

## Leveraging Secure Hash Algorithm for Securing IPv6 Protocols SLAAC and DAD

**Jithender Reddy Machana**

Assistant Professor, Computer Science & Engineering, Vasavi College of Engineering  
(Autonomous),

Hyderabad, Telangana, 500010, India

[m.jitenderreddy@staff.vce.ac.in](mailto:m.jitenderreddy@staff.vce.ac.in)

**Dr G Narsimha**

Professor, Computer Science & Engineering, JNTU, Sultanpur.

Hyderabad, Telangana, 502319, India

[narsimha06@gmail.com](mailto:narsimha06@gmail.com)

### Abstract

In the recent past, the IPv6 protocol has gained importance in the industry. The IPv6 protocol is considered more reliable and secured when compared to its 32-bit counterpart. The IPv6 has increased the address length from 32 to 128 bits to address the exhaustion of IPv4 address space. It provides more addresses through address hierarchy and a simpler address autoconfiguration through SLAAC, SLAAC with DHCPv6, and DHCPv6 server. The IPv6 Neighbor Discovery protocol does duplicate address detection, determines neighbor MAC address, finds out the next-hop router, and checks neighbor unreachability. The node comes pre-configured with an IPv6 address. An IPv6 address is made up of two parts: the prefix and the interface id. It is possible to generate the interface-id using an extended unique identifier or at random. The address has to be unique on the local link. The duplicate address detection process tests the address uniqueness on the link. This process is susceptible to many attacks, such as DOS attacks [23], replay attacks. To secure is the main objective in IPv6 networks. We have introduced a novel approach, which optimizes NDP and DAD process security. It employs SHA-512 to check the authenticity of NDP messages on the link. This technique implemented programmatically to secure the DAD process and estimate the resources utilized at a given node. We have also discussed the existing flaws in CGA and proposed two modifications, i.e., replacing the present public key cryptography scheme and hash function. Instead, Elliptical Curve Cryptography (ECC) for the NIST P-384 curve, is recommended for the ECDSA key generation process and SHA-512 in place of SHA-1. ECC with SHA-512 proves to be highly secured and optimal in terms of the consumption of computational resources at the nodes.

**Keywords:** Neighbor 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

The Internet Protocol Version 6 [22] succeeds IPv4 protocol. IPv6 has a 128-bit address length and can provide 340 undecillion addresses. Though NAT is a Temporary solution, it worked fine, but the enormous growth of routing tables has always been an issue with IPv4. With an increase in the Internet population and the advent of IoT and NAT issues, transitioning to IPv6 is no surprise. Hence, it provides one end to another end connection while discarding NAT. IETF has fixed some of the limitations of IPv4 in IPv6. Firstly, IPv6 provides address resolution and address autoconfiguration with ICMPv6 protocol. Secondly, IPv6 decreases routing table size and increases

routing efficiency. Further, the IPv6 header has a simpler size with no checksum, makes the packet processing a bit more efficient at intermediate nodes from source 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 is more efficient than IPv4 ARP address resolution. The reason, it uses a 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 uses the Neighbor Discovery Protocol (NDP) and enables a node to get a unique address in the IPv6 network. However, NDP inherently assumes that all nodes are trusted nodes, but with the advent of insecure wireless networks, the rogue device with minimum credentials can become part of the local link and launch an attack. Therefore, an IPv6 network is susceptible to DoS and DDoS attacks [23] during the DAD process and several other attacks [25] during the NDP process as a whole. In this paper, we have discussed the security problems associated with the IPv6 link-layer. The drawbacks of the CGA address generation process discussed. This paper aimed to propose an algorithm for securing the local-link communications by leveraging Secure HashFunction-512 (SHA-512). Also, dual cryptography with Elliptical Curve Cryptography (ECC) for the NIST (National Institute of Standards and Technology) P-384 curve with SHA-256 is discussed and evaluated. We can implement the DAD process programmatically. The results of securing NDP presented along with computational resources (Time and space complexity) required at a given node.

## 2. Related Work

Many researchers have discovered attacks on IPv6 networks [25] in various scenarios, especially on Duplicate Address Detection (DAD) and NDP processes. As a result, these have attracted researchers over the years; many researchers have proposed algorithms and novel approaches to secure IPv6 link-local communications involving DAD and NDP processes.

In [2], the authors have proposed a mechanism for generating an address. This method takes lesser processing time than the standard Cryptographically Generated Address (CGA) [3] approach, and it is not secured as it employs SHA-1 encryption, which is obsolete as per Google security reports in 2015. It is also susceptible to collision attacks for this reason. In [11], authors have used a new approach to secure the DAD process. They used an alternative approach to CGA and SEND (Secure Neighbor Discovery) [2] protocols, with a security level limitation. In [4], the authors have introduced an algorithm to secure IPv6 addresses by modifying RFC 3972 standard. They have reduced the sec granularity from 16 to 8 and used ECC instead of RSA but implemented SHA-256 hashing. SHA-256 may be compromised soon. To secure the DAD process for vehicular networks [5], introduced another methodology, such as the secure address auto-generation protocol. However, this is only useful when a vehicle and its serving AP are one-hop apart. To verify the received message integrity, the authors [10] have proposed an SDN-controller- based mechanism. It has its limitations and is not efficient. Trust-ND [26] is another approach to secure the DAD. But tests have proved their limitations. We have proposed an algorithm to provide DAD process security and verify the integrity of NDP messages. The results have indicated that it can be optimized by hashing the newly generated IPV6 address with SHA-512 and include it in Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages. It also leverages ECC with SHA-256 for the existing CGA process, using RSA and SHA-1 to improve the algorithm.

