An Efficient CGA Algorithm against DoS Attack on Duplicate Address Detection Process

Cui Zhang†, Jinbo Xiong*, Qiong Wu ‡
†State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
*Faculty of Software, Fujian Normal University, Fuzhou, China
‡National Mobile Communications Research Laboratory, Southeast University, Nanjing, China
Email: zhangcui@iie.ac.cn, jbxiongfj@gmail.com, qiongwu@seu.edu.cn

***Abstract*—Neighbor Discovery Protocol (NDP) is significant in mobile network, which enables mobile node randomly access to foreign network by Stateless Link Address Autoconfiguration (SLAAC). However, the NDP initially offers no protection mechanism and is prone to address spoofing and Denial of Service (DoS). Secure Neighbor Discovery Protocol (SeNDP) is proposed to solve these NDP threats. Recently there are many solutions presented in SeNDP which relies on special IPv6 addresses named Cryptographically Generated Address (CGA). But there is little work to defend DoS attack on Duplicate Address Detection (DAD). In our paper, we focus on the problems of CGA and propose a novel time-based monitoring DoS attack. The conventional DoS defense mechanisms are realized by monitoring the packet rating and observing connection delay to analyze various DoS attack. Hence, we adopt a delay as an indication to distinct the DoS attack. We set a timer to control the address generation for monitoring abnormal attack to protect each address configuration. In addition, we adopt SHA-224 hash function instead of SHA-1 to improve the security of address generation. Considering the computation overhead, we decrease the hash matching factor from 16 bits to 8 bits. We develop our scheme using the Network Simulator (NS2) and the OpenSSL library. Finally, experiment results prove our scheme can provide more efficient IP generation. Compared with the CGA algorithm in SeNDP, our time consumption decreases to 10%. From the view of defense attack, our scheme can control DoS attack.**

***Index Terms*—Neighbor Discovery; Stateless Link Address Autoconfiguration; Secure Neighbor Discovery Protocol; Denial of Service**

I. Introduction

With the development of mobile computing, smartphones and PDAs are becoming more and more popular in our life. Every device needs an IP address to access the Internet. However, the traditional IPv4 addresses have been depleted. To address this problem, IPv6 suites [1] are proposed for expanding the network address spaces. Additionally, IPv6 suites enrich the performances including mobility, quality of service and security. ICMPv6 is a part of IPv6 protocol which is used to feed information back to IP nodes regarding network errors, resource constraints, neighbor discovery and mobile IPv6. A node automatically realizes Stateless Address Autoconfiguration (SLAAC) function by Internet Control Message Protocol version 6 (ICMPv6) [2]. SLAAC function means that a node accesses an IPv6 network without manual configuration. It gets its own address based on the local link information. SLAAC uses EUI-64 (Extended User Interface-64 bits) format for address assignment. This means that IPv6 addresses will be constructed from a combination of the Layer 3 subnet prefix and the MAC address of a node. When a node prepares to connect an IPv6 network, it first configures its IP address using Neighbor Discovery Protocol (NDP). In NDP, a node sends a link-local router solicitation multicast request for its configuration parameters; routers respond to such a request with a router advertisement packet that contains network layer configuration parameters. In IPv6 protocol suites, NDP [3] alternates the Address Resolution Protocol (ARP) in IPv4 protocol suites and provides Duplication Address Detection (DAD) function.

Owing to the lack of the authentication, message protection and router authority, the NDP is vulnerable to spoofing attack, DoS attack and forge router attack. To address the shortcoming of NDP, the IETF working group proposed SeNDP [4]. The protocol is composed of Cryptographically Generated Address (CGA), a digital signature and X.509 certification to protect NDP. However, SeNDP lacks mature implementations by network device manufactures and operating system developers. It consumes intensive computing and bandwidth resources. Hence, we consider more efficient method to reduce resource consumption. As an essential part of SeNDP, CGA is proposed to prevent address stealing. It realizes IPv6 address authentication without requiring Trusted Third Party (TTP). CGA address is an IPv6 address which includes subnet network prefix address and Interface Identifier (IID). The IID is generated by hashing a public key and other public data parameters. By the means of CGA mechanism, a node's IPv6 address is bound to its public key. The receiver can verify this binding relationship by recomputing the hash value and comparing the hash result with the sender's IPv6 address. However, the standard CGA exists lacks, e.g., high computation consuming and DoS attack. DoS attack is still a troublesome threat to the reliability of the Internet. Its characteristic is an explicit attempt to prevent the legitimate use of a service. Current DoS defense methods are applied to wired communication. Although some works have been done for enhancing the SLAAC security, most of the works could not solve the problem when the node is not connected in the network. Because conventional DoS defense methods need the monitor network flow such

