An Algorithm for Securing IPv6 SLAAC and DAD

**Neeraj Kumar Cheryala, Nikhil Madhavaneni, Jithender Reddy Machana**

1[neerajcher@gmail.com](mailto:neerajcher@gmail.com)

2nikhil.madhavaneni@gmail.com

3[jeethreddy1@gmail.com](mailto:jeethreddy1@gmail.com)

***Abstract--* Over the last few years, IPv6 has gained momentum in the industry with deployments of decreasing complexity, improving security, and offering 340 sextillion unique IP addresses. The IPv6 protocol is considered more reliable and secure when compared to its 32-bit counterpart. The most important part of IPv6 protocol is the Neighbor Discovery (NDP) which enables a node to communicate with other devices on the same LAN. The IPv6 SLAAC where there are no central address configuration servers is also based on the NDP process which uses Duplicate Address Detection to verify the uniqueness of generated addresses on the local link. However, this process is susceptible to many attacks and making it secure should be one of the primary concerns to address the related issues in small IPv6 networks. In this paper, we introduce a new approach to optimize the security of NDP and DAD processes. This method is based on SHA-512 to verify the identity of NDP messages on the local link. The technique is programmatically implemented to illustrate the secure DAD process and estimate the resources utilized at a given node. We also discuss the existing flaws in CGA and propose two major modifications i.e. replacing the present public key cryptography scheme and the hash function. Elliptical Curve Cryptography (ECC) for the NIST P-384 curve is recommended for the ECDSA key generation process and SHA-512 instead of SHA-1. ECC with SHA-512 proves to be secure and optimal in terms of the consumption of computational resources at the nodes.**

***Keywords-- Neighbour Discovery Protocol, SEcure Neighbor Discovery Protocol, DAD Attack, DoS Attacks, SHA-512, Cryptographically Generated Address, ECDSA, NIST P-384, GNS3, Docker, Scapy, Wireshark.***

1. INTRODUCTION

Internet Protocol Version 6 is designed to succeed in the IPv4 protocol and the depletion of IPv4 address space has been a motivating factor to move to IPv6 addresses. Temporary solutions such as Network Address Translation (NAT) worked fine but the enormous growth of routing tables has always been an issue with the IPv4. With an increasing internet population, issues with NAT, and the advent of IoT, transitioning to IPv6 is no surprise. IPv6 has a larger 128-bit address space providing 340 sextillion possible addresses. Hence it supports end-to-end connectivity while discarding NAT. Apart from the address space, Internet Engineering Task Force (IETF) also fixed some of the limitations of IPv4 when designing IPv6. These are: IPv6 comes with Internet Control Message Protocol Version 6 (ICMPv6) which includes address resolution and address autoconfiguration not found in ICMPv4. Secondly, IPv6 reduces the size of routing tables and makes routing more efficient and hierarchical. Furthermore, the IPv6 header has a simpler size with no checksum making the packet processing a bit more efficient at each hop along a path to the destination. It also offers in-built security with IPSec for network operations. However, this protocol does not suit well for communications on the local link. Lastly, IPv6 neighbor discovery [1] is more efficient than IPv4 ARP address resolution as it uses the solicited-node multicast MAC address for the resolution process which doesn't require layer-3 processing of the packet by each node as in the case of IPv4 ARP. IPv6 makes use of the Neighbor Discovery Protocol (NDP) and allows a node to get connected with the IPv6 network. However, NDP inherently assumes that the network only consists of trusted nodes but with the advent of public insecure wireless networks, any rogue device with minimal authentication can become a part of the local link and launch an attack. Therefore an IPv6 network is prone to DoS attacks on DAD and several other attacks on the NDP process as a whole. In this paper, we discuss the security problems associated with IPv6 link-layer communications and the flaws present in the current CGA address generation process. The main aim of the paper is to propose an algorithm for securing the local-link communications by leveraging Secure Hash Function-512 (SHA-512). Also, employing dual cryptography with Elliptical Curve Cryptography (ECC) for the NIST (National Institute of Standards and Technology) P-384 curve in combination with SHA-512 is discussed and evaluated. The proposed technique is programmatically implemented for a DAD scenario and the results of securing the NDP among the nodes are presented along with the required computational resources (Time complexity) at a given node.

1. RELATED WORK

Attacks on IPv6 networks have been discovered in various scenarios especially on Duplicate Address Detection (DAD) and NDP processes. As a result, these have been interesting research cases and over the years, many researchers have been proposing algorithms and novel approaches to secure IPv6 link-local communications involving DAD and NDP processes.

In [8], the authors have proposed an algorithm for address generation. This method has lesser computation cost when compared to the standard Cryptographically Generated Address (CGA) approach but the methodology is based on SHA-1 encryption which is mostly obsolete as per Google security reports in 2015. It is also susceptible to collision attacks for this reason. In [15], the authors have utilized a novel approach for securing the DAD process. They made use of an alternative approach to CGA and SeND (Secure Neighbor Discovery) protocols with a limitation to the security level using SHA-1. In [6], the authors have introduced an approach to secure IPv6 addresses by modifying RFC 3972 standard. They have reduced the granularity factor of a sec from 16 to 8 and replaced RSA with ECC but used the SHA-256 function. SHA-256 may be compromised in the near future. Another methodology such as securing address configuration protocol for vehicular networks [9] was introduced to secure the DAD process. However, this is only useful when a vehicle and its serving AP are one-hop apart. In [10], the authors have proposed an SDN-controller based mechanism to verify the integrity of received messages. However, this method is not so efficient and has its own limitations. Another approach was taken in [11] to secure the DAD which is called Trust-ND. But the experiments show the limits of this method.

In this paper, we suggest a new algorithm to secure the DAD process and verify the integrity of NDP messages. The results illustrate that the DAD process can be optimized by adding a new field in Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages i.e. the hash code (from SHA-512) of the new device's IP address. Also, ECC with SHA-512 can be leveraged for the existing CGA process which uses RSA and SHA-1 to improve the robustness of the algorithm. Section-III briefly describes the IPv6 Neighbor Discovery Protocol along with the ICMPv6 packets involved in the process. We look at several NDP attacks in Section-IV. Section-V discusses the major limitations of existing CGA mechanism and subsequently, we propose the required changes to the same in Section-VI. In Section-VII, we describe the proposed algorithm for IPv6 address generation and verification process along with the screenshot of the python function. Nextup, we present the overall implementation of the SLAAC process in Section-VIII using Docker and GNS3. Section-IX compares the performance of proposed algorithm with that of standard CGA.

III. NEIGHBOR DISCOVERY PROTOCOL

Neighbor discovery protocol (NDP) [1] is a protocol for the Internet protocol version 6(IPV6), it is a key protocol for the functioning Internet Protocol suite. It uses the Internet Control Message Protocol(ICMPV6) to perform various actions like Address Resolution Protocol (ARP) for IPv6 which is similar to ARP for IPv4, makes Stateless address autoconfiguration possible to do the configuration of hosts, gathers various information about neighboring nodes, Router discovery and many more. NDP ICMPv6 is used which is more secure and robust than the ICMPv4.

NDP uses ICMPv6 packets to exchange information in some of the key functionalities are: (RFC 2461)

*Router Solicitation:*

In this, the host uses the ICMPv6 packet with type 133 and sends a Router Solicitation message to all the routers in the link requesting for the information then a router responds with the Router Advertisement message. It uses the destination address as All router's multicast address ff02::2.

