IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 3, 2020, accepted January 26, 2020, date of publication January 31, 2020, date of current version February 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970787

Match-Prevention Technique Against Denial-of-Service Attack on Address Resolution and Duplicate Address Detection Processes in IPv6 Link-Local Network

AHMED K. AL-ANI, MOHAMMED ANBAR, AYMAN AL-ANI, AND DYALA R. IBRAHIM
National Advanced IPv6 Centre, Universiti Sains Malaysia, Gelugor 11800, Malaysia

Corresponding author: Mohammed Anbar (anbar@nav6.usm.my)

This work was supported by the Fundamental Research Grant Scheme (FRGS), Ministry of Higher Education, Malaysia, under Grant FRGS 203.PNAV.6711836.

ABSTRACT Address Resolution (AR) and Duplicate Address Detection (DAD) are considered the most important processes in Neighbour Discovery Protocol (NDP), which occurs frequently from each Internet Protocol version 6 (IPv6) host communicating with other neighbouring hosts. Two NDP messages are used during AR and DAD to communicate with one another in the same IPv6 link-local network, namely Neighbour Solicitation (NS) and Neighbour Advertisement (NA) messages. However, NDP messages have non-secure designs and lack verification mechanisms for authenticating whether incoming messages originate from a legitimate or illegitimate node. Therefore, any node in the same link can manipulate NS or NA messages and then launch a Denial-of-Service (DoS) attack. Techniques proposed to secure AR and DAD include Secure NDP (SeND) and Trust-NDP (Trust-ND); however, these techniques either entail high processing time and bandwidth consumption or are vulnerable to DoS attacks because of their designs. Therefore, to secure AR and DAD, this study aims to introduce a prevention technique called Match-Prevention, which secures target IP addresses and exchange messages (i.e. NS and NA). The processing time, bandwidth consumption and DoS prevention success rate of Match-Prevention in different scenarios are evaluated, and its performance is compared with those of existing techniques, including Standard-Process (i.e., Standard-AR and Standard-DAD), SeND and Trust-ND. Results show that Match-Prevention requires less processing time during AR and DAD processes and less bandwidth consumption compared with other existing techniques. In terms of DoS prevention success rate, the experiments show that Standard-Process and Trust-ND are unable to secure AR and DAD from DoS attacks, whilst SeND is vulnerable to flooding attacks. By contrast, Match-Prevention allows IPv6 nodes to verify the incoming message, discard the fake message before further processing and prevent a DoS attack during AR and DAD in an IPv6 link-local network.

INDEX TERMS IPv6 link-local Network, neighbour discovery protocol, duplicate address detection, address resolution.

I. INTRODUCTION

The worldwide system of interconnected computer networks known as 'the Internet' has become significantly prevalent in our daily lives. The Internet has become an essential need for human beings, similar to other basic needs such as water and electricity [1]. The huge numbers of Internet users and their devices, which are connected through the Internet Protocol version 4 (IPv4), has depleted the IPv4 [2]. Therefore, Internet Protocol version 6 (IPv6) was introduced by the Internet Assigned Numbers Authority (IANA) to tackle the main problem of address exhaustion in IPv4 [3]. According to

The associate editor coordinating the review of this manuscript and approving it for publication was Sharief Oteafy.

Google statistics, over 30% of its users accessed Google via IPv6 in January 2020 [4].

IPv6 introduced a new protocol, namely Neighbour Discovery Protocol (NDP) as specified in RFC 4861, to mitigate the security issues in a link-local network [5]. NDP is a key protocol in the IPv6 network, and it has many processes, such as Address Resolution (AR), Neighbour Unreachability Detection (NUD) and Duplicate Address Detection (DAD). The designer of the IPv6 network presumes that the local area network (LAN) comprises trusted nodes. Therefore, every node inside the LAN is trusted by the NDP. This condition makes the network vulnerable to various attacks, such Denial Of Service (DoS), which is the most dangerous attack on the IPv6 link-local network [6], [7]. Given the important processes of NDP and its vulnerability to DoS attacks, numerous techniques have been proposed to secure these processes. In this paper, the most common techniques are presented and evaluated in terms of processing time, bandwidth consumption, and prevention success against DoS attacks.

The rest of this paper is organised as follows. Section II illustrates the NDP functions. Sections III and IV discuss the security issues of AR and DAD, respectively. Section V elaborates on the related work. Sections VI and VII present the design and workflow of the proposed technique, respectively. Section VIII shows the security analysis of the Match-Prevention technique. Section IX presents the implementation and setup of the test-bed environment. Section X and XI elaborate on and discuss the results of the experiment, respectively. Finally, Section XII presents the conclusion and future work.

II. NEIGHBOUR DISCOVERY PROTOCOL

NDP represents several messages and processes for establishing communication amongst nodes, routers and hosts located in the same IPv6 network. It replaces some protocols found in IPv4, including router discovery, AR Protocol (ARP), Internet Control Message Protocol (ICMP) and ICMP redirect. IPv6 NDP allows nodes to detect neighbours on a similar LAN and let their existence be known to their neighbours. NDP also has important processes, such as AR, DAD and NUD [8].

In an IPv6 link-local network, all nodes can automatically configure their addresses by using Stateless Address Autoconfiguration (SLAAC), which is one of the NDP functions, without the need for Dynamic Host Configuration Protocol version 6 (DHCPv6) [9], which is one of the NDP processes. SLAAC provides the IPv6 host with the power to generate a link-local network and a global address without the need for manual intervention. For the purpose of performing the mentioned processes between nodes (i.e. routers and IPv6 hosts), five messages are used during these processes as listed below:

- Router Solicitation (RS) – type 133: The hosts usually begin the RS when the system starts up so that an immediate router advertisement message is taken rather than waiting for the following scheduler timer.
- Router Advertisement (RA) – type 134: RA is sent periodically from routers or in response to RS to advertise presence and link-specific parameters, such as link prefixes, link maximum transmission units and hop limits.
- Neighbour Solicitation (NS) – type 135: The hosts send the NS to request the medium access control (MAC) address of another host and for functions such as AR and NUD.
- Neighbour Advertisement (NA) – type 136: NA is transmitted to change the host MAC address, announce IP addresses or respond to NS messages through the AR, DAD or NUD processes.
- Redirect Message (RM) – type 137: RM is sent from routers to redirect user traffic from one path to another significant path.

Although the NDP considers the essential protocol in IPv6, it does not have a sufficient security strategy to verify and authenticate messages exchanged among hosts located on the same link. In addition, the abovementioned messages are not secured by their designs; thus, any attacker placed on the same network may join any NDP process (e.g. AR, DAD, etc.) and disturb this process by manipulating the messages and carrying out a DoS attack [10]. The section below discusses the vulnerabilities to a DoS attack of two common NDP processes performed frequently from each of the IPv6 hosts in the network.

III. ADDRESS RESOLUTION PROCESS AND ITS SECURITY CHALLENGES

Internet Protocol (IP) addresses cannot be directly used to communicate in the network. The IP address of a host is just an abstract network layer address. If a host wants to transmit the network layer's packet to the target host, then the packet needs to be converted into a MAC frame that can be sent to the actual network [10]. Therefore, regardless of whatever protocols the network layer used, the physical address of the target in the data-link layer must be known to achieve transmission. Thus, AR becomes an important part of the IP [11].

Each host in LAN has an AR-cache table that stores other hosts' IP and MAC address mapping. If Host_A needs to send packets to Host_B, A will initially search its AR-neighbour cache table to check whether Host_B has an IP address. If an address is found, then Host_A will send packets with a corresponding MAC; if no address is found, then Host_A will carry out an AR by multicasting an NS message that requests Host_B to provide its MAC address. All hosts in LAN can receive the NS message, but only Host_B will answer an AR via an NA message that contains Host_B's IP and MAC addresses, while Host_A receives the NA message and updates its AR-neighbour cache table and then sends data to Host_B following the MAC of Host_B [12]. The AR-neighbour cache table can be shown by using these commands in the command prompt of the Windows operating

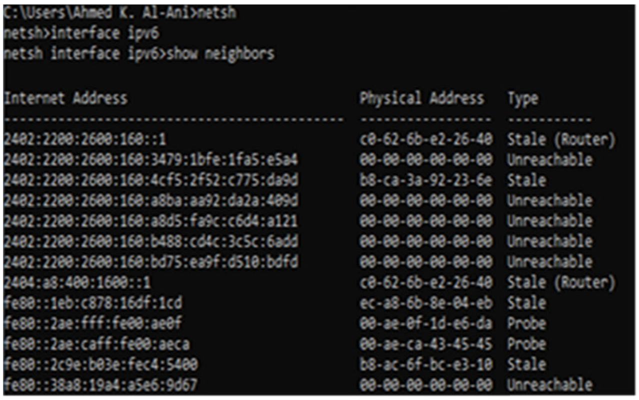

FIGURE 1. AR-neighbour cache Table.

system, as shown in Figure 1.

$$netsh \rightarrow interface\ ipv6 \rightarrow show\ neighbours$$

The AR-neighbour cache has different states of the IP address on the basis of their reachability. These states are shown in Table 1.

TABLE 1. IP address states.

Type (states)	Definition
Unreachable	The IP address is unreachable.
Incomplete	AR is in progress, and NUD has not determined the link-layer address of the neighbour.
Probe	The neighbour is no longer known to be reachable, and NUD is sending unicast NS probes to verify reachability.
Delay	The neighbour is no longer known to be reachable, and traffic was recently sent to the neighbour. NUD does not probe the neighbour immediately. NUD delays sending probes for a short time so that the upper layer protocols can provide reachability confirmation.
Stale	The neighbour is no longer known to be reachable, and until traffic is sent to the neighbour, NUD makes no attempt to verify its reachability.
Reachable	The neighbour is known to have been reachable recently, that is, within the last minute.
Permanent	The neighbour is statically provisioned and will not expire unless you remove it through configuration.

The AR process is simple and efficient but has many security risks. Firstly, it assumes that all hosts on the same link are trustworthy, but the reality is that a deceptive host caused by a virus or a malicious program exists in the network. Secondly, AR lacks an authentication mechanism. Specifically, if Host_A wants to communicate with Host_B, after performing AR multicasting, if a reply packet is received, then Host_A will not check whether an AR request was sent or an AR reply is real or not; Host_A will just update its AR-neighbour cache table as long as the target address is a MAC [13].