978-1-4673-8666-1/16/$31.00 ©2016 IEEE

as limiting network packet rate during TCP, UDP or ICMP link connection [5]. These defense methods could waste much network system resources and can not solve a node without starting network services. Hence, we proposed timer-based method. This way can control the DoS attack happening before a node becomes online state. To the best of our knowledge, there is no scheme using timer to defend DoS in DAD process. This is our motivation.

In our paper, we propose an approach quickly generated a CGA address with a low hash matching without affecting the security level. We decrease the time consumption to few milliseconds by optimizing some algorithms in the battery limitation situation. A modified CGA generation algorithm named time-based CGA is proposed in [6]. By decreasing the hash matching granularity from 16 to 8, we minimize the final generation time.

At the same time, the security performance of CGA relies on the SHA-1 hash function which may be broken by brute-force break. For protecting CGA against this attack, we design our CGA algorithm using SHA-224.

The contributions of this paper are summarized as follows:

- We propose SHA-224 as an alternative hash function in the standard CGA. In the perspective of computation complexity, there is no brute-force method to break SHA-224. In addition, the computational overhead of algorithm used SHA-224 function is similar to that used SHA-1 function in the view of our experiment results.
- We adopt an efficient factor 8 rather than 16 during matching modifier value. Random modifier matching time in the situation is less than 16*sec when sec equals to 1.
- We monitor DAD DoS attack using a controlled timer rather than collision count. Although an address collision count is designed in CGA, the generated processing will stop and report the error after three collisions. The collision count can not distinguish the normal address confliction with the malicious DAD DoS attack.

The rest of this paper is organized as follows. In Section II, we introduce the related works. Our mechanism is presented in Section III. We provide security analysis in Section IV and evaluation results in Section V. Finally, we draw a conclusion in Section VI.

II. Related work

A. Secure neighbor discovery

Neighbor Discovery Protocol (NDP) is used to discover and get the link addresses of the neighbors of a node or router. It includes Neighbor Solicitation (NS), Neighbor Advertisement (NA), Router Solicitation (RS) and Router Advertisement (RA). They can realize the SLAAC, DAD and Destination Unreachable functions. However, it takes no account of security and is vulnerable to many attacks. SeNDP is proposed to secure the NDP using CGA, signature and authentication. CGA algorithm has been wildly studied. Aura proposed CGA standard [7]. CGA standard uses RSA public key, subnet prefix, collision count and random number as modifier parameters input a hash function. Tony et al. proposed Elliptic Curve Cryptography (ECC) instead of RSA in CGA generation to solve energy and storage limitation problem in mobile environments [8]. Zhou defined Mhash-method Extension and Hash Algorithm Identity Parameter to enable multiple hash functions support in CGAs in case the current hash function does not satisfy future requirements [9]. Alsadeh proposed a time-based CGA algorithm [6]. The time-based CGA takes the upper bound of CGA running time as an input and set a sec value to save the hash function computing time. Rafiee proposed a secure identifier interface generation algorithm based on ECC and bound the MAC address for privacy extension [10]. They call this approach a Simple Secure Addressing Scheme for IPv6 Auto-configuration (SSAS) which mitigates many attacks against a NDP-enabled node and decreases the complexity of the IID generation.

