IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2022, accepted 25 July 2022, date of publication 3 August 2022, date of current version 12 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3196028

RESEARCH ARTICLE

NDPsec: Neighbor Discovery Protocol Security Mechanism

AYMAN AL-ANI[1], **AHMED K. AL-ANI**[2], **SHAMS A. LAGHARI**[3], **SELVAKUMAR MANICKAM**[3], **KHIN WEE LAI**[4], **(Senior Member, IEEE), AND KHAIRUNNISA HASIKIN**[4]

[1]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Sabah 88400, Malaysia
[2]School of Computing and Data Science, Xiamen University Malaysia, Sepang, Selangor 43900, Malaysia
[3]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM), Gelugor, Penang 11800, Malaysia
[4]Department of Biomedical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia

Corresponding authors: Ahmed K. Al-Ani (ahmedkhallel.alani@xmu.edu.my) and Khairunnisa Hasikin (khairunnisa@um.edu.my)

This work was supported by the School of Computing and Data Science, Xiamen University Malaysia, under Grant XMUMRF/2022-C9/IECE/0028.

ABSTRACT Internet Protocol version 6 (IPv6) is envisioned as the cornerstone for future internet connectivity and information technology (IT) expansion. Due to its enormous address pool, extendable headers, high level of security, and mobility, IPv6 is positioned as the next-generation Internet Protocol. NDP is an integral component of IPv6 since it resolves addresses, locates routers, and finds duplicated addresses in a local-link network. Because NDP is based on the premise that all nodes in the network are trustworthy, it is subject to a variety of attacks, including Denial of Service (DoS) on Duplicate Address Detection (DAD) attacks (aka. DoS-on-DAD), Address Resolution-based attacks, Router Advertisement (RA) based attacks, and Redirect attacks. This paper proposes an NDP security (NDPsec) mechanism based on the Ed25519 digital signature to authenticate IPv6 hosts to prevent unauthorized devices from joining the network. The proposed NDPsec mechanism is evaluated and compared to Secure NDP (SeND), Match-Prevention, and Trust-ND mechanisms. The performance is measured in terms of processing time, traffic overhead, and resilience against network-based attacks. The results obtained from the experiments showed that NDPsec successfully prevented cyberattacks, with approximately 144% less processing time and over 50% less traffic overhead compared to SeND (the default security mechanism for NDP protocol). The proposed NDPsec mechanism has remarkable superiority in terms of resilience against attacks compared to Match-Prevention and Trust-ND mechanisms.

INDEX TERMS IPv6, NDP, denial of service, RA flooding, security, authentication, MITM.

I. INTRODUCTION

The rise of the digital-world enabled by cutting-edge technologies such as 5G networks, the Internet of Things (IoT), and Cloud Computing has resulted in a massive increase in the number of devices connected to the Internet [1]. The projected expansion of IoT devices is increasing, and Cisco research indicates that there will be 27.1 billion networked devices in 2021, equating to 3.5 networked devices per person worldwide [2]. The Internet Protocol version 4 (IPv4) barely provides enough address space to connect the Internet's approximately 4.3 Billion devices, which has long been recognized as being depleted and in need of replacement with a protocol that provides a larger pool of address space to meet the demands of today's digital world [3]. The Internet Protocol version 6 (IPv6) is the next generation of the Internet Protocol that will supersede the IPv4 protocol [4]. The percentage of devices accessing Google via IPv6 has exceeded 33 percent, according to Google data [5]. Compared to IPv4, IPv6 offers a modest improvement in network security as well as in service quality. IPv6 does, however, continue to face a number of security problems, including Denial of Service (DoS) and Man-in-the-middle (MITM) attacks [6].

In order to mitigate security problems in a link-local network, IPv6 introduces a new protocol, known as the Neighbor Discovery Protocol (NDP), which is defined in

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

RFC 4861 [7]. NDP is a critical protocol in the IPv6 network, and it performs a variety of tasks, including Address Resolution (AR), Neighbor Unreachability Detection (NUD), router discovery, and Duplicate Address Detection (DAD). IPv6 was designed on the premise that devices connected to a Local Area Network (LAN) are trustworthy and reliable. As a consequence, the NDP treats every device connected to the LAN as trustworthy and lacks security measures for situations in which a malicious host enters the network and initiates network attacks. This situation renders the network vulnerable to a variety of attacks, including DoS and MITM, which is the most severe attack on an IPv6 link-local network. Given the importance of NDP's processes and its susceptibility to attacks, a plethora of techniques have been proposed to protect these processes from being compromised. In this article, the most commonly used techniques are presented and assessed in terms of processing time, bandwidth usage, and effectiveness in preventing attacks on NDP [8]. Thus, this paper introduces a new mechanism called NDPsec that enables authentication for NDP communications in an IPv6 link-local network.

The paper is organized as follows: Section II offers context for the NDP process and elaborates on the threat model. Section III discusses the security threats posed to NDP, and Section IV presents IPv6 address privacy concerns, while Section V summarizes related studies. Section VI discusses the design of NDPsec. Section VII covers the implementation details of the NDPsec protocol. The experiments, as well as the evaluation of the proposed mechanism, are presented in Section VIII. Section IX discusses the findings. Section X discusses the conclusion and recommendations for future research.

II. BACKGROUND

NDP is used in combination with IPv6 and performs a variety of functions formerly performed by different IPv4 protocols. For instance, it supersedes IPv4-specific protocols such as router discovery, Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and ICMP redirection [9]. NDP enables nodes to discover and notify their neighbors on the same LAN of their presence. Additionally, NDP consolidates critical network management operations such as router discovery, AR, and DAD [10]. NDP is a collection of messages and procedures used to establish communication between nodes, routers, and hosts in a single IPv6 network. NDP makes use of the following five ICMPv6 messages:

- Router Solicitation (RS) – type 133: RS messages are used by the hosts to determine the presence of routers connected to the link. Routers immediately generate RA messages upon receiving packets not addressed to them; thus, avoid delaying advertisement for the next scheduled timer.
- Router Advertisement (RA) – type 134: Routers advertise is used to distribute link-specific parameters such as hop count limits, configuration flag, network prefixes, etc., and it is multicast periodically or in response to RS messages.
- Neighbor Solicitation (NS) – type 135: NS messages are sent when a host needs to know the medium access control (MAC) address of another host on the same network.
- Neighbor 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 (R) – type 137: R message is sent from routers to redirect user traffic from one path to another significant path.

While the NDP is considered to be the core protocol of IPv6, it lacks a suitable security mechanism for verifying and authenticating messages exchanged between hosts connected through the same connection. Additionally, the abovementioned messages are not protected by design; as a result, an attacker who enters the network may interfere with any NDP process (e.g., AR, DAD, Router Discovery, etc.) by altering the messages (i.e., five NDP messages) and launching DoS and MITM attacks.

III. NDP THREATS

Despite the fact that the NDP is often regarded as the most significant and vital protocol in IPv6, it lacks a suitable security mechanism for verifying and authenticating messages sent between hosts that are connected by the same network. An attacker with access to the same network may participate in any of the NDP processes and cause disruption by altering the messages exchanged between the hosts and may launch DoS and MITM attacks whenever desired [9]. Thus, cyberattacks on NDP functions and processes are possible, and attack actors may jeopardize network security. The detailed discussion on various potential attacks that can be carried out against NDP processes is listed in the following sub-sections. The following sub-sections provide a more in-depth insight into the different types of cyberattacks that may be launched against NDP processes.