For example, when Host_A wants to communicate with Host_B but does not have Host_B's MAC address at the AR-neighbour cache table, it performs AR process by multicasting an NS message, which asks what the MAC address is for this IP 'Host_B's IP address'. If there is an attacker on the same LAN, can reply the NS message via a fake NA message. Given that AR does not have any verification mechanism to authenticate the incoming message (i.e. NA message), Host_A will receive and accept the fake NA message and update its AR-cache table with fake information. This attack, named DoS-on-AR, is shown in Figure 2. Table 2 shows the IP and MAC addresses of the hosts.

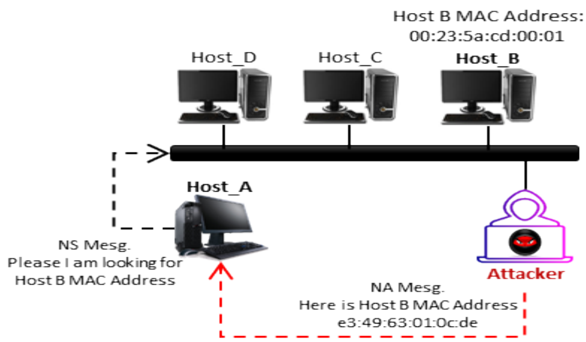

FIGURE 2. DoS-on-AR process.

TABLE 2. IP and MAC addresses of hosts.

Host Name	Host IP	Host MAC
Host_A	fe80::428d:5cff:fee5:f406	01:00:5e:40:44:f8
Host_B	fe80::7892:e2ff:fe97:d520	00:23:5a:cd:00:01
Host_C	fe80::ee34:02ff:fe90:8246	12:55:e8:c2:23:27
Host_D	fe80::4680:085f:f010:2780	89:36:e0:07:d2:11
(Attacker)	fe80::b342:5882:fe00:f701	e3:49:63:01:0c:de

IV. DUPLICATE ADDRESS DETECTION AND ITS SECURITY CHALLENGES

Every IPv6 host performs the DAD process before assigning IPv6 addresses to ensure that none of the hosts in the same link-local network shares the same IP address. The IP conflict is low because of the immensity of the address space. However, this will not be the case in the future as the number of devices exponentially increase due to the emergence of novel technologies, including the Internet of Things and cloud computing.

Two types of NDP messages, NS and NS, are used during the DAD in an IPv6 link-local network. When the target host intends to join an IPv6 local-link network or when a host in the same link generates a new IP address as a tentative IP address, the uniqueness of this generated address needs to be verified. To this end, the target host multicasts a number of NS messages to all hosts in the same link. In case the tentative address has already been assigned by another host in the same link, an NA message should be sent by the existing host as a reply to the NS message indicating that the generated tentative IP address is not unique. Afterwards, the target host must generate a new tentative IP address and re-perform the DAD to verify the uniqueness of this generated address until NA messages are no longer received [14].

The DAD presumes that all neighbour hosts in an IPv6 link-local network are reliable. Therefore, upon receiving an NA message from other hosts during the address verification process, the target host can act accordingly regardless of the validity of the message. In this scenario, a malicious host may respond to an NS message by sending a fake NA message claiming that the generated tentative IP address has already been assigned; doing so will prevent IPv6 hosts from configuring this unique IP address. Therefore, these hosts are unable to join the IPv6 network and communicate with other hosts in the network. This type of attack is referred to as a DoS-on-DAD, which prevents hosts from configuring IP addresses in an IPv6 link-local network, as revealed in previous studies [15]. Figure 3 illustrates a DoS-on-DAD attack in an IPv6 link-local network.

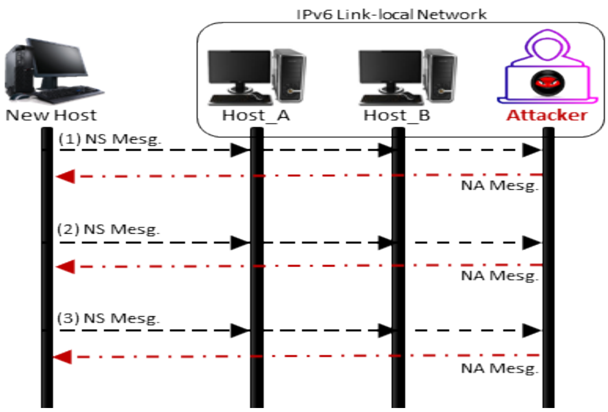

FIGURE 3. DoS-on-DAD attack.

V. RELATED WORKS

As shown in Section IV, AR and DAD are completely vulnerable to DoS attacks. These processes are considered essential in an IPv6 link-local network, which needs to perform the processes from each host frequently. Therefore, many studies have proposed different techniques based on various methods to secure AR and DAD. This section presents some important works related to AR/DAD security along with their limitations.

A. NETWORK MONITORING TECHNIQUES

Some techniques attempt to secure the AR/DAD in an IPv6 link-local network by monitoring the network traffic and sending a notification to the administrator for any abnormal behaviour, such of the mechanisms NDP monitoring (NDPmon) and Intelligent NDP monitoring (INDPmon).

NDPmon, introduced by Beck in 2007 [16], is a tool that is very similar to Arpwatch for IPv4 in terms of features but provides additional attack-detection capabilities. NDPmon requires the deployment of a central server in a LAN and has been utilised by network hosts to detect NDP anomalies. It sends emails to the administrator via a report or by writing on the system log. NDPmon is based on three phases, namely training, learning and monitoring. During the training phase, NDPmon assumes that all nodes and network activities are legitimate; that is, this reporting mechanism is assumed to be disabled. Some studies [17], [18] have highlighted the disadvantages of using NDPmon. For instance, a compromised node during the training phase may cause a total detection failure. NDPmon is also vulnerable to potential false positive alarms if changes are made to the network interface card of a node. In addition, detection schemes may be bypassed when malicious nodes break the ICMPv6 packets that carry NDP information [19]. Although NDPmon prevents the attacks launched by illegal users, the attacks inflicted by legal users are very difficult to detect. Song and Ji [20] revealed that although NDPmon can detect attacks and send warnings for ordinary network behaviour, this tool is unable to stop these attacks.

Another technique for security monitoring, called INDPmon, was introduced in 2015 [21]. This technique uses the extended finite state machine (EFSM) to model the main NDP processes and detects abnormal behaviour on the basis of strict anomaly detection. INDPmon monitors network traffic by investigating the NDP behaviour. The strict anomaly detection technique is employed to observe any contravention in NDP. Given that the abnormality features deviate from standard behaviour, new failure states are defined to hinder any disallowed event or illegal transaction. Strict anomaly detection can define failure states in EFSM and report any violation of the protocol fundamentals. These violations often result from protocol misconfiguration or attacks. Therefore, INDPmon can also warn the network of NDP attacks because the majority of these attacks violate the protocol rules. Previous studies [10], [15] show that INDPmon can only identify NDP attacks that violate the fundamental protocols (i.e. protocol-rule-related violations). However, other violations, such as spoofed IP or MAC addresses, are difficult to detect. For instance, attackers can flood a network with NDP messages by using a spoofed IP address, thereby disabling INDPmon from differentiating a real IP address from a spoofed one. In sum, INDPmon cannot secure the AR against DoS attacks.

B. EXTEND NDP-HEADER TECHNIQUES

Some studies have attempted to secure AR and DAD in an IPv6 link-local network by extending the NDP header and adding a new option to the header. The most common techniques used in NDP header extension are Secure NDP (SeND) and Trust-NDP (Trust-ND).

SeND [22] adds several new options to NDP, including the cryptographically generated address (CGA) option to verify CGA senders. SeND uses CGAs to ensure the ownership of the claimed IPv6 address, as defined in RFC 3972 [23]. Other options include the Rivest–Shamir–Adleman (RSA) cryptosystem signature option, which attaches a public key-based signature and time stamp, and the Nonce option, which validates unsolicited advertisements, redirects all unanswered messages and validates advertisement messages sent as responses to solicited messages to prevent replay attacks. SeND also introduces two new ICMPv6 message types, including certificate path solicitation and certificate path advertisement. SeND aims to prevent NS and NA spoofing, NUD failure, DoS-on-AR and NDP DoS attacks [24]. However, previous studies [7], [10], [15] reveal that the security options of SeND, including CGA and RSA, hinder the implementation of this mechanism as an NDP extension in an IPv6 link-local network. Moreover, CGA cannot establish the identity of a legitimate host, thereby allowing attackers to capture the NDP messages, modify the CGA parameters and, eventually, exploit the target hosts. Another limitation of SeND is that it introduces a considerable amount of process overhead because of its design. With CGA and RSA as the main components of SeND, this mechanism requires additional processing time and consumes CPU resources and network bandwidth, thereby increasing its complexity [25]. Given these drawbacks of SeND, especially its complexity, malicious hosts can launch DoS attacks, such as flooding attacks, during NDP processes (i.e. AR and DAD) in an IPv6 link-local network.

Praptodiyono *et al.* [25] proposed a new technique called Trust-ND for protecting AR and DAD processes and securing the exchange of NDP messages amongst hosts in an IPv6 link-local network. Trust-ND has a light design and uses the SHA-1 hash algorithm to achieve the required security [26]. It also uses a new security option called the *trust*-option, which is attached to NDP messages to guarantee secure communication amongst hosts. Trust-ND depends on the concept of trust; it requires the host to verify NDP messages upon their receipt. This mechanism performs address verification much faster than either SeND because it is based on the SHA-1 hash function. Some researchers claimed that Trust-ND is a light security method for IPv6 DAD, whilst others [27], [28] argued that the SHA-1 hash function is vulnerable to hash collision attacks. These studies show that any malicious host can launch hash collision attacks against SHA-1. Given that Trust-ND depends on the SHA-1 hash function to achieve security, this mechanism is susceptible to collision and DoS attacks during AR and DAD in an IPv6 link-local network. Therefore, by design, Trust-ND is unsuitable for securing AR in an IPv6 network.

Table 3 summarises the limitation of the proposed techniques for securing AR and DAD processes in an IPv6 link-local network.