B. Denial of Service on Duplicate Address Detection in neighbor discovery

Current NDP is vulnerable for DoS attack. There are abundant researches on the security of NDP in recent years. Barbhuiya et al. proposed an attack detection mechanism for neighbor solicitation spoofing and neighbor advertisement spoofing in NDP [11]. The mechanism contains an Intrusion Detection System (IDS) for detection NS/NA spoofing attacks. However, they take no account of some attacks against DAD processing. Feng et al. analyzed security for IPv6 NDP and described DoS attacks against NDP, i.e., forging fake prefix address and network configuration parameters, even launching DoS attack based on DAD [12]. However, they only proposed DoS attack but not how to defense. Rehman et al. proposed a rule-based mechanism to detect a DoS attack on DAD process in IPv6 link local communication [13]. This mechanism designs a rule-based controller to reply a "unique" or "duplicate" address to the requested node. The node relies on the response to decide whether to use the tentative IP address as preferred one or to generate a new IP address. However, it needs to add a controller server. There are some researches on the security problem of SeNDP [14]. In our paper, we design a mechanism without adding device to detect the DoS attack against SeNDP DAD processing.

III. Our Proposed Protocol

One advanced function of IPv6 is auto-configuration function. The auto-configuration function is implemented through Dynamic Host Configuration Protocol version 6 (DHCPv6) and SLAAC. DHCPv6 is used to manage the IP address. This method requires human assistant with respect to the installation and administration of DHCPv6 servers. However, the DHCPv6 severs are hard to deploy in mobile environment. Hence, we consider the generation of a link local address without configuration servers using SLAAC to generate a global address and verify the uniqueness of the address in IPv6 networks. This mechanism is mainly realized by SeNDP. First, we describe secure neighbor discovery. Then, we describe our proposed CGA scheme.

Modifier(128bits)	
Subnet prefix(64bits)	
Collision count(8bits)	Public key(1024bits)
Extension(variable)	

Fig. 1: Cryptographically generated address (CGA) data parameters

Secure neighbor discovery protocol provides address ownership, router authentication and message authenticity. It contains four new functions, i.e., RSA signature, nonce option, timestamp option and CGA option. RSA signature is used to authenticate the sender's identity. Nonce option is a random number to prevent replay attack. Timestamp is a synchronized clock for the requested message and responded message. The CGA option associates CGA data parameters with public key. It ensures a receiver can validate the proper binding between the public key and the CGA. The CGA algorithm is a part of secure neighbor discovery protocol. We describe the CGA algorithm in the following.

A. Cryptographically Generated Address

In this paragraph, we will introduce CGA. The CGA data structure is composed of subnet prefix and interface identifier similar to common IPv6 address. The interface identifier is generated by computing a hash function from a public key and auxiliary parameter. Using CGA algorithm, a receiver is able to verify a message from a real sender without public key infrastructure. The CGA algorithm can compute an interface identifier in IPv6 suites efficiently with a special mechanism in IPv6 address. The data structure includes modifier, subnet prefix, collision count, public key and extensions. The data structure is shown as Fig. 1. The CGA generates a unique IPv6 address according to the following steps.

(1) Initialize CGA parameter date structure. Set modifier as a random or pseudo-random 128-bit value. Set subnet prefix as 0, counter as 0 and public key as 1024-bit value by RSA.

(2) Concatenate each segment of data structure from left to right and construct a string. Input the string to SHA-1 function. Compute SHA-1 function and get a result Hash2.

(3) Hash2 is a 160-bit value. Select a sec value between 0 to 7 before the algorithm is operated. We select a 112-bit value from left to right in Hash2.

(4) Compute the value of 16*sec and judge whether all the left 16*sec bits of the 112 bits selected from Hash2 equal to 0. If all the 16*sec bits of the 112 bits equal to zero, save the random 128-bit value in step 1 as the modifier value, else increase the value of modifier by one and jump to step 2 to re-calculate Hash2.

(5) Concatenate the 128-bit obtained in step 4 with the 64-bit subnet prefix, the 8-bit collision counter and the 1024-bit public key as the inputs of the SHA-1 function. Then we can compute Hash1 using SHA-1 function.