A. DUPLICATE ADDRESS DETECTION THREAT

To participate in the IPv6 network, all the participating hosts must have a unique IP address. In contrast to IPv4, IPv6 does not need DHCP to get a unique IP address prior to joining the network. In an IPv6 network, hosts may self-assign an IP address without the need for a DHCP server. However, since it is unknown if the selected IP address is unique in the network, it is necessary to query the whole network about the chosen IP address's uniqueness. Thus, before initiating communications with the network, the host that wishes to join must start the DAD procedure.

Typically, in an IPv6 link-local network, two types of NDP messages are utilized during the DAD: NS and NA. Verifying the uniqueness of an IPv6 local-link network address is necessary when the host joins an IPv6 local-link network, the host creates a tentative IP address. As a result, the host

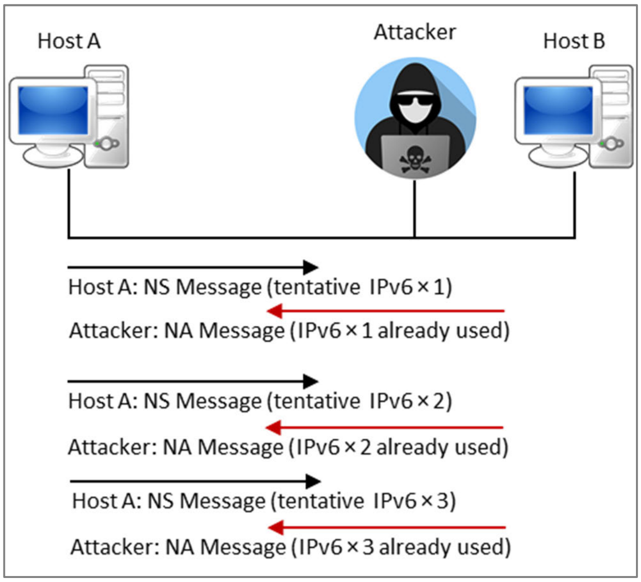
Host A
Attacker
Host B
Host A: NS Message (tentative IPv6 × 1)
Attacker: NA Message (IPv6 × 1 already used)
Host A: NS Message (tentative IPv6 × 2)
Attacker: NA Message (IPv6 × 2 already used)
Host A: NS Message (tentative IPv6 × 3)
Attacker: NA Message (IPv6 × 3 already used)

FIGURE 1. DoS-on-DAD process.

multicasts NS messages to all the hosts connected to the same link-local network to check the tentative IPv6 address is not used by another host in the network. In the event that the tentative IP address has already been allocated to another host on the same network, the current host that uses the tentative IP, should send a NA message in response to the NS message, indicating that the produced tentative IP address is not unique and that the tentative IP address should be changed. Later, the target host must create a new tentative IP address and repeat the DAD process in order to ensure that the newly produced address is unique until no more NA messages are received [11].

In an IPv6 link-local network, the DAD procedure presupposes that all neighboring hosts are reliable. When a host receives a NA message from other neighboring hosts as part of the address verification procedure, it replies to the message in accordance with the requirements, unaware that the message was sent by a legitimate host or an attack actor. In this case, an attacker might reply to an NS message by returning a bogus NA response stating that the produced tentative IP address has already been taken; thus, it is not unique and cannot be used by the requesting host. Although the IP address is unique, the reply received from the malicious host would prohibit requesting IPv6 host from self-assigning this unique IP address. As a result, the host is unable to join the network and communicate with other hosts. As exhibited in previous studies, this kind of attack is referred to as Denial of Service (DoS) on Duplicate Address Detection (DAD) attacks (DoS-on-DAD) since it prevents hosts from self-assigning IP addresses in an IPv6 link-local network [12]. Figure 1 depicts a scenario wherein Host A has self-assigned a tentative IP address; however, the attacker has responded that the requested IP is already taken and cannot be used to join the network.

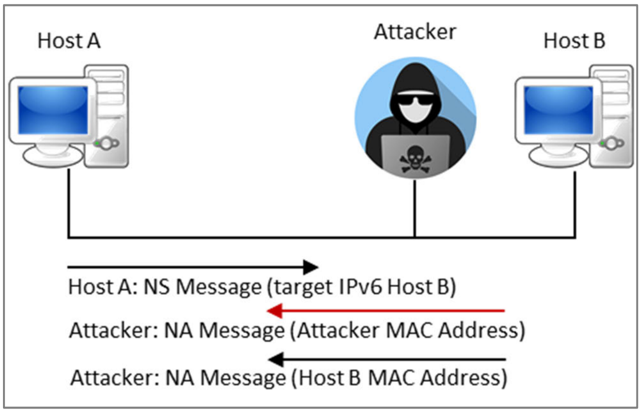
Host A
Attacker
Host B
Host A: NS Message (target IPv6 Host B)
Attacker: NA Message (Attacker MAC Address)
Attacker: NA Message (Host B MAC Address)

FIGURE 2. Attack on neighbor discovery.

B. ADDRESS RESOLUTION THREAT

In an IPv6 network, the host does not obtain the MAC address of another host using the ARP protocol but rather via NDP. When an IPv6 host requires to perform the AR process, it multicasts NS message in the link-local network. The NS message payload contains the target IPv6 address. The host that owns the target IPv6 address replies with a NA message containing its MAC address in the payload. By doing so, the host can get the MAC address of the IPv6 address [13].

From the preceding description, it appears that NDP authenticates neither the requestor (sends the NS) nor the responder (sends the NA). Thus, NDP for IPv6 is performing similarly to how ARP does for IPv4. The attacker can reply to an NS instead of the real host, as shown in Figure 2. So, the victim will send its packets to the attacker instead of the real host. The attack can be even worse when the spoofed node is the default router, which allows a MITM attack for sniffing, altering, and dropping all packets leaving the subnet.

C. RA MESSAGE THREAT

In the IPv6 network, a router located on the link sends RA messages frequently, and based on the RA messages, the host will configure itself with network configuration parameters such as Local prefixes, Router link-layer address, Maximum transmission unit (MTU), etc. The NDP does not have a mechanism to verify the source of the RA message; therefore, the attacker can spoof the RA message and configure the host with the attacker parameter, which can cause DoS, MITM attacks. Further, the attacker can send thousands of RA messages to all hosts in the IPv6 network, making hosts configure themself with RA messages again and again to exhaust the hosts' resources which eventually leads to a DoS attack on the entire IPv6 link-local network. This type of attack is called an RA flooding attack. Figure 3 illustrates how the scenario wherein the attacker replies with the RA message [13].

