanism to some extent addressed the issue of complexity by introducing a new scheme compared to the SeND mechanism. However, SSAS still requires significant amount of time to process the ND messages [12]. Recently, Trust-ND has been proposed that claims to be the lightweight mechanism compared to SeND and SSAS schemes. However, the issue with Trust-ND mechanism is that it is built on SHA-1 hashing algorithm which has been found vulnerable to hash collision attacks [13, 14]. Thus, due to its design it can induce DoS attack during DAD process.

This paper introduces a new mechanism know as Secure-DAD which is faster in terms of processing time and effective enough to prevent DoS attack during DAD process. The rest of the paper is organized as follows: Section 2 will present an overview of DAD process and its security issues. Section 3 will discuss the related work. Section 4 explains the design and implementation of Secure-DAD mechanism. Section 5 will present a Test-bed setup environment. Section 6 will discuss the evaluation procedure of Secure-DAD mechanism. Section 7 provides the experimental results and discussion. And finally, Section 8 will present the conclusion and future work.

II. IPv6 DAD Process and its Security Issues

In order to be able to communicate on the same network, IPv6 host(s) has to verify the uniqueness of its self-generated IP address which is the final stage of address auto-configuration [1, 5, 10]. This verification procedure is being executed through Duplicate Address Detection process. New host performs DAD process by sending Neighbor Solicitation (NS) message to all node multicast address (FF02::1) so that existing hosts can receives NS message. NS message carry the tentative IP address that new host has generated and would like to assign it as a preferred address. If that tentative address is configured already by any other host in the network then that particular host will reply back with a Neighbor Advertisement (NA) message. Hence, new host repeats the DAD process again, in case if there is no response to its generated NS message; then it will consider the generated IP address is unique [1, 5, 15]. Thus, a new host can use it as a preferred IP address. Figure 1 describes the DAD process in IPv6 network.

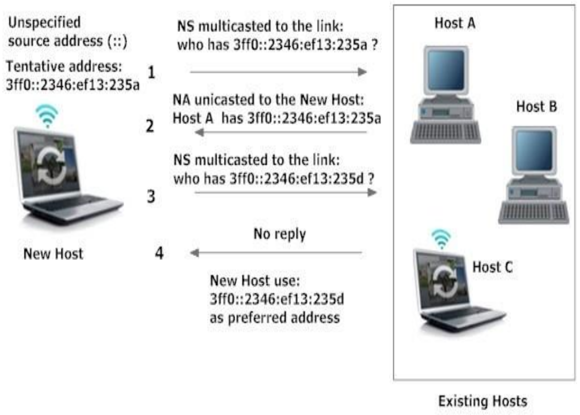

Unspecified
source address (::)
Tentative address:
3ff0::2346:ef13:235a
NS multicasted to the link:
who has 3ff0::2346:ef13:235a ?
1
NA unicasted to the New Host:
Host A has 3ff0::2346:ef13:235a
2
NS multicasted to the link:
who has 3ff0::2346:ef13:235d ?
3
No reply
4
New Host use:
3ff0::2346:ef13:235d
as preferred address
New Host
Host A
Host B
Host C
Existing Hosts

Fig. 1. Duplicate address detection process [15]

In IPv6 link local communication any existing IPv6 host can participate in DAD process. Since, ND messages such as: NS/NA messages are insecure by design. Thus, an attacker can easily exploit the DAD process by fabricating NA message and reply it to every NS message received. This can disrupt DAD process and cause DAD failure. Hence, new host will not be able to obtain a valid IP address. As a result, new host cannot communicate in IPv6 link local network. This attempt of DoS attack is known as DoS-on-DAD attack. Figure 2 illustrates the DoS attack on DAD process in IPv6 network.

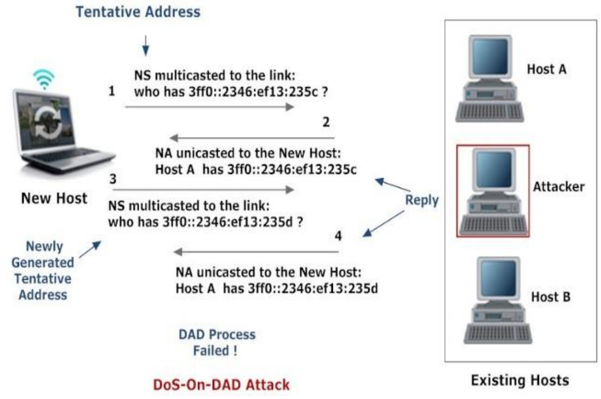

Tentative Address
NS multicasted to the link:
who has 3ff0::2346:ef13:235c ?
1
2
NA unicasted to the New Host:
Host A has 3ff0::2346:ef13:235c
3
NS multicasted to the link:
who has 3ff0::2346:ef13:235d ?
4
NA unicasted to the New Host:
Host A has 3ff0::2346:ef13:235d
Reply
New Host
Newly
Generated
Tentative
Address
DAD Process
Failed !
DoS-On-DAD Attack
Host A
Attacker
Host B
Existing Hosts

