Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 177 – 185

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Optimizing Security and Address Configuration in IPv6 SLAAC

Junaid Latief Shah* and Javed Parvez

University of Kashmir, Hazratbal, Srinagar 190 006, India

Abstract

The Neighbor Discovery Protocol (NDP) is the predominant component in IPv6; the next generation internet protocol providing for stateless address auto configuration of nodes (SLAAC), resolution of link layer addresses and neighbor unreachability detection. The stateless address auto configuration is designed for self configuration of nodes and achieving plug and play support for network devices. The protocol is rooted on the assumption that network consists of trusted nodes, however 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. The attacker can impersonate legitimate nodes and launch Man-in-the-Middle (MITM), Denial of Service (DoS), and other network related attacks. The 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. This paper presents a novel technique for address generation having a minimal computation cost as compared to CGA. The technique generates a highly randomized Interface Identifier that helps maintain nodes privacy and allows the nodes to ascertain the uniqueness on the link. It also provides robust security against DoS attacks during the DAD process of IPv6 SLAAC.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information
Processing-2015 (IMCIP-2015)

Keywords: CGA; DAD; IPv6; SLACC; NDP.

1. Introduction

Internet Protocol Next Generation (IPng) or IPv6 dispenses an agile and supple architecture framed to vanquish the limitations of IPv4. It provides a flexible architecture upon which network services and applications can be deployed[1]. The main driving force in the evolution of IPv6 was the exhaustion of IPv4 address space. While IPv4 extensions like NAT, CIDR worked for limited short term scenarios, the features and solutions required by the modern internet cannot be fully provided by IPv4. The IPv4 lacks in deployed security infrastructure and entails rapid growth of routing tables. The solution lies in the ultimate migration to IPv6. IPv6 protocol is based on 128 bit address providing 2^{128} i.e. practically unlimited number of addresses for each and every device on the earth. The rapid expansion of the protocol maintains the end-to-end connectivity principle by banishing the requirement for NAT[2]. Additionally, IPv6 also provides a strong QoS support with the help of Flow Label and Traffic Class fields in packet header. IPv6

*Corresponding author. Tel.: +97-979-772-2303.
E-mail address: junaidlatiefshah@gmail.com

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.020

has a simplified packet header which aids in faster convergence of routing packets. The implementation of IPSec has been made mandatory in the form of extension headers. IPSec being an intrinsic element of IPv6 achieves end-to-end security and provides for confidentiality, data integrity and authentication with the help of Encapsulating Security Protocol (ESP) and Authentication Header (AH)[4]. The Neighbor Discovery Protocol (NDP) being a part of Internet Control Message Protocol for IPv6 (ICMPv6) is one of the intrinsic components of IPv6[5] and uses ICMPv6 message format[6]. The protocol provides several services like discovering nodes in the neighborhood, resolving their link layer addresses (congruent to ARP in IPv4), determining routers in local link, maintaining reachability information and identifying any duplicate addresses. However the protocol is not immune to attacks. The NDP is based on the assumption that all nodes on the local link are trustworthy. The assumption doesn't hold good in case of wireless networks where any node can join the link with minimum authentication. The malicious user could craft MITM attack where in legitimate traffic to the node could be redirected. This usually happens when the user's neighbor cache is poisoned with spoofed Neighbor Advertisement messages of ND Protocol. The malicious users could also launch DoS attack against Duplicate Address Detection Process in IPv6 Stateless Address Auto configuration impeding the legitimate nodes from joining the link.

This paper explicates discussion over the IPv6 Stateless Address Auto configuration Mechanism and explains the problems associated with it. The main aim of this paper is to propose a new address generation technique having 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 remaining paper is structured as follows. Section 2 explains IPv6 Stateless Address Auto Configuration process. Section 3 will propound on IPv6 privacy issues and the drawbacks of earlier approaches. Section 4 will discuss about Duplicate Address Detection attack in IPv6 SLAAC process. In Section 5 we introduce our proposed address generation technique which is effective against DoS attacks in SLAAC. Section 6 implements and evaluates our approach. Finally in Section 7, we summarize our conclusions.