D. REDIRECT THREAT

Redirection is a straightforward process that is based on the NDP, enabling a router to suggest a better route to a

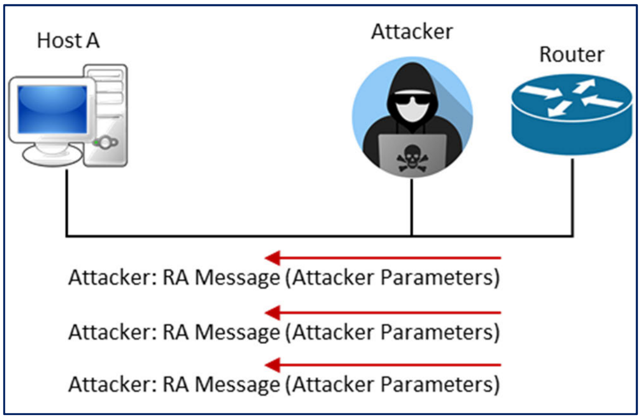
Host A
Attacker
Router
Attacker: RA Message (Attacker Parameters)
Attacker: RA Message (Attacker Parameters)
Attacker: RA Message (Attacker Parameters)

FIGURE 3. RA flooding attack.

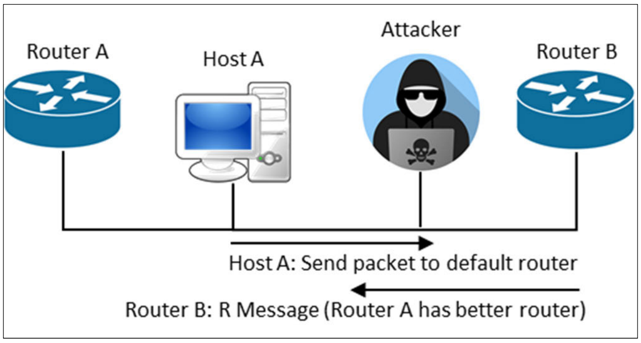
Router A
Host A
Attacker
Router B
Host A: Send packet to default router
Router B: R Message (Router A has better router)

FIGURE 4. NDP redirect message.

destination host, if available, in an IPv6 link-local network. The R message is required for the Redirection procedure to function properly. Figure 4 shows a situation in which Host-A has a default route to Host-B via Router-A; however, there is a better route through Router-B that Host-A is unaware of, as shown in the example below. To communicate with another network host, a host send a packet to the default Router's MAC address, which in this case is Router-B. When Router-B receives packets from the host, it checks its own forwarding table and determines that there is a better route available for the packet; as a result, Router-B will immediately send an NDP R message to the source of the packet (i.e., Host-A) informing it that there is a better route available through Router-A. The Host-A will react accordingly and will update its routing table with the information provided by Router-B, which will eventually allow Host-A to communicate with Host-B via a shorter path than that which was previously available [14].

Because the NDP redirect process does not have any authentication mechanism in place to verify the authenticity of the sender of the messages, it is obvious that an attacker can easily spoof the NDP's R messages in order to launch attacks such as DoS and MITM attacks in the link-local networks.

IV. IPv6 ADDRESS PRIVACY CONCERNS

In the IPv6 network, hosts use stateless address autoconfiguration (SLAAC) to generate an IP address and configure itself after checking the DAD process. The host generates an IPv6 address using a combination of link-local-prefix or network-prefix, which is locally available information advertised by the router, with addresses or interface identifiers (IIDs), as shown in Figure 5. IPv6 hosts use IEEE identifiers for generating IIDs, they allow correlation and location tracking for the lifetime of the device since IEEE identifiers last long, and their structure makes address scanning and device exploits possible. The IID remains the same regardless of the subnet it connects to in the SLAAC method. The default addressing scheme of IEEE identifiers referred to as the 64-bit extended unique identifier (EUI-64), uses the MAC address as the IID. This would result in a huge exposure to the attacker, who would know the list of subnets and the host MAC addresses. This would also mean that the addresses and hosts are vulnerable to tracking and easy targets for attacks from anywhere worldwide. Several studies address the issue of using the IEEE Identifiers mechanism and the effect of using the same IIDs with different subnet networks [15].

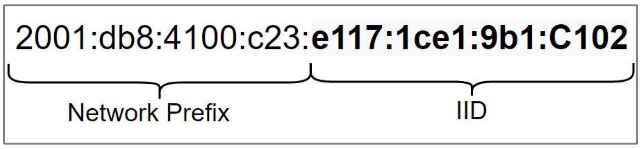

FIGURE 5. Formation of IPv6 address as network prefix and IID.

Therefore, Internet Engineering Task Force (IETF) proposed Privacy Extensions mechanism to generate IPv6. The working mechanism of Privacy Extensions is explained in RFC 4941 [16]. This mechanism attends to generate temporary addresses every time the IPv6 host connect to the network, thus the attack cannot track the host, and exploit the IPv6 to threat the host privacy. However, this mechanism was not designed to prevent spoofing IPv6 address.

V. RELATED WORK

Researchers have proposed several mechanisms to secure NDP in order to protect it from threats such as DAD threat, AR threat, RA message threat, and redirect threat. Some of the mechanisms proposed to address security concerns in NDP use a monitoring strategy in which network traffic is monitored, and administrators are immediately notified of any suspicious behavior detected. NDP monitoring (NDPmon) and Intelligent NDP monitoring (INDPmon) are two such methods (INDPmon) used to overcome security issues found in NDP. NDPmon, developed by Beck in 2007 [17], is a program that is functionally identical to Arpwatch for IPv4 but has additional attack detection capabilities. NDPmon is often deployed on a centralized server on a LAN, and network hosts utilize it to monitor NDP processes for suspicious activities. When NDPmon detects unusual behavior on the network, it notifies administrators through email or by writing logs to storage devices. NDPmon is a three-phase process that includes training, learning, and monitoring. NDPmon makes the assumption that all nodes and network actions are legit during the training phase, due to which the reporting function is disabled during this phase. According to studies [18], the drawbacks of this default behavior include the possibility of a compromised node causing a complete detection failure if it joins the network during the training phase. Additionally, NDPmon may generate false positive alerts when modifications are made to the Network Interface Card (NIC) of a legitimate computer on the network. Furthermore, detection methods may be circumvented when attacker nodes intercept the ICMPv6 packets carrying NDP data [19]. Although NDPmon prevents attacks launched by unauthorized users, it completely ignores the possibilities of attacks launched by legitimate users, which are very difficult to detect. Song and Ji [20] demonstrated that although NDPmon can detect threats and issue alerts for regular network activities, it fails to prevent these attacks.

INDPmon is yet another security mechanism proposed in 2015 [21] for monitoring network traffic in order to detect attacks launched on NDP processes. This mechanism models the major NDP processes using an extended finite state machine (EFSM) and identifies aberrant behavior by employing strict anomaly detection. Since the abnormality behaviors vary from conventional behavior, additional failure states are created to thwart unauthorized activities or transactions. Strict anomaly detection enables the defining of failure states in EFSM and the reporting of any protocol fundamental violation. These violations often occur as a consequence of protocol misconfiguration or an attack. INDPmon may also alert the network to potential NDP attacks, as most of these attacks violate protocol rules. INDPmon can only detect NDP attacks that break basic protocols, according to previous research [14]. Other violations, on the other hand, such as spoofed IP addresses or MAC addresses, are more challenging to detect. For example, attackers may flood a network with NDP messages using a spoofed IP address, preventing INDPmon from differentiating between legitimate and spoofed IP addresses. In summary, INDPmon cannot secure NDP messages.