## 3. Neighbor Discovery Protocol

Neighbor Discovery Protocol (NDP) is an IPv6 key protocol. It uses the ICMPV6 to resolve the IPv6 address to MAC address at the link layer. It is similar to ARP for IPv4. It makes Stateless address auto configuration possible to configure hosts and gather information about neighboring nodes and discover routers. NDP uses ICMPv6, which is more secured and robust than ICMPv4. Some key functionalities of NDP (RFC 2461) are:

### 3.1. *Router Solicitation*

The host sends a router solicitation (RS) message when an interface is enabled. It requests routers to send router advertisement immediately rather than sending at a scheduled time. The host uses the ICMPv6 packet with type 133 and sends an RS message to all the routers in the link using the destination address as all routers' multicast address ff02::2.

### 3.2. *Router Advertisement:*

Routers advertise RA packets periodically for every 200 seconds. It uses ICMPv6 packet type 134 to provide prefix, prefix length, DNS address and a default route to IPV6 enabled hosts. It uses all node multicast address ff02::1 as a destination address.

### 3.3. *Neighbor Solicitation:*

IPv6 hosts use ICMPv6 packet type 135 NS (Neighbor Solicitation) for duplicate address detection, resolve the link-layer address, neighbor address unreachability, and default route. Once the host configures with an IPv6 address, the node forwards the NS packet.

### 3.4. *Neighbor Advertisement:*

NDP protocols Neighbor advertisement sent in response to the Neighbor solicitation. It is ICMPv6 packet type 136. The nodes respond with NA messages when any node changes its MAC address.

### 3.5. *Redirect:*

Whenever Routers find the best hop route, it informs nodes using ICMPv6 packet type 137 known as redirect message. It is similar to IPv4 ICMP redirect message.

Whenever a new host joins the link, it requests the router for addressing information by sending an RS message. The router responds with a RA message stating to use stateless or stateful address configuration. In stateless auto-configuration, the host creates the new interface-id with EUI 64 bit or generates MAC address randomly and uses the prefix part provided by the router's RA message. IPv6 uses Duplicate Address Detection (DAD) to check IPv6 address uniqueness. The host forwards the NS packet with a new address as a target address. If the address matches with any device, it responds with an NA packet. Else, the host waits for a certain period and assumes the new address as a unique address. This paper mainly focused on the DAD process to get rid of attacks. And implement them as efficiently as possible.

## 4. *NDP Attacks*

The NDP plays a significant role in the functioning of the Link layer. The major drawback is it is prone to attacks. It disrupts the performance of the network. Some of the NDP attacks are MiTM, DoS, and Spoofed Router Redirect Message attacks.

### 4.1. *A Man-in-the-Middle attack (MiTM):*

The hacker sends NS and NA forged messages and poisons the hosts' cache. The attacker gains network access as a trustable host. He enters the middle of the conversation between two hosts. He disrupts by modifying the traffic between them. Here three kinds of attacks are possible: Spoofed ICMPv6 NA and ICMPv6 RA and Replay Attack.

### 4.2. *Spoofed ICMPv6 NA attack:*

In a scenario of 3 hosts in a network, host A wants to know the MAC address of B for the corresponding IP address. It sends an NS message using an all-node multicast message address "ff02::1" as a destination address. The host which belongs to the multicast group responds, if a rogue host is present, it claims that the IP address belongs to it and overrides the B's NS message by setting the flag to '0'. In this way, all the traffic intended for B goes to C.

#### 4.3. *Spoofed ICMPv6 RA attack:*

RA messages are forwarded periodically or in response to solicitation messages. By default, the router sends RA messages every 200 seconds. It is addressed to the FF02::1 multicast address group to configure prefix, prefix length, default gateway, DNS server address, and lifetime.

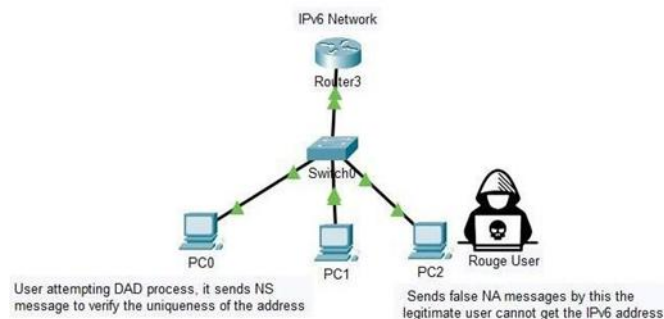
Any device can pretend like a router and send routing advertisements periodically. It can act as a default gateway and can see the traffic which flows through the IPv6 Network.

#### 4.4. *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 message and try to gain network control.

#### 4.5. *DoS Attack:*

In this attack, a rogue user denies services to other hosts. It deliberately disconnects users from accessing websites and network services. It degrades the performance of the network, and legitimate users do not get network access.



