Chapter 6

Securing IPv6 Stateless Address Auto Configuration

6.1 Introduction

The driving force in minimizing the deployment cost as well as increasing the usability of naive users is adopting the Self configuration of nodes.Varied network technologies are already into self configuration like Ethernet. With introduction of IPv6, prodigious amount of devices are getting connected to the Internet, thus necessitating demand for plug and play support in the network. However this is not true for backbone and multiservice networks. The support forself configuration of nodes increases the number of network elements managed per person or the time required to operate the network. Moreover they elevate the user friendliness for inexperienced users, thus enabling them to run large networks with trivial networking knowledge. In general, self or auto configuration increases the autonomy of network elements by taking away significant amount of network management functions, thus minimizing the synergy level between centralized server and the node(Tobella et al,2004).

The motivating factor towards adoption of IPv6 has been the provision for Auto Configuration. The introduction of Stateless Address Auto Configuration in IPv6 allows the hosts automatically configure their interface identifiers. Using Auto Configuration, innumerable number of hosts successfully configure their interfaces and get connected to global internet without interfacing with any centralized DHCP server. This approach makes it a user-friendly and economically feasible technique. However the auto configuration process is not free from security vulnerabilities present in Neighbor Discovery Protocol (NDP)(Baig&Adeniye, 2011).

The protocol is rooted on the assumption that network consists of trusted nodes, but with emergence of public wireless networks; any node can join the link with minimal authentication and the condition changes drastically. With no inclusion of central address configuration servers or trusted authorities, the process is vulnerable to malicious activities against the legitimate nodes that are in the process of joining the network(Rostanski&Mushynskyy, 2013). ThroughDenial of Service (DoS) on Duplicate Address Detection (DAD), the malicious nodes in the network can hinder legitimate nodes to join the network by broadcasting incorrect responses to node requests for verifying the uniqueness of an address. Also Man-in the-Middle (MITM) attacks can be launched by which a node will be configured with incorrect Link Layer

Address.This usually happens when the user’s neighbor cache is poisoned with spoofed Neighbor Advertisement messages of ND Protocol. Thus access to the link can be blocked and the network traffic can be redirected without the knowledge of users. To overcome the above problem, RFC 3971 suggests the use of Cryptographically Generated Addresses (CGA) which is an innate component of Secure Neighbor Discovery (SEND).Although CGA provides for message integrity, authentication and mitigating address impersonation, the process is computation intensive with higher bandwidth consumption and harbors some other limitations(Alsadeh et al, 2012). This chapter explicates discussion over the IPv6 Stateless Address Auto configuration Mechanism and explains the problems associated with it. We also highlight the IPv6 privacy issues and the drawbacks of earlier problem solving approaches. Later, in this chapter we propose a new address generation technique that has a minimum computational cost and time complexity as compared to CGA. The technique maintains nodes privacy and is also secure against DoS attack during the Duplicate Address Detection phase. In particular, the technique is effective in mitigating duplicate addresses in local subnet. The proposed technique is implemented and evaluated in C#.NET.

6.2 Stateless Address Auto Configuration (SLAAC)

The IPv6 Neighbor Discovery Protocol (NDP) is an indispensable component used for critical functions such as auto configuration, detecting end systems on the local link, resolving Link Layer Addresses and obtaining network prefix values. Among the key functions is the IPv6 SLAAC (Stateless Address Auto Configuration) which forms an integral component of Neighbor Discovery protocol (Narten,Thomson &Jinmei, 2007).

SLAAC adds a unique characteristic in IPv6 networks. In IPv6, a node configures its address through one of the following ways (Ladid, 2009); using manual or fixed address configuration, stateful disposition using DHCPv6 or stateless auto configuration. In static/manual address configuration, the node configures its address using the configuration file present on the system. Stateful configuration makes use of DHCPv6 servers which maintain database of IP address assignments and provides mechanism of passing reusable IPv6 addresses and other configuration parameters to network nodes. This method of configuration is similar to IPv4 where IP addresses assigned to routers and servers rarely change. In stateful configuration, when a client node starts booting in IPv6 network, it generates its Link Local Address (LLA) and multicasts a Router

Solicitation (RS) message to the all routers multicast address FF02::2. If a router replies with Router Advertisement (RA) with its ‘Managed Configuration Flag’ set or if there is no Router Advertisement after sending several Router Solicitations, then the node proceeds by sending a DHCP solicit message to all-DHCP agents multicast address FF02::1:2 to track available DHCP servers in the network. If DHCP server is found to be on the same network as the requesting node, the server then responds with a DHCP advertise message, or else a DHCP Relay forwards the request to any of the available DHCP servers on the network site multicast address FF05::1:3. A DHCP server then responds to the client via the relay (Kaur & Maini, 2010).