Some researchers have attempted to secure the NDP process in an IPv6 link-local network by mapping the MAC address to the IP address. For example, Source Address Validation Architecture (SAVA) has been proposed by different researchers from Tsinghua University to address security issues in the NDP processes [22]. SAVA aims to authenticate the source IP address of a packet whilst considering the aspects of authorization, uniqueness, and traceability. In SAVA, validation occurs at the first hop of a local network. Local network validation aims to prevent spoofing from another host with the same IPv6 prefix. This is accomplished by adding a switch port and a legitimate source IPv6 address or by linking the link-layer address to the IP address and switch port. The study [22] argued that SAVA is an ideal way to stop receiving spoofed packets from many network scopes, but their study did not examine other NDP vulnerabilities, such as DoS attack on DAD, Address Resolution, RA threats, etc.

Another study by [23] has proposed an improved version of SAVA known as Source Address Validation Improvement (SAVI) to secure the link-local communication from source address spoofing. SAVI establishes associations amongst the link-layer address, source IPv6 address, and switch port. SAVI also introduces a filtering policy that forwards and filters identical packets that may have been abandoned [24]. SAVI has limitations as stated in [25], for instance, dynamic address arrangement problems, such as SLAAC and DHCPv6, may also occur in the access network when implementing SAVI, given the difficulty of binding creation (anchor information) when an IP address is changed. Additionally, the binding process in SAVI is also prone to vulnerabilities when LAN devices have multiple IPv6 addresses (e.g., router and multi-LAN hosts and firewalls). Another limitation highlighted by [23] is the presence of several SAVI devices in a network, where each SAVI device operates separately and does not exchange information with other SAVI devices, exposing other devices to traffic spoofing. Therefore, SAVI cannot effectively prevent attacks on the NDP processes in an IPv6 link-local network.

Researchers have explored other venues in order to secure NDP messages in IPv6 link-local networks. Some of these studies have proposed that higher security can be achieved by adding new NDP header options. Among these mechanisms, the most common mechanisms are Secure NDP (SeND) and Trust-ND. SeND [24] adds several new options to NDP, including the cryptographically generated address (CGA) option to verify CGA senders. SeND uses CGAs to generate IPv6 and ensure the ownership of the claimed IPv6 address, as defined in RFC 3972 [25]. Other options include the Rivest–Shamir–Adleman (RSA) cryptosystem signature option, which attaches a public key-based signature, 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 secure all NDP messages [26]. However, several studies [27] show that SeND mechanism introduces a considerable processing 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 [28]. Further, Generating IPv6 using CGA requires high processing time; therefore, the SeND does change IP address which can expose the privacy of hosts in the network. Given these drawbacks of SeND, especially its complexity, malicious hosts can launch DoS attacks, such as flooding attacks, during NDP processes in an IPv6 link-local network.

Praptodiyono *et al.* proposed a new mechanism called Trust-ND for protecting NDP messages 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 [29].

It also uses a new security option called the trust-option 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 SeND because it is based on the SHA-1 hash function. Some researchers claimed that Trust-ND is a light security method for NDP [30]. However, the attacker can spoof the hash value and evade the security mechanism; therefore, Trust-ND mechanism is still vulnerable to various types of NDP threats in an IPv6 link-local network. Therefore, by design, Trust-ND is unsuitable for securing NDP messages in the IPv6 network [29].

[31] proposed a mechanism called Match-Prevention to secure NDP processes (DAD and AR processes) by utilizing SHA-3 to hash a part of the tentative IP address (the node's IP address) and appending the hash value to a verification option termed match-option. In this mechanism, the host performs verification on both messages (NS and NA messages). The sender must first generate the NS-match message with the match-option and related fields when performing AR and DAD. In order to transform this message into an NS-match message, the match-option must be appended to each NS message. The match-option containing the NS-match message is sent using a multicast address in a network. The received hosts must verify the NS-match message. If hash values are valid, hosts conduct AR or DAD and reply by sending an NA-match message. The sender host receives NA-match as a response to the NS-match request. The sender host should verify the match-option. If the hash result matches, the NA-match message is from a legal host; otherwise, the sender host deems it illegitimate. Match-Prevention is not designed to secure all NDP messages, and it cannot secure the AR process if the attackers know the sender IPv6 address.

After outlining the security issues associated with NDP processes and identifying shortcomings in existing mechanisms, it is necessary to build an effective mechanism for securing NDP processes in an IPv6 link-local network to prevent DoS attacks.

VI. PROPOSED NDPsec

The mechanisms discussed in the preceding section to solve security problems with NDP processes require an excessive amount of time to process NDP messages, which attackers may use to flood the network with these messages and launch a DoS and MITM attacks in the link-local network. Thus, the main goal of this study is to propose a new mechanism (NDPsec) that, by using digital signatures, overcomes the shortcomings of prior mechanisms and offers increased security for NDP messages in an IPv6 link-local network.

A. GENERATING IPv6 ADDRESS (STEP ONE)

This step aims to generate IPv6 address for the host in the link-local network. In IPv6 network, SLAAC is used to generate an IP address and configure itself after checking the DAD process. In general, there are two possible methods to generate IID addresses: utilizing the EUI-64 mechanism or the Privacy Extensions mechanism. As stated in the previous sections (section IV), the problem with EUI-64 is the IPv6-driven MAC, which will raise privacy issues for many users, as node packets may be tracked back to a physical machine, and nodes can easily be recognized across several networks or renumbering. The second method, i.e., the Privacy Extensions mechanism, is intended to preserve IPv6 host privacy. However, the NDPsec cannot leverage these two approaches for generating IPv6 addresses since none of these approaches can be used to verify that the host really owns the IPv6 address.

Other scheme (e.g. SeND) generating IPv6 addresses employ both RSA and CGA, which are considered computation-intensive; as a result, SeND do not generate new IPv6 addresses frequently when a host joins the network; instead, it retains the generated IPv6 address for an extended period of time [14].

In general, an IPv6 address is composed of two components: a link-local prefix or network prefix assigned by a router and an IID. The NDPsec algorithm derives the IID for a particular host using the public key of a digital signature. Each IPv6 host that joins the network through the proposed NDPsec mechanism is required to generate public and private key pair using a digital signature algorithm. The Ed25519 digital signature algorithm is selected for this research because it generates the public and private key pair fast and has a smaller public key size than other digital signature algorithms. Additionally, the Ed25519 algorithm is a digital signature technique that employs a twisted Edwards curve variant of the Schnarr signature. It is designed to be faster than existing digital signature schemes while maintaining the same level of security [32].

Once the IPv6 host generates the key pair, which consists of a public-key and a private key pair, the IPv6 host stores the private key for the signing process, while the public key is split into two segments of 8 bytes and 24 bytes, referred to as Left-Fragment (LF) and Right-Fragment (RF), respectively. IPv6 addresses of the host are composed of a series of 16-byte hexadecimal numbers, with the left-most eight bytes representing the link-local or network prefixes with the right-most 8 bytes representing the IID. The NDPsec mechanism takes advantage of this by treating LF as an IID and uses it as an IPv6 address for the host intending to join the network. The RF is subsequently used in conjunction with the IID (i.e., LF) to reconstruct the public key on the receiving hosts, which is used to validate the integrity of the message. Thus, the IPv6 address will associate the public key with the host's IP address. Additionally, since the Ed25529 key pair is fast to generate, the host may utilize it to generate an IID without depleting host resources. When connecting to the same network, the host uses the same IID and only produces a new IID when connecting to a new network. Thus, the host is neither overwhelmed nor are its computing resources squandered. Figure 6 illustrates the process of generating the key pair, IID (i.e., LF) and RF.