Fig. 2. Denial of service attack on DAD process [15]

III. Related Work

Considering the security concern with IPv6 DAD process, existing mechanisms such as: SeND, SSAS, and Trust-ND have been proposed to address this problem in IPv6 link local network. However, these mechanisms have some issues due to their designed mechanism which restrains their implementation on DAD process in IPv6 network. This section describes these issues and limitations with existing mechanisms as follows:

A. Secure Neighbor Discovery (SeND)

SeND was introduced to address the security issues related with NDP messages. It introduces four NDP options; CGA option, Nonce option, Timestamp option, and RSA signature option as well as two ICMPv6 messages; Certificate Path Solicitation (CPS) and Certificate Path Advertisement (CPA) as specified in RFC 3971 [10]. Although, SeND was able to prevent malicious attacks on IPv6 neighbor discovery. However, researches have proven [11, 12] that SeND has a drawback like high computation to generate the options especially the CGA option and RSA signature. Thus, it consumes higher computation time. Based on the previous research, SeND mechanism adds significant processing time and it takes 367.59 milliseconds to perform the message verification operation [12]. Hence, if SeND is implemented, its processes i.e. authorization and certificate validation function can add delay and increase complexity during DAD process as highlighted by the researchers [7]. Thus, any malicious host can exploit this mechanism and can cause DoS attack against the SeND mechanism itself by engaging the victim host in message verification processing.

B. Simple Secure Addressing Scheme (SSAS)

In order to address the issues with SeND mechanism, another mechanism known as Simple Secure Addressing Scheme (SSAS) was proposed which is considered as an improved version of SeND mechanism on securing ND messages during DAD process in IPv6 network [11]. SSAS introduces alternative addressing scheme by employing elliptic curve cryptography (ECC) algorithm rather than RSA as used by SeND mechanism for address configuration process. In other words, SSAS mechanism is lightweight version of SeND mechanism. In order to protect ND message from spoofing attacks SSAS uses Signature and Timestamp options which are appended to ND messages during DAD process. Although, SSAS has reduced some complexity and resulted in decreased message processing time compared to SeND mechanism. Since this method relies on signature and key exchange processes, hence the complexity issue still exists [12]. Based on the research conducted by Praptodiyono et al. in 2015 [12], SSAS mechanism takes 223.1 milliseconds to generate an interface identifier which is a considerable amount of processing time. Thus, due to its complexity issue, SSAS mechanism can also induce DoS attack on DAD process by delaying the message verification process during address configuration in IPv6 link local network.

C. Trust-ND

Recently, researchers have claimed a lightweight mechanism for DAD process in IPv6 network known as Trust-ND [12]. The main focus of this mechanism has been the complexity of the ND message processing. Trust-ND has significantly reduced the processing time of ND messages during DAD process compared to existing mechanisms such as: SeND and SSAS in IPv6 network. In Trust-ND, message authentication is a result of SHA-1 operation as a message integrity check. Thus, Trust-ND mechanism relies on SHA-1 hash function to satisfy the security requirements. Although, the authors claims that Trust-ND is a lightweight security mechanism for IPv6 DAD process. However, researches [13, 14] have shown that SHA-1 and MD5 hash functions are susceptible to hash collision attacks. Since, Trust-ND's security is based on SHA-1 hash function therefore, any malicious host can exploit this weakness to generate hash collision attack against this mechanism that can cause DoS attack on DAD process in IPv6 network. Thus, due to this security vulnerability Trust-ND might not be a suitable mechanism for IPv6 DAD process.

Due to the constraints possessed by existing security mechanisms as aforementioned. The implementation of the security mechanisms for IPv6 DAD process has been limited. As a result, IPv6 DAD process is still unprotected and prone to be exploited by malicious hosts. Therefore, we proposed a new mechanism known as Secure-DAD to secure ND messages during DAD process. Due to its design, Secure-DAD mechanism can protect NS/NA messages from any kind of exploitation such as: spoofing attack, man-in-the-middle attack (MITM), replay attack or hash collision attacks which are responsible for causing DoS attack during DAD process in IPv6 network. The following Section will explain the design and implementation processes of Secure-DAD mechanism.

IV. Design and Implementation of Secure-DAD Mechanism

In case of IPv6 DAD process, authentication is required to protect NS and NA messages from several types of attacks such as: masquerade, content modification, sequence modification and timing modification which eventually leads to DoS attack [16]. Here, DoS attack relates to the absence of the services i.e. to configure unique IP addresses rather than service unavailability due to flooding attacks. In order to authenticate NS and NA messages, research [17] has recommended using the most appropriate hash function which is resistant to hash collision attacks and can also be faster in computation. Researches [17, 18] have proven that Universal Hashing (UMAC) is efficient algorithm and secure than existing hash functions such as: SHA-1 and MD5. Thus, the most suitable and available hash function algorithm has been selected which can satisfy this security requirement. UMAC can provide message integrity to prevent any tempering with NS and NA messages content as the security requirement. Secure-DAD mechanism is built on UMAC hash function algorithm to ensure that the proposed mechanism is reliable and effective enough to secure a DAD process in IPv6 network.