(6) Hash1 is a 160-bit value. Extract the leftmost 64-bit from the 160-bit of Hash1. In the obtained 64-bit value, the leftmost three bits stand for the security level flag sec, the sixth and seventh bit stand for the identifier of local and group address respectively. The obtained 64-bit value is an interface identifier. Concatenate the subnet prefix with interface identifier and form 128-bit value as an IPv6 address.

(7) Execute the duplicate address detection. If an address collision is detected, increase the collision count by one and return to step 4, else the IPv6 address is efficient. Here an efficient standard IPv6 address is obtained.

However, the efficiency of CGA is quite low when the security level sec is higher. When sec exceeds three, the generation time will be closer to two hours. It is hard for a user to endure in practical use. Hence, for improving the time efficiency, some researches have been proposed, e.g., stopping time cryptographically generated address. We introduce stopping time CGA in the next paragraph.

B. Stopping time Cryptographically Generated Address

In this paragraph we will introduce stopping time CGA. To guarantee that CGA generation process can be terminated after a certain time, the stopping time CGA takes time as an input to determine the CGA termination time. The stopping time CGA is shown in Fig. 2.

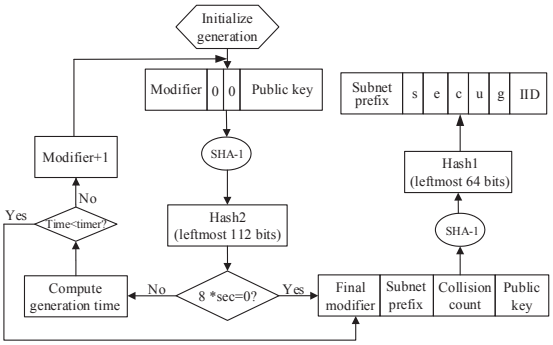

Fig. 2: Stopping time cryptographically generated address

Time parameter is used as an input in the CGA data parameters. If time parameter has not been exceeded, the algorithm increases the modifier by one and compute a new Hash2. After generating every Hash2 value, the numbers of zero are counted from the big-endian, i.e., counting from the leftmost bit. Only a matching modifier will satisfy the restricted condition and get the finial modifier value. Then the node computes the generation time, compare it with a timer. Once the time parameter is exceeded, the loop stops the modifier computation.

In addition, the time stopping CGA is decreasing the brute-force level with 8*sec. Because this way saves much time for matching a proper modifier value. Faster devices are able to find a better sec value than slower ones during the same time. Hence, the scheme is better than the standard CGA in wireless or energy limited environment.