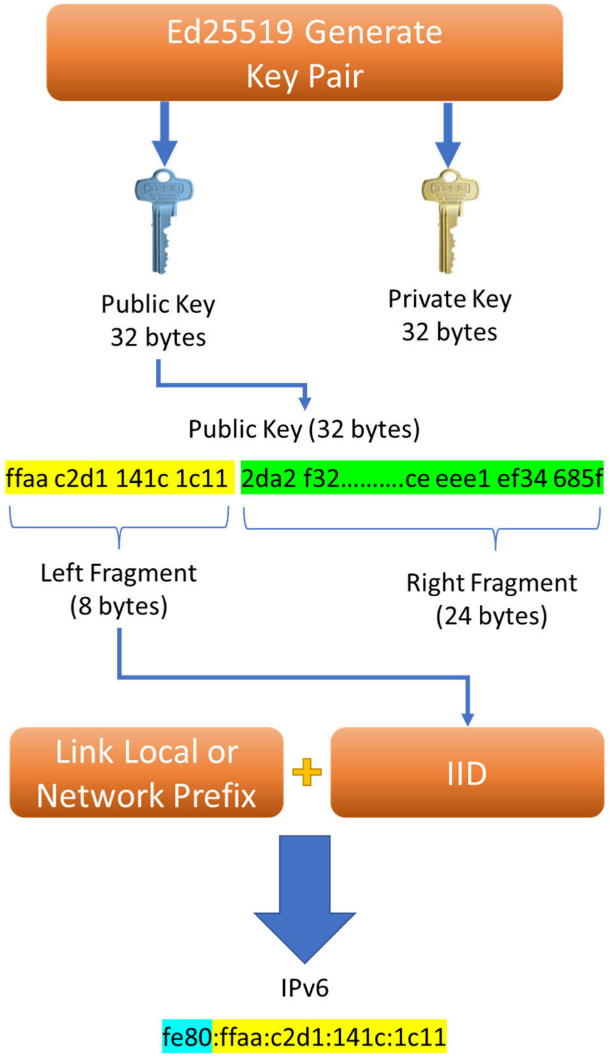

FIGURE 6. Generating the key pair, IID and RF.

B. SECURE-OPTION (STEP TWO)

The step aim to append NDPsec parameters with NDP messages without compromising the original structure of NDP messages. NDPsec uses a digital signature to sign the NDP messages and hence, achieves authentication and integrity required to protect NDP messages from being attacked. Each NDP message is required to carry additional information such as a digital signature, the public key, etc., which is usually not seen in standard NDP messages.

The standard NDP messages have the option field, which can be used to append other useful information. Therefore, this study proposes to utilize this option (the option is named Secure-option) to convey the digital signature and the LF. Secure-option consists of four main fields, namely, RDM, RDM-Info, LF, and DS. The primary objective of proposing this option is to differentiate legitimate NDP messages from invalid ones. Figure 7 illustrates the structure and format of the Secure-option and its associated fields.

Type	Length	RDM (2bytes)
RDM-Info (4 bytes)		
RF (24 bytes)		
DS (32 bytes)		

FIGURE 7. Secure-option and its fields.

To maintain the original structure of NDP messages, the NDP-option design (Figure 6) follows the option format of RFC 4861 [7]. Given that all NDP options should include the type and length fields, the proposed Secure-option also contains these fields. The Secure-option comprises 68 bytes divided into five fields as follows:

- Type: 1-byte identifier that indicates the option type carried by the NDP message. The Secure-option type is 253 because this option is used for experimentation.
- Length: 1-byte field that indicates the total length of the Secure-option, including the type and length fields in an 8-byte unit (64 bits). Given that the total length of the Secure-option is 68 bytes, the value of the Length field is set to 8.
- RDM: 2-bytes field that indicates to replay detection method used in this Secure-option.
- RDM-Info: 4-bytes field that indicates replay info for the Replay Mode.
- RF: This is a 24-bytes field that holds RF generated during step one, which will be combined with IPv6 address.
- DS: This is a 32-bytes field that carries the signature value, which results from the network layer.

The Secure-option must be appended into all NDP messages (i.e., NS, RA, RS, etc.) to validate the NDPsec messages, and messages without the secure-option field must be discarded.

C. SINGING NDP MESSAGES (STEP THREE)

This step aims to explain how the sender of the IPv6 host signs the NDP message. The NDPsec messages must be digitally signed with the private key generated in step one to prevent them from being altered in transit or spoofed by threat actors in order to launch cyber-attacks.

In NDPsec mechanism, when the IPv6 host generates the NDP message, the mechanism generates Secure-option, which is then appended to the NDP message. The RDM and RDM-Info fields should be configured as explained in step five. The RF field of Secure-option contains 24 bytes of RF that was generated in step one. The signature field is filled with zeros during the initialization process. The sending host then signs the entire message using the sender's private key (this includes information about the data link layer, the network layer, the NDP message contents, and the NDPsec fields). Consequently, the signature field is replaced with the resultant signature, which was previously filled with zeros. The signature is used to protect the integrity of the NDP message; since attackers are unaware of the target host's private key, they cannot generate the NDP message on behalf of the victim host and claim ownership of the address.

D. VERIFYING NDP MESSAGES (STEP FOUR)

This step attempts to verify the NDP message by validating the digital signature that was included in the message. Upon receiving the NDP messages, the receiving host must first verify that the NDPsec option field is present in the NDP message; otherwise, the message must be regarded as an attack initiated by the threat actors and must be rejected immediately.

To validate received NDP messages, the receiving host must first extract the public key from the message. NDPsec achieves this by combining IID with the RF value contained within the message. The successful extraction and construction of the public key from the received IPv6 message ensures that the message is sent by the legitimate owner of the IPv6 address. Afterwords, as stated in step five, the receiving host checks for the replay attack by inspecting the RDM and RDM-Info fields. Following that, the receiving host checks the digital signature that was attached to the message for authenticity by using the constructed public key. If the message fails the verification, it indicates that the message is suspicious and may have been sent by an attacker and must thus be rejected. Once the message has passed all of these checks and is deemed genuine, it will be processed in accordance with RFC 4861 [7].

E. PREVENTING REPLAY ATTACK (STEP FIVE)

Replay attacks are one of the most common types of attacks against authentication security mechanisms. In replay attacks, the attacker uses an old authentication message to configure the victim with old configuration information, leading to a DoS attack or MITM. NDPsec makes use of the RDM and RDM-Info fields (as shown in Figure 7) of the Secure-option to prevent replay attacks. Nonce and timestamp are the two prevalent ways of preventing replay attacks. When transmitting and receiving NA and NS messages, the nonce mode is used. While sending an NA or NS message, the host sets the Replay mode to 1 and the Replay Info to a random integer number. The receiver should first check if the nonce has been received from the same sender (i.e., indicate the message has been re-sent by the attacker), and if it has, the message should be discarded; otherwise, the message should be retained. The host then sends a response message to the sender using the same nonce; the sender should also check to see whether the sender nonce is the same as the received nonce. If equal, the message is accepted; otherwise, the message is discarded.