**Fig. 1. Understanding DoS attacks during DAD**

When the host configures its interface with an IPv6 address, it checks for the address uniqueness on the link. The Host interface id can be configured manually or through EUI, and random number generation. However the interface is configured, it has to check for the address uniqueness in the link. To check, it uses the DAD process, which is vulnerable to attacks. A rogue user exploits the DAD process with a DOS attack. When the host sends a NS message as part of the DAD process, as discussed earlier, the attacker replies with an NA message. On receiving the NA message, the host doesn't configure a tentative address and generates a new address. Then the DAD process is initiated again. The rogue user repeats with an NA message and makes sure that the host cannot configure with an interface address.

### 5. Existing CGA Mechanism

RFC 3972 specifies a method to bind an IPv6 address with a cryptographic public key in the SeND protocol. The interface- id generated cryptographically using a hash function. CGA implementation requires no third-party authorization. At receiving end, it verifies by recomputing the address binding and public-key. The messages are encrypted with the nodes private key before transmitting on the local-link.

The receiver has to know the source address and public key to authenticate and decrypt the message. An RFC on CGA was drafted in the year 2005.

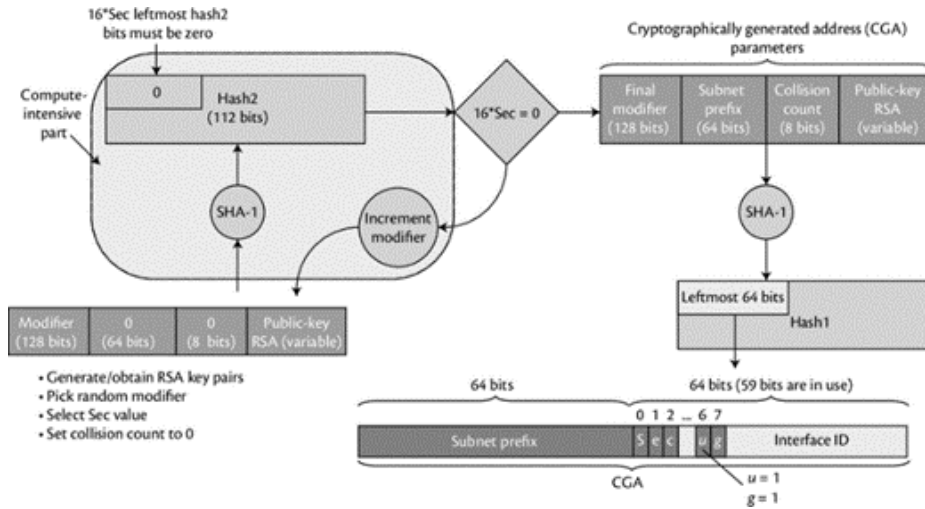


Fig. 2. The Standard CGA Process

“SeND” has many limitations though it addresses many security issues of Neighbor discovery protocol. Firstly, CGAs require heavy computational time and are impractical for sec(security parameter) values greater than 1. Secondly, the public-key selection plays a role 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 take shorter generation time, usually provide medium security, which is the case with the standard CGA mechanism. Finally, the hash function selection impacts security. It also affects the performance of CGAs. CGA employs SHA-1, which uses a 160-bit hash function. which uses an hash function of 160 bit. It is prone to collision-free attacks. CGAs are also prone to Denial-of-Service (DoS) attacks, as mentioned in the previous section. A node is flooded with constant messages at high frequency to disable it from providing services.

## 6. Restrictions Of SeND Protocol

SeND has quite a few limitations which actually restrict its deployment in IPv6 networks. CGA generation is processing intensive which increases the processing costs for both attacker and the legitimate generator of the address. Using brute-force search, the owner of the address requirements  $O(2^{16*sec})$  hash meaning computations towards discovery the correct transformer worth besides content the Hash2 complaint in CGA for their part, an attacker may use a network node's public/private key information to assume their identity. There must be a 1:1 match between the hash 1 values of this key pair and the imitated lump. To get there, though, the attacker needs to achieve  $(16*sec+59)$  hash function computations addition a charge of  $(2^{16*sec+59})$ . For the calculation of the right Hash 1 by the attacker, time required is  $2^{59}*T1$  where T1 is the time required to compute Hash 1 with a given key pair. Once the right Hash 1 is obtained, a valid modifier is to be obtained by satisfying the conditions of Hash 2 which receipts  $2^{16*sec}$  hash purpose computations. Thus the entire period required by the attacker when starting from Hash 1 equals:

$$TH = (2^{59} * T1 + T2)2^{16*sec}, \quad (1)$$

where T2 is the time required to compute Hash 2.

When beginning with Hash 2 evaluation, it takes the attacker  $2^{16*sec} * T2$  time to satisfy the Hash 2 conditions of CGA. Once satisfied, Hash 1 can be verified in  $2^{59}$  computations. Thus the total time required by the attacker when beginning with Hash 2 equals:

$$T_H = (2^{16*sec} * T_2 + T_1)2^{59}$$

(2)

Hence the final impersonation time required by the attacker is given by:

$$T_I = \min \{ (2^{59} * T_1 + T_2)2^{16*sec}, (2^{gf*sec} * T_2 + T_1)2^{59} \}$$

(3)

Where  $T_I$  is the impersonation time required and  $gf$  is the granularity factor selected for the CGA algorithm.