After the security concerns of AR and DAD in an IPv6 link-local network are consolidated, an efficient technique for securing such processes (i.e. AR and DAD) and the exchange of NS and NA messages from DoS attacks must be developed. This technique should be able to address the main drawbacks of existing techniques and satisfy the following requirements:

TABLE 3. Limitations of related works.

Mechanism Name (Author/Year)	Limitations
NDPmon Beck et al. (2007)	• Unable to distinguish the normal and abnormal behaviour of IP and MAC addresses • Encounters problems during the training phase • Requires database support to monitor network traffic • Requires a third-party device to provide additional services
INDPmon Firas et al. (2015)	• Can only detect attacks and not prevent them • Shows weaknesses against spoofed IP attacks. • Cannot be easily deployed • Requires additional resources to achieve complete security
SeND Arkko et al. (2005)	• Has high computational cost, especially for the CGA and RSA options • Unable to identify the CGA address utilised by the legitimate node • Cannot utilise CGA for static address configuration. • Shows high complexity • Increases network overhead and bandwidth consumption
Trust-ND Praptodiyono et al. (2016)	• Degrades network bandwidth performance • Vulnerable to DoS attacks due to its design • Shows vulnerability to collision attacks due to its design

I. Lightweight: Message verification must be accomplished with minimal process overhead. The complexity of any technique draws upon the algorithm that they use for the computation.
II. High security: The exchange of the target IP address amongst hosts should be protected by securing NDP messages from MITMA, hash collision, brute-force and IP spoofing attacks, all of which can trigger a DoS attack against AR and DAD.
III. Self-decision: The technique must be able to make independent decisions in securing the AR/DAD instead of relying on a third party to protect IPv6 hosts from malicious threats to facilitate the verification of their addresses.

VI. PROPOSED MATCH-PREVENTION TECHNIQUE

To fulfil the aforementioned requirements (i.e. lightweight, high security and self-decision), Match-Prevention is designed to secure AR and DAD against DoS attacks in an IPv6 link-local network. In designing the proposed technique, four objectives must be achieved as listed below:

1. To propose a cryptographic mechanism for preventing the disclosure of target IP addresses during the AR and DAD processes in an IPv6 link-local network.
2. To propose a mechanism for securing NS and NA messages by utilising the experimental option of NDP without jeopardising the original structure to security challenges.

3. To design a rule-based mechanism with the aim of preventing DoS attacks against AR and DAD in an IPv6 link-local network.
4. To evaluate the performance of the proposed technique in terms of its processing time, bandwidth consumption and DoS prevention success rate.

The architecture of the proposed technique comprises three main stages (Figure 4). This section describes the main stages, components and steps of this mechanism.

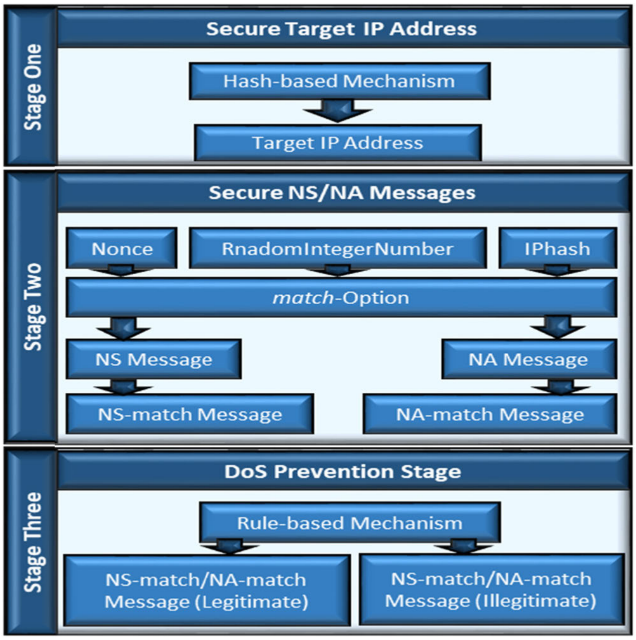

FIGURE 4. Architecture of the match-prevention technique.

A. SECURE TARGET IP ADDRESS (STAGE ONE)

This stage aims to achieve the first objective of this research, which is to prevent the disclosure of the target IP address during AR and DAD in an IPv6 link-local network. To achieve this goal, the hashing mechanism is applied to hide the address for security purposes. Firstly, all the hosts should generate their IP addresses by using the privacy extension method [29] instead of the EUI-64 bit method [30]. The privacy extension method randomly generates the Interface ID (IID) of an IPv6 address, thus helping to prevent an attacker from predicting the IP address and maintaining the privacy of the host [31].

Moreover, there are two methods for encrypting (hiding) data: encryption cryptography, which uses a secret key called 'symmetric encryption' and a public/private key called 'asymmetric encryption', and hash-function cryptography. The key difference between encrypting and hashing data is that only the former can decrypt data. Many studies have likewise shown that encryption cryptography requires extensive calculation. Therefore, encryption cryptography is not appropriate for the proposed technique, which aims to achieve less processing time requirement. Hash-function cryptography is also more lightweight than encryption cryptography and, as some researchers [32] argued, is faster in terms of processing. Therefore, this study adopts hash-function cryptography.

A cryptographic hash function maps a random length input (message or plaintext) into a fixed-length output (message digest or hash value). This function has an easily computable design and achieves specific security properties, such as preimage resistance, second preimage resistance and collision resistance. These properties are explained in detail as follows (Rogaway & Shrimpton, 2004):

(i) Preimage resistance – Given digest $D = H(m)$ for a message m, locating any message that provides this digest is computationally difficult.
(ii) Second preimage resistance – Given message m, locating a different message m' that gives the same digest is computationally difficult. For instance, for a message m', we have $H(m) = H(m')$.
(iii) Collision resistance – Locating two messages m, m^' that give the same digest is computationally difficult. For instance, for a couple of messages m and m', we have $H(m) = H(m')$.

In these properties, H represents the hash function, which takes a message (plaintext) m of arbitrary length and creates a fixed-length digest (hash value) represented by D.

One of the most common attacks being faced by the cryptographic hash function is the brute-force attack [14]. When launching this type of attack, attackers try every possible input value until they generate the expected output. For instance, the brute-force preimage search attack requires attackers to check the average (2^{n-1}) values before finding the right message, and a long input cannot be broken within a short period.

Numerous hash-function algorithms have been proposed over the past few years, including MD4, MD5 and secure hash algorithms (e.g. SHA-1, SHA-2 and SHA-3). Previous study [33] revealed that all these algorithms are vulnerable to hash collision attacks except for SHA-3, for which no security gap can be found. According to [34], SHA-3 is a new generation of SHA that produces promising results by utilising a fast sponge to create hash values (thereby leading to its speed advantages) and generates a random output length unlike the traditional hashes being used today. SHA-3 also demonstrates remarkable security strength levels against attacks and has flexible performance implementation options and security trade-offs. Thus, based on the literature review, SHA-3 is the most hash-function algorithm that can be adopted in our proposed Match-Prevention technique.

SHA-3 has six hash functions, including four cryptographic hash functions (i.e. SHA3-224, SHA3-256, SHA3-384 and SHA3-512) and two extendable-output functions (i.e. SHAKE128 and SHAKE256). Match-Prevention utilises SHAKE128 given its 128-bit output. In this way, the proposed technique consumes less processing time for extensive calculations during the generation verification of

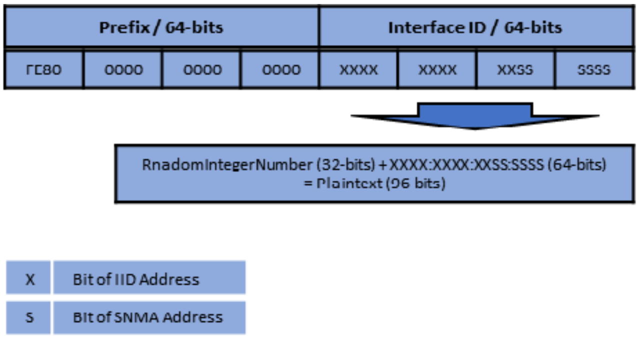

FIGURE 5. Plaintext combination.

NS and NA messages through the AR/DAD processes in an IPv6 link-local network.

However, using the sole IID (64 bits) as an input 'plaintext' for SHAKE128 is vulnerable to brute-force attacks. Therefore, aside from IID (64 bits), a random integer number (RIN) from 0 to 32 bits are used as inputs to prevent any type of attack, including brute-force attacks, against the hash function, as revealed in the security analysis of brute-force attacks in Section VIII. The input 'plaintext' for SHAKE128 is a combination of IID (64 bits) and a RIN (32 bits), thereby leading to 96 bits. Figure 5 presents the plaintext combination.

As shown in Figure 5, the 96-bit plaintext is a combination of the first 64 bits of IID of the target IP address and the RIN (32 bits). The SHAKE128 hash function should then be applied on the plaintext as illustrated in Equation (1).

$$Hashvalue = hash\left(TargetIPAddress + RIN\right), \quad (1)$$

where *Hashvalue* denotes the output of the hash calculation, *hash* is the SHAKE128 hash-function algorithm, *RIN* is a random integer number that ranges from 0 to 32 bits and *TargetIPAddress* is the 64-bit IID from the target IP address. After the SHAKE128 hash function is applied on the plaintext, the output, which is usually referred to as the hash value (128 bits), is carried by the IPhash field in the *match*-option as detailed in the following subsection.

B. SECURE NS AND NA MESSAGES (STAGE TWO)

This stage aims to achieve the second objective of this study, which is to secure NS and NA messages without jeopardising the original structure. Match-Prevention introduces NS-match and NA-match messages without changing the original structure of NDP but uses the NDP experimental option. Therefore, NS-match and NA-match messages are secured during AR and DAD in an IPv6 link-local network, unlike the Standard-Process (Standard-AR and Standard-DAD) messages (i.e. NS and NA) which do not have any authentication mechanism to validate the originality of incoming NS and NA messages. To achieve this objective, the *match*-option is proposed and explained in detail below.

The experimental NDP option, *match*-option, comprises three main fields, namely Nonce, RIN and IPhash, and aims to distinguish valid NS/NA messages from bogus ones.

Type (1 byte)	Length (1 byte)	Nonce (2 bytes)
RIN (4 bytes)		
IPhash (16 bytes)		

FIGURE 6. Design of the *match*-option.