2. IPv6 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[15]. The auto configuration of networks is the key to reduce operational deployment cost and improve the overall network design[7]. A variety of network topologies including Ethernet are self-configurable. The self configuration provides provision for Plug and Play support for network devices and as such increases the number of network elements that can join the link. In general auto-configuration increases the self-sufficiency of network devices by taking considerable amount of management functions and without relying on the central or additional configuration servers. SLAAC adds a unique characteristic in IPv6 networks. In IPv6, a node configures its address through one of the following ways[1]: 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.

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[12]. In SLAAC, when a node joins a link; it autonomously generates its Link Local Address (LLA) and 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

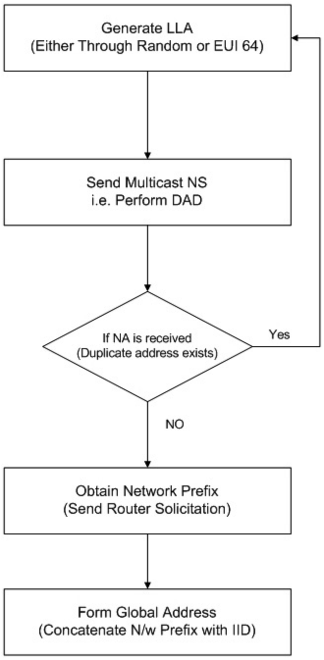
Generate LLA
(Either Through Random or EUI 64)
Send Multicast NS
i.e. Perform DAD
If NA is received
(Duplicate address exists)
Yes
NO
Obtain Network Prefix
(Send Router Solicitation)
Form Global Address
(Concatenate N/w Prefix with IID)

Fig. 1. IPv6 SLAAC process.

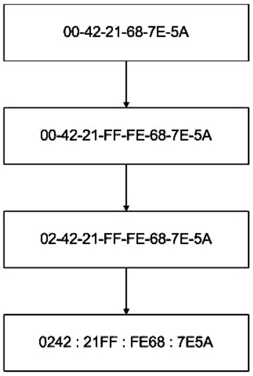

Fig. 2. EUI-64 interface identifier generation.

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 Fig. 1. 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.

3. IPv6 SLAAC Privacy Implications

An IPv6 address is comprised of two elements; a 64 bit network prefix and a 64 bit Interface Id. A standard for generating the 64 bit Interface Id (IID) has been proposed by IEEE standards association[8] 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:: with the IID results in the formation of Link Local Address FE80:: 0242:21FF:FE68:7E5A. This shown in Fig. 2. This method of interface id generation has some privacy issues and drawbacks. 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[10]; 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. To overcome this; the approach may be to use a Cryptographically Generated Address (CGA) which generates a randomized IID rooted on node's public key. CGA's are used to provide address ownership proof and to avert IP spoofing. However CGA implementation 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. The drawbacks and discussions on using different privacy extension mechanisms and techniques are further explained in[9,11].

4. Duplicate Address Detection (DAD) Attack 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[13]. 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 Fig. 3.

The attacker can assert ownership of address in one of two ways[14]. 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

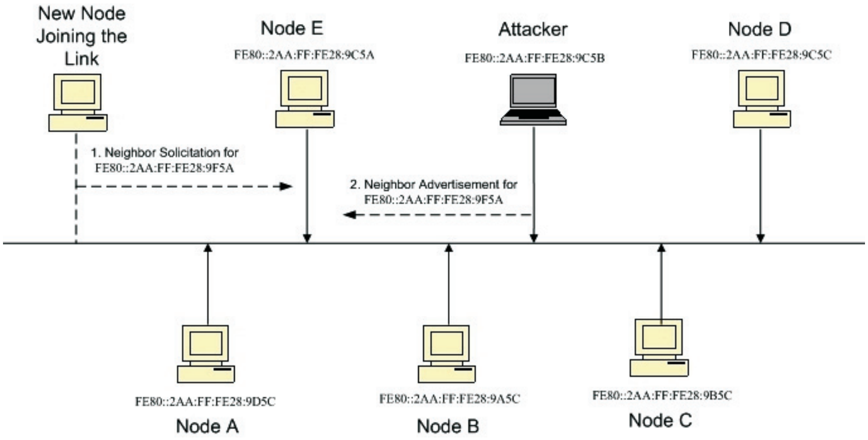

Fig. 3. DAD attack in IPv6 SLAAC.