Sec clearly has an impact on the CGA's overall strength. It takes longer to create an address if the sec value is large, and it is less secure if the sec value is low.

## 7. CGA Limitations

### 7.1. Safety Constraint or sec Value

The robustness of the CGA method and the time it takes to generate addresses are directly influenced by the security parameter. For every additional one second, address creation time increases by a factor of one  $2^{16}$ . To keep network Quality of Service high, handover operations must be executed in milliseconds or less. Mobile devices, with their constrained processing power, bandwidth, and energy, also need efficient resource use. As found by [6], this limitation can only be fulfilled on mobile devices if the sec value is 0. In accordance with [6] a desktop computer can't handle a second number greater than 1.

### 7.2. RSA Key Pair Crypto System

Not only the sec value but also the assortment of a proper public key crypto organization plays a huge character in security also computational delay injected by the CGA algorithm. The CGA generation time increases with an increase in key size. Hence the CGA compares time is essentially predisposed by the key-pair size. Due to this reason, the authors suggest the use of ECDSA and ECC (Elliptic Curve Cryptography) as a substitute to RSA which reduces the key generation time owing to its smaller key lengths which provide the same level of security. It also helps in reducing the packet size in low-bandwidth applications.

RSA Key length (bits)	ECC Key length (bits)
1024	160
2048	224
3072	256
7680	384
15360	512

**Table 1. Key-size equivalence between RSA and ECC.**

<b>RSA Key Length</b>	<b>1024</b>	<b>2048</b>	<b>3072</b>	<b>7680</b>
DER Encoded RSA public key length (bytes)	160	292	420	996
CGA parameter data structure length (bits)	1480	2536	3560	8168
Number of 512 bit blocks	4	6	8	17
<b>ECC Key Length</b>	<b>160</b>	<b>224</b>	<b>256</b>	<b>384</b>
Octet Encoded Public Key Length (bytes)	66	80	88	120
CGA parameter data structure length (bits)	728	840	904	1160
Number of 512 bit blocks	2	2	2	3
Number of bits saved in ECC (bits)	752	1696	2656	7008

**Table 2. Comparison of CGA parameter data structures lengths using RSA vs ECC public keys.**

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 named Curve parameter in the ECC parameters field of AlgoIdentifier.

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 [8] the authors have proposed plausible software techniques for accelerating cryptographic operations using the P-384 curve. The equation of the curve is given by:

$$(4) \quad y^2 = x^3 + ax + b$$

Where

b=27580193559959705877849011840389048093056905856361568521428707301988689241309860865  
136260764883745107765 439761230575

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

### 7.3. 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 rightly points out the chances of collision attacks. Hence, replacing it with more secure functions such as SHA-256 or SHA- 512 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.



Algorithm	Output Size (Bits)	Internal State Size (Bits)	Block Size (Bits)	Max message Size (Bits)	Rounds
SHA 1	160	160	512	$2^{64} - 1$	80
SHA512	512	512	1024	$2^{128} - 1$	80

**Table 3. Simple comparison between SHA1 and SHA512.**

#### 7.4. DoS and Other Attacks

DoS attacks may cause a denial of service (DoS) on a CGA, according to [16]. A denial-of-service attack may be launched in a variety of methods by an attacker: DoS against DAD-CGA or a replay attack to bring the node down. Each time a node sends a tentative address, the attacker acknowledges it with an acknowledgement. This exploit will prevent the targeted node from setting up an IP address after three unsuccessful tries. According to [16], the DAD should be destroyed after three failed attempts using the same tentative address and CGA parameters. It's very unlikely that two nodes in a network would have the identical CGA data structure.

The probability that two nodes would produce the same address using values obtained from the [19]th birthday conundrum is provided by:

$$(n, k) \leq 1 - \left( \frac{n - k + 1}{n} \right)^{k-1} \tag{5}$$

Where  $n = 2^{59}$  also  $k$  is the amount of boundaries on the connection. For a large subnet, let's say  $k=100000$  then  $P(2^{59}, 100000) \leq 1.7e^{-08}$ . This means the heuristics are correct since the value is low. Another method an attacker may use to take down a system is to send a large number of legitimate or erroneous messages in quick succession across the network to a CGA node. The host node is kept active throughout verification, wasting computer resources and time.

### 8. Proposed Algorithm

Now let's comprehend the algorithm: The details of securing the DAD process and mitigating DOS attacks during the duplicate address detection process when we use stateless address auto-configuration (SLACC) for configuring the IPv6 address. These two processes are involved, one at the source end and another at the destination end. The source forwards NS, and the receiver responds with an NA if the target address match, else it simply discards.

#### 8.1. Algorithm Explanation

##### 8.1.1. Acronyms Used

- NC = Number of collisions
- UF = Unique Flag
- NS\_C = Neighbor Solicitation Collisions
- RaNo = 64 bit random number
- CT = Clock time or the time at the generation
- I\_ID = Interface Identifier of the link-layer address
- LLA = Link Layer Address
- T\_IP\_IID = Interface Identifier in the target IP header field
- LLA\_D\_IP = Destination link layer address
- IID\_DT = Destination interface Identifier in the target IP field.