*Router Advertisement:*

In this RA messages are sent by IPv6-enabled routers using ICMPv6 packet with type 134. For every 200 seconds to provide addressing information to IPv6-enabled hosts in the local link. The RA message can include addressing information for the host such as the prefix, prefix length, DNS address, and domain name. It sends RA messages for every RS message the routers receive instead of waiting for the periodic timer.

It uses the destination address as All node's multicast address ff02::1.

*Neighbor Solicitation:*

Neighbor solicitations are used by IPv6 enabled hosts. They use ICMPv6 packet type 135 to determine the link-layer address of a neighbor or for duplicate address detection and neighbor unreachability detection.

*Neighbor Advertisement:*

Neighbor advertisements are used by IPv6 enabled hosts. They use ICMPv6 packet type 135  to respond to a Neighbor Solicitation message or whenever a device changes its mac address it sends out an NS message to the multicast group.

*Redirect:*

Routers may inform hosts of a better first-hop router for a destination in the same way as redirects are used in ICMP for IPv4 to redirect traffic from one router to another.

Whenever a new host joins the link, it requests the router for addressing information by sending an RS message. The router responds by sending a RA message stating to use stateless or stateful address configuration. In the case of stateless auto-configuration, the host creates the new interface-id using the EUI 64 bit process or a randomly generated mac address and uses the prefix part provided by the routers RA message.

In order to verify that the generated IPv6 address is unique, it uses a process called Duplicate Address Detection(DAD).To check the uniqueness of an address, the device will send an NS message with its own IPv6 address as the targeted IPv6 address, as shown in the figure. If another device on the network has this address, it will respond with an NA message. This NA message will notify the sending device that the address is in use. If a corresponding NA message is not returned within a certain amount of time, the unicast address is unique and acceptable for use. Our paper mainly focuses on this DAD process as it is susceptible to many attacks and making the implementation as efficient as possible.

IV. NDP ATTACKS

NDP plays a crucial role in the functioning of the Link layer. But there are vulnerabilities in NDP [4] [14] that can cause adverse effects on the network. Some of the NDP attacks are  MiTM, DoS, and Spoofed Router Redirect Message attacks.

A Man-in-the-Middle attack (MiTM): In MiTM, the attacker enters into the network as a trustable host and enters himself in the middle of the conversation of the two hosts. To achieve this the attacker sends forged NS/NA packets to both the hosts and hence poisoning their caches of the respective hosts. The attacker will then forward the frames between the two target hosts by modifying traffic between the two hosts.

Here three kinds of attacks [12] can be possible: Spoofed ICMPv6 NA, Spoofed ICMPv6 RA, Replay Attack.

Spoofed ICMPv6 NA attack: In a scenario of 3 hosts in a network, where hosts a wants to know the mac address of B for the corresponding IP address, it sends out an NS message to find out the mac address of the host it sends an all-nodes multicast message with destination address as ff02::1. Now every host checks and response, if a rogue host is present it claims that the IP address belongs to it and it overrides the B's NS message by setting the flag to '0'. In this way, all traffic which is intended for B goes to C.

Spoofed ICMPv6 RA attack: In an IPv6 network, RA messages are sent out by router periodically every 200sec to give information about network prefix, lifetime, and configuration address to all the FF02::1 multicast nodes group. Now all the IPv6 enabled hosts to configure accordingly.

Here in this network, anyone can claim to be the router and send the periodic RA to the hosts in the network. Thus, anyone can be the default gateway on the network, act as a router, and see the traffic that is flowing from the network.

Replay Attack: In this attack, the rogue host can use the NDP messages for the latter purpose and send it. He might even alter the messages and try to gain network control.

DAD attack: In a DoS attack, the rouge host makes network services unavailable to other hosts. A DoS attack is to deliberately cut off a web site or network from its hosts or users qualifies as a DoS attack. Similarly, DoS attacks are possible in IPv6 enabled networks.

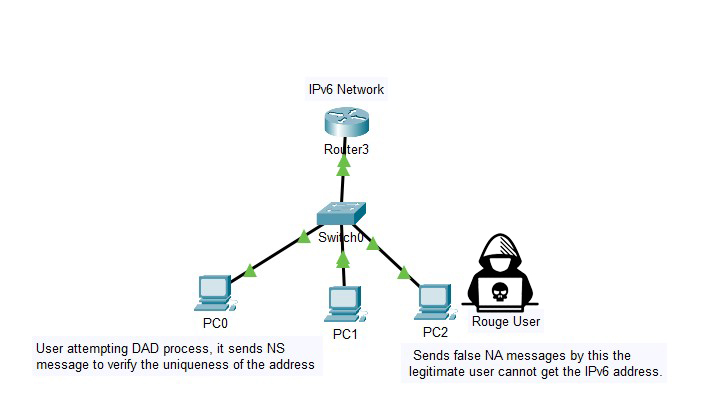


Figure 1: Understanding the DoS attacks during DAD.

Whenever a host configures it's IPv6 address it has to check that no host is using that address to do this; it undergoes a DAD process, which isn't secure. By exploiting this the rouge host can perform a DoS attack. When the host sends an NS message for the DAD process as discussed earlier, the attacker can reply with an NA stating that it has already taken the address into use.  On receiving that message, the host generates a new address again and performs the DAD process again as the IP is already taken. Again a similar process is done by the rouge host, by this, the user cannot connect to the network.

V. LIMITATIONS OF EXISTING CGA MECHANISM

RFC 3972 specifies a method to bind a cryptographic public key with the IPv6 address in the SeND [2] protocol. The interface identifier i.e. IID (last 64-bits of the IPv6 address) is generated bypassing the public key of the corresponding node to a hash function. CGA operates independently of any third party authorization while helping the address generator to claim ownership of the address. The association of the public key and generated address is confirmed by re-computation at the receiver's end. The messages are encrypted with the node's private key before transmitting on the local-link. Hence the verifier must know the public key and source address to decrypt and authenticate. The concept of CGA was drafted successfully by Aura in 2005 [3] resulting in an RFC.

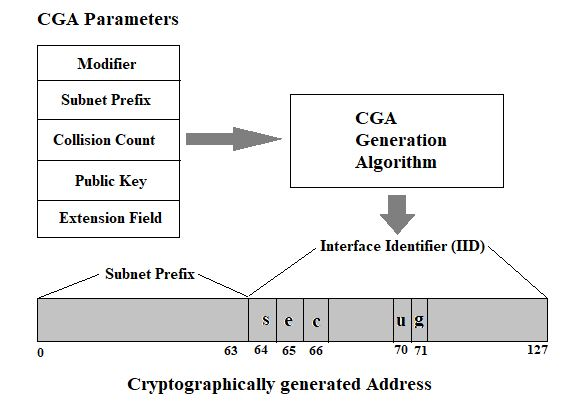


Figure 2: The standard CGA process (CGA Generation algorithm is SHA 1).

Although SeND [2] addresses security issues with Neighbor discovery, it has a number of limitations. Firstly, CGAs require heavy computational resources and are impractical for sec (security parameter) values greater than 1. Secondly, the selection of a public-key cryptosystem plays a key role [5] in the overall performance and security of messages and operations. The CGA generation time also depends on the time required to generate key pairs. RSA key pairs taking shorter generating times usually provide medium security which is the case with the standard CGA mechanism. Finally, the selection of the hash function also impacts the security and performance of CGAs. CGA uses SHA-1 which is a 160-bit hash function. However, it is susceptible to collision-free attacks and it has been even reported that SHA-1 was compromised in practice by Google and CWI institute. Additionally, it is obvious that CGAs are vulnerable to Denial-of-Service (DoS) attacks as mentioned in the previous section. A node can be flooded with messages sent in high frequency across the network. All the destination nodes need to verify and eventually waste their computational resources.