The Stateless Address Auto configuration is a decentralized mechanism unlike DHCPv6. The potential of stateless auto configuration mechanism to operate without involving any central address configuration servers or trusted authorities makes it an ideal choice for users (Blanchet, 2009).Using SLAAC; the node configures its address by following steps:

A node first forms its Link Local Address. The LLA of a node is generated by concatenating the 64 bit interface identifier with 64 bit link local prefix FE80::/64.A standard for generating the 64 bit Interface Id(IID) has been proposed by IEEE standards association (IEEE,2012) known as EUI-64 (Extended Unique Identifier).They are created by the combination of 36 bit OUI (Organizationally Unique Identifier) which is allocated by IEEE RA(IEEE Registration Authority) and 28 bit extension identifier allocated by the hardware manufacturers. The bits 7 (Universal/Local (U/L) bit) and 8 (Individual/Group (I/G) bit) of the leftmost byte of EUI are set to one. This resultant forms a 64 bit interface identifier. However if we have only 24 bit OUI and 24 bit extension identifier (i.e. a 48 bit MAC address), then a IEEE reserved hexadecimal value (0xFFFE) is inserted between the third and fourth byte to form a 64 bit EUI. For example; if we have a node with MAC address equal to 00-42-21-68-7E-5A, then by embedding FF-FE into the middle of 48-bit MAC address and inverting the uniqueness bit to 1 will result in 64-bit IID 0242:21FF:FE68:7E5A. Thereafter by concatenating Network prefix FE80::/64 with the IID results in the formation of Link Local Address FE80:: 0242:21FF:FE68:7E5A. This is shown in figure 6.1.

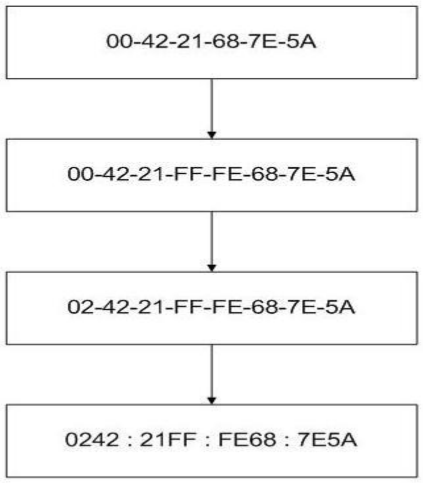


Figure 6.1 EUI-64 Interface Identifier Generation

After forming the address; the node performs uniqueness test on its generated address by performing Duplicate Address Detection (DAD) check. The node sends a Neighbor Solicitation message to corresponding solicited node's Multicast Address. The solicited node's multicast address is created by extracting low-order 24 bits of link local address and affixing those bits with prefix FF02:0:0:0:0:1:FF. For Example, if we have our link local address as FE80::2CB:DA:EC37:8D4A, then its equivalent solicited node's multicast address would be FF02::1:FF37:8D4A. This process ascertains the uniqueness of its generated Link Local Address (also known as Duplicate Address Detection check). However, if another node in the network with the same Link Local Address exists, Neighbor Advertisement message is sent back to the source node to inform it about the Duplicate Address Detection. If Duplicate address is not found, the source node assumes the address to be unique and therefore progresses to acquire the network prefix value. This is done by directing a Router Solicitation Message with destination set to FF02::2 (All routers on the Local Link).The network prefix value is obtained from the Router Advertisement message sent by the routers to FF02::1 (All nodes Multicast Address), for enabling a node to form its Global Unicast IP address. This process is depicted in figure 6.2. Address Collisions are very unlikely to be encountered during the auto configuration mechanism unless there is an attack being leveraged. This is because the interface or host identifier unit of address is formed with the help of unique 48 bit hardware address (used when the Interface

Identifier is generated using EUI-64) or with randomized interface identifier obtained by applying a Pseudorandom function.