#### 8.2. The Process at the sender node:

- 1) Initially set the values as  $NC=0, NS\_C=0, UF=1, CT$ .
- 2) Generate a 64 bit random number and assign it to RaNo.



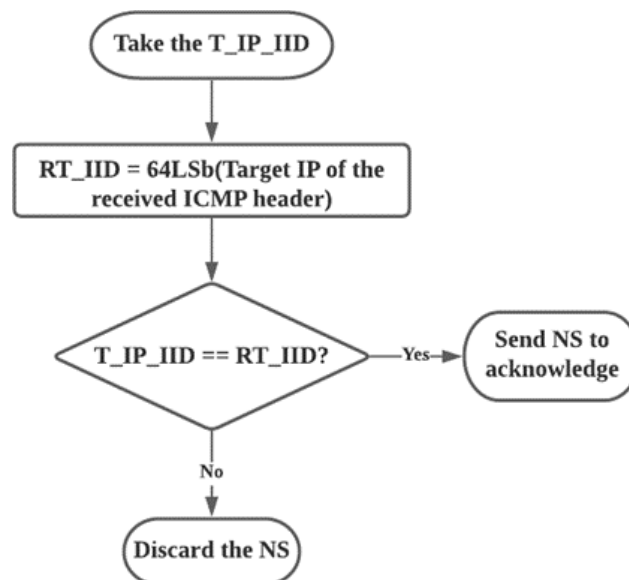
- 3) Apply SHA-1 to concatenate (NC, NS\_C, CT, and RaNo), and the result isHash-1.
- 4) Divide Hash-1 into two equal parts as Div\_hash1 and Div\_hash2.
- 5) Create an interface-id(I\_ID) including 20MSB from Div\_hash1 and Div\_hash2, and 24 LSB from the generated Rano given in Step 2.
- 6) Now combine the interface Id (I\_ID) with the network prefix and perform the DADprocess.
- 7) Apply SHA-512 on the interface-id(I\_ID), and the result obtained is Hash-2.
- 8) Now form the T\_IP\_IID that is the target IPv6 address in the ICMPv6 header field by taking 40 MSB of SHA-512 and 24 LSB of the generated link-local address.
- 9) The mark IP address field in the icmpv6 header is formed through concatenating the local system precede with IP\_T\_I\_ID to become a 128-bitaddress.
- 10) And now, perform the DAD process on the generated address for its uniqueness on the local link by sending a Neighbor Solicitation(NS)message.

The ICMPv6 type 134 message sends the source address as unspecified address and destination address as solicited-node multicast address FE80::1.

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 in the flowchart. [16] shows that the probability of occurrence of two nodes generating the same interface identifiers is given by

$$P(n,k) \leq 1 - \left( \frac{n-k+1}{n} \right)^{k-1} \tag{6}$$

where n = number of possible address combinations, k = total number of interfaces on the same link. In other words, finding three address collisions in the NC or NS C variables implies malicious behavior.