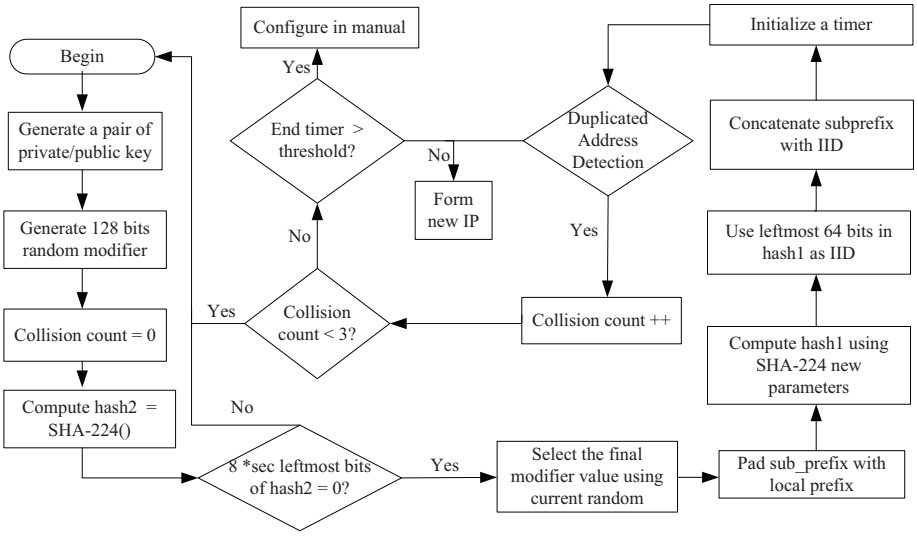

Fig. 3: DoS-defense cryptographically generated address

C. DoS-defense Cryptographically Generated Address

In this paragraph, we will introduce DoS-defense Cryptographically Generated Address. The current CGA takes no account of the DoS attack on DAD and its efficiency is low. Hence, we have proposed our DoS-defense Cryptographically Generated Addresses. During duplicate detection processing, if CGA collision counter exceeds 3, the generated address would always recycle from the beginning. The computation will consume the CPU resource and the router cannot afford the normal service for other requests. What is DoS attack on DAD? DAD is performed on unicast addressed prior to assigning them to an interface. If there is a duplicate address, the address cannot be assigned to the interface. The use of DAD brings the possibility of DoS attacks. In several reactive defense mechanisms, the attack detection is threshold-based, i.e., whenever traffic rate at the victim server exceeds a threshold rate, the defense mechanism is triggered. Any malicious node always says "I have the IP address", causing other node to reject the address assignment even though there is a collision count parameter in the standard CGA algorithm. The collision count can not distinguish the normal address confliction with the malicious DAD DoS attack. Hence, we use a delay metric to indicate the DoS attack. Once every generation address cannot satisfy a security level, a node configuring IP will waste much time to compute a proper IP address so that we cause a DAD DoS attack. We first define two time parameters.

(1) The generation time is the complete duration of the generation of a CGA algorithm from the generation of the public key up to the computation of the interface identifier including Hash1 and Hash2.

(2) The final modifier generation time corresponds to the time spent computing a Hash2 value that matches the condition on the first bit. We use a time index to modify the standard CGA algorithm. At the same time, for preventing the abnormal computation from generating a valid address, we set a new monitor in the node for controlling the time. There are two advantages in this way. The first one is that we can detect the DoS attack and second one is that we can defend DoS attack. Therefore, we design a timer monitor. This time value derives from current rational experiment result. If the timer is time-out, we stop the address generated processing. In order to assure the node normally access the network, we configure a tentative address in manual method.

The data parameter structure to input the algorithm is $\langle modifier, subnet\ prefix, collision\ count, public\ key \rangle$. The address generation processing is shown in Fig. 3.

(1) Generate a public/private key, a 128-bit random modifier value and collision count with 0.

(2) Construct data parameters above-mentioned as an input for SHA-224 and set the hash result as Hash2.

(3) Match the leftmost 8-bit value with 8-bit zero when the value of sec is 1.

(4) Once the condition of step 3 is satisfied, the finial modifier value is static. Use the current modifier value to form a new data structure.

(5) Pad subnet prefix value with the local-link prefix. Then use the new data structure formed in step 4 as the input of SHA-224 and compute it as Hash1. Get the 64 bits from the whole Hash1 as the interface identifier.

(6) Concatenate the subnet prefix and interface identifier as an IP address and initialize a timer. The current time is T1.

(7) Let the IP address match with the local IP. If the duplicated detection is true, collision count is increased by one, regenerate an IP address. Loop the detection process until collision count exceeds the limit value.

(8) Compute the end generation time Tn. Get the whole generation time GMT. If the time exceeds the time threshold Th, the node judges some DoS attack at the node and re-configures a default IP address, else restart the whole generation algorithm:

GMT = Tn - T1. Set Th = 30 seconds as a threshold for terminating the proceeding. If GMT exceeds Th, the host is judged to be attacked by DoS. A malicious node hinders the

legitimate address generation using dead loop. We use the timer to stop whole generation address and configure some tentative addresses.

IV. Security analysis and complexity

Theorem 1: Our scheme ensures data integrity. Any address is generated by HASH transformation. Once HASH verification is right, the process would prove whether the data is complete.