To ensure that only valid hosts can communicate in an IPv6 link-local network, the *match*-option must be appended onto the NS and NA messages, which are then transformed into NS-match and NA-match messages, respectively. Figure 6 presents the *match*-option and its respective fields.

To maintain the original structure of NDP messages, the *match*-option design (Figure 6) follows the option format of RFC 4861 [35]. Given that all NDP options should include the type and length fields, the proposed *match*-option also contains these fields. The minimum length of the NDP option should be 8 bytes (64 bits); otherwise, the option must be padded. The *match*-option also comprises 24 bytes divided into five fields as follows:

- ➢ Type: 1-byte identifier that indicates the option type carried by the NDP message. The *match*-option type is 253 because this option is used for experimentation [36].
- ➢ Length: 1-byte field that indicates the total length of the *match*-option, including the type and length fields in an 8-byte unit (64 bits). Given that the total length of the *match*-option is 24 bytes, the value of the length field is 3.
- ➢ Nonce: This is a 2-byte field because the NDP messages are presented in the form of request and response (i.e. NS and NA messages, respectively). Therefore, using the Nonce option or sequence number can ensure that a reply message is sent for the corresponding solicitation message [22].
- ➢ RIN: This is a 4-byte field that generates a RIN ranging from 0 to 32 to serve as an input with 64-bit IID, thereby preventing attackers from breaking it. This field holds the generated RIN, which will then be used by the receiver side for verification purposes.
- ➢ IPhash: This is a 16-byte field that carries the hash value of 128 bits, which result from hashing the IID (64 bits) with a RIN (32 bits) to verify the NS and NA messages during AR and DAD in an IPv6 link-local network by matching the hash values between the sender and receiver. Specifically, this process determines whether the message was generated by a legitimate or fake host. This field is also considered the main field of the *match*-option.

The *match*-option is appended into each NS and NA message to transform them into NS-match and NA-match messages, respectively, during AR and DAD. Each message without the *match*-option should be discarded, and those messages with this option should undergo a computation hash, where the hash values of the sender and receiver are matched.

C. DOS PREVENTION (STAGE THREE)

This stage aims to prevent a DoS attack on the AR and DAD processes in an IPv6 link-local network by using a rule-based mechanism to allow the receiver and sender hosts to verify NS and NA messages without any third-party intervention. Therefore, the third requirement, namely self-decision by validating NS or NA messages, is satisfied.

In this mechanism, the host performs verification on both messages (NS and NA messages). When performing AR and DAD, the target host (sender) must firstly generate the NS-match message with a *match*-option and its corresponding fields, specifically, Nonce, which has a sequence number value; RIN, which is a random integer number; and IPhash, which carries the hash value.

The *match*-option must be appended to each NS message to transform this message into an NS-match message. Afterwards, the NS-match message is multicast to the solicited-node multicast address (SNMA) based on the last 24 bits of the target IP address. Using the SNMA instead of an all-node multicast can help reduce the network overhead in a wide-scale network. After the NS-match message is multicast to the SNMA, all existing hosts that share the same SNMA will receive the NS-match message. These hosts should then verify the NS-match message by checking the existing *match*-option, performing the computational hash and matching the hash values. If the hash values match, then the hosts perform the process (i.e. AR or DAD) reply by sending an NA-match message. The rule-based codes for verifying the NS-match message is elaborated below.

Verification (NS-match Message)
receive NS-match message
if NS-match message does not have a *match*-option, ***then***
return false // Illegitimate NS-match Message
if Nonce of NS-match is empty, ***then***
return false // Illegitimate NS-match Message
if RIN of NS-match is empty, ***then***
return false // Illegitimate NS-match Message
if IPhash of NS-match is empty, ***then***
return false // Illegitimate NS-match Message
else
return true // Legitimate NS-match Message

Afterwards, the target host receives the NA-match message as a response to its NS-match message. The target host should first check the existing *match*-option before performing the computational hash. On the basis of the computational hash, in case the result is matched, the received NA-match message is sent from a legitimate host; otherwise, the target host considers the NA-match message to be illegitimate. In addition, if 3 seconds have passed, the target host should discontinue the process and stop receiving any NA-match messages. The rule-based codes for verifying the NA-match message is elaborated below.

Verification (NA-match Message)
receive NA-match message
if time out, ***then***
return false // Illegitimate NS-match Message
if NA-match message does not have a match-option, ***then***
return false // Illegitimate NS-match Message
if Nonce of NA-match $\neq$ Nonce of NS-match, ***then***
return false // Illegitimate NS-match Message
if RIN of NA-match = RIN of NS-match, ***then***
return false // Illegitimate NS-match Message
if IPhash of NA-match $\neq$ IPhash of NS-match, ***then***
return false // Illegitimate NS-match Message
else
return true // Legitimate NA-match Message

This process allows the target host to configure its IP address and prevent a DoS-on-DAD attack against the DAD in an IPv6 link-local network. The next section explains the workflow of the proposed Match-Prevention mechanism in detail.

VII. WORKFLOW OF MATCH-PREVENTION IN THE IPV6 LINK-LOCAL NETWORK

As mentioned in literature review, the Standard-Process for both AR and DAD is vulnerable to DoS attacks that can be launched by sending fake NA messages as a response to NS messages during the process. Many techniques have been proposed to secure these processes in an IPv6 link-local network, but these techniques are vulnerable to DoS attacks because of their designs.

Match-Prevention secures the AR and DAD processes in an IPv6 link-local network against DoS attacks. The proposed *match*-option should be appended to all NS and NA messages, which are then transformed into NS-match and NA-match messages, respectively. The validation is performed by the receiver and the sender by distinguishing legitimate messages from illegitimate ones. This security process covers the exchange of target IP addresses amongst hosts during the AR and DAD processes and prevents DoS attacks on the sender and receiver hosts. Meanwhile, Nonce appears in every (NS/NA)-match message to ensure that this message is a fresh response to the message request. The workflow of the Match-Prevention technique is similar for both processes (i.e. AR and DAD) as presented below:

1. Firstly, all the IPv6 hosts should generate their IP address by using the privacy extension method instead of EUI-64.
2. The target host should generate a random integer, and then the hash algorithm should be applied on this random integer with the 64-bit IID from the target IP address. The hash value (128 bits) of the process is inserted into the IPhash field in the *match*-option,

whilst the RIN field carries the integer used in hashing.

3. The Nonce field is set with a random number to ensure that a reply message is sent for the corresponding solicitation message.
4. IPhash, RIN and Nonce fields are combined to create the *match* -option.
5. The *match*-option is appended to each NS message, which then becomes an NS-match message. Afterwards, this message is multicast to the SNMA according to the last 24 bits of the target IP address (FF02::1:FFSS:SSSS).
6. All existing hosts on the same IPv6 link with the same SNMA will receive the NS-match message.
7. The existing hosts check whether the NS-match message comes from a legitimate host on the basis of the existing *match*-option. If the *match*-option is not found in the NS-match message, then the host discards this message.
8. In case the *match*-option exists, the receiving hosts should extract the random integer found in the NS-match message with a 64-bit IID and then apply the hash on them. Afterwards, the hosts should match the hash value results with the hash value existing in the IPhash field in the NS-match message. In case of a match, the host should reply by sending an NA-match message; otherwise, the host should discard the NS-match message.
9. The process for sending the NA-match message is the same as that for sending the NS-match message, that is, after a new random integer is generated and inserted into the RIN field, this integer is hashed with the 64-bit IID from the tentative IP address and the hash value is inserted into the IPhash field. The value of Nonce should be the same as that of the Nonce field of the NS-match message to indicate that the message is replying to it.
10. The three fields should be combined to create a *match*-option that is appended to the NA message, which then becomes the NA-match message, and then sent to the SNMA.
11. The target host receives the NA-match message. If 3 seconds have passed since the NS-match message was sent, then the NA-match message should be discarded; otherwise, the NA-match message should be processed for further verification.
12. The existence of the *match*-option should be checked. If the option is found, then the process continues; otherwise, the message is discarded.
13. The value in the RIN field should also be checked. It should be different from that in the RIN field of the NS-match message sent earlier to avoid a reply attack.
14. When the *match*-option exists, the target host should do the same procedure by performing the hashing and matching the hash values with the hash value in IPhash in the NA-match message. In case of a match, then the message is considered to come from a legitimate node; otherwise, the NA-match messages should be discarded.

In this way, both hosts (sender and receiver) can verify the incoming messages (i.e. NS and NA messages) and AR and DAD can be performed successful in an IPv6 link-local network. Figure 7 shows the workflow of the proposed Match-Prevention technique for securing AR and DAD in an IPv6 link-local network.

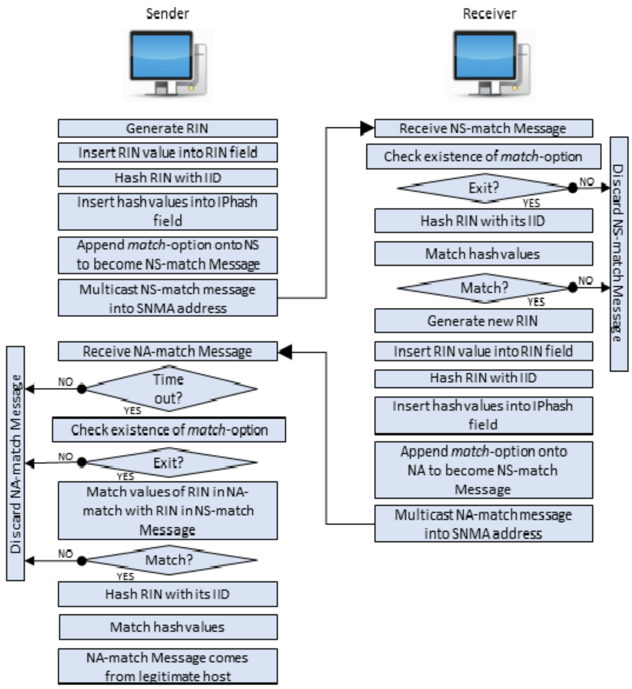

FIGURE 7. Workflow of match-prevention technique.