VI. MODIFYING THE EXISTING CGA MECHANISM

*Replacing RSA with ECC*

For standard CGA, we use RSA (Rivest-Shamir-Adleman) key size of 1024 bits. The bandwidth can be easily consumed by minimizing the existing CGA parameters. Using an alternate cryptosystem such as Elliptical Curve Cryptography or Elliptical Curve Digital Signature Algorithm (ECDSA) [8] is more plausible for low resource-constrained networks. This is because ECDSA takes less time for the generation of key-pairs. Secondly, ECC key size is lesser than that of RSA offering the same level of security. This is where bandwidth consumption comes into picture. Table-1 shows an equivalence between RSA and ECC security levels with ECC having smaller key sizes. Table-2 [8] shows the comparison of CGA parameter data structures lengths using RSA vs ECC public keys.

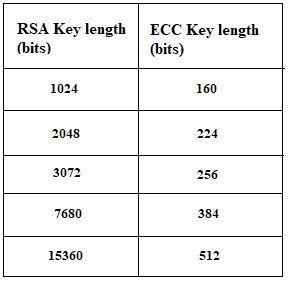
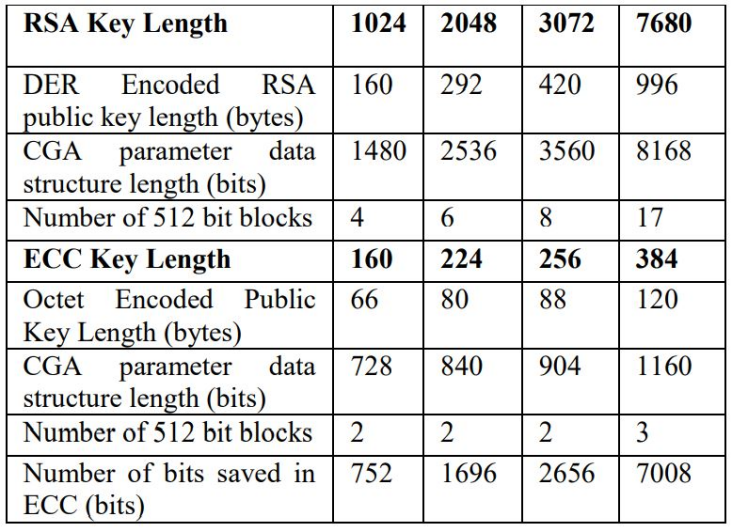
 

Table-1 Table-2

As per [18], National Institute of Standards and Technology (NIST) recommends the use of only three curves including NIST P-256, NIST P-384 and NIST P-521 when going for ECDSA. With ECC, the CGA generation and verification remains the same as described by RFC3972. But we can extend Section-3 since it illustrates the RSA mechanism. When ECC is used, the AlgoIdentifier in ASN.1 data structure of type SubjPublicKeyInfo must be the id-ecPublicKey algorithm identifier which is OID 1.2.840.10045.2.1 and the SubjPublicKey becomes an ECC Public key specified in RFC5480. ECC key lengths are identified by the namedCurve parameter in the ECC parameters field of AlgoIdentifier. Figure 3 shows the common sizes of the Digital Signature field when using ECDSA.

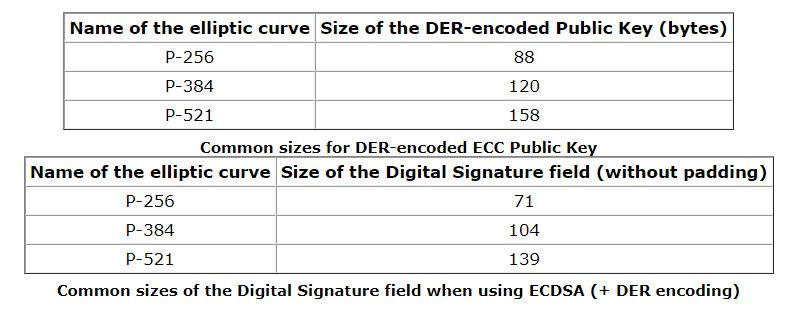


Figure 3

Where b=27580193559959705877849011840389048093056905856361568521428707301988689241309860865136260764883745107765439761230575

We have chosen the NIST P-384 curve as it is recommended by NSA to be used until the dawn of post-quantum cryptographic methods. It provides a 192-bit security and has also got a lot of research work to make the key generation process efficient. For instance, in [16] the authors have proposed plausible software techniques for accelerating cryptographic operations using the P-384 curve. The equation of the curve is given by:

Where:

The original value of a=-3 is chosen for efficiency reasons as per IEEE Std 1363-2000.

b=27580193559959705877849011840389048093056905856361568521428707301988689241309860865136260764883745107765439761230575

*Replacing the existing hash function*

The original CGA uses SHA-1 for getting the hash code in the address generation process, however it is soon to become obsolete as [17] rightly points out the chances of collision attacks. Hence, replacing it with more secure functions such as SHA-256 or SHA-512 [7] should be one of the most important modifications to the algorithm. Since the CGA also has the overhead of generating the cryptographic keys, SHA-512 happens to be a good companion in terms of time taken for address generation. As a result, ECDSA using NIST P-384 along with SHA-512 provide a good trade-off between the security and the computational resources which is the actual need of the hour in link-local communications of the present day. Consider the following comparison between the hash functions shown in figure .

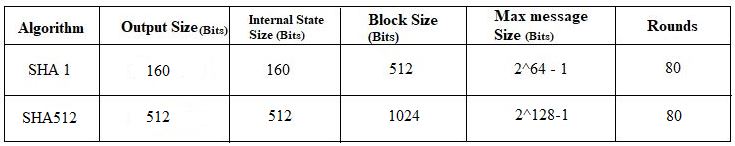


Figure 4: Simple comparison between SHA1 and SHA512.

VII. PROPOSED ALGORITHM

It is the need of the hour that SLAAC phase should take as small time as possible. This is because in the present day networks, devices constantly get added and migrate to different networks very often. Here we propose an algorithm which satisfies this need without the overhead of cryptographic key generation. Let's have a look into the algorithm, the details of securing the DAD process, and mitigating DoS attacks during duplicate address detection process when we use stateless address autoconfiguration (SLACC) for configuring the IPv6 address. During the DAD process in this algorithm, we have two processes that have to be performed: one at the node which sends the NS message, and the second process at the receiver nodes which sends the NA message if the address matches it or else it simply discards it.

***Algorithm Explanation:***

ACRONYMS USED:

**NC=Number of collisions**

**UF=Uniqueness Flag**

**NS\_C=Neighbour Solicitation collisions**

**RaN0=64 bit random number**

**CT= Clock time or the time at the generation**

**I\_ID= Interface Identifier of the IPv6 link local address**

**LLA= Link Local Address**

**IP\_  T\_I\_ID=Interface Identifier in the target IP header field**

**LLA\_R\_IP=Received link local address**

**IID\_RT=Received target interface Identifier in the target IP field.**

*At the sender node:*

1) Initially set the values as NC=0, NS\_C=0, UF=1, CT..