for some time until the new address is generated. The new address should again be verified using DAD[15]. Repeated acknowledgements to generate link local addresses may deny the new node from joining the link and the node would thus remain uninitialized. A substantial amount of research is being done to address this issue 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[14]. 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 3971[17] defines 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. Further discussion on CGA constraints can be found in[6,9,11]. Rafiee *et al.* in[18], have implemented the SEND protocol for the windows platform. Although claiming to be the first SEND implementation for windows, the approach suffers the drawback of being platform specific and thus requires adding an update in every individual host to support it. However, there are some limitations for executing the attacks like having access to the link and replying with NA message well before the value of RetransTimer variable has expired.

4.1 The SEND approach

As discussed above, IPv6 SLAAC is susceptible to malicious threats and attacks. To counter these challenges, RFC 3971[17] mandates the use of SEND as a security extension to Neighbour Discovery Protocol. The protocol introduces four different options (CGA, RSA Signature, Nonce, and Timestamp) and two new ICMPv6 messages (CPS and CPA) which aid in providing address ownership proof mechanism, message integrity/identity and authorization of routers. Despite its innumerable tangible benefits, SEND faces major challenges including intense computation, vast implementation, deployment and security issues.

- *Cryptographically Generated Address (CGA)*

The CGA is an innate component of SEND protocol used to prevent address impersonation. CGA authenticates IPv6 addresses without requiring installation of third party services or additional servers. CGA's are computed by cryptographic one way hash function applied on public key and auxiliary data structure[23]. The process certifies that IPv6 address is bound to the public key it uses. This binding can later be verified by re-computation of hash digest and comparing it with Interface Identifier (IID) of the IPv6 address. CGA's can also be used for authentication and proving that sender of the packet is the actual owner of the packet. For this, the packet has to be signed by sender's private key. The public key, signature and auxiliary parameters are then sent with the packet to the receiver. The receiver can then verify the signature and confirm the sender of the packet is owner of CGA. The CGA algorithm begins by generating owner's public key and choosing a proper sec value. Then Hash-2 evaluations occur in a loop until a final modifier value is obtained. The Hash-2 value is obtained by applying SHA-1 over the CGA data structure (128 bit randomly generated modifier, 64 bit subnet prefix, 8 bit collision count, public key, extension field). The generation function tries different modifier values until 16 × sec leftmost bits of Hash-2 equals zero. Once the final modifier is found, the loop terminates and it serves as an input for Hash-1 calculation. From Hash-1, IID is derived and value of sec is encoded in its three leftmost bits. Finally DAD process is executed to ensure the address uniqueness. CGA's find their main application in mitigating Denial-of-service attacks during the IPv6 SLAAC process. Today CGA's are even used in mobile computing security. A node proves address ownership by using its private key to sign the DAD and ND messages that it sends. The main disadvantage of CGA is its computational cost involved in Hash-2 calculations. The sec value ranges from 0 to 7. Using a sec value greater than 1 result's in exceptionally longer address generation time. To impersonate a given node by using brute force search takes an average of O ($2^{16\times \text{sec}+59}$) hash function evaluations. However on the other hand, address owner tries O ($2^{16\times \text{sec}}$) iterations to find the right modifier value that satisfies

Hash-2 condition. Another drawback of CGA is that it's not certified. The attacker can create a new valid CGA and start the communication. The attacker can thus impersonate other node addresses from a valid public key but cannot sign the other node's messages.

5. Proposed Technique

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

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
```

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.*[19] 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.

Table 1. Comparison of CGA and proposed technique.

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

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
```

6. 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 1 below.

As shown in Table 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 μs for address generation and 112 μs 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 μs for RSA key pair generation, 290 μs for address generation and 132 μs for address verification adding a total time of 642 μs. 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 $(2^{16\times \mathrm{sec}})$ iterations to satisfy the Hash2 condition in CGA. For an attacker to break the CGA, cost of the brute force search takes $O(2^{16\times \mathrm{sec}+59})$ iterations[11]. If focus is on performance, lower sec value should be preferred which means compromising on security.

7. Conclusion

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 paper discussed the IPv6 SLAAC process and highlighted some of its critical issues related to privacy and security. The paper 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.