Secure-DAD mechanism introduces a concept of Secure-tag option which will be appended to each ND message i.e. NS and NA messages exchange between the hosts during DAD process in IPv6 network. This Secure-tag comprises of message authentication code (MAC) to distinguish the valid messages from the fake ones. After the addition of the Secure-tag, these ND messages i.e. NS and NA messages are named as Secured NS and Secured NA messages. Figure 3 and Figure 4 presents the Secured NS and Secured NA messages format respectively.

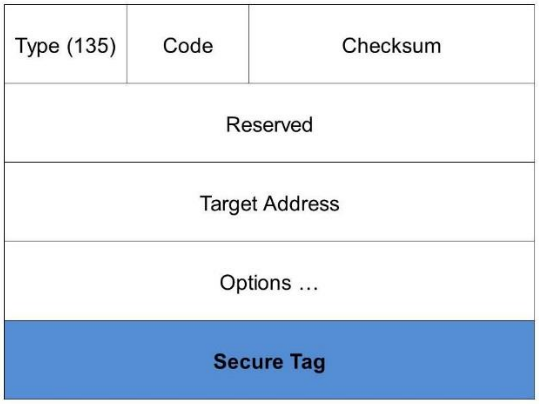


Fig. 3. Secured NS message format

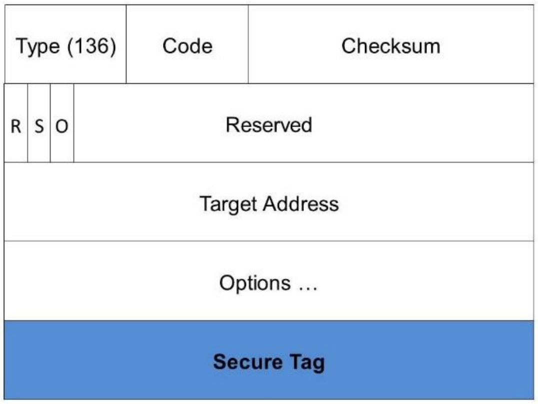


Fig. 4. Secured NA message format

In Secure-DAD mechanism, when a new host performs DAD process it will generate a Secure-tag, appends onto NS message and sends it to multicast address group i.e. FF02::1. Upon receiving NS message existing host(s) will match this Secure-tag option with its self-generated Secure-tag. After the computation process, if these Secure-tags match, then it will perform DAD process and can reply via Secured NA message i.e. NA message appended with Secure-tag. Similarly, upon receiving the NA message, new host performs the same procedure i.e. matching of Secure-tags, else if no match of Secure-tags is found then new host will simply discard the received NA message. Hence, in this manner, new host can perform DAD process successfully. Thus, new host can configure a unique IPv6 link local address. Figure 5 illustrates the Secure-tag generation and verification processes between the hosts in IPv6 link local network.

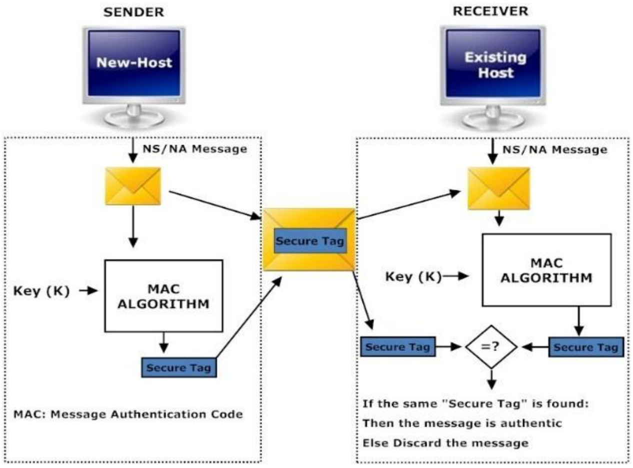


Fig. 5. Secure-tag generation and verification process

V. Test-Bed Setup Environment

In order to evaluate the performance of secure-DAD mechanism in terms of processing time and effectiveness a Test-bed setup has been deployed at NAv6 research Centre in University Science Malaysia (USM). Figure 6 shows the topology of the Test-bed setup environment.