VIII. SECURITY ANALYSIS OF MATCH-PREVENTION

A. SECURITY ANALYSIS OF TARGET IP ADDRESS

The target IP address is important information in both AR and DAD processes in an IPv6 link-local network. In a Standard-Process (i.e. Standard-AR and Standard-DAD), the host multicasts the target IP address through NS messages in plaintext. Hence, all the hosts located on the same link will receive the NS messages, including the attacker, who can then launch a DoS attack by sending fake NA messages because it knows the target IP address.

In Match-Prevention, the target IP address is hidden via a cryptographic hash function. Thus, the attacker will not be able to know the target IP address and perform its attack.

B. SECURITY ANALYSIS OF PROBABILITY OF COLLISION ATTACK

Take for instance a bandwidth (BW) on the network of 10 Gbyte, a message (Mz) size of 102 bytes for NS-match and NA-match and the time (T) of 3 seconds. Equation (2) can be used to calculate how many messages (Nm) the attacker can send at most as follows:

$$N = \frac{BW}{Mz} * T$$
$$Nm = \frac{10 * 2^{30}}{102} * 3 \; second = 315,806,418, \tag{2}$$

where (315, 806, 418) is the number of messages that can be sent by the attacker on the LAN within 3 seconds. Equation (3) illustrates the probability of a collision attack P, where n is the number of bits of plaintext:

$$P = \frac{Nm}{2^n}$$
$$P = \frac{315,806,418}{2^{96}} = 3.9 * 10^{-21}\%. \tag{3}$$

Accordingly, the successful collision attack is ignored in the proposed Match-Prevention technique.

C. SECURITY ANALYSIS OF BRUTE-FORCE ATTACK

In a brute-force attack, the attacker attempts to get all probabilities for breaking the hash within 3 seconds. Let us assume that computers are used to their full computing power for hashing, with a total of (U) CPUs, each with (Nc) cores running a frequency (Fr) (in Hertz) for total time (t) (in seconds), to hash (M) messages each b-byte, with each hash produced requiring (Cy) cycles of one execution thread of one core of one CPU. Therefore, the t need can be derived from Equation (4) below:

$$T = \frac{M * Cy}{U * Nc * Fr} \tag{4}$$

where the values of the Cy cycles are obtained from the SHA-3 benchmark indexed by a machine. Five machines are selected in this study to calculate the time needed for a brute-force attack on Match-Prevention. Table 4 lists the machines' details:

TABLE 4. Machine details.

Machine No.	CPU; Cores in machine × MHz	Cycles
1	Intel Xeon Gold 6130; 32 × 2100	930
2	Intel Xeon Gold 6150; 18 × 2700	940
3	Intel Xeon E5-2609 v4; 8 × 1700	1584
4	NVIDIA Tegra X1; 4 × 1734	1679.04

Table 4 and Equation 4 indicate that the time needed for a brute-force attack for machine 1 is

$$T = \frac{2^{96} * 930}{1 * 32 * 2100 * 10^9} = 35,251,452,470 \; years.$$

Therefore, a brute-force attack against Match-Prevention is impossible. Figure 8 shows the brute-force attack time of other machines.

D. NETWORK OVERHEAD ANALYSIAS

To measure the performance of the proposed Match-Prevention technique on the network with multiple machines existing on the same network, assumed 10 machines exist in the network and that a host wants to perform AR or DAD. In Match-Prevention, the NS-match message should multicast to the SNMA on the basis of the last 24 bits of the target IP address. Thus, the probability of existing hosts that will receive the NS-match message is low. To calculate the probability of machines that have the same SNMA, Equation (5) below is used as follows:

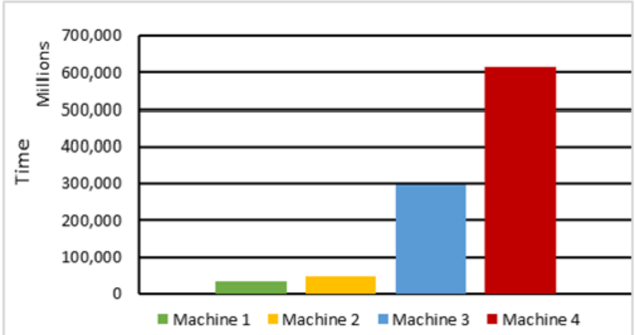

FIGURE 8. Time needed to launch a brute-force attack on four machines.

$$Probability = \frac{N_m}{2^{24}} \tag{5}$$

where $Probability$ is the probability of the hosts having the same SNMA, and N_m is the number of machines existing in the link. In the experiment, 10 hosts exist in the IPv6 link:

$$Probability = \frac{10}{2^{24}} = 5 * 10^{-6}\%.$$

Thus, the probability of existing machines having the same SNMA in the link is 5×10^{-6}, which is insignificant. Therefore, using Match-Prevention does not cause any overhead on the IPv6 network.

IX. IMPLEMENTATION AND TEST-BED SETUP ENVIRONMENT

A. TEST-BED SETUP ENVIRONMENT

Match-Prevention was implemented in accordance with the current IPv6 network architecture to guarantee that the NS and NA messages are recognised by all IPv6 hosts without extra changes or without adding any third party to the architecture. The test-bed has six nodes, including one gateway router, one switch, two existing hosts (EH_1 and EH_2), one new host (NH) and one attacker (AT). Figure 9 shows the experimental test-bed network topology.

The specifications of the hardware and software used for deploying the test-bed environment were selected in accordance with their requirements and in consideration of their availability in the IPv6 environment of the National Advanced IPv6 Centre (Nav6) of Universiti Sains Malaysia [37] to successfully conduct the experiments. These specifications are presented in Table 5.

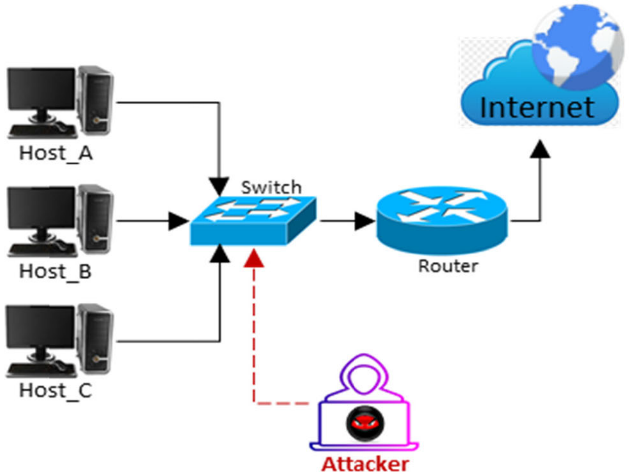

FIGURE 9. Test-bed setup environment.

TABLE 5. Hardware and software requirements for the test-bed environment.

Item Name	CPU	Memory	Operating System
Host_A	Intel(R) Core (TM)2 CPU Q8400 @ 2.66 GHz × 4	5.00 Gb	Windows 10 Pro
Host_B	Intel(R) Core (TM) i7-3770M CPU @ 3.40 GHz × 8	8.00 Gb	Ubuntu 16.04
Host_C	Intel(R) Core (TM) i7-2640M CPU @ 2. 30 GHz × 4	4.00 Gb	Ubuntu 16.04
Attacker	Intel(R) Core (TM) i7-2640M CPU @ 2.80 GHz × 4	4.00 Gb	Ubuntu 16.04
Item Name	**Type**		
Switch	Cisco Catalyst 2960 Fast Ethernet		
Router	Cisco Router C7200		

B. IMPLEMENTATION TOOLS

The Python programming language (Python, 1990) was used to build the Match-Prevention technique in the same way that comparative techniques (i.e. SeND and Trust-ND) were developed. Furthermore, to analyse the network traffic and the exchange of NDP messages (i.e. NS and NA messages) amongst IPv6 hosts in a link-local network, the Wireshark tool was used. A hacking tool must also be used to launch a DoS attack against AR and DAD in an IPv6 link-local network as well as ensure the performance and capability of the proposed technique in identifying incoming fake NDP messages and preventing a DoS attack. In this work, The Hacker Choice (THC) developed by Van Hauser in 2006 [38] was used. THC is specifically designed for IPv6 attacks and presents a complete tool kit for identifying the weakness of the existing IPv6 implementation. In addition, Scapy tool was used to craft NDP messages (i.e. NA messages) and perform an advanced attack. Scapy is a packet manipulation tool designed specifically for IPv6 networks. It was utilised in this study to craft NDP messages depending on the requirements of the attack scenario.

X. EXPERIMENTS AND RESULTS

Several evaluation metrics were applied using two experiment scenarios (i.e. normal and attack) to evaluate the proposed Match-Prevention technique and check whether it fulfils the study requirements. These metrics, including processing time, bandwidth consumption and DoS prevention success rate, were the same as those used in previous researches [20], [25], [39].

A. NORMAL SCENARIO

In the first scenario, Match-Prevention was implemented in a normal condition without any attacks. The goal of this scenario is to ensure that Match-Prevention is implementable. In this scenario, the bandwidth consumption and processing time for verifying NS-match and NA-match messages at the sender and receiver sides were measured. Such measurements can determine whether Match-Prevention satisfies the requirement of less complexity. Furthermore, to ensure a better evaluation of Match-Prevention in terms of processing time, the experiments performed in this scenario were examined 20 times to satisfy the computer science test requirements (Devore, 2015). In this way, Match-Prevention addresses the complexity issue, which remains one of the key challenges facing the AR and DAD processes in an IPv6 link-local network when under a DoS attack.

1) PROCESSING TIME

A long processing time indicates the complexity of the adopted mechanism. Complexity is regarded as a key measure for evaluating the security of a network (Erdheim, 2012). A more complex technique is generally more difficult to manage, whereas a less complex technique can operate faster and requires a shorter processing time, thereby improving the network performance. This study aims to secure the AR and DAD processes in an IPv6 link-local network by securing the exchange of NS and NA messages amongst IPv6 hosts during AR and DAD via a less complex security technique, which is one of the requirements of this study. To measure the complexity of the mechanism, its behaviour during AR and DAD should be analysed.

Figure 10 shows the total processing time for the Match-Prevention technique in comparison with the performance of existing techniques (i.e. Standard-Process, SeND and Trust-ND) on both AR and DAD.