2) Now generate a 64 bit random number and assign it to RaNo.

3) Apply SHA-1 to the concatenation of (NC, NS\_C, CT and RaNo) and take the result to Hash-1.

4) Now break the hashed result which is in SHA-512 into two equal parts and put in Div\_hash1 and Div\_hash2.

5) Now form the interface id (I\_ID), by taking 20MSB of Div\_hash1 and Div\_hash2 and 24 LSB of the generated RaNo at the 2 step.

1. Now the interface Id (I\_ID)is generated , lets combine it with the network prefix part which is given to us. For this LLA address, we have to perform the DAD process.

7) Now perform SHA-512 on the interface id (I\_ID) and take the result into Hash-2.

8) Now we have to form the IP\_T\_I\_ID for the icmpv6 header field target ip address by taking 40 MSB of SHA-512 and 24 LSB of the generated link-local address.

9) Now for the target IP address field in icmpv6 header is formed by concatenating the local network prefix with IP\_T\_I\_ID to form a 128 bit address.

10) Now we have to perform the DAD process on the generated link local address whether it is unique or not by sending a Neighbor Solicitation (NS) message.

Now an ICMPv6 type 134 message is sent with the source address as unspecified address and destination address as solicited node multicast address FE80::1.

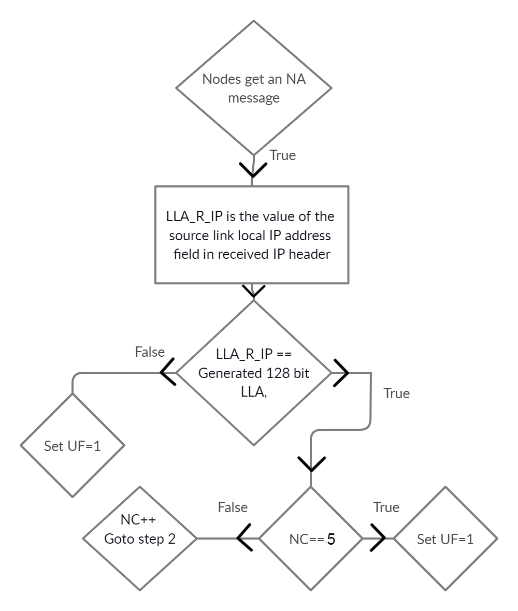


Figure 5: NA received by the sender.

Figure 5 describes what happens at the sender node when it receives an NA message for the generated IPv6 address.

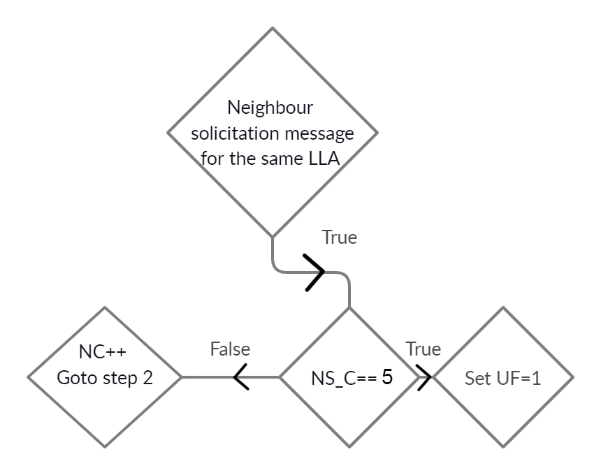


Figure 6: NS received by the sender.

1. The source node will receive an NS message for the same LLA whenever a rogue node is trying to perform the DAD process for the same LLA, by this he can carry out the DAD process. Each time this happens the collision count is increased by 1. If the value equals 3 it means that an attack is being carried out by any rouge node. This step is depicted by Figure as a flowchart.

*At the receiver node:*

Now the nodes that receive the solicited-node multicast NS message initiate the following process:

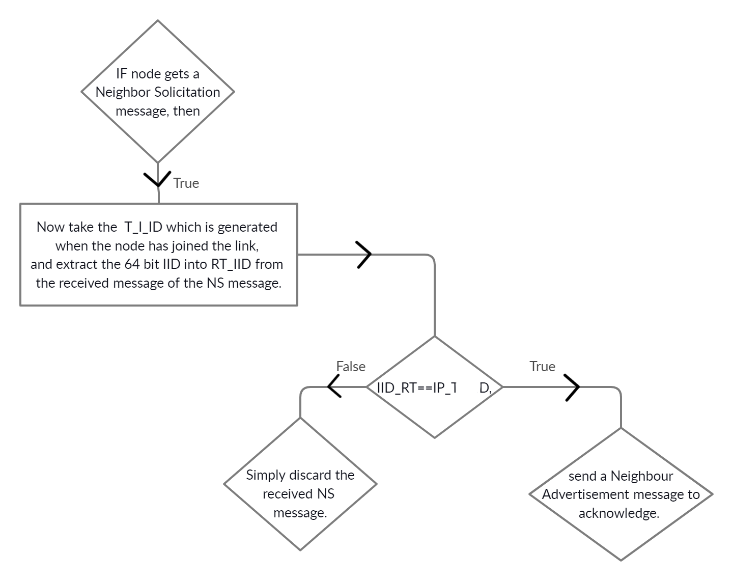


Figure 7: Flowchart depicting the process at the receiver node as part of the proposed algorithm.

Figure 8 shows the screenshot of the Python function that implements the proposed algorithm. It contains the address generation phase. Figure 9 depicts what happens at the sender when it receives a Neighbor Advertisement or a Neighbor Solicitation for the same generated address.

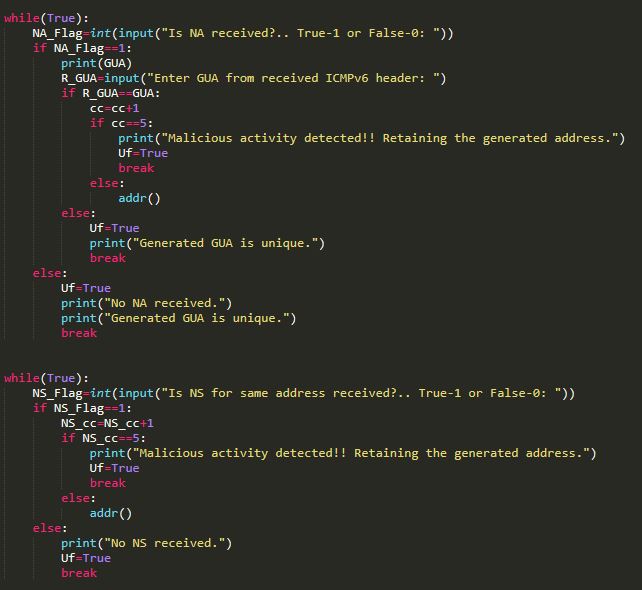
 

Figure 8: Python function that implements the address Figure 9: NA or NS received at the sender node. generation

VIII. IMPLEMENTATION

This section gives detailed explanation of the implementation. To demonstrate, we have used a topology comprising of a Router, Switch and 3 Hosts (sender, receiver, and a hacker). Figure 10 depicts the topology.