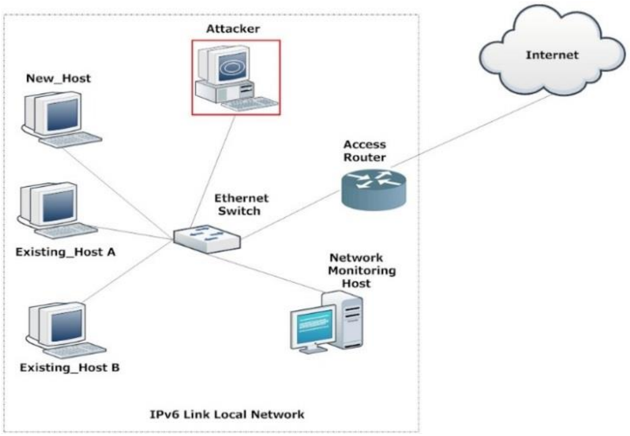


Fig. 6. Test-bed setup environment

The attack could be coming from any type of host's i.e. Windows, Linux etc., since we are using Kali [19] for that purpose attacker host is Linux host. A packet capturing tool known as Wireshark [20] has been used to capture and analyse the network traffic. Moreover, the hardware and software specifications have been selected based on the availability and support for IPv6 environment at NAv6 research Centre to conduct the experiments successfully. The details of the required hardware and software specifications for Test-bed environment setup are presented in Table 1 and Table 2 respectively.

TABLE I. Hardware Requirements for the Experiments

Hardware		Details
Computer Hardware @ per (Host)	CPU	Intel® Core™2Duo CPU E6750 @ 2.66GHZ
	Memory	1 GB Ram
	Network Interface Card	Intel® 82579LM Gigabit1 Ethernet LAN 10/100/1000
	Network Patch cables	Digitus UTP Cat5e
Other Network Devices	Switch	Cisco Catalyst 2960 Fast Ethernet
	Access Router	Cisco Router C7200

TABLE II. Software Requirements for the Experiments

Operating System		Role	Tools
Microsoft Windows	Windows 7 Ultimate 64-bit (version: 6.1.7601.17514)	Network Monitoring Host	Wireshark
		New_Host	-
		Existing_Host A	-
		Existing_Host B	-
Linux Distributions	Kali Linux (version 3.18.0-amd64)	Attacker Host	THC IPv6 Attack Toolkit 2.7

VI. Evaluation of Secure-DAD Mechanism

In order to evaluate the proposed Secure-DAD mechanism, Network security experts have specified a standard criterion known as Information Technology Security Evaluation Criteria (ITSEC) [21]. Therefore, ITSEC has been used to assess the Secure-DAD mechanism. ITSEC presented three metrics for evaluation i.e. Operation, Effectiveness and Functionality [22]. According to ITSEC, any security mechanism that can fulfill these three parameters is considered applicable. Since, Secure-DAD is defined to prevent ND messages from any exploitation which can induce DoS attacks on DAD process by Secure-tag option. Hence, the performance of the Secure-DAD mechanism was evaluated based on these recommended criteria as described in the following Section.

VII. Experimental Results and Discussion

Secure-DAD mechanism is implemented based on the Test-bed environment as presented in Figure 6. In order to make sure that the proposed Secure-DAD mechanism works properly and satisfies the security requirements, the implementation was done in two scenarios. The reason behind that was to measure the performance of Secure-DAD in terms of processing time in first scenario and also, the effectiveness, and functionality of the mechanism in second scenario.

A. Experiments in First Scenario

In first scenario experiments were conducted to examine the performance of Secure-DAD mechanism in terms of processing time. In order to fulfill these requirements, Secure-DAD was performed on Test-bed environment setup to measure the Secured ND messages processing time i.e.

Secured NS and Secured NA messages between the sender and receiver hosts. In addition, same experiments were also conducted for the standard DAD process and Trust-ND mechanism on same Test-bed environment. The purpose of conducting these experiments on standard DAD, Secure-DAD, and Trust-ND were to obtain the results of NS and NA messages processing time between IPv6 hosts during DAD process in IPv6 link local network. These results were then analyzed by comparing the three mechanisms to justify the performance of Secure-DAD mechanism. The obtained results are discussed in the following sub-section.

B. *Results Analysis and Discussion*

This section provides the results analysis and discussion of the operation of Secure-DAD compared against the standard DAD and Trust-ND mechanism. The metric to measure the performance of the Secure-DAD operation along with standard DAD process and Trust-ND mechanism is the processing time of received NS and NA messages at the sender (New_Host) and receiver (Existing_Host) during DAD process respectively. The measurement of Secure-DAD processing time was done by subtracting the end time with the start time of the NS and NA messages verification process at the receiving host. Similarly, the same operation was performed with standard DAD and Trust-ND respectively. It was conducted for each of the NS/NA message for 10 (times) experiment. The comparative results are presented in a graphical form as shown in Figure 7 and Figure 8 for standard DAD process, Secure-DAD and Trust-ND mechanism respectively.