Figure 10 reveals that, amongst the compared techniques, SeND consumes the longest processing time for both processes (AR and DAD) at 147.3878 and 151.9802 milliseconds on average, respectively, followed by Trust-ND (9.291 milliseconds on AR and 10.9775 on DAD). These long processing times can be ascribed to the extensive calculation requirements of these mechanisms when generating and verifying the messages being exchanged between hosts. By contrast, the proposed Match-Prevention technique only consumes 6.6349 and 7.7809 milliseconds of

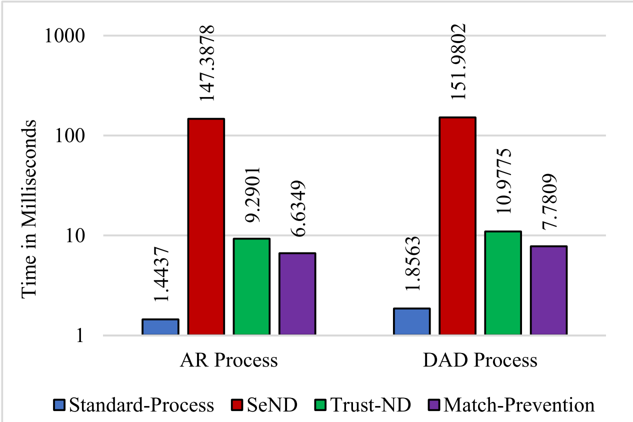

FIGURE 10. Total processing time of AR and DAD.

processing time on average for AR and DAD, respectively. Furthermore, the Standard-Process for both AR and DAD consumes only 1.4437 and 1.8563 milliseconds, respectively, given its lack of any security verification and no additional options to standard messages (i.e. NS and NA messages). In sum, Match-Prevention reduces the complexity of AR and DAD by reducing the time for generating and verifying the NS and NA messages at the sender and receiver sides compared with the other techniques.

2) BANDWIDTH CONSUMPTION

The exchange of NDP messages during AR and DAD processes in the IPv6 link-local network occurs frequently, which inevitably consumes the available bandwidth. For example, each host who wants to join an IPv6 link-local network must perform the DAD by sending NS and NA messages. Furthermore, each host will perform AR to map the IP with the MAC address by also using the NS and NA messages. Therefore, when introducing a security technique in NDP processes such as AR and DAD, the bandwidth consumption must be considered. To evaluate Match-Prevention in terms of its bandwidth consumption, the message size must be calculated. Accordingly, this section calculates the message size for Match-Prevention and the other compared techniques (i.e. SeND and Trust-ND). Afterwards, the bandwidth consumptions of these mechanisms are analysed by performing a normal scenario.

The size of NDP messages for each technique differs depending on their usage of the NDP option. Standard NDP is used as the baseline in this experiment because it carries either one or several existing options as listed in RFC 4861. Table 6 shows the message size of each technique.

Match-Prevention and Trust-ND each have one option, namely *match*-option and *trust*-option, respectively, whilst SeND carries four options, including CGA, Timestamp, Nonce and RSA signature. Wireshark is used to capture the NS and NA messages. The message-capturing results reveal that the NS-SeND/NA-SeND message has four options in addition to ICMPv6, thereby explaining its large size

TABLE 6. Message size.

Technique Name	Message Size (bytes)	
	NS Message	NA Message
Standard-Process	86	86
SeND	454	454
Trust-ND	118	118
Match-Prevention	102	102

of 454 bytes. Meanwhile, Match-Prevention and Trust-ND each have one option with different sizes depending on their design.

A reduction in message size can greatly influence the operation of the aforementioned techniques for NS and NA messages during the AR and DAD processes in an IPv6 link-local network. Firstly, given that message processing is performed based on byte, a larger message requires a longer generation time. In this case, Match-Prevention outpaces the other techniques in generating NS and NA messages by reducing the sizes of these messages and, consequently, reducing the time for processing these messages on both sender and receiver hosts. Secondly, the traffic overhead is affected by the message size and the number of IPv6 hosts located in the network, where NS and NA messages are exchanged during the DAD. Figure 11 shows the traffic overhead (in kilobyte), which is directly proportional to the number of IPv6 hosts.

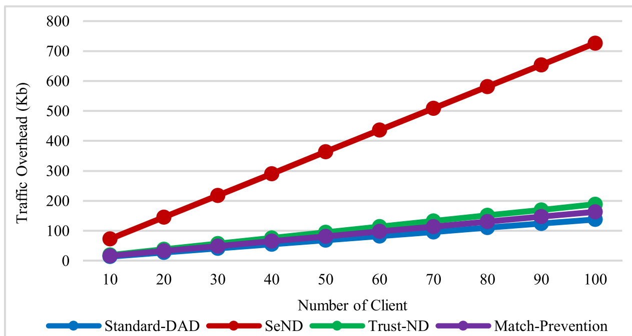

FIGURE 11. Traffic overhead.

Furthermore, the message size overhead is calculated by using Equation 6 as shown below:

$$MZoverhead = (MZ\ of\ Xtechnique) - (MZ\ of\ Standard - Process) \quad (6)$$

where *MZoverhead* is the message size overhead, *MZ of Xtechnique* is the summation messages (NS and NA) of the compared mechanisms (i.e. SeND, Trust-ND and Match-Prevention) and *MZ of Standard − Process* is the summation messages (NS and NA) of the Standard-Process. Table 7 presents the message size overhead from the baseline.

As shown in Figure 11, when 100 IPv6 hosts are exchanging NS and NA messages during the AR and DAD processes in the network, Match-Prevention reduces the traffic overhead by 563 and 25 kb compared with SeND and

TABLE 7. Message size.

Technique Name	Summation Message (NS+NA) Size (byte) during AR/DAD	Traffic Overhead
Standard-Process	172	0 (baseline)
SeND	908	736
Trust-ND	236	64
Match-Prevention	204	32

TABLE 8. Bandwidth consumption.

Message Type	Number of Messages
Total Messages	28,574
IPv6 Messages	1,223
ICMPv6 Messages	1,077
NS Messages	382
NA Messages	233

Trust-ND, respectively. Thirdly, transmitting a small message size consumes only a small amount of bandwidth, thereby conserving the bandwidth available in an IPv6 link-local network, as explained in the following.

Regarding bandwidth consumption, the IPv6 packet traffic was captured by using Wireshark at the NAv6 laboratory (National Advanced IPv6 Centre, 2009) for 17.3 minutes on April 24, 2019 in a normal scenario where no attack takes place. Amongst the 28,574 messages captured, 1,223 IPv6 packets were detected. Amongst these packets, 1,077 were ICMPv6 messages. The following filtrations were used to know the NS and NA messages:

➢ *icmpv6.nd.ns.target_address* (For NS Messages)
➢ *icmpv6.nd.na.target_address* (For NA Messages)

After these filtrations were used, 382 NS messages and 233 NA messages remained. Table 8 below shows the message type and the number of messages for each type.

Bandwidth consumption is calculated using Equation (7) below [25]:

$$BC = \frac{\sum Mz}{\sum T} \tag{7}$$

where BC is the bandwidth consumption; Mz is the sum of the NDP message size (i.e. NS and NA messages), including the IPv6 and ICMPv6 headers; and T is the total duration of the experiment.

According to Equation (7), two parameters should be considered, namely message size and transmission time. Message size denotes the total number of bytes of all NDP messages by using all options of the defined ICMPv6, whilst total transmission time should be the same for all NDP messages because these messages were generated in the same local network by using the same medium. Each NDP message has a different length among the mechanisms, which are summarised in Table 9.

Table 9 uses Standard-Process as the measurement baseline. Wireshark is employed for the measurement owing to its capability to show the number of bytes in the network media with the link layer and IPv6 headers as well as the number of NDP messages (NS and NA messages). With similar transmission times of 17.3 minutes, each message bandwidth consumption relies on the message size. The bandwidth consumption of SeND exceeds that of Standard-DAD by approximately 427% because of its message size. By contrast, the bandwidth consumptions of Trust-ND and Match-Prevention exceed that of Standard-Process by only 37% and 18%, respectively, because of their message sizes.

TABLE 9. Bandwidth consumption.

Technique Name	NDP Message Type (Size)		Bandwidth Consumption	
	NS	NA	NS	NA
Standard-Process (AR/DAD)	86	86	1.8989	1.1582
SeND	454	454	10.0247	6.1145
Trust-ND	118	118	2.6055	1.5892
Match-Prevention	102	102	2.2522	1.373

B. ATTACK SCENARIO

As indicated in the literature review, several techniques have failed to secure the NDP processes (i.e. AR and DAD) against DoS attacks. To examine the effectiveness of the proposed Match-Prevention technique in preventing DoS attacks, these attacks were carried out using THC tools and Scapy whilst the host was performing AR and DAD in an IPv6 link-local network. The experiments were repeated 20 times, and the below Equation (8) was used to calculate the success rate for AR and DAD:

$$ProcessSR = 1 - \frac{F}{N} \tag{8}$$

where *ProcessSR* is the process success rate, N is the number of process times and F is the number of failed process times. According to the definition of *ProcessSR*, if *ProcessSR* is equal to 1, then the attack is completely prevented. However, if *ProcessSR* is equal to 0, then the technique cannot prevent attacks. Therefore, *ProcessSR* can be used to measure the ability of each technique.

1) LAUNCHING DOS-ON-DAD ATTACK

A DoS-on-DAD attack targets a new host that joins an IPv6 link-local network and aims to disturb this network in such a way that the hosts are prevented from performing the DAD and obtaining a unique IP address. An attacker can launch a DoS-on-DAD attack by responding to each NS message sent by the new host to perform the DAD process and verify the uniqueness of a tentative IP address. This attack is carried using the THC tool and Scapy on Standard-DAD and Match-Prevention as well as on existing techniques (i.e. SeND and Trust-ND). Table 10 shows the results of the DoS-on-DAD attack experiment.

TABLE 10. DoS-on-DAD attack experiment.

Technique Name	Number of Experimental Runs	Success DAD	Failure DAD	ProcessSR
Standard-DAD	20	0	20	0
SeND	20	20	0	1
Trust-ND	20	0	20	0
Match-Prevention	20	20	0	1

2) LAUNCHING DOS-ON-AR ATTACK

A DoS-on-AR attack targets a host that is performing the AR to get another host's MAC address for future communication. This attack aims to disturb the network in such a way that the hosts get another MAC address which may lead to a Man-in-the-Middle attack. An attacker can launch a DoS-on-AR attack by responding to each NS message sent by the host for performing the AR. This attack was carried using the THC tool and Scapy on Standard-AR and the proposed Match-Prevention technique as well as on the existing techniques SeND and Trust-ND. Table 11 shows the results of the DoS-on-AR attack experiment.