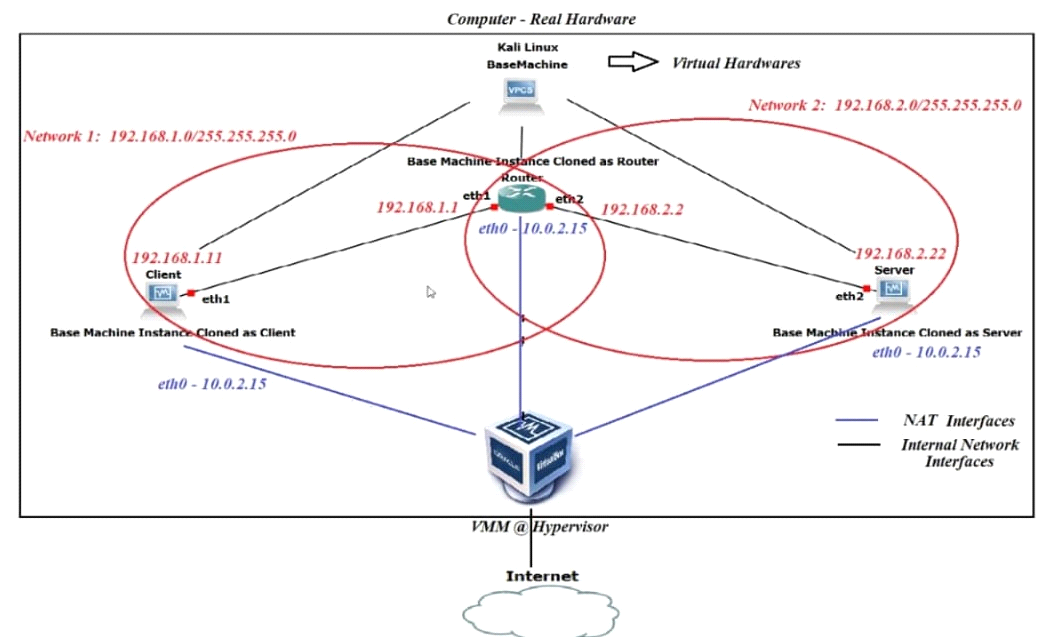
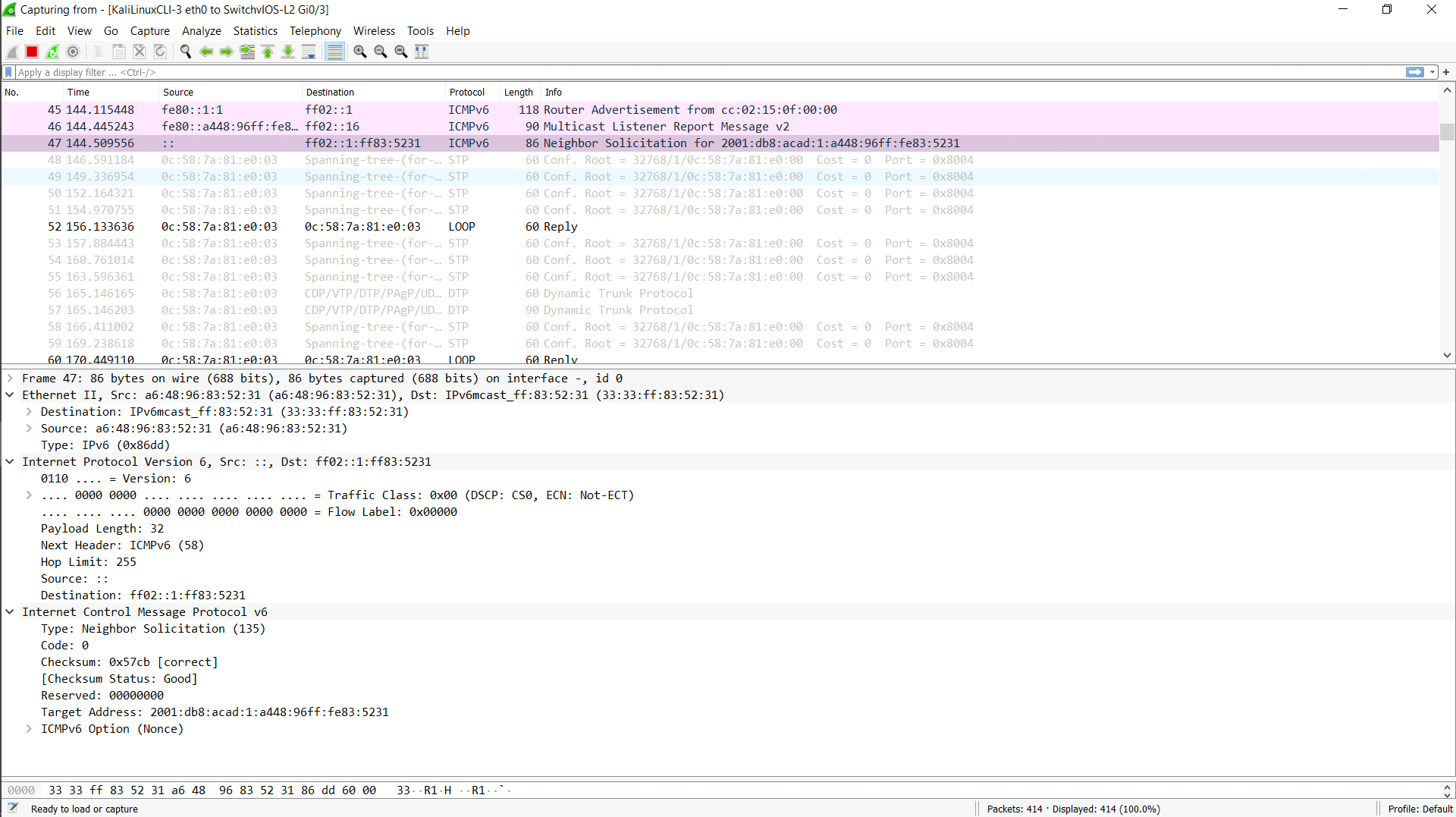


Figure 10 : Topology designed for testing the algorithm in GNS3.

To demonstrate the idea, the above topology is designed using the emulation tool GNS3, having a router, a switch and three host machines. the Cisco IOS 3660 image is installed on the router. And Docker machines are used as hosts in the topology. To support the virtualization, hypervisor “VMVare” is installed. The reason behind using Docker machines is it consumes less primary memory than QEMU/ Virtualization technologies. After trying out many tools and python modules, we came to conclusion that the networking tool “Scapy” is good to craft and manipulate packets. It served our purpose to demonstrate DAD process, the DOS attack and repay attacks. Figure 11 includes the screenshots of original DAD protocol captured using sniffing tool “Wireshark”.