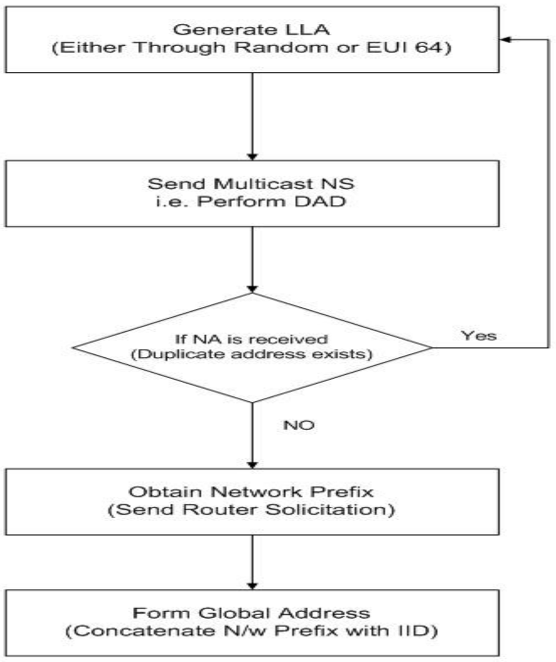

Generate LLA
(Either Through Random or EUI 64)
Send Multicast NS
i.e. Perform DAD
If NA is received
(Duplicate address exists)
Yes
NO
Obtain Network Prefix
(Send Router Solicitation)
Form Global Address
(Concatenate N/w Prefix with IID)

Figure 6.2 IPv6 SLAAC Process

6.3 Security and Privacy Implications in SLAAC

As discussed before, in order for a node to validate the uniqueness of its generated link local address, the node must execute the Duplicate Address Detection (DAD) process. Since the SLAAC process assumes that network consists of trusted nodes, this may create avenues for malicious activities and serve as a launch pad for attacks. One such attack is the Denial of Service (DoS) attack on IPv6 SLAAC (Caicedo,Joshi &Tuladhar, 2009). In this attack, a node trying to generate an address for itself may be blocked from forming such an address by a malicious user in the network. For example; if an attacker is successful in responding i.e. sending Neighbor Advertisement (NA) to every Neighbor Solicitation (NS) message (during the DAD process) sent by a new node, the node may not be able to obtain the address. This is shown in figure 6.3.The attacker can assert ownership of address in one of two ways (Barbhuiya et al, 2013).

The attacker can reply with a NA message indicating that the address is already been assigned. The attacker can reply with a NS message indicating that it is also performing the DAD process for the same address. In that case, both the nodes should drop the address and wait for some time until the new address is generated. The new address should again be verified using DAD (Narten et al, 2007). Repeated acknowledgements to generate link local addresses may deny the new node from joining the link and the node would thus remain uninitialized.