Theorem 2: Our scheme is IP spoofing resistant. The protocol adopts RSA signature and every source address is converted to a hash value. It is difficult for an attacker to distinguish the real address when they receive a packet from CGA address node without the key. During the CGA generation processing, a modifier parameter adds a random number to defend the replay attack. Every same public key will generate the different addresses so that this method protects the privacy. Every node generates a pair private/public key preparing for RSA signature in SeNDP. The sender node signs the outgoing messages with a private key. And the private key is only owned by the sender. So the node can keep from spoofing CGA address by RSA signature.

Theorem 3: Our scheme is DoS on DAD resistant. Once a malicious node takes the DoS attack by constructing a number of false packets and waste the resources of the victim node, our scheme can monitor the attack. Owing to the CGA generation special processing, it is easy to compute the generation time for autoconfiguration. Our scheme sets ending time as a threshold turn-off to terminate DoS attack.

The complexity. We can infer that the time complexity of the computation is bound to the input size k. In the context of CGA, we use SHA-224 hash function whose computation cost equals to $O(2^{64})$. Data structures and the length of the structures have a direct impact on the computation time of the digests. We select 1 as sec value so that the final modifier time dramatically decreases. During Hash2 generation processing, we match a number with 8 zeros rather than 16 zeros for saving generation cost. The generation cost of our algorithm is shown in TABLE I. Once the sec increases by one, the generation time will increase 256 times. Hence, we consider the sec value less than 3.

TABLE I: Average CGA generation time (us)

sec value	Average CGA generation time (us)
0	313
1	15034.6
2	114697.5
3	29362432

V. Simulation evaluation

We use a SeNDP implementation (www.ohloh.net/p/ipv6-send-cga) in the Linux kernel IPv6 module. At first, we set the simulation environment in the OpenSSL library in NS2 with the 1024 key length. First we embed IPv6 SeNDP protocol code into NS2. By modifying the generation parameters, we debug CGA source code in Eclipse environment. The simulation network environment is made of two networks

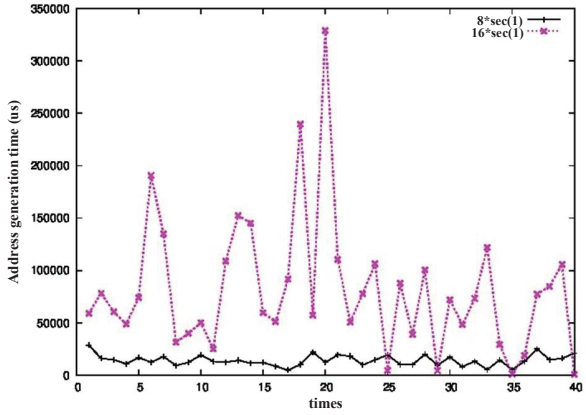

Fig. 4: Generation time in two sec values

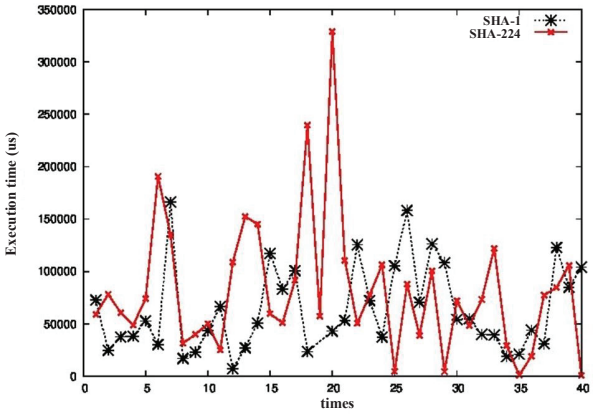

Fig. 5: The execution time in SHA family

installing SeNDP protocol. One is a LAN composed of a router and five nodes. The other network contains a malicious node and a legitimate node. The malicious node attacks the router by raising DoS attack. The most efficient algorithm concerning the final modifier generation time is the currently used SHA-1 algorithm. In our scheme, the generation time is computed in the algorithm. Random modifier matching time is less than 16*sec when sec equals to 1. With different security levels, the modifier time difference is tremendous. We set the security level value of sec equals to 1 for saving whole generation time. Fig. 4 compares the two situations. From Fig. 4 we can observe that 8*sec matching time is less ten times than 16*sec. We keep the security level at one level and dramatically decrease the hash matching time.