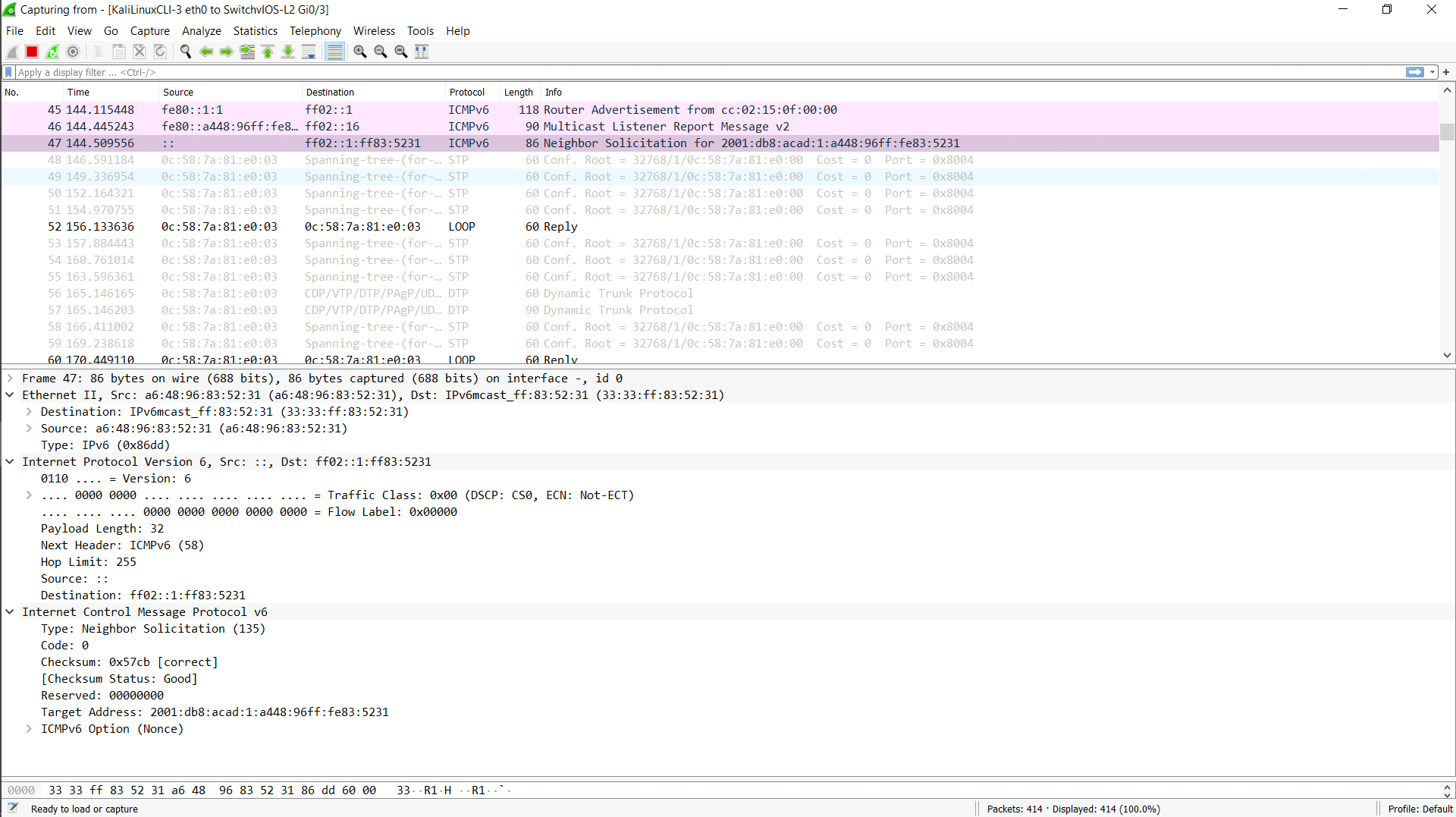
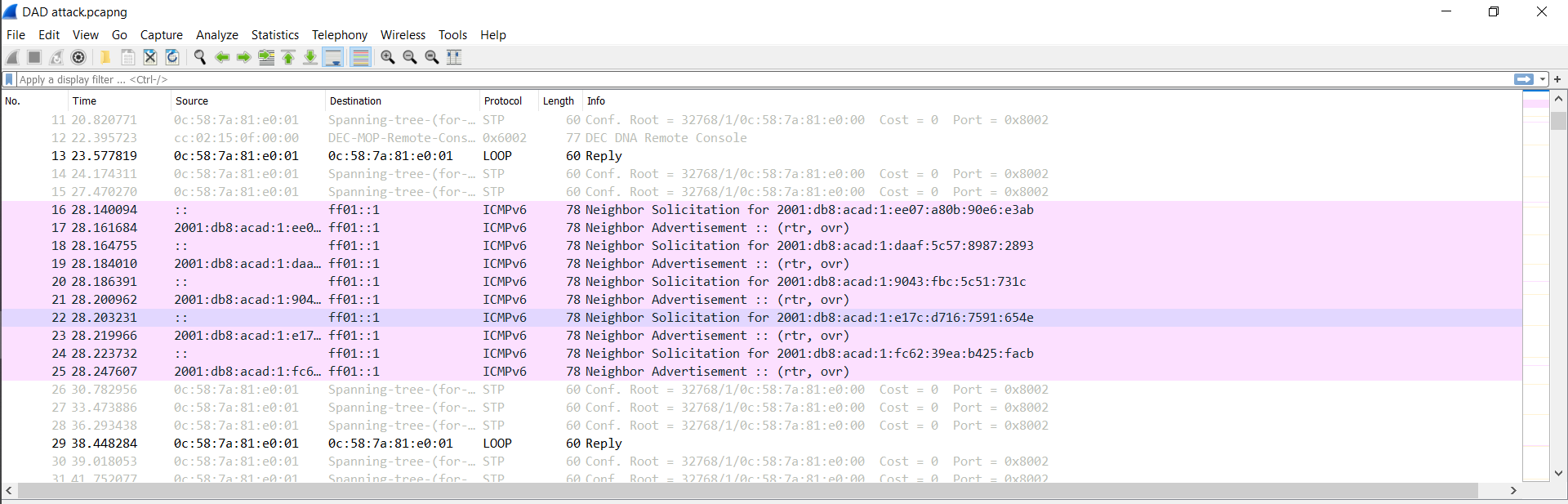
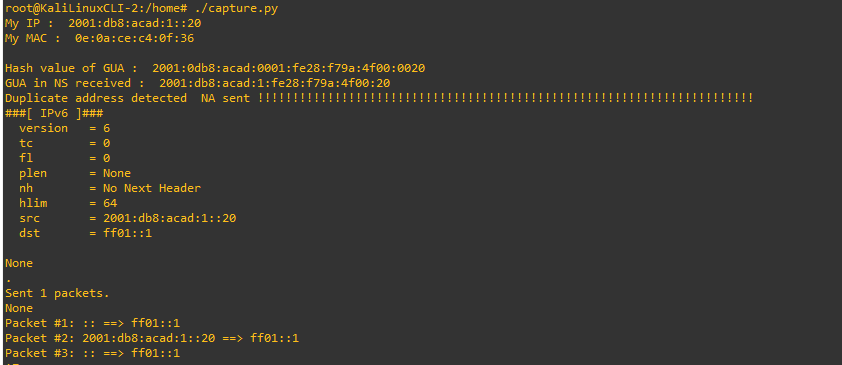


Figure 11: DAD protocol captured using Wireshark tool.

The limitations of DAD protocol are shown by running a script at hacker side. The script does the DoS attack acknowledging each neighbour solicitation message sent by the host, its objective is not to allow the host to be part of the network. With that the host fails to configure tentative address as an IID. Figure 12 includes the required screenshots.