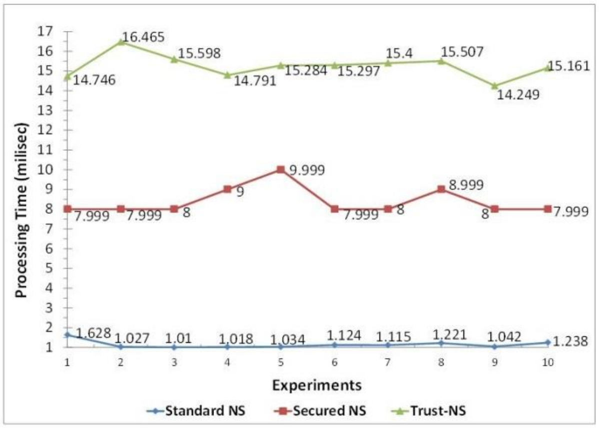


Fig. 7. Comparative NS messages processing time

Figure 7 presents the NS messages processing time at the receiver side i.e. Existing_Host. For each message i.e. Standard NS, Secured NS, and Trust-NS messages experiments were repeated 10 times separately. The purpose for doing this was to find the level of consistency i.e. the average processing time of the NS messages processing time performed for each attempt. Figure 7 also depicts the amount of processing time taken by the three different messages types for each experiment. It shows the level of consistency of the message processing time taken by these message types at the receiver host i.e. Existing_Host.

Table 3 presents the average processing time of the 10 experiments conducted on each message type, as well as the overhead introduced in each Secured NS and Trust-NS messages respectively. The overhead was calculated by putting the standard NS messages average processing time as the baseline. Later, it was compared with Secured NS and Trust-NS messages processing time at the receiver host respectively. Secured NS message processing time is 7.253 milliseconds in average. However, it was also noticed that the Trust-NS message processing time is higher that reaches to 15.250 milliseconds in average. Thus, from the experimental results, it is clear that Secured NS messages consumes less processing time than the Trust-NS messages, which consumes more processing time at the receiver host.

TABLE III. NS MESSAGES PROCESSING TIME AT RECEIVER HOST

Processing Time of NS messages (milliseconds)			
Receiver (Existing_Host)	**Standard NS**	**Secured NS**	**Trust-NS**
Mean	1.146	8.399	15.250
Overhead	Baseline	7.253	14.104

Likewise, sender host i.e. New_Host performs the message verification for all incoming NA messages. The incoming NA message is the response to its NS message sent earlier to Existing_Hosts on a same link to complete the DAD process in IPv6 link local network. Similarly, the sender host i.e. New_Host goes through the same message verification process as performed by the Existing_Host. Therefore, in case of Secured NA message, New_Host verifies the Secure-tag option and its message content. Whereas, in case the incoming message is Trust-NA, it verifies the Trust option and its message content. For standard NA, message processing takes place without the verification of message content. Since standard NA message does not contain any such option to be processed.

Figure 8 depicts the NA messages processing time at the sender side i.e. New_Host. Again for each message type i.e. Standard NA, Secured NA, and Trust-NA messages, individual experiments were conducted 10 times for each mechanism. Figure 8 demonstrates the different processing time for each message types which were carried out 10 times for each experiment. It also presents the level of consistency performed by each message types during the message processing at the sender host i.e. New_Host.

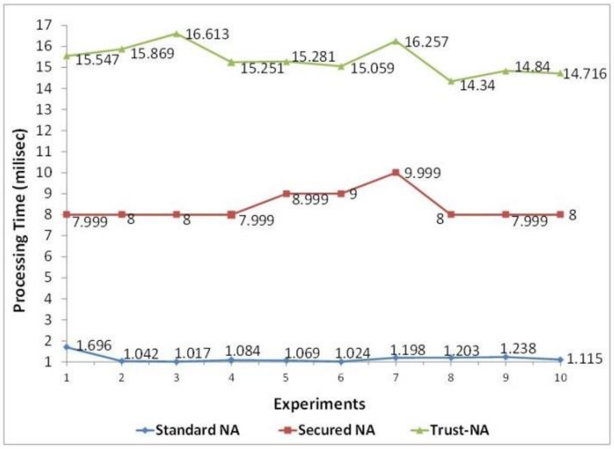


Fig. 8. Comparative NA messages processing time

Table 4 depicts the average processing time consumed by each message types at the sender host i.e. New_Host. Experiments were carried out 10 times on each message types. In addition, to the overhead introduced in average by each Secured NA and Trust-NA messages are also presented. The overhead was estimated by placing the standard NS messages average processing time as a baseline. In this manner, Secured NA and Trust-NA messages processing time were calculated accordingly.

TABLE IV. NA MESSAGES PROCESSING TIME AT RECEIVER HOST

Processing Time of NA messages (milliseconds)			
Sender (New_Host)	**Standard NA**	**Secured NA**	**Trust-NA**
Mean	1.169	8.499	15.377
Overhead	Baseline	7.330	14.208

Table 5 shows the overall processing time differences between the standard DAD, Secure-DAD, and Trust-ND mechanisms. The processing time of ND messages i.e. NS and NA messages between the IPv6 hosts represents the computational efficiency of security mechanism. Therefore, by comparing the processing time of Secure-DAD and Trust-ND mechanisms with the standard DAD as a baseline, effects of these two mechanisms on DAD process in IPv6 network can be distinguished.