In addition, in the perspective of computational complexity, no brute-force method to break SHA-224 is designed in current time. Algorithm computation time is similar to SHA-1 as Fig. 5. The Fig. 5 compares the algorithm execution time of two different hash function. The average generation time of one address is about 95ms when we execute SHA-224. Hence, the security level is increased by using SHA-224. In the Fig. 5, the generation time is slightly increased.

VI. Conclusions

Although SeNDP is proposed to protect address ownership and neighbor discovery message, there are some attacks such as DoS attack, man-in-the-middle attack, replay attack and forge IP or router. Especially DAD process in CGA can be used as a trigger of DoS. A novel CGA scheme based on timer has been proposed to defend DoS on DAD process. We have designed a timer threshold to control DoS attack. Once the generation time exceeds the threshold, we configure a tentative valid address for the host. Compared with the previous CGA algorithm, the algorithm is efficient in defense DoS attack. At the same time, our proposed CGA sets the security level at 1 to increase difficulty during matching hash for higher security performance. In addition, the proposed method saves time overhead as much as 90% by half adjusting factor. The novelty is the time monitor for DAD DoS attack during SeNDP. We can defend the DoS rather than add any intermediate applications or devices. Hence it is significant in some energy limit and mobile situation.

VII. acknowledgement

This work was supported by National Natural Science Foundation of China under Grant 61402109, Natural Science Foundation of Fujian Province under Grant 2015J05120.

References

[1] S. Deering and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification," *RFC 2460, IETF: The Internet Engineering Taskforce, 1998*.

[2] A. Conta, and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," *RFC 4443, IETF: The Internet Engineering Taskforce, 2006*.

[3] T. Narten, E. Nordmark, W. Simpson and H. Soliman, "Neighbor Discovery for IP Version 6 (IPv6)," *RFC 4861, IETF: The Internet Engineering Taskforce, 2005*.

[4] J. Arkko, J. Kempf, B. Zill and P. Nikander, "SEcure Neighbor Discovery (SEND)," *RFC 3971, IETF: The Internet Engineering Taskforce, 2005*.

[5] J. Mirkovic and P. Reiher, "D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks," *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, No. 3, 2005, pp. 216-232.

[6] A. Alsadeh, H. Rafiee and C. Meinel, "Stopping Time Condition for Practical IPv6 Cryptographically Generated Addresses," in Proc. of *IEEE ICOIN 2012*.

[7] T. Aura, "Cryptographically Generated Addresses (CGA)," *RFC 3972, IETF: The Internet Engineering Taskforce, 2005*.

[8] T. Cheneau, A. Boudguiga and M. Laurent, "Significantly Improved Performances of the Cryptographically Generated Addresses Thanks to ECC and GPGPU," *Elsevier Computer & Security*, Vol. 29, No. 2010, 2009, pp. 419-431.

[9] S. Zhou and R. Zhang, "Another Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)," *Internet-Draft*, 2012.

[10] H. Rafiee and C. Meinel, "SSAS: A Simple Secure Addressing Scheme for IPv6 Autoconfiguration," in Proc. of *IEEE PST 2013*.

[11] F. A. Barbhuiya, S. Biswas and S. Nandi, "Detection of Neighbor Solicitation and Advertisement Spoofing in IPv6 Neighbor Discovery Protocol," in Proc. of *ACM SIN 2011*.

[12] X. Feng, J. Lin and S. Jia, "Security Analysis for IPv6 Neighbor Discovery Protocol," in Proc. of *IEEE IMSNA 2013*.

[13] S. U. Rehman and S. Manickam, "Rule-Based Mechanism to Detect Denial of Service (DoS) Attacks on Duplicate Address Detection Process in IPv6 Link Local Communication," in Proc. of *IEEE ICRITO 2015*.

[14] A. AlsaDeh and C. Meinel, "Secure Neighbor Discovery: Review, Challenges, Perspectives, and Recommendations," *IEEE Security Privacy Magazine*, Vol. 10, No. 4, 2012, pp. 26-34.