TABLE 11. DoS-on-AR attack experiment.

Technique Name	Number of Experimental Runs	Success DAD	Failure DAD	ProcessSR
Standard-AR	20	0	20	0
SeND	20	20	0	1
Trust-ND	20	0	20	0
Match-Prevention	20	20	0	1

The experimental results reveal that SeND and Match-Prevention can verify the fake NA messages and prevent DoS-on-AR and DoS-on-DAD attacks throughout the AR and DAD processes in an IPv6 link-local network, respectively. By contrast, Trust-ND and Standard-Process (Standard-AR and Standard-DAD) are ineffective in securing AR and DAD. Moreover, even though SeND can stop DoS-on-AR and DoS-on-DAD attacks, this technique is vulnerable to flooding attacks, as explained in the following section.

3) FLOODING ATTACKS AGAINST SEND AND MATCH-PREVENTION TECHNIQUES

A flooding attack is amongst the most dangerous attacks that target IPv6 link-local networks. This type of attack freezes and/or crashes the victim machine and effectively ceases its operations. To investigate the sustainability of Match-Prevention compared with SeND during a flooding attack, these mechanisms are implemented on a machine that is targeted by thousands of NS and NA messages.

A flooding attack cannot be easily prevented. Therefore, the measurements are performed at a definite time needed by each technique to process attacking messages. SeND is a complex mechanism that requires extensive competitions, making it vulnerable to flooding attacks. In the experiment, a flooding attack is carried out by multicasting (NS-SeND and NA-SeND) messages as well as (NS-match and NA-match) messages to a machine that runs SeND and Match-Prevention, respectively. Figures 12 and 13 respectively present the experimental results of flooding NS and NA messages on SeND and Match-Prevention.

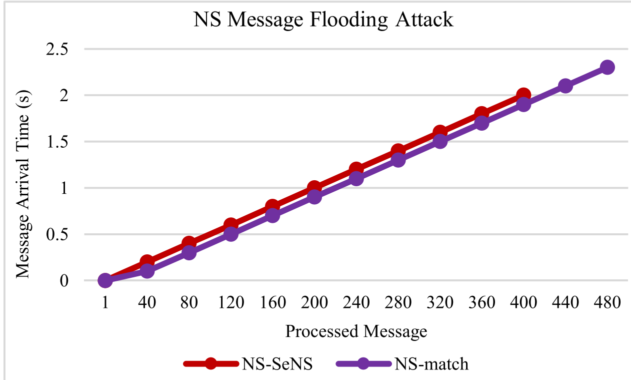

FIGURE 12. NS message flooding attack against the receiver host.

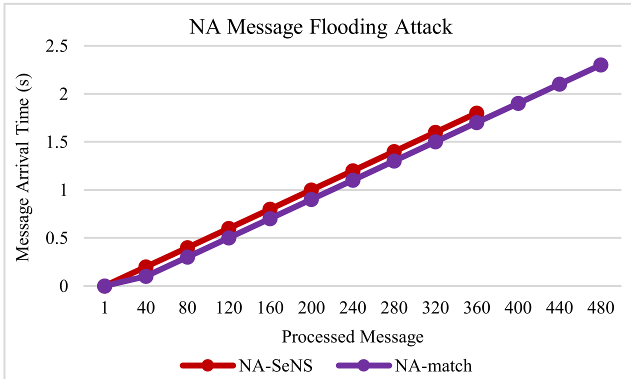

FIGURE 13. NA message flooding attack against the receiver host.

The machine that implements SeND crashed after processing many SeND messages. This machine can only process an average of 398 NS-SeND and 356 NA-SeND messages at 2 seconds and 1.8 seconds, respectively, whilst the machine that implements Match-Prevention can process all NS-match and NA-match messages without crashing or freezing. These results are in line with the previous discussions on the complexity of SeND. In sum, an IPv6 host that implements SeND is vulnerable to flooding attacks. Therefore, Match-Prevention shows better sustainability against flooding attacks compared with SeND.

XI. DISCUSSION

Match-Prevention technique is proposed to secure AR and DAD processes and is evaluated using the metrics of processing time, bandwidth consumption and DoS prevention success rate. Experiments are performed on the proposed Match-Prevention technique, Standard-Process (AR/DAD) and other existing techniques, including SeND and Trust-ND, to prove whether the proposed technique fulfils the three requirements mentioned in Section V, including less complexity, high security and self-decision.

A. PROCESSING TIME

The experimental results reveal that Match-Prevention has less complexity in terms of processing time compared with SeND and Trust-ND. Therefore, Match-Prevention fulfils the less complexity requirement.

The comparison results presented in Figure 10 reveal that SeND consumes more processing time for AR and DAD at 147.3878 and 151.9802 milliseconds on average, respectively, because this technique appends four options to the NS and NA messages, thereby extending the time required for processing to generate and verify NS and NA messages. Furthermore, Trust-ND has shorter processing times of 9.291 and 10.9775 milliseconds on average on AR and DAD, respectively, by reducing the number of options appended to the NS and NA messages from four to one. However, hashing all the NS and NA messages during AR and DAD consumes a considerable amount of processing time.

By contrast, Match-Prevention demonstrates lower complexity in terms of processing time by appending only one option to each NS and NA message and applying a hash function only on part of a target IP address instead of the entire message. Additionally, the hash function used does not require a key for encryption or decryption, and Match-Prevention directly discards those messages that lack the *match*-option, thereby shortening the message verification process without the need for further processing. Altogether, these ways significantly reduce the processing time for generating and verifying NS and NA messages during AR and DAD in an IPv6 link-local network. The experiment results reveal that Match-Prevention only spends 6.6349 and 7.7809 milliseconds of processing time on average for AR and DAD, respectively, in an IPv6 link-local network. In sum, the proposed Match-Prevention technique demonstrates the best performance in terms of processing time amongst existing techniques.

B. BANDWIDTH CONSUMPTION

A large message size can significantly influence the bandwidth availability in a network. Therefore, using a small message size can improve the performance of this network. The existing techniques have large message sizes due to their designs. For example, SeND uses four options, namely CGA, Timestamp, Nonce and RSA Signature, all of which increase the size of NS and NA messages to 454 bytes. Meanwhile, Trust-ND uses SHA-1 as a hash-function algorithm for hashing NS and NA messages, leading to a 160-bit SHA-1 that reduces the size of NS and NA messages to 118 bytes. Given the large size of their messages, SeND and Trust-ND consume large amounts of bandwidth. Specifically, compared with Standard-DAD, SeND and Trust-ND increase the bandwidth consumption to approximately 427% and 37%, respectively.

Meanwhile, the bandwidth consumption of Match-Prevention is only 18% higher than that of Standard-DAD for two reasons. Firstly, unlike SeND, Match-Prevention uses only one option instead of four. Secondly, unlike Trust-ND, Match-Prevention uses SHAKE128, which only produces 128 bits of hash values, thereby reducing the NS and NA message size to 102 bytes and, consequently, the bandwidth consumption.

C. DOS PREVENTION SUCCESS RATE

In terms of security, Standard-Process (AR and DAD) and Trust-ND failed to prevent a DoS attack against the AR and DAD processes in an IPv6 link-local network. The Standard-Process for both AR and DAD is to send the target IP address in plaintext. Thus, any attacker located on the link can utilise this address and send a fake message (i.e. NA message); besides, the standard messages (i.e. NS and NA messages) are not secure because of their design, which can be modified by the attacker and disturbs the AR and DAD. Furthermore, Trust-ND is hashing the entire message i.e. NS message without hiding the information i.e. target IP address. Thus, any attacker placed on the same link receives the NS message will be able to craft the NS message and generate a fake NA message with the trust-option and send it as a reply to NS message. By doing this, the target host will receive the NA message and processing it as a legitimate message but in fact, it is an illegitimate message, and this can disturb the whole process i.e. AR and DAD. In this scenario, Trust-ND fails to secure AR and DAD in an IPv6 link-local network. Trust-ND is also vulnerable to hash collision attacks, which can induce DoS attacks against the AR/DAD processes.

In sum, both Standard-Process of AR and DAD and Trust-ND fail to prevent DoS attacks against AR and DAD in an IPv6 link-local network, whilst SeND, as recommended by RFC, can prevent DoS attacks on AR and DAD. However, the high complexity of this technique leads to a long processing time for verifying NS and NA messages, thereby affecting the security provided by SeND. Specifically, an attacker on the same link can bombard a victim with SeND-NS and SeND-NA messages, keeping this victim preoccupied with processing SeND messages and increasing its vulnerability to DoS attacks, such as flooding attacks that can freeze and prevent the victim machine from processing further messages.

In conclusion, the proposed Match-Prevention technique can be applied to secure AR and DAD in an IPv6 link-local network without being vulnerable to brute-force or collision attacks. When this technique is implemented on IPv6 hosts, these hosts can distinguish whether the incoming NS and NA messages are sent by legitimate or illegitimate hosts. Therefore, IPv6 hosts can secure their IPv6 link-local addresses from DoS attacks and communicate with other IPv6 hosts. Match-Prevention then achieves the other two requirements of high security and self-decision. This technique also demonstrates better performance compared with

SeND and Trust-ND in terms of processing time and bandwidth consumption.

XII. CONCLUSION

AR and DAD are considered the most important processes of NDP in an IPv6 link-local network. Both processes occur frequently from each IPv6 hosts. Two messages are used during the AR and DAD processes, namely NS and NA messages. Although AR and DAD are crucial in an IPv6 network, they are vulnerable to DoS attacks. The Standard-Process of AR and DAD does not have any verification mechanism to validate incoming messages. Moreover, with all the information sent in plaintext (including the target IP address), any attacker can manipulate these messages (i.e. NA) and disturb the whole process.