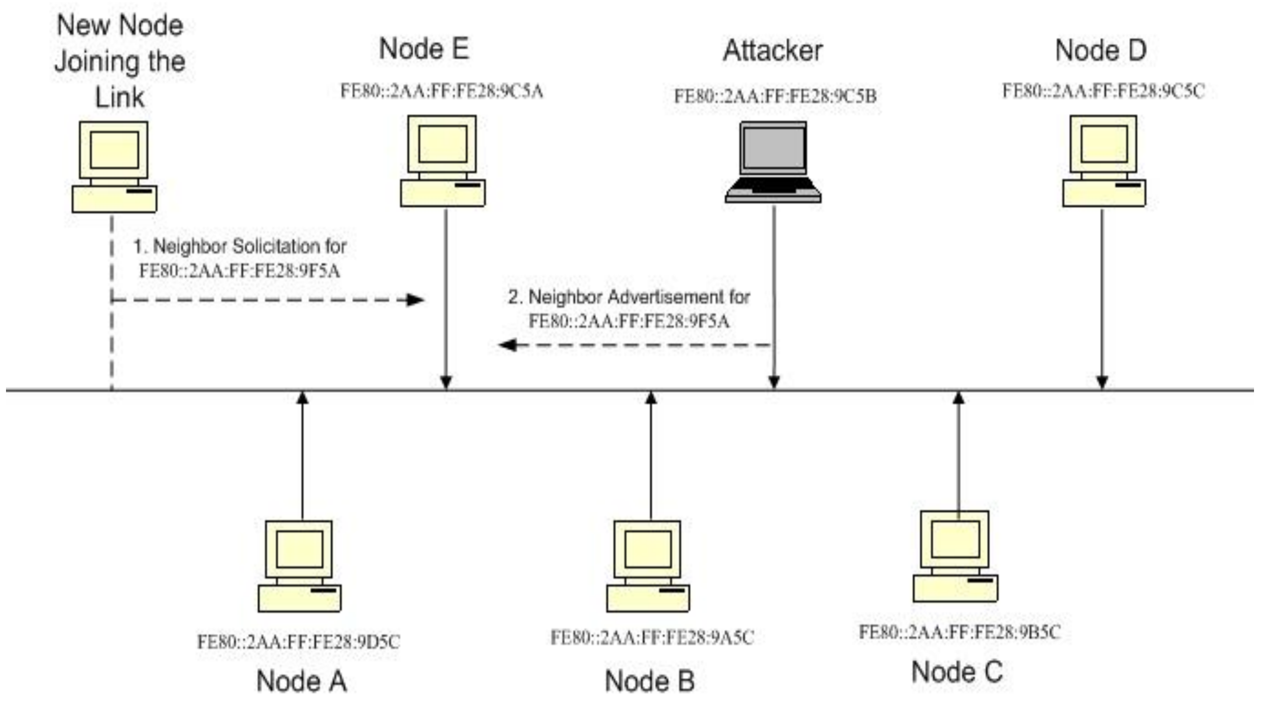


Figure 6.3 DAD Attack in IPv6 SLAAC

In addition to DoS attack on DAD, the IPv6 SLAAC method of interface id generation has some privacy issues and drawbacks. For example; using MAC address for generating Interface Id (IID) results in the formation of a static IID which does not change over time. This issue makes it vulnerable to some privacy attacks. The attackers can have enough time to track the node by capturing the network traffic. Once the node gets identified, the attackers can launch different types of attacks. To resolve this issue, RFC 4941 ”Privacy Extensions for Stateless Address Auto configuration” suggested the use of Randomized Interface Identifiers that change over time. Two techniques for generation of IID were proposed. The first mechanism requires the presence of a stable storage area. A node chooses the IID value from history value of the preceding iteration of the algorithm. If stable storage is absent i.e. implying no history value; then node chooses a randomized value. The IID value so obtained is combined with the EUI generated value. The node computes the MD5 message digest over the quantity created in previous step. Thereafter it extracts 64 leftmost bits of the MD5 message hash and sets leftmost bit 7 to zero indicating the local significance.

The node then compares the generated identifier against a list of reserved IID's and to those already assigned. If a match occurs, 64 rightmost bits of the message digest are saved to history and algorithm restarts. If match doesn't occur; it will use this as final IID.

In the absence of a stable storage area, no history value is available and must be generated randomly by using a good pseudorandom number generator function. A number of different approaches are possible (Narten et al, 2007); for example the nodes do have configuration information (e.g. user id, security keys, serial numbers etc) that vary from one machine to another. This information can be appended with some random data and the MD5 hash can be computed. The main drawback with privacy extension approaches is that they fail to prevent IP spoofing attacks and are unable to provide proof of address ownership by a node.

A substantial amount of research is being done to address the above mentioned drawbacks with the focus on mitigating and discovering new preventive mechanisms. The earlier NDP architecture mandated for utilizing the IPSec services to shield NDP messages; however there are some potential problems like bootstrapping, because Internet Key Exchange (IKE) entails for a working IP stack (Barbhuiya et al, 2013).To configure security associations manually in IPSec is a cumbersome and unrealistic task considering the bulk amount of messages in NDP. Therefore before using IKE, the nodes should be addressable and must have a valid IPv6 address. RFC 3971defines the use of SEND (Secure Neighbor Discovery) protocol for protecting IPv6 SLAAC. SEND offers features like address ownership claim mechanism, NDP message protection and router authorization. The SEND protocol comes with four different option headers like CGA, RSA Signature, Nonce and Timestamp. To counter DoS attack in IPv6 SLAAC, it was proposed that CGA along with signed DAD and NA messages should be used. These messages are subject to validity check and if the validation fails, the node simply drops the NA message. However the protocol is yet to attain a maturity level given the overheads associated with the protocol, which result in DoS itself. The CGA-DAD messages can be subjected to DoS attacks using CGA or non-CGA addresses. The CGA implementation also may not be suitable because they require a node to have a set of Public/Private keys. Thus the nodes can still be identified by their public key. Also CGA generation is a computationally intensive task and may not be feasible. Further discussion on CGA constraints and the drawbacks and discussions on using different privacy extension mechanisms and techniques are further explained in (Alsadeh et al, 2012).