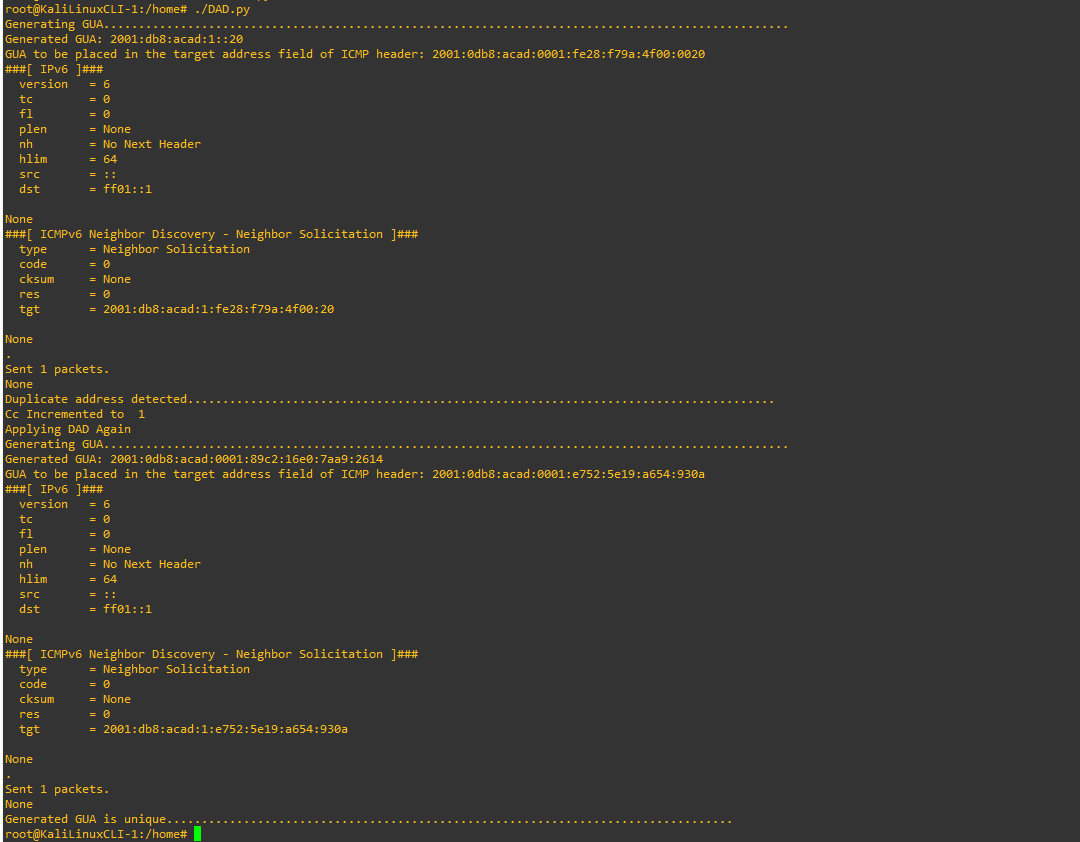
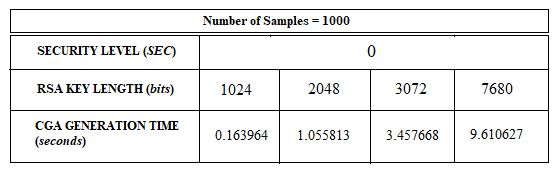
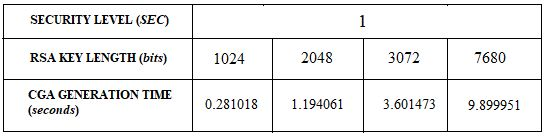


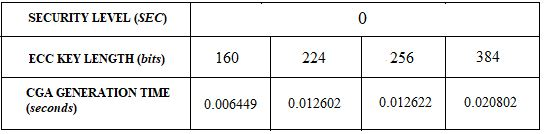
Figure 12: DoS attacks and prevention using Wireshark and GNS3.

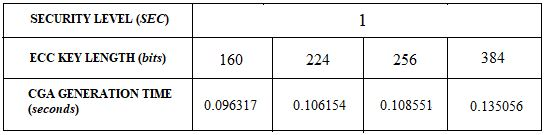
IX. RESULTS AND COMPARISON

In this section, we present the comparison of the performance of standard CGA to that of the proposed algorithm for IPv6 address generation. This comparison is mainly based on address generation times. As discussed in the previous section, the implementation is carried out in a virtual environment with three hosts running Kali 2020.1b version. For calculation and comparison of address generation times of CGA and our own algorithm, the tests have been carried out on an i5-1035G1 processor which operates at a clock frequency of 1 GHz. Note that the CGA algorithm is realized for SEC values of 0 and 1 complemented with RSA and ECDSA of varying key lengths. The following tables give the results thus obtained.









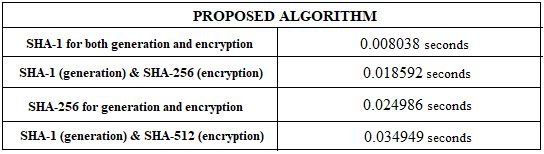


Figure 13: Comparing the time complexity of the CGA with that of proposed algorithm.

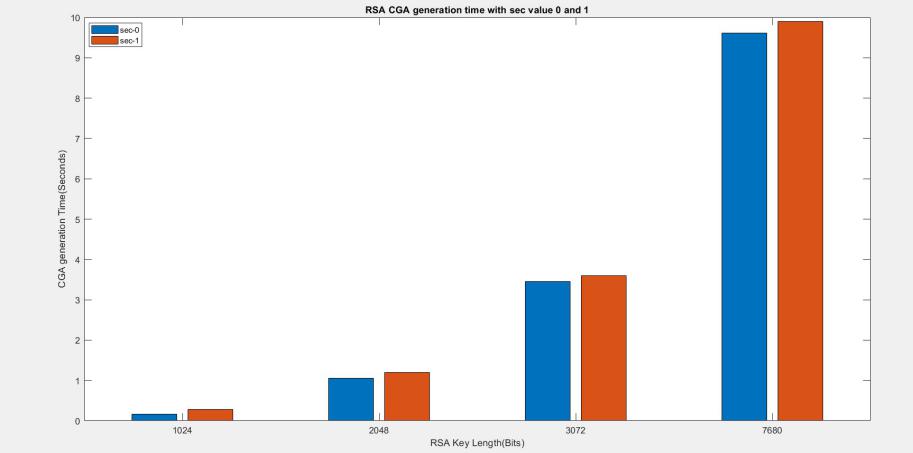


Figure 14: Address generation time for RSA with *sec* values, 0 and 1.

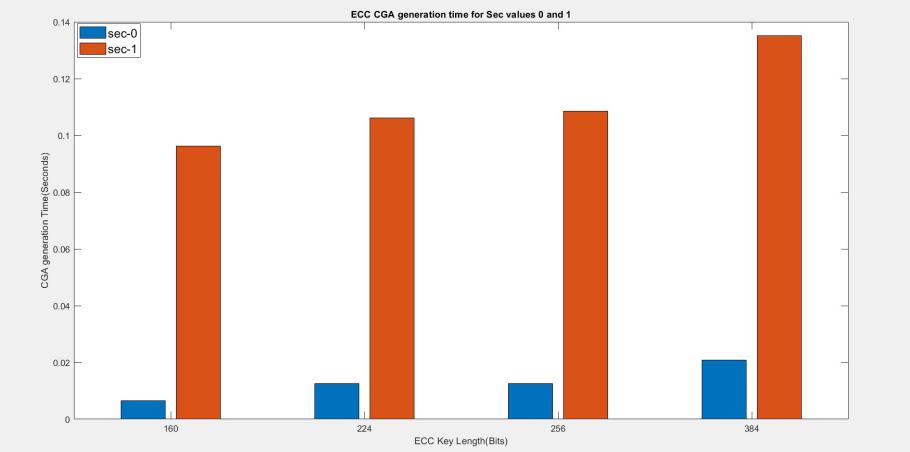


Figure 15: Address generation time for ECC with *sec* values, 0 and 1.