Match-Prevention is proposed to prevent DoS attacks during AR and DAD in a IPv6 link-local network by hiding the target IP address using the SHA-3 hash-function algorithm. Security and comparative analyses are carried out on existing techniques, such as SeND and Trust-ND. The implementation is carried out in two scenarios to measure the performance of Match-Prevention and compare its results with those of the existing techniques. The obtained results revealed that Match-Prevention consumes less processing time and bandwidth consumption and is more efficient compared with SeND and Trust-ND. Match-Prevention technique resists various types of attacks, such as collision and brute-force attacks. Consequently, Match-Prevention can efficiently prevent DoS attacks during AR and DAD. In the context of this study, Match-Prevention is implemented on a small-area IPv6 network. For further studies, the authors recommend that Match-Prevention should also be considered for securing other NDP processes.

REFERENCES

[1] M. Sánchez-Valle, M. V. Abad, and C. Llorente-Barroso, ''Empowering the elderly and promoting active ageing through the Internet: The benefit of e-inclusion programmes,'' in *Safe at Home with Assistive Technology*. Cham, Switzerland: Springer, 2017, pp. 95–108.

[2] *Internet Control Message Protocol*, InterNet Netw. Work. Gr., document RFC 792, 1981.

[3] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, document RFC 2469, Dec. 1998.

[4] *IPv6—Google Statistic*. Accessed: Jan. 2, 2020. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[5] T. Narten, E. Nordmark, and W. Simpson, *Neighbor Discovery for IP Version 6 (IPv6)*, document RFC 4861, 2007.

[6] E. Mahmood, A. H. Adhab, and A. K. A. Ani, ''Review paper on neighbour discovery protocol in IPv6 link-local network,'' *Int. J. Serv. Oper. Informat.*, vol. 10, no. 1, p. 65, 2019.

[7] O. E. Elejla, M. Anbar, and B. Belaton, ''ICMPv6-based DoS and DDoS attacks and defense mechanisms: Review,'' *IETE Tech. Rev.*, vol. 34, no. 4, pp. 390–407, Jul. 2017.

[8] M. Anbar, R. Abdullah, R. M. A. Saad, and I. H. Hasbullah, ''Review of preventive security mechanisms for neighbour discovery protocol,'' *Adv. Sci. Lett.*, vol. 23, no. 11, pp. 11306–11310, Nov. 2017.

[9] A. Al-Ani, M. Anbar, I. H. Hasbullah, R. Abdullah, and A. K. Al-Ani, ''Authentication and privacy approach for DHCPv6,'' *IEEE Access*, vol. 7, pp. 73144–73156, 2019.

[10] A. S. A. Mohamed Sid Ahmed, R. Hassan, and N. E. Othman, ''IPv6 neighbor discovery protocol specifications, threats and countermeasures: A survey,'' *IEEE Access*, vol. 5, pp. 18187–18210, 2017.

[11] G. Song and Z. Ji, ''Anonymous-address-resolution model,'' *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 10, pp. 1044–1055, 2016.

[12] A. K. Al-Ani, M. Anbar, S. Manickam, A. Al-Ani, and Y. B. Leau, ''Preventing denial of service attacks on address resolution in IPv6 link-local network: AR-match security technique,'' in *Computational Science and Technology*. Singapore: Springer, 2019, pp. 305–314.

[13] A. AlSa'deh, H. Rafiee, and C. Meinel, ''SEcure neighbor discovery: A cryptographic solution for securing IPv6 local link operations,'' in *Theory and Practice of Cryptography Solutions for Secure Information Systems*. Philadelphia, PA, USA: IGI Global, 2013, pp. 178–198.

[14] A. K. Al-Ani, M. Anbar, S. Manickam, and A. Al-Ani, ''DAD-match; Security technique to prevent denial of service attack on duplicate address detection process in IPv6 link-local network,'' *PLoS ONE*, vol. 14, no. 4, Apr. 2019, Art. no. e0214518.

[15] A. K. Al-Ani, M. Anbar, S. Manickam, C. Y. Wey, Y.-B. Leau, and A. Al-Ani, ''Detection and defense mechanisms on duplicate address detection process in IPv6 link-local network: A survey on limitations and requirements,'' *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3745–3763, Apr. 2019.

[16] F. Beck, T. Cholez, O. Festor, and I. Chrisment, ''Monitoring the neighbor discovery protocol,'' in *Proc. Int. Multi-Conf. Comput. Global Inf. Technol. (ICCGI)*, Mar. 2007, p. 57.

[17] F. A. Barbhuiya, G. Bansal, N. Kumar, S. Biswas, and S. Nandi, ''Detection of neighbor discovery protocol based attacks in IPv6 network,'' *Netw.Sci.*, vol. 2, nos. 3–4, pp. 91–113, Jul. 2013.

[18] N. Kumar, G. Bansal, S. Biswas, and S. Nandi, ''Host based IDS for NDP related attacks: NS and NA Spoofing,'' in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2013, pp. 1–6.

[19] A. Herrera, ''How secure is the next-generation internet? An examination of IPv6,'' Defence Sci. Technol. Organisation Edinburgh (Australia) Cyber Electron. Warfare Division, Tech. Rep. DSTO-GD-0767, 2013.

[20] G. Song and Z. Ji, ''Novel duplicate address detection with hash function,'' *PLoS ONE*, vol. 11, no. 3, Mar. 2016, Art. no. e0151612.

[21] F. Najjar, M. Kadhum, and H. El-Taj, ''Neighbor discovery protocol anomaly detection using finite state machine and strict anomaly detection,'' in *Proc. 4th Int. Conf. Internet Appl., Protocols Services*, 2015, pp. 967–978.

[22] J. Arkko, J. Kempf, B. Zill, and P. Nikander, *Secure Neighbor Discovery (SEND)*, document RFC 3971, Mar. 2005, pp. 2003–2005.

[23] A. Kukec, M. Bagnulo, and M. Mikuc, ''SEND-based source address validation for IPv6,'' in *Proc. 10th Int. Conf. Telecommun.*, Jun. 2009, pp. 199–204.

[24] A. AlSa'deh, H. Rafiee, and C. Meinel, ''IPv6 stateless address autoconfiguration: Balancing between security, privacy and usability,'' in *Proc. Int. Symp. Found. Pract. Secur.*, 2012, pp. 149–161.

[25] S. Praptodiyono, I. H. Hasbullah, and M. M. Kadhum, ''Securing duplicate address detection on IPv6 using distributed trust mechanism,'' *Int. J. Simul.–Syst., Sci. Technol.*, vol. 17, no. 26, pp. 1–9, 2016.

[26] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varici, and I. Verbauwhede, ''SPONGENT: A Lightweight hash function,'' in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2011, pp. 312–325.

[27] E. Andreeva, B. Mennink, and B. Preneel, ''Open problems in hash function security,'' *Des., Codes Cryptogr.*, vol. 77, nos. 2–3, pp. 611–631, Dec. 2015.

[28] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, ''Quark: A Lightweight Hash,'' in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2010, pp. 1–15.

[29] T. Narten, R. Draves, and S. Krishnan, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, document RFC 3041, Jan. 2001.

[30] J. Arkko, J. Kempf, B. Zill, and P. Nikander, *Secure Neighbor Discovery (SEND)*, document RFC 3971, Mar. 2005, pp. 2003–2005.

[31] H. Rafiee and C. Meinel, ''SSAS: A simple secure addressing scheme for IPv6 autoconfiguration,'' in *Proc. 11th Annu. Conf. Privacy, Secur. Trust*, Jul. 2013, pp. 275–282.

[32] J. S. Coron, Y. Dodis, C. Malinaud, and P. Puniya, ''Merkle-Damgård revisited: How to construct a hash function,'' in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, Aug. 2005, pp. 430–448.

[33] V. Aurora. (2017). *Lifetimes of Cryptographic Hash Functions*. Accessed: Sep. 12, 2017. [Online]. Available: http://valerieaurora.org/hash.html

[34] K. Dunham, ''A fuzzy future in malware research,'' *ISSA J.*, vol. 11, no. 8, pp. 17–18, 2013.

[35] W. Simpson, D. T. Narten, E. Nordmark, and H. Soliman, *Neighbor Discovery for IP Version 6 (IPv6)*, document Internet Engineering Task Force, RFC2461, Dec. 1998.

[36] B. Fenner, *Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers*, document RFC 4727, Nov. 2006.
[37] (2009). *National Advanced IPv6 Centre*. Accessed: May 20, 2019. [Online]. Available: https://www.nav6.usm.my/
[38] T. H. C. Van Hauser, *Attacking the IPv6 Protocol Suite*, 2006.
[39] A. El Ksimi and C. Leghris, "Towards a new algorithm to optimize ipv6 neighbor discovery security for small objects networks," *Secur. Commun. Netw.*, vol. 2018, pp. 1–11, Jun. 2018.

AHMED K. AL-ANI received the B.S. degree in computer technique engineering from the University of Al-Ma'mun, in 2013, and the M.Sc. degree in information technology from Universiti Utara Malaysia, in 2016. He is currently pursuing the Ph.D. degree with the National Advanced IPv6 Centre, Universiti Sains Malaysia. He is currently a Computer Engineer. His research interests include computer network security, the Internet security, network communication protocols (IPv4/IPv6), and IPv6 security.

MOHAMMED ANBAR received the bachelor's degree in computer system engineering from Al-Azhar University, Palestine, the M.Sc. degree in information technology from Universiti Utara Malaysia, Malaysia (UUM), and the Ph.D. degree in advanced Internet security and monitoring from University Sains Malaysia (USM). He currently serves as a Senior Lecturer with the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. His research interests are malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, the Internet of Things (IoT), and IPv6 security.

AYMAN AL-ANI received the B.S. degree in computer engineering from the University of Technology and the M.Sc. degree in information technology from Universiti Utara Malaysia, in 2016. He is currently a Ph.D. Fellow with the National Advanced IPv6 Centre, Universiti Sains Malaysia. His research interests include computer networks, network security, and software-defined networks.

DYALA R. IBRAHIM received the B.S. degree in computer technique engineering from Al-Tafila University, in 2012, and the M.Sc. degree in security from Al-Zaytoonah University, in 2016. She is currently pursuing the Ph.D. degree with the National Advanced IPv6 Centre, Universiti Sains Malaysia. She is currently a Computer Engineer. Her research interests include security, the Internet, image processing, and artificial intelligence.