6.4 Proposed Model

In this section; we introduce our technique which generates a highly randomized interface identifier needed for maintaining privacy. The technique guides through the address generation process and is secure against the DoS attacks during IPv6 Duplicate Address Detection phase in SLAAC. The technique is based on the assumption that by making Interface Identifier very difficult to approximate, the attacker will be unsuccessful in determining node's location and hence unable to leverage attacks on the node. The technique is composed of two parts; address generation which is done at the sender node (the node which sends the neighbor solicitation) and address verification which is carried at receiver node (the node that processes the neighbor solicitation sent by other node).We also assume that validity of IID is for a limited time period after which that node generates a new IID. This step increases the complexity level for attacker thereby making it arduous to guess the address of the new node and thus prevent eavesdropping. To generate a unique and robust link local address, the node needs to perform the following steps:

Acronyms Used:

Cc: Collision_Count	LLA: Link Local Address
Uf: Uniqueness_Flag	T_IP: Target_IP_Address
NS_cc: Neighbor_Solicitation_Collision_Count	CRn: Concatenated_Random_number
Rn: Random_number	T_IP_IID: Target_IP_Interface_Identifier
Ts: Current_Timestamp	R_LL_IP: Received_Link_Local_IP
IID: Interface_Identifier	RT_IID: Received_Target_IP_Interface_ID

Table 6.1 Proposed Algorithm Acronyms

At Sender Node:

1. Set Cc = 0, Uf = 0, NS_cc = 0.
2. Set Rn = 64 bit random number (by Pseudo random number generator).

3. Set CRn = Concatenation of [Cc, NS_cc, Ts and Rn] where Ts is the current timestamp of the system.
4. Apply SHA-1 to CRn and put result in Hash-1
5. Break Hash-1 into two equal parts: Sub_Hash-1 and Sub_Hash-2
6. Form IID by concatenating 20 MSB of Sub_hash-1, 20 MSB of Sub_hash-2 and 24 LSB of original 64 bit Random number i.e. Rn.
7. Concatenate 64 bit Link Local Network prefix FE80:: with IID to form 128 bit LLA (Link Local IP address). This address is the temporary generated Link Local IP Address of the node. This will be become permanent after executing duplicate address detection.
8. Apply SHA-1 to IID and put result in Hash-2.
9. Form T_IP_IID for Target IP address field of ICMP header by concatenating 40 MSB of Hash-2 and 24 LSB of Generated 128 bit Link Local IP address.(LLA) .The 40 MSB of generated IID in step 6 are now encrypted. Save the value of T_IP_IID to file. The value is used later for sending Neighbor Acknowledgments for nodes that are soliciting for generated addresses.
10. Concatenate 64 bit Link Local Network prefix FE80:: with T_IP_IID to form 128 bit T_IP field of ICMP header. This is the Target IP address of ICMP header.
11. Perform DAD (Duplicate Address Detection) on T_IP i.e. Send Neighbor Solicitation (NS) for T_IP. This T_IP is verified by nodes for duplicate address that receive the multicast transmission. The actual address that is being verified are 40 MSB of generated IID in step 6 which are encrypted in the form of 40 MSB of T_IP_IID. The source address of IP packet header will contain unspecified address (::) because the node is doing DAD. The destination address will be set to solicited node multicast address corresponding to target IP address (T_IP).The value of generated T_IP will be placed in the target address field of NS message. Hence 40 MSB of IID generated in step 6 are actually encrypted.
12. ***If*** Neighbor Advertisement(NA) for sent T_IP is Received ***Then***
    a. Copy the value of Source Link Local IP address field in Received IP header and Put it in R_LL_IP.
    b. ***If***R_LL_IP = Generated 128 bit LLA ***Then***
        I. ***If*** Cc = = 3 ***Then***

```
                Set Uf = 1
              *Else*
          I.  Increment Cc by 1
         II.  Goto Step 2.
              *End*
      *Else*
          I.  Set Uf = 1
      *End*
  *Else*
          I.  Set Uf = 1
  *End*
```

In this step; if some other node in the network has generated same address, the source node will receive the Neighbor Advertisement and the condition 12(b) is checked. If it matches and if the value of collision count equals 3, then it's probably an attack. If the value of collision count is not equal to 3, then collision count is incremented by one and process repeats from step 2.