**Fig. 3. Process at the receiver node.**

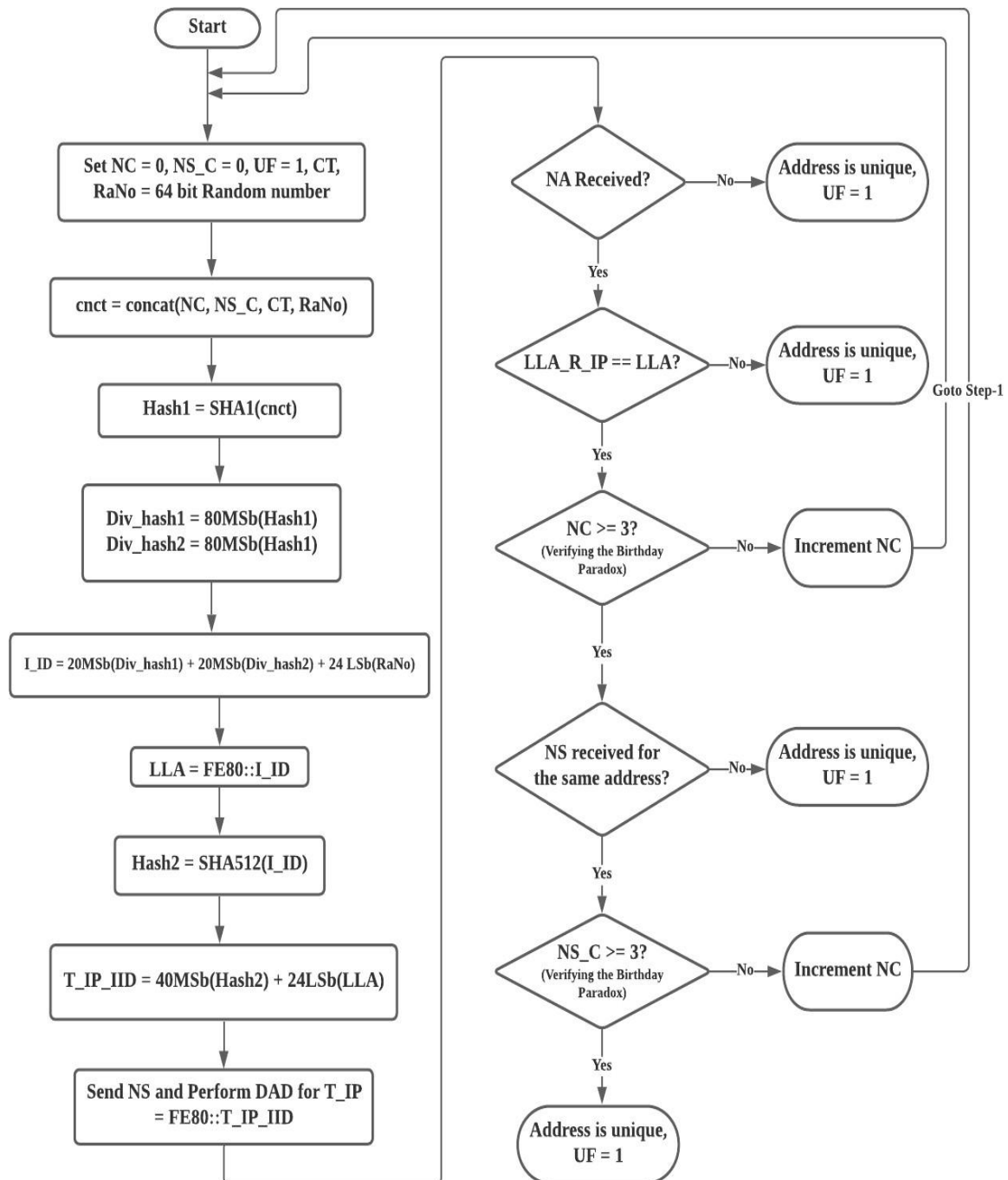
### 8.3. At the receiver node

The nodes that receive the solicited-node multicast NS message do the following process:

- 1) You must assign an RT IID to each IP address you send and receive data from.

2) Get the value of T\_IP\_IID from the file that was generated when this node joined the network and formed its address for the first time.  
 If T\_IP\_IID = RT\_IID, then send the Neighbor Advertisement message to acknowledge the new node. Else discard the Neighbor Solicitation.