References

[1] L. Ladid, IPv6-the Next Big Bail-Out: Will IPv6 save the Internet? In *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, ACM*, pp. 2, June (2009).

[2] Shah, Junaid Latief and Javed Parvez, An Examination of Next Generation IP Migration Techniques: Constraints and Evaluation, *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), IEEE*, (2014).

[3] Z. A. Baig and S. C. Adeniye, A Trust-Based Mechanism for Protecting IPv6 Networks Against Stateless Address Auto-Configuration Attacks, In *17th IEEE International Conference on Networks (ICON), IEEE* pp. 171–176, December (2011).

[4] IPsec, (2015, January 8), In Wikipedia, The Free Encyclopedia, Retrieved 16:08, January 19, (2015).

[5] Narten, Thomas, *et al.* Neighbor Discovery for IP Version 6 (IPv6), (2007).

[6] A. AlSa'deh, and C. Meinel, Secure Neighbor Discovery: Review, Challenges, Perspectives, and Recommendations, *Security & Privacy, IEEE*, vol. 10(4), pp. 26-34, (2012).

[7] J. J. S. Tobella, M. Stiemerling and M. Brunner, Towards Self-Configuration of IPv6 Networks, In *Network Operations and Management Symposium, NOMS 2004. IEEE/IFIP, IEEE*, vol. 1, pp. 895–896, April (2004).

[8] IEEE Standards Association, (2012), http://standards.ieee.org/develop/regauth/tut/eui64.pdf

[9] H. Rafiee and C. Meinel, Privacy and Security in IPv6 Networks: Challenges and Possible Solutions, In *Proceedings of the 6th International Conference on Security of Information and Networks, ACM*, pp. 218–224, November (2013).

[10] T. Narten, R. Draves and S. Krishnan, Privacy Extensions for Stateless Address Auto Configuration in IPv6, (2007).

[11] H. Rafiee and C. Meinel, SSAS: A Simple Secure Addressing Scheme for IPv6 Auto Configuration, In *Eleventh Annual International Conference on Privacy, Security and Trust (PST), 2013, IEEE*, pp. 275–282, July (2013).

[12] M. Blanchet, Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks, John Wiley and Sons, (2009).

[13] C. E. Caicedo, J. B. Joshi and S. R. Tuladhar, Ipv6 Security Challenges, *Computer*, vol. 2, pp. 36–42, (2009).

[14] F. A. Barbhuiya, G. Bansal, N. Kumar, S. Biswas and S. Nandi, Detection of Neighbor Discovery Protocol based Attacks in IPv6 Networks, *Networking Science*, vol. 2(3-4), pp. 91–113, (2013).

[15] T. Narten, S. Thomson and T. Jinmei, IPv6 Stateless Address Auto-Configuration, (2007).

[16] T. Aura, RFC 3972, Cryptographically Generated Address (CGA), (2005).

[17] J. Arkko and J. Kempf, RFC 3971-Secure Neighbor Discovery (SEND), (2005-03).

[18] H. Rafiee, A. Alsa'deh and C. Meinel, Winsend: Windows Secure Neighbor Discovery, In *Proceedings of the 4th International Conference on Security of Information and Networks ACM*, pp. 243–246, November (2011).

[19] Bagnulo, Marcelo, *et al.*, Random Generation of Interface Identifiers, draft-soto-mobileip-random-iids-00.txt, Internet Draft, IETF, (2002).

[20] J. L. Shah and J. Parvez, Performance Evaluation of Applications in Manual 6in4 Tunneling and Native IPv6/IPv4 Environments, In *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), IEEE*, pp. 782–786, July (2014).

[21] J. L. Shah and J. Parvez, Evaluation of Queuing Algorithms on QoS Sensitive Applications in IPv6 Network, In *International Conference on Advances in Computing, Communications and Informatics (ICACCI, 2014), IEEE*, pp. 106–111, September (2014).

[22] Shah, Junaid Latief and Javed Parvez, Security Issues in Next Generation IP and Migration Networks, *IOSR Journal of computer Engineering (IOSR-JCE)*, 17.1, pp. 13–18, (2015).

[23] Aura, Tuomas, Cryptographically Generated Addresses (CGA), (2005).

[24] Moore, Nick, Optimistic Duplicate Address Detection (DAD) for IPv6, (2006).