TABLE V. OVERALL PROCESSING TIME AT SENDER AND RECEIVER HOSTS

DAD Process	**Processing Time (milliseconds)**		
	Standard DAD	**Secure-DAD**	**Trust-ND**
Sender (New_Host) NS	1.146	8.399	15.250
Receiver (Existing_Host) NA	1.169	8.499	15.377
Total	2.315	16.898	30.627
Overhead	Baseline	14.583	28.312

The overall processing time of standard DAD, Secure-DAD, and Trust-ND mechanisms are 2.315, 16.898, and 30.627 milliseconds respectively. Hence, the total overhead introduced by Secure-DAD mechanism is 14.583 milliseconds in average. Whereas, Trust-ND mechanism is 28.312 milliseconds in average. Thus, the overhead introduced by Secure-DAD is lesser as compared to Trust-ND mechanism.

Table 6 depicts the saved processing time on the implementation of Secure-DAD against Trust-ND mechanism. Secure-DAD is able to save time up to 13.729 times, which means processing time reduction of 45.1% compared to Trust-ND correspondingly for NS and NA messages processing time during address verification process between hosts in IPv6 link local network.

TABLE VI. PROCESSING TIME SAVED BY SECURE-DAD

DAD Process	**Processing Time (milliseconds)**		**Saving Time (milliseconds)**
	Trust-ND	**Secure-DAD**	
Sender (New_Host) NS	15.250	8.399	6.851
Receiver (Existing_Host) NA	15.377	8.499	6.878
Total	30.627	16.898	13.729

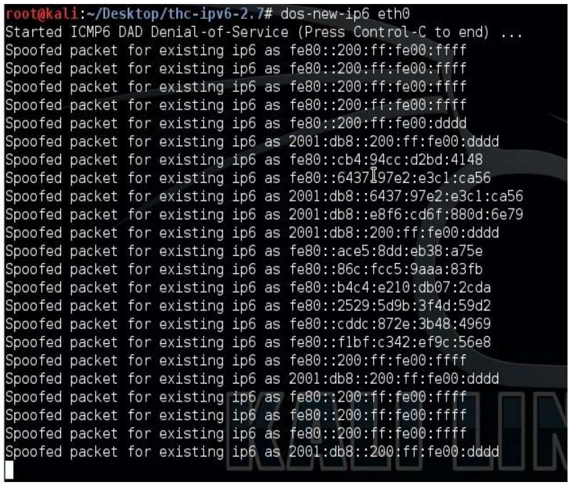
Thus, from the results it is clear that the proposed Secure-DAD mechanism is able to reduce the level of complexity i.e. the processing time of NS and NA messages verification at the hosts during DAD process in IPv6 link local network. This is in contrast to the Trust-ND mechanism and other existing mechanism such as: SeND, SSAS that possess the high level of complexity as stated by the researchers [12].

C. Experiments in Second Scenario

The second scenario was conducted to validate the effectiveness of Secure-DAD under the attacking situation. This scenario was examined to ensure that Secure-DAD mechanism is capable of protecting NS and NA messages from fabricating during DAD process which can eventually causes DoS attack. In order to test the Secure-DAD, the attacking approach was performed by running dos-new-ip6 attack tool [23]. The purpose of carry out denial of service attack was to measure the effectiveness of Secure-DAD mechanism to satisfy the functionality requirement under attack condition this was done by using the dos-new-ip6 attack tool.

- Attack Generation on DAD Process

The main purpose of attacking a New_Host is to cause the host initialization failure. In order to achieve this aim, Attacker host uses dos-new-ip6 tool to generate a NA message to answer whatever tentative address is being generated by the New_Host. This is intended to cause DAD process failure which can deny New_Host to obtain a unique IPv6 address. Figure 9 depicts the DoS-on-DAD attack generation against DAD process in IPv6 network.

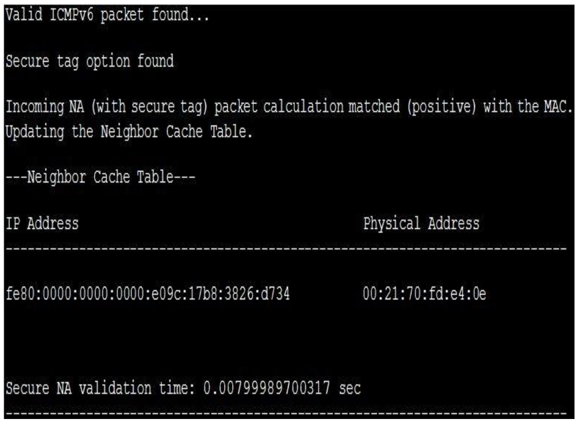


```
root@kali:~/Desktop/thc-ipv6-2.7# dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as fe80::200:ff:fe00:dddd
Spoofed packet for existing ip6 as 2001:db8::200:ff:fe00:dddd
Spoofed packet for existing ip6 as fe80::cb4:94cc:d2bd:4148
Spoofed packet for existing ip6 as fe80::6437:97e2:e3c1:ca56
Spoofed packet for existing ip6 as 2001:db8::6437:97e2:e3c1:ca56
Spoofed packet for existing ip6 as 2001:db8::e8f6:cd6f:880d:6e79
Spoofed packet for existing ip6 as 2001:db8::200:ff:fe00:dddd
Spoofed packet for existing ip6 as fe80::ace5:8dd:eb38:a75e
Spoofed packet for existing ip6 as fe80::86c:fcc5:9aaa:83fb
Spoofed packet for existing ip6 as fe80::b4c4:e210:db07:2cda
Spoofed packet for existing ip6 as fe80::2529:5d9b:3f4d:59d2
Spoofed packet for existing ip6 as fe80::cddc:872e:3b48:4969
Spoofed packet for existing ip6 as fe80::f1bf:c342:ef9c:56e8
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as 2001:db8::200:ff:fe00:dddd
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as fe80::200:ff:fe00:ffff
Spoofed packet for existing ip6 as 2001:db8::200:ff:fe00:dddd
```