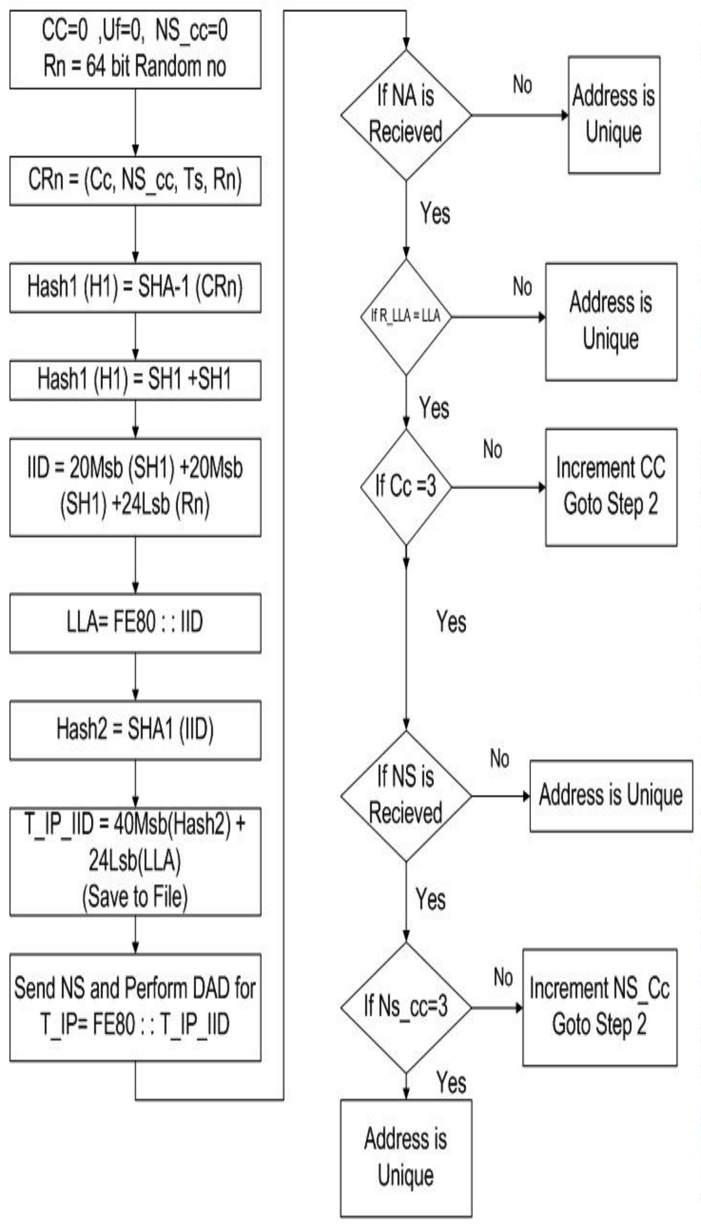
```
13. *If* Neighbor Solicitation(NS) for Generated LLA is received *Then*
     a. *If* NS_cc = = 3 *Then*
      I.  Set Uf =1
        *Else*
      I.  Increment NS_cc
     II.  Goto Step 2
       *End*
  *End*
```