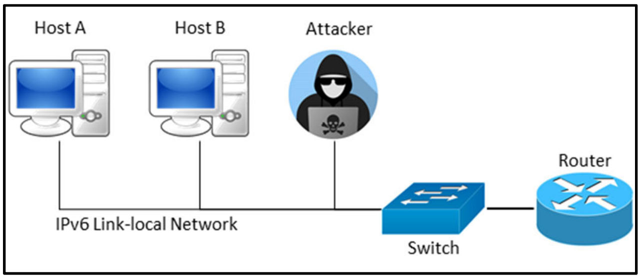

FIGURE 8. Network topology and device specification.

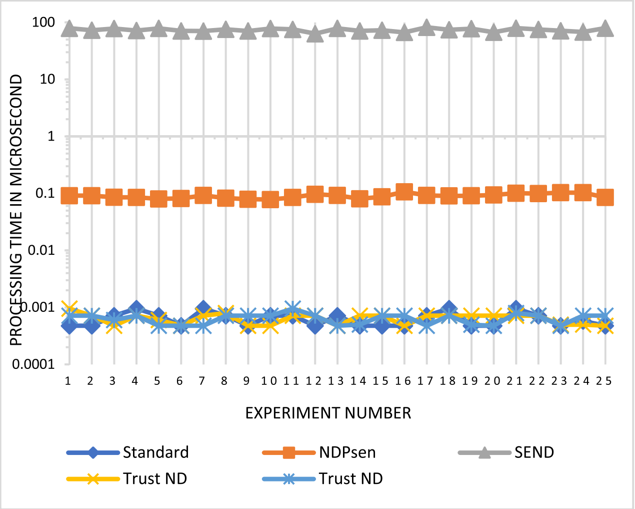

FIGURE 9. Processing time for generating IPv6 address.

On the other hand, the timestamp mode is employed for sending and receiving RA, RS, and R messages. The sender should set the Replay mode to 2 (i.e., Timestamp mode) and set the Replay Info to timestamp. Each host in the NDPsec mechanism must keep a copy of the timestamp of the last message it received from any other host in the network. This is important because, in order to prevent replay attacks, the receiving host must check that the timestamp in the received message is greater than the previous timestamp it saved; failure to comply with this check will result in the rejection of the message because it is obvious that the attacker has sent the same message which has previously been received. There is one exception to this rule, and that is when a message is received for the first time from a host. Because there is no copy of the previous timestamp associated with this host, the message will be accepted, and the timestamp will be stored for future use.

VII. IMPLEMENTATION OF THE PROPOSED MECHANISM NDPsec

Experiments on a local network were conducted to evaluate the functioning and performance of the proposed NDPsec mechanism. The network topology, which comprises two hosts, one router, and one attacker, is shown in Figure 8.

Figure 9 depicts the network topology wherein all devices are connected directly to the switch. The hosts and router

TABLE 1. Hardware and software requirements for the testbed environment.

Item Name	CPU	Memory	Operating System
Host A	Intel(R) Core(TM) i7-2640M CPU @ 2.3 GHz × 4	4.00 Gb	Ubuntu 16.04
Host B	Intel(R) Core(TM) i7-3770M CPU @ 3.40GHz × 8	8.00 Gb	Windows 10 Pro
Attacker Host	Intel(R) Core(TM) i7-2640M CPU @ 2. 30GHz × 4	6.00 Gb	Ubuntu 16.04
Switch (SW)	Cisco Catalyst 2960 Fast Ethernet		
Router	Cisco Router C7200		

were modified to use NDPsec. The NDPsec is developed by using the python programming language [33]. The Python-based cryptography library is utilized for the Ed25519 digital signature algorithm [34]. The attacker runs on Kali Linux, which is used for Penetration Testing. IPv6 attacks are carried out using Scapy and flood_router26.c, whereas Wireshark is used to monitor network activities [35]. In order to protect the RA message, the router's public key is pre-configured on all hosts in the network as the proposed mechanism does not provide a mechanism to distribute the public key of the router. The specifications of the hardware and software used for deploying the testbed environment are presented in Table 1.

VIII. EXPERIMENTS AND EVALUATION

The proposed NDPsec is designed to authenticate NDP messages. Accordingly, the NDPsec is evaluated and compared against Standard NDP, SeND, Match-Prevention, and Trust-ND mechanisms in terms of (i) performance which includes processing time for generating IPv6 address, verifying and generating NDP messages, and (ii) penetration test. This section shows the NDPsec experiments.

A. PROCESSING TIME FOR GENERATING IPv6 ADDRESS (PERFORMANCE)

This experiment aims to measure the processing time of generating an IPv6 address. The experiment has been applied to Standard NDP, SeND, Match-Prevention, Trust-ND, and NDPsec. The processing time has been calculated by subtracting the ending time of generation IPv6 from starting time of the verification process. Due to the possibility that other processes running on the operating system may have an effect on the processing time, the experiments are repeated 25 times to ensure the findings are reliable. Figure 9 shows a line chart for generating an IPv6 address. Based on the results, the SeND has the highest processing time since it makes use of the RSA digital signature and CGA, both of which are considered compute-intensive operations. In contrast, Standard NDP, Match-Prevention, and Trust-ND have almost the same processing time because they use the Privacy Extensions mechanism. NDPsec has slightly more processing time because it uses Ed25519 for generating the IPv6.

B. PROCESSING TIME FOR GENERATING AND VERIFYING NDP MESSAGES (PERFORMANCE)

This experiment aims to measure the total process time while generating and verifying NDP messages. The experiment has been applied for Standard NDP, SeND, Trust-ND, and NDPsec messages which are NA, NS, RA, RS, R messages. Additionally, because the Match-Prevention mechanism is proposed to secure only the DAD and Address Resolution processes, only NS and NA were calculated, while the remaining NDP messages (i.e., RS, RA, and R) were left unprotected and handled in the same way as the Standard NDP. The total processing time (Pt) between a sender and a receiver is calculated by subtracting the starting time (St) from the ending time (Et) of the generating process (Gp) and verifying process (Vp) for the NDP messages, followed by the summation of the generating and verifying processes for the NDP messages, as shown in Equation (1):

$$PT = \left(Et_{(Gp)i} - St_{(Gp)i} + Et_{(Vp)i} - St_{(Vp)i}\right) \tag{1}$$

Sometimes, the processing time is influenced by other operating system activities, which may affect the results; thus, the experiments are repeated 25 times to verify the reliability of the results. The mean, standard deviation (STDVE), and overhead for generating and validating NDP messages are shown in Table 1. The overhead is calculated by using the total Standard NDP message size as a baseline. The experiments (results from Table 2) show that the overall processing time of SeND is higher than NDPsec. This is because the SeND method utilizes both RSA digital signatures and CGA, while NDPsec uses only the Ed25519 digital signature, which requires less processing time for generating and verifying NDP messages. Besides, SeND appends four options to the NDP messages, thereby extending the time required for processing to generate and verify the messages.

Additionally, the Match-Prevention and Trust-ND mechanisms outperformed both the SeND and NDPsec mechanisms since these mechanisms employ hashing, which is significantly faster than signature-based mechanisms and consumes less processing time. Moreover, Due to the lack of an authentication mechanism, standard NDP has the lowest total processing time of all the tested security mechanisms.