Fig. 9. Carrying out DoS-on-DAD attack

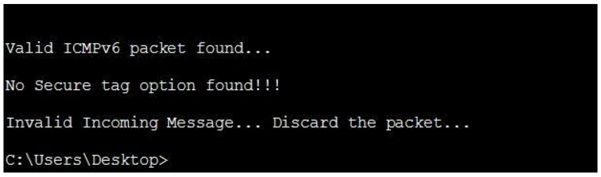
- Prevention Approach

In order to prevent the occurrence of DoS-on-DAD attack, New_Host was enabled with Secure-DAD mechanism that wants to join the IPv6 link local network. To prevent itself from DoS attack, it performed the validation check on every incoming NA message as aforementioned in Section 4. New_Host discarded all ND message such as; in coming NA messages except Secured NA messages appended with Secure-tag option from the sender (Existing_Host), while conducted Secure-tag matching process for all incoming NA messages with its self-generated Secure-tag. For instance, when New_Host received any NS message from the Existing_Hosts, It performed Secure-tags matching process. It entertained only those incoming NA messages that contains Secure-tag option while rest of the incoming NA messages were discarded. Figure 10 and Figure 11 presents the Secure-tag validation performed by New_Host upon receiving the valid Secured NA message from the valid host and fake NA message from an attacker host respectively.



```
Valid ICMPv6 packet found...

Secure tag option found

Incoming NA (with secure tag) packet calculation matched (positive) with the MAC.
Updating the Neighbor Cache Table.

---Neighbor Cache Table---

IP Address                                Physical Address
----------------------------------------------------------------------------

fe80:0000:0000:0000:e09c:17b8:3826:d734         00:21:70:fd:e4:0e



Secure NA validation time: 0.00799989700317 sec
----------------------------------------------------------------------------
```

Fig. 10. Secure-tag validation for incoming NA message



```
Valid ICMPv6 packet found...

No Secure tag option found!!!

Invalid Incoming Message... Discard the packet...

C:\Users\Desktop>
```

Fig. 11. Secure-tag validation process failure

Hence, from the experimental tests and results, it is clear that the Secure-DAD is an improved mechanism both in terms of processing time and effectiveness to prevent DoS attacks during DAD process in IPv6 link local network. The results have also proven that the Secure-DAD consumes less processing time to perform DAD process as compared with the existing mechanisms such as; SeND, SSAS, and Trust-ND. Moreover, Secure-DAD is effective enough to prevent DoS attack on DAD process. Figure 12 depicts the comparatives analysis of all mechanisms (SeND, SSAS, Trust-ND, and Secure-DAD) in terms of processing time to perform DAD process in IPv6 link local network. Thus, Secure-DAD is a suitable mechanism for IPv6 hosts to perform a secure link local communication in IPv6 network.

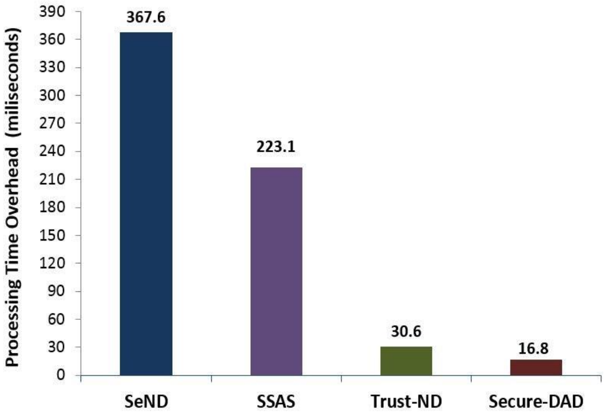


Fig. 12. Comparative results of Secure-DAD with existing mechanisms in terms of processing time overhead (SeND and SSAS processing time results were adopted from [12])

VIII. Conclusion and Future Work

This paper presented an improved mechanism to prevent DoS attack on DAD process in IPv6 network. A Test-bed was designed to allow the authors to evaluate the effectiveness of the mechanism by carrying out DoS attacks and comparing the performance of Trust-ND and Secure-DAD mechanisms. The experimentations were conducted on standard DAD, Secure-DAD, and Trust-ND mechanisms to justify the performance of Secure-DAD. The results showed that Secure-DAD consumed less processing time compared to Trust-ND mechanism. Moreover, Secure-DAD possessed less complexity compared to existing mechanisms such as; SeND, SSAS, and Trust-ND. Therefore, Secure-DAD is computationally efficient compared to existing mechanisms. In addition, experimented results also proved that the Secure-DAD mechanism is resistant to different types of attacks which can induce DoS attacks directly or indirectly on DAD process in IPv6 link local network i.e. effective and functional.