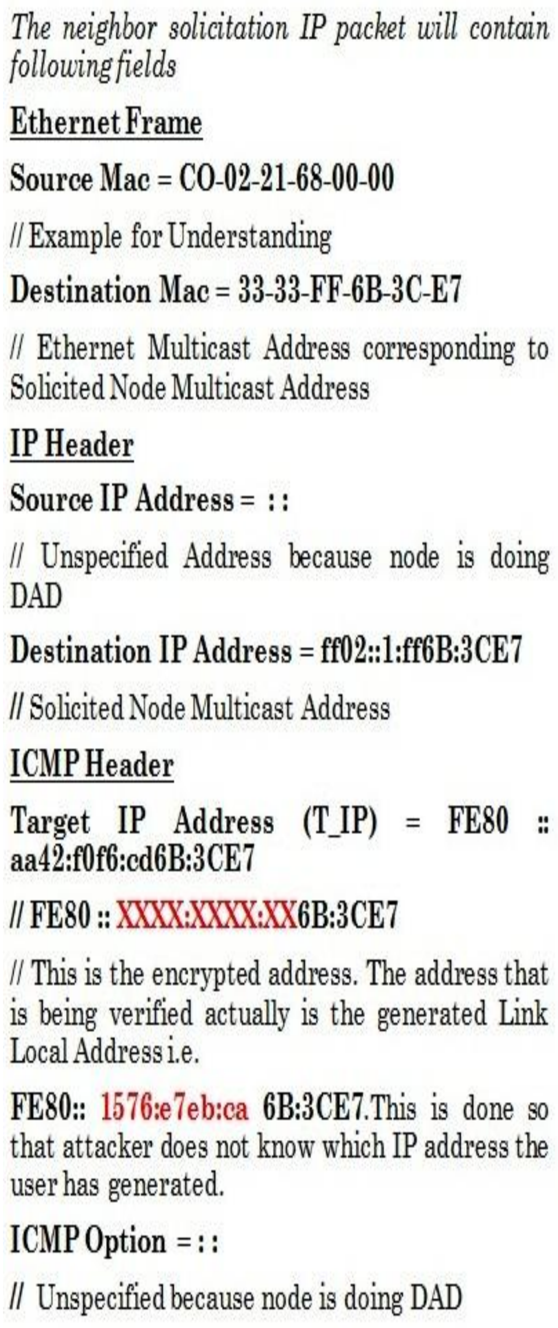
This step is carried because an attacker in addition to sending Neighbor Advertisement for asserting ownership of address can also send Neighbor solicitation indicating that it's also performing DAD process. To overcome this problem, we introduce a variable NS_cc which tracks how many times address collision occurs. If the collision happens more than 3 times, then it's probably an attack.

14. ***If***Uf = 1 ***Then***

    Generated LLA is Unique and valid.

    End





The neighbor solicitation IP packet will contain following fields

Ethernet Frame

Source Mac = CO-02-21-68-00-00

// Example for Understanding

Destination Mac = 33-33-FF-6B-3C-E7

// Ethernet Multicast Address corresponding to Solicited Node Multicast Address

IP Header

Source IP Address = : :

// Unspecified Address because node is doing DAD

Destination IP Address = ff02::1:ff6B:3CE7

// Solicited Node Multicast Address

ICMP Header

Target IP Address (T_IP) = FE80 :: aa42:f0f6:cd6B:3CE7

// FE80 :: XXXX:XXXX:XX6B:3CE7

// This is the encrypted address. The address that is being verified actually is the generated Link Local Address i.e.

FE80:: 1576:e7eb:ca 6B:3CE7.This is done so that attacker does not know which IP address the user has generated.

ICMP Option = : :

// Unspecified because node is doing DAD

Figure 6.4 Proposed Sender Node Algorithm

Figure 6.5 Sender NodeExample for Understanding

In Step 12 and 13 above; the heuristic solution for determining that an attack is being executed is by checking whether the value of Cc or NS_cc has reached three. This is based on the fact that probability of two nodes generating same address based on the values of Cc, Ns_cc in Timestamp 'Ts' is very low. (Bagnulo et al, 2002) shows that probability of occurrence of two nodes in network generating identical interface identifiers is given by:

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

Where n = number of address combinations possible, k = total number of interfaces on the same link. In our case; n = 2^{40}, and we assume the value of k= 1000; this gives us the probability as Pb (2^{40}, 1000) = 9.076e-7.This is a very small probability and thus validates heuristic theory. Thus encountering address collisions three times as indicated by Cc or NS_cc variables is a clear indication of malicious activity.

At Receiver Node:

The following algorithm is executed by only those nodes that have already joined the link and have a valid IP address. It is used for Processing Received Neighbor Solicitation for Duplicate Address Detection.

```
1. Extract 64 bit Interface Identifier from Target IP address field in Received ICMP header
   of Neighbor Solicitation and Put it in RT_IID
2. Obtain the value of T_IP_IID from file and compare it with RT_IID. The value of
   T_IP_IID has been generated during its address generation phase when the node had
   joined the link.
3. If T_IP_IID = = RT_IID Then
     I.   Send Neighbor Advertisement and inform the node that address already exists.
   Else

     I.   Discard Received Neighbor Solicitation.
   End
```