**8.4. Proposed Algorithm Flowchart**

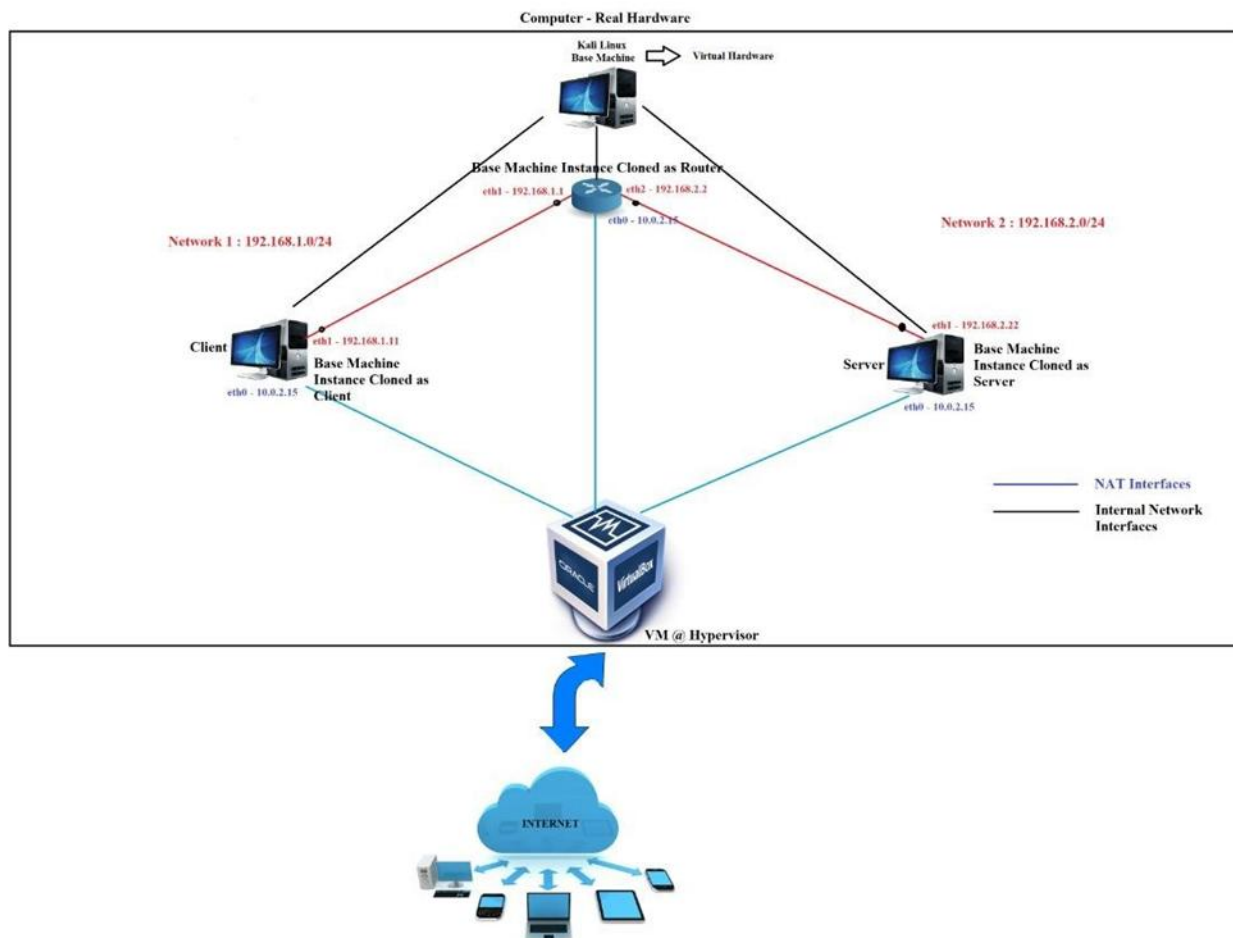


## 9. Implementation

This section gives a detailed explanation of the implementation. To demonstrate, we have used a topology comprising of a Router, Switch, and 3 Hosts (sender, a receiver, and a hacker). Fig. 4. depicts topology.

To demonstrate the idea, designed the topology using the emulation tool GNS3. It consists of a router, a switch, and three host machines. "Cisco IOS image 3660" was installed on the router. Docker machines consume less primary memory than QEMU/Virtualization technologies, used as hosts in the topology. The hypervisor "VMware", was installed to support virtualization. We have used the networking tool Scapy to craft and manipulate packets.

It served our purpose to demonstrate: the DAD process, the DOS attack, and replay attacks. The script running on the hacker side shows the shortfalls of the DAD protocol.



**Fig. 4. Topology For Implementation**

The hacker side shows the shortfalls of the DAD protocol. The script implements the DoS attack acknowledging each neighbor's 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 a tentative address as an IID.

## 10. Results and Comparison

The below table shows the comparison of standard CGA and the algorithm employed for IPv6 address generation. The comparison is on address generation times. As discussed in the previous section, it is implemented, in a virtual environment, with three hosts running Kali Linux. To

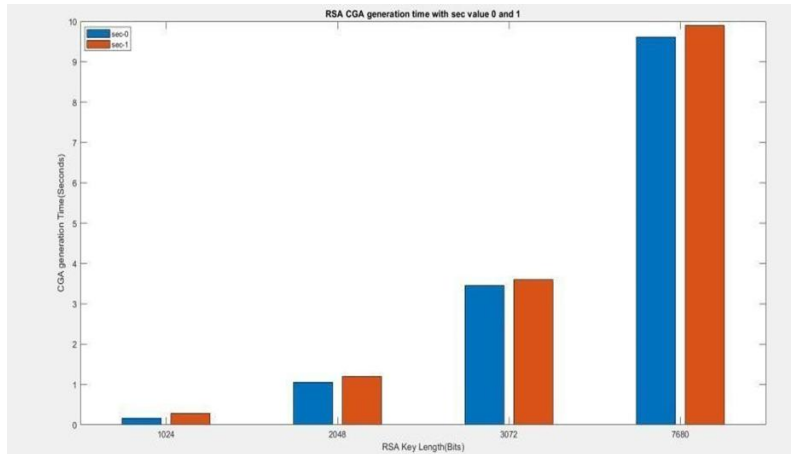
## Leveraging Secure Hash Algorithm for Securing IPv6 Protocols SLAAC and DAD

calculate and compare address generation times of CGA and the proposed algorithm, the tests were conducted on an i5-1035G1 processor. That operates at a clock frequency of 1GHz. The CGA algorithm, implemented for SEC values of 0 and 1 and complemented with RSA and ECDSA of varying key lengths, is shown.

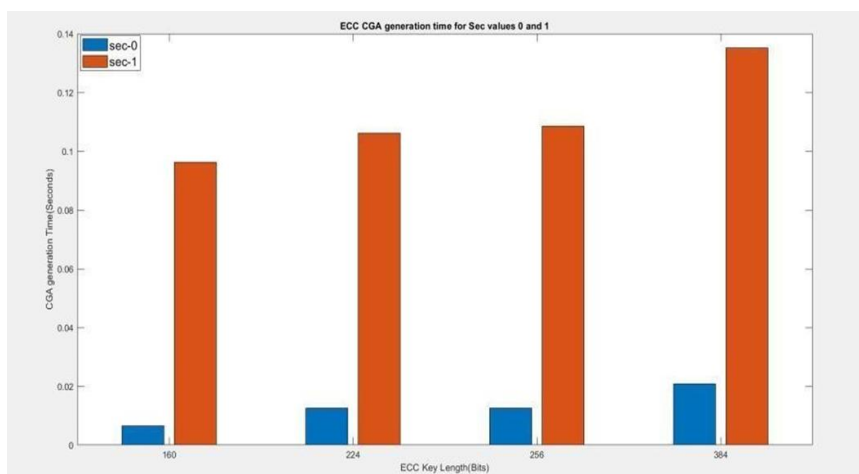
The following tables give the results thus obtained.