Hence, from the experimental tests and results, it was evaluated that the Secure-DAD mechanism not only performed better in terms of processing time, but also was effective and functional during attack conditions. Currently, the Secure-DAD mechanism was implemented on a small scale private IPv6 network. Therefore, our future work will be to optimize the Secure-DAD mechanism so that it can be applicable for the large scale public area IPv6 network.

ACKNOWLEDGMENT

This research was supported by National Advanced IPv6 Centre (NAV6), Universiti Sains Malaysia (USM) and Ministry of Higher Education Malaysia.

REFERENCES

[1] Thomson S, Narten T, Jinmei T. IPv6 Stateless Address Auto-configuration. Internet RFC 4862, 2007.

[2] Deering S, Hinden R. Internet protocol version 6 (IPv6) specification. Internet RFC 2460, 1998.

[3] Li, S., Da Xu, L., & Zhao, S. The internet of things: a survey. Information Systems Frontiers, Springer, Science & Business Media, vol. 17(2), pp. 243-259, 2015.

[4] Botta, A., de Donato, W., Persico, V., & Pescapé, A. Integration of cloud computing and internet of things: a survey. Future Generation Computer Systems, Elsevier, 56, pp.684-700, 2016.

[5] Rehman SU, Manickam S. Significance of duplicate address detection mechanism in Ipv6 and its security issues: A survey. Indian Journal of Science and Technology, vol. (8)30, 2015.

[6] Narten T, Simpson, WA, Nordmark E, Soliman H., Neighbor discovery for IP version 6 (IPv6), 2007.

[7] AlSa'deh A, Meinel C. Secure neighbor discovery: Review, challenges, perspectives, and recommendations. IEEE Security & Privacy, vol. 10, pp. 26-34, 2012.

[8] Conta A, Gupta M. Internet control message protocol (ICMPv6) specification. Internet RFC 4443, 2006.

[9] Dawood, H. IPv6 Security Vulnerabilities. International Journal of Information Security Science, vol. 1(4), pp.100-105, 2012.

[10] Arkko J, Kemp f J, Zill B, Nikander P. Secure neighbor discovery (SEND). Internet RFC 3971, 2005.

[11] Rafiee H, Meinel C. SSAS: A simple secure addressing scheme for IPv6 autoconfiguration. Eleventh Annual IEEE International Conference on Privacy, Security and Trust (PST), pp. 275-282, 2013.

[12] Praptodiyono S, Murugesan R K, Hasbullah IH., Wey CY, Kadhum MM, Osman A. Security mechanism for IPv6 stateless address autoconfiguration. 2015 IEEE International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), pp. 31-36, 2015.

[13] Andreeva E, Mennink B, Preneel B. Open problems in hash function security. Designs, Codes and Cryptography, vol. 77, pp. 611-631, 2015.

[14] Bhargavan K, Leurent G. Transcript collision attacks: Breaking authentication in TLS, IKE, and SSH. NDSS, 2016.

[15] Rehman SU, Manickam S. Denial of Service Attack in IPv6 Duplicate Address Detection Process. International Journal of Advanced Computer Science & Applications, vol. 7, pp. 232-238, 2016.

[16] Moore D, Shannon C, Brown D J, Voelker GM, Savage S. Inferring Internet denial-of-service activity. ACM Transactions on Computer Systems (TOCS), vol. 24. Pp. 115-139, 2006.

[17] Shoup V, fast and provably secure message authentication based on universal hashing. In Advances in Cryptology—CRYPTO'96, pp. 313-328, 1996.

[18] Krovetz T. UMAC: Message authentication code using universal hashing. Internet RFC 4418, 2006.

[19] Kali Linux Penetration Testing and Ethical Hacking Linux Distribution. https://www.Kali.org.

[20] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, and Y. Xiao, "Network forensics analysis using Wireshark," International Journal of Security and Networks, vol. 10(2), pp. 91–106, 2015.

[21] Woodcock, J., Stepney, S., Cooper, D., Clark, J., & Jacob, J., The certification of the Mondex electronic purse to ITSEC Level E6. Formal Aspects of Computing, vol. 20(1), pp. 5-19, 2008.

[22] Saleem S, Popov O, Dahman R. Evaluation of security methods for ensuring the integrity of digital evidence. 2011 IEEE International conference on Innovations in information technology (IIT), pp. 220-225, 2011.

[23] THC-IPv6 Attack Tool-kit. https://www. aldeid. Com/wiki/THC-IPv6-Attack-Toolkit.