C. TRAFFIC OVERHEAD (PERFORMANCE)

This experiment aims to measure the traffic overhead associated with Standard NDP, SeND, Trust-ND, and the proposed NDPsec mechanism. The total message size for NDP messages has been calculated by summing the message sizes for the different mechanisms, which include NA, NS, RS, RA, and R. The traffic overhead is calculated by comparing it to the total size of Standard NDP messages (i.e., baseline).

Table shows that a larger message size can significantly influence network traffic. Therefore, using a small

TABLE 2. Total processing time (NANOSECOND).

	Standard NDP		SeND			Match-Prevention			Trust-ND			NDPsec		
	STDVE	Mean	STDVE	Mean	Overload	STDVE	Mean	Overload	STDVE	Mean	Overload	STDVE	Mean	Overload
NA	0.05	0.71	42.77	741.68	740.97	1.15	2.05	1.34	0.16	2.14	1.43	8.08	146.87	146.16
NS	0.07	0.67	40.68	742.63	741.96	0.93	1.99	1.32	0.2	2.01	1.34	1.32	140.19	139.52
RS	0.05	0.71	11.8	701.69	700.98	0.08	0.75	0.04	0.16	2.16	1.45	5.17	143.79	143.08
RA	0.05	0.69	43.55	790	789.31	0.06	0.70	0.01	0.14	2.08	1.39	1.85	139.88	139.19
R	0.08	0.62	59.51	771.05	770.43	0.09	0.64	0.02	0.25	1.86	1.24	10.91	147.05	146.43

TABLE 3. Message size and traffic overhead (in Bytes).

	Standard-NDP	SeND	Match-Prevention	Trust-ND	NDPsec
NS	86	454	102	118	154
NA	78	446	102	110	146
RA	118	486	118	150	186
RS	70	438	70	102	138
R	174	542	174	206	242
Summation	526	2366	566	686	866
Overhead	-	1840	40	160	340

message size can improve the performance of the network. The existing mechanisms have large message sizes due to their designs. For example, SeND employs four options: CGA, Timestamp, Nonce, and RSA Signature. Each of these options increases the size of NDP messages, resulting in an increase in traffic overhead that is 1840 bytes greater than the traffic overhead for all other mechanisms. Meanwhile, Trust-ND uses SHA-1 as a hash-function algorithm (lightweight compared to the digital signature algorithm) for hashing NDP messages. This significantly decreases the message size and reduces the traffic overhead to 160 bytes. Besides, Match-Prevention got less overhead than the other mechanisms due to the used hash method, and RA, RS, and R messages were not considered/secured by the abovementioned mechanism; thus, it was the same as the Standard NDP message size. For the proposed NDPsec mechanism, the traffic overhead is 340 bytes due to the mechanism design. NDPsec uses only one option instead of four options compared to SeND; however, employing a digital signature led to more traffic overhead than Trust-ND and Match-Prevention mechanisms.

TABLE 4. DoS-on-DAD experiment results.

Mechanism Name	Successful Number	Successful rate
Standard NDP	10	100%
SeND	0	0%
Match-Prevention	0	0%
Trust-ND	10	100%
NDPsec	0	0%

D. DUPLICATE ADDRESS DETECTION (PENETRATION TEST)

The purpose of this experiment is to determine the ability of various preventing mechanisms studied in this study (i.e., Standard NDP, SeND, Match-Prevention, Trust-ND, and NDPsec) to prevent DOS attacks on the DAD process. This experiment is performed when host A connects to the network and tries to configure itself with a new IPv6 address by using the DAD process. The attacker attempts to prevent the new host from obtaining an IPv6 address, resulting in the host being unable to gain access to the network. This experiment has been repeated ten times to ensure the robustness of the proposed mechanism. If the IPv6 host could not configure itself with an IPv6 address, the attack is considered successful; otherwise, the attack was regarded as a failure. The success rate of the attacks has been calculated based on Equation (2).

$$SR = \frac{Sn}{n} * 100 \tag{2}$$

where SR refers to success rate, Sn represents Successful number, and n is the number of experiments. Table shows the results of the experiment.

As results indicate, Standard NDP failed to prevent the DoS-on-DAD attack because Standard NDP lacks a security mechanism to prevent the attacks. Trust-ND is based on a

TABLE 5. Address resolution attack results.

Mechanism Name	Successful Number	Successful Rate
Standard NDP	70	70%
SeND	0	0%
Match-Prevention	0	60%
Trust-ND	60	60%
NDPsec	0	0%

trust mechanism, enabling an attacker to evade the Trust-ND security mechanism by simply hashing NDP messages without requiring a key. Match-Prevention succeeds in securing the DAD process due to hashing the tentative IP address during the exchange of NS and NA via the entire process. NDPsec successfully prevented attacks on the DoS-on-DAD process because the attacker cannot claim ownership of the IPv6 address since the attacker does not have the private key of the IPv6 host. Further, SeND can also prevent the DoS-on-DAD attack because it uses RSA digital signature and CGA.

E. ADDRESS RESOLUTION (PENETRATION TEST)

In an AR spoofing attack, the attacker aims to spoof the NS message to inject the attacker's MAC address. If Host A saves the attack's MAC address in the neighbor cache table, the attack is deemed successful; otherwise, the attack is considered unsuccessful, and the message is discarded. The experiments are repeated ten times, and the success rate is calculated using Equation (2). Table 5 summarizes the outcome of the experiments.

The attack success rate for Standard NDP Match-Prevention, and Trust-ND is 70% 60% and 60%, respectively. The attacks sometimes fail because both mechanisms (Standard NDP Match-Prevention, and Trust-ND) cannot distinguish between messages sent by a legitimate host and those sent by an attacker, in consequence, the host configures itself with a legitimate message. Furthermore, the SeND and NDPsec successfully thwarted AR attacks because an attacker cannot spoof IPv6 addresses without having a valid key to sign the messages.

F. REDIRECT ATTACK (PENETRATION TEST)

This experiment aims to test secure Redirect attacks. Redirection is a simple mechanism based on NDP that allows a router to suggest a better route to an IPv6 host. The attacker can exploit the redirect mechanism by spoofing the Redirect message and redirecting the host traffic from a router to a specific address in order to launch a DoS or MITM attack. The victim host will accept the Redirect message because it lacks a mechanism to validate the origin of the received message. The attack is considered successful if host A configures redirect the traffic to the attacker; otherwise, the attack will be considered as failed. Likewise, to other experiments, the experiment was repeated ten times, and Equation (2) is used to compute the success rate of the attacks. Table shows the experiment results.

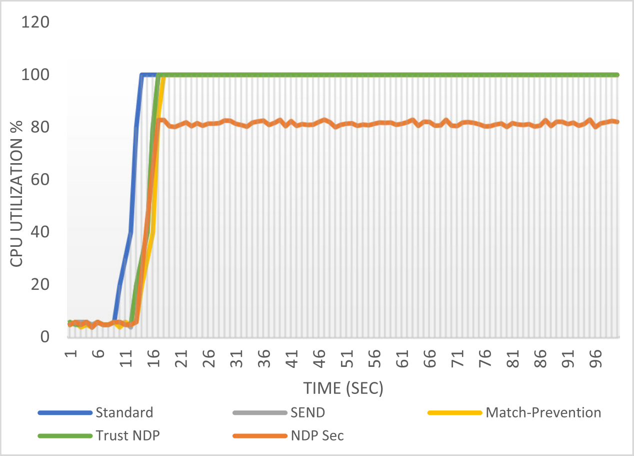

FIGURE 10. CPU utilization during the RA flooding attack.