Ethernet Frame	
Source Mac	co-03-21-68-00-00
Destination Mac	33-33-00-00-00-01 //Ethernet All Nodes Multicast Address
IP Header	
Source IP Address	fe80 :: 1576:e7eb:ca6b:3ce7 //Actual Link Local Address
Destination IP Address	ff02 :: 1 //All nodes Multicast Address
ICMP Header	
Target IP Address	fe80 :: 1576:e7eb:ca6b:3ce7 //Link Local Address for which solicitation was sent
ICMP Option	
Source Link Layer Address	co-03-21-68-00-00

Table 6.2 Receiver Node Header Structure

6.5 Implementation and Evaluation

Our technique is executed on a machine running windows 7 operating system using language C#.net. The machine has 4 GB of Ram and 2.5 GHz Intel i5 processor. For performance evaluation and comparison, we also implemented CGA algorithm on our system.CGA also provides for address security and maintenance of privacy. Both the address generation algorithms were executed 20 times and average value was calculated. In our experiment, we do not consider the time spent in sending Neighbor Solicitation and reception of Neighbor Advertisement because that depends on network bandwidth and speed which may vary over time. The results are shown in table 6.3.

	Address-Generation Time (ms)		Address Verification (ms)		RSA Key Generation (ms)	Total Time(ms)	
	Sec = 0	Sec = 1	Sec = 0	Sec = 1	1024 bits	Sec = 0	Sec = 1
CGA	290 ms	4881 ms	132 ms	210 ms	220 ms	642 ms	5311ms
Proposed Technique	190 ms		112 ms		-n/a-	302 ms	

Table 6.3 Comparison of CGA and Proposed Technique

As shown in table 6.1, our proposed technique does not require the overhead of generating RSA keys as compared to CGA. This results in faster computation of interface identifier without processing bottlenecks. Our proposed technique takes a total time of 190 ms for address generation and 112 ms for verification. In our approach, if an attacker claims for a valid address generated by other node, he will have to provide exact value for IID that is sent to him in encrypted form in NS message. Even if dictionary attack is executed to search for hash pair values, it will require a database space for storing 2^{40} hash pair values. Searching such a large number of records for a match will take considerable amount of time and such a large space may not be feasible. Since we are also assuming that node will maintain IID for a limited period of time; this approach makes it extremely difficult for an attacker to eavesdrop on the link. Also in our approach, it’s very unlikely that other node will have used the same value for Cc, NS_cc and Ts while generating the address unless there is an attack being leveraged upon. For CGA, we are using RSA key size of 1024 bits. With sec value equal to 0, it takes 220 ms for RSA key pair generation, 290 ms for address generation and 132 ms for address verification adding a total time of 642 ms. For sec value greater than 0, the computational time increases exponentially. Here we are not considering the overhead of signature generation time for CGA, which is required for signing DAD messages. The robustness of CGA hinges on the value of sec but there is a tradeoff. If node needs robust security, higher value of sec should be used but that result in higher computational time which is not feasible. Using a sec value greater than zero, the brute force search takes an average of O ($2^{16\times sec}$) iterations to satisfy the Hash2 condition in CGA. For an attacker to break the CGA, cost of the brute force search takes O ($2^{16\times sec+59}$) iterations. If focus is on performance, lower sec value should be preferred which means compromising on security.

6.6 Summary

Security has been the prime motivation in the design of IPv6 protocol. While IPv6 has banished the shortcomings of its earlier predecessor, it has also introduced some parlous issues that require elaborate solutions. This chapter discussed the IPv6 SLAAC process and highlighted some of its critical issues related to privacy and security. The chapter also proposed a novel and highly randomized technique for address generation that safeguards node’s privacy and asserts address uniqueness on the link. The technique has a minimal computational cost and provides robust security against DoS attacks during the DAD process. For comparative performance analysis, we compared our technique with CGA algorithm. The results show that proposed technique improves computational time as compared to CGA. Since our technique uses SHA-1 hash encryption which is vulnerable to collisions attacks, as a part of our future work, the technique can be improved by using SHA-256 hash encryption. Also instead of RSA, Elliptic Curve Cryptography (ECC) can be used for public key generation.ECC uses shorter key size for the same level of security as RSA. This results in faster address generation. The technique can also be enhanced to mitigate other attacks like Address Spoofing, Man-in the-Middle and Router Authorization in order to secure the neighbor discovery protocol in IPv6.