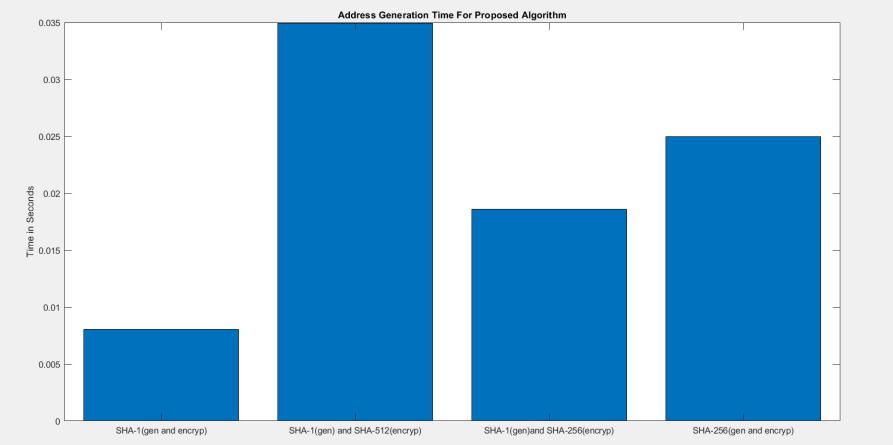


Figure 16: Address generation times for proposed algorithm based on the function used for address generation and encryption.

We observe that the proposed algorithm takes at least 100 msec less for address generation and encryption when compared to the traditional CGA with ECDSA. This is mainly because the algorithm does not have an overhead of crypto key pair generation. Also, the difference between the generation times is decreasing as we increase the strength of the hash functions for address generation and encryption.  The added advantage with our algorithm is that the hacker must generate the encrypted addresses by performing SHA-512 for  a brute force attack. This should be done for each possible network prefix and the interface identifier. In addition, this requires a large storage to impersonate a random node in the network. Hence it’s not so easy for a hacker to carry out an attack on the DAD process with nodes running our algorithm.

X. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new algorithm for address generation that leverages SHA-1 and SHA-512 functions. These functions help to generate a unique IPv6 IID and also carry out the DAD procedure securely without the overhead of generation of cryptography key-pairs. Also, the substitution of ECC for RSA in the original CGA has been discussed. The results show that the method is robust to DoS attacks, Spoofing attacks and Man-in-the-Middle attacks. As part of our future work, we intend to make use of Software-Defined Networking to monitor the DAD and Neighbour Discovery protocols. SDN in recent times has been a pioneer networking procedure which handles critical tasks smoothly without the administrator having to monitor the results constantly.

REFERENCES

# [1] T. Narten, E. Nordmark, W. Simpson, H. Soliman. *Neighbor Discovery for IP Version 6 (IPv6).* *Internet Engineering Task Force.* September. 2007. RFC 4861, 2007.

[2] J. Arkko, J. Kempf, B. Zill, P. Nikander. *SEcure Neighbor Discovery (SEND). Internet Engineering Task Force.* March. 2005. RFC 3971, 2005.

[3] Aura, Tuomas. *Cryptographically Generated Addresses (CGA)*. *Internet Engineering Task Force*. Mar. 2005. RFC 3972, 2005.

[4] Alsadeh, Ahmad, Hosnieh Rafiee, and Christoph Meinel. "Cryptographically Generated Addresses (CGAs): Possible attacks and proposed mitigation approaches." *Computer and Information Technology (CIT*), 2012 *IEEE 12th* *International Conference on.* IEEE, 2012.

[5] AlSa'deh, Ahmad, and Christoph Meinel. "Secure neighbor discovery: Review, challenges, perspectives, and recommendations." *Security & Privacy, IEEE* 10.4 (2012): 26-34.

[6] Qadir, Sana, and Mohammad Umar Siddiqi. "Cryptographically generated addresses (CGAs): a survey and an analysis of performance for use in mobile environment." *IJCSNS Int. J. Comput. Sci. Netw. Secur* 11.2 (2011): 24-31.

[7] C. Dobraunig, M. Eichlseder, and F. Mendel, “Analysis of SHA-512/224 and SHA-512/256,” *in Proceedings of the 21st International Conference on the Teory and Application of Cryptology and Information Security*, Advances in Cryptology – ASIACRYPT 2015, Auckland, New Zealand.

[8] J. L. Shah and J. Parvez, “IPv6 cryptographically generated address: analysis and optimization,” in Proceedings of the AICTC ’16 *Proceedings of the International Conference on Advances in Information Communication Technology & Computing,* vol. 13, 2016.

[9] X. Wang, Y. Mu, G. Han, and D. Le, “A secure IPv6 address confguration protocol for vehicular networks,” *Wireless Personal Communications*, vol. 79, no. 1, pp. 721–744, 2014.

[10] Y. Lu, M. Wang, and P. Huang, “An SDN-based authentication mechanism for securing neighbor discovery protocol in IPv6,” *Security and Communication Networks*, vol. 2017, pp. 1–9, 2017.

[11] S. Praptodiyono, I. H. Hasbullah, M. M. Kadhum, R. K. Murugesan, C. Y. Wey, and A. Osman, “Improving security of duplicate address detection on IPv6 local network in public area,” in *Proceedings of the 2015 9th Asia Modelling Symposium* (AMS), pp. 123–128, Kuala Lumpur, Malaysia, September 2015.

[12] F. A. Barbhuiya, G. Bansal, N. Kumar, S. Biswas, and S. Nandi, “Detection of neighbor discovery protocol based attacks in IPv6 network,” *Networking Science*, vol. 2, no. 3-4, pp. 91–113, 2013.

[13] R. Hassan, A. S. Ahmed, and N. E. Osman, “Enhancing security for IPv6 neighbor discovery protocol using cryptography,” *American Journal of Applied Sciences*, vol. 11, no. 9, pp. 1472– 1479, 2014.

[14] M. Anbar, R. Abdullah, R. M. A. Saad, E. Alomari, and S. Alsaleem, “Review of security vulnerabilities in the IPv6 neighbor discovery protocol,” *Lecture Notes in Electrical Engineering*, vol. 376, pp. 603–612, 2016.

[15] J. L. Shah, “A novel approach for securing IPv6 link local communication,” *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 136–150, 2016.

[16] Armando Faz-Hernandez, Julio Lopez, “Speeding up Elliptic Curve Cryptography on the P-384 Curve”, *Soc. Bras. de Computacao, SBC,* 2016.

[17] Wang, X., Yin, Y. L., & Hongbo, Y. (2005).Finding collisions in the full SHA-1. Advances in Cryptology–CRYPTO. Berlin/Heidelberg: Springer-Verlag.

[18] Tony Cheneau, Maryline Laurent, Sean Shen, Michaela Vanderveen, “ECC public key and signature support in Cryptographically Generated Addresses (CGA) and in the Secure Neighbor Discovery (SEND),” *draft-cheneau-csi-ecc-sig-agility-01.* [Online]. Available: <https://tools.ietf.org/id/draft-cheneau-csi-ecc-sig-agility-01.html#rfc.authors.>