Number of Samples = 1000				
SECURITY LEVEL (SEC)	0			
RSA KEY LENGTH (bits)	1024	2048	3072	7680
CGA GENERATION TIME (seconds)	0.163964	1.055813	3.457668	9.610627
SECURITY LEVEL (SEC)	1			
RSA KEY LENGTH (bits)	1024	2048	3072	7680
CGA GENERATION TIME (seconds)	0.281018	1.194061	3.601473	9.899951
SECURITY LEVEL (SEC)	0			
ECC KEY LENGTH (bits)	160	224	256	384
CGA GENERATION TIME (seconds)	0.006449	0.012602	0.012622	0.020802
SECURITY LEVEL (SEC)	1			
ECC KEY LENGTH (bits)	160	224	256	384
CGA GENERATION TIME (seconds)	0.096317	0.106154	0.108551	0.135056
<b>PROPOSED ALGORITHM</b>				
SHA-1 for both generation and encryption	0.008038 seconds			
SHA-1 (generation) & SHA-256 (encryption)	0.018592 seconds			
SHA-256 for generation and encryption	0.024986 seconds			
SHA-1 (generation) & SHA-512 (encryption)	0.034949 seconds			

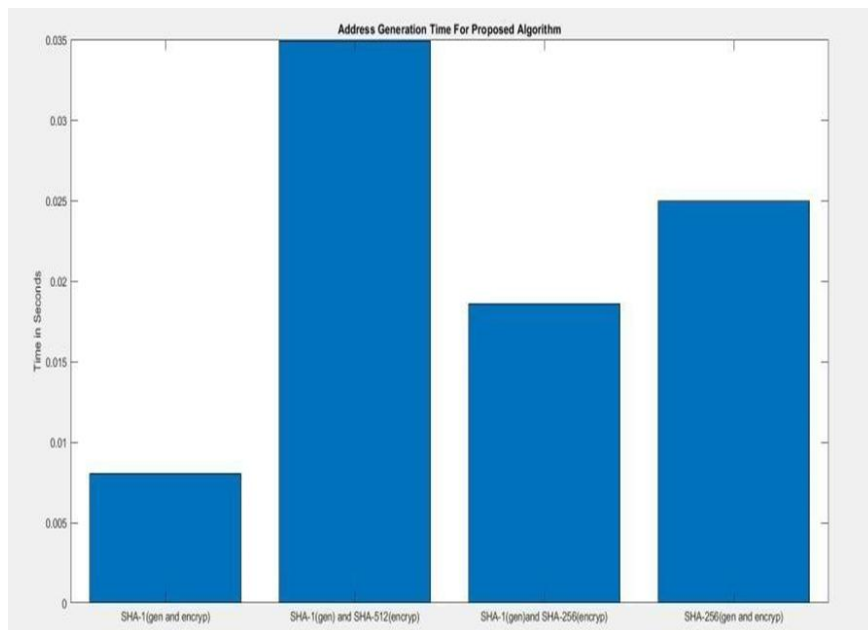
**Table 4. CGA and proposed algorithm time complexity.**



**Fig. 5. Address generation time for RSA with sec values, 0 and 1.**



**Fig. 6. Address generation time for ECC with sec values, 0 and 1.**



**Fig. 7. Address generation times for the proposed algorithm based on the function used for address generation and encryption.**

We observed that the proposed algorithm takes at least one hundred msec less for an address generation and encryption as compared to the traditional CGA with ECDSA. It is mainly because the algorithm does not have an overhead of a crypto-key pair generation. Also, the difference between the generation time decreases 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. The attacker should do for each possible network prefix and the interface identifier. Besides, to impersonate and attack a node in the network, more storage is required. The conclusion is that the hacker cannot attack with the nodes employing proposed algorithm.

## 11. Conclusions And Future Scope

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 a cryptography key- pairs generation. Also discussed the substitution of ECC for RSA of the CGA algorithm. 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 DAD and Neighbor Discovery protocols. In recent times, SDN has been a pioneer networking procedure. It handles critical tasks smoothly without the administrator's intervention.

### Acknowledgements:

I am grateful to the Computer Science Head of the Department, Principal, and Management of Vasavi College of Engineering, Hyderabad, Telangana, for providing the resources to conduct the research work.

### References

- [1] T. Narten, E. Nordmark, W. Simpson, H. Soliman. *Neighbor Discovery for IP Version 6 (IPv6)*. Internet Engineering Task Force.
- [2] J. Arkko, J. Kempf, B. Zill, P. Nikander. *SEcureNeighbor 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] 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.
- [6] 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 Theory and Application of Cryptology and Information Security, Advances in Cryptology –ASIACRYPT 2015*, Auckland, New Zealand.
- [7] 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.
- [8] X. Wang, Y. Mu, G. Han, and D. Le, "A secure IPv6 address configuration protocol for vehicular networks," *Wireless Personal Communications*, vol. 79, no. 1, pp. 721–744, 2014.
- [9] Y. Lu, M. Wang, and P. Huang, "An SDN-based authentication mechanism for securing neighbor discovery protocol in IPv6,"
- [10] *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*-[Online]. Available: <https://tools.ietf.org/id/draft-cheneau-csi-ecc-sig-agility-01.html#rfc.authors>.
- [19] Bagnulo, M., Soto, I., Garcia-Martinez, A., & Azcorra, A. (2002). Random generation of interface identifiers. [Online] Available: <https://tools.ietf.org/html/draft-soto-mobileip-random>
- [20] S. Thomson, T. Narten and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", *RFC4 862*, September 2007.
- [21] Christian Vogt, *Source Address Validation Improvement Protocol Framework*, [online] Available: <http://tools.ietf.org/id/draft-vogt-savi-framework-01.txt>