The results show that both NDPsec and SeND mechanisms can prevent the DoS attack. In contrast, the Trust-ND cannot prevent the DoS attack, and that is due to the attacker in the Trust-ND can authenticate any Redirect message by just hashing the message; therefore, the victim host will be accepted. On the other hand, NDPsec and SeND mechanisms both provide the authentication for the IPv6 address; thus, the attacker cannot spoof the source of the IPv6 address. The Match-Prevention has failed to secure this kind of attack because the R message was not secure/addressed by the mechanism.

G. RA FLOODING (PENETRATION TEST)

This experiment aims to verify the immunity of the different mechanisms against the RA flooding attacks. Aside from being a RA flooding that can cause DoS to the IPv6 host, it is also considered fatal to the entire IPv6 link-local network. In this experiment, we measured the CPU consumption under the RA flooding attack and system failure. IPv6 attacks were launched using the toolkit flood_router26.c tool and Scapy. Figure 10 shows the CPU utilization during the RA flooding attack.

All mechanisms compared in this study have a processing time of 100%, with the exception of the proposed NDPsec mechanism, which has a processing time of around 82%. The standard NDP is affected by the RA message flooding attack because the client has to configure itself with attacker RA messages. Every RA message has around 12 prefixes, which is required to perform 12 DAD processes. Furthermore, the Trust-ND that relies on a trust mechanism, the Trust-ND host accepted all the RA flooding attacks because all the RA flooding Trusted NDP option with a hash value. On the other hand, the SeND mechanism does not accept the RA flooding attack; instead, it verifies the CGA and RSA signature, considered

TABLE 6. Redirect attack results.

Mechanism Name	Successful Number	Successful rate
Standard NDP	10	100%
SeND	0	0%
Match-Prevention	10	100%
Trust-ND	10	100%
NDPsec	0	0

compute-intensive operations. Meanwhile, the CPU utilization for the proposed NDPsec mechanism is recorded at 82% because the IPv6 host can prevent the RA flooding attacks, as it is required to verify only the Ed25519 signature. Also, Match-Prevention mechanism could not prevent the RA flooding attack as it is not designed for it. Accordingly, the NDPsec can mitigate the RA flooding attack impact on the IPv6 host.

IX. DISCUSSION

The proposed NDPsec mechanism has almost the same processing time for generating IPv6 addresses as compared to other mechanisms, including the Standard NDP, Trust NDP and Match-Prevention mechanisms. The SeND requires a high processing time for generating IP addresses because it employs RSA and CGA algorithms. The NDPsec's IP generation process relies on the Ed25519 digital signature keypairs, which is considerably faster than the SeND's RSA and CGA.

Based on the experiments, it is clear that NDPsec can reduce the complexity of the process while generating and verifying NDP messages compared with SeND mechanism. The NDPsec is around five times faster than SeND. Consequently, this leads to limitations for using SeND in mobile or IoT devices with limited resources. Besides reducing the complexity, the NDPsec also reduced the traffic overhead by around 57.08% compared to the SeND mechanism. Hence, NDPsec significantly reduces communication cost and bandwidth utilization. Furthermore, the Standard NDP, Match-Prevention, and Trust-ND mechanisms have less processing time when generating and verifying messages and generate less traffic size. However, the Trust-ND failed to prevent all NDP attacks, and Match-Prevention was only able to secure DAD but not the rest of the NDP processes. On the other hand, the SeND prevents DoS-on-DAD, Address Resolution, and Redirect attacks, like the proposed mechanism. However, SeND failed to prevent RA flooding attacks. Although SeND prevents the attacker from injecting RA messages into the IPv6 host, the attacker consumes the host's CPU, causing the host to stop responding to any valid incoming message while under attack. In contrast, the proposed NDPsec mechanism is immune against all the NDP attacks, including the RA flooding attack, as it is designed to be a lightweight process and preserve security; accordingly, using NDPsec can provide authentication for the NDP messages. The summary of the penetration experiments, as well as the comparisons, are provided in Table 7.

TABLE 7. Experiment results summary and comparison.

	DoS-on-DAD	AR	Redirect Attack	RA Flooding
Standard	No	No	No	No
Trust-ND	No	No	No	No
Match-Prevention	Yes	No	No	No
SeND	Yes	Yes	Yes	No
NDPsec	Yes	Yes	Yes	Yes

*No: Failed to prevent the attack, and Yes: Succeed to prevent the attack

X. CONCLUSION AND FUTURE WORK

Securing NDP messages is essential for IPv6 link-local networks. The Standard NDP protocol does not have any verification mechanism to validate incoming messages, therefore any attacker that exists on the network can exploit the NDP message to lunch its attacks. Thus, the NDPsec is considered a sophisticated and modern mechanism to secure NDP messages in the IPv6 network. It is designed and implemented to overcome the issues of other mechanisms. It is designed by using the Ed25519 digital signature to generate IPv6 and provide authentication. The experiments show clear superiority in NDPsec in most of the experiments as compared to SeND Match-Prevention, and Trust-ND. Therefore, NDPsec is considered a better alternative than SeND Match-Prevention, and Trust-ND. Some potential future works include; making the proposed mechanism work with stateful mode (i.e DHCPv6) [36], [37] and proposing a mechanism to distribute the router's public key in the link-local network.

REFERENCES

[1] Y. Huang, S. Nazir, X. Ma, S. Kong, and Y. Liu, "Acquiring data traffic for sustainable IoT and smart devices using machine learning algorithm," *Secur. Commun. Netw.*, vol. 2021, pp. 1–11, Jun. 2021.

[2] S. Suryavansh, A. Benna, C. Guest, and S. Chaterji, "Ambrosia: Reduction in data transfer from sensor to server for increased lifetime of IoT sensor nodes," 2021, *arXiv:2107.05090*.

[3] K.-H. Li and K.-Y. Wong, "Empirical analysis of IPv4 and IPv6 networks through dual-stack sites," *Information*, vol. 12, no. 6, p. 246, Jun. 2021.

[4] A. Al-Ani, M. Anbar, S. A. Laghari, and A. K. Al-Ani, "Mechanism to prevent the abuse of IPv6 fragmentation in OpenFlow networks," *PLoS ONE*, vol. 15, no. 5, May 2020, Art. no. e0232574.

[5] Google. (2021). *Google IPv6 and Google*. Accessed: Sep. 2, 2019. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[6] L. Ubiedo, T. O'Hara, M. J. Erquiaga, and S. Garcia, "Current state of IPv6 security in IoT," 2021, *arXiv:2105.02710*.

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

[8] 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.

[9] E. Mahmood, A. H. Adhab, and A. K. Al-Ani, "Review paper on neighbour discovery protocol in IPv6 link-local network," *Int. J. Services Oper. Inform.*, vol. 10, no. 1, pp. 65–78, 2019.

[10] F. Abusafat, T. Pereira, and H. Santos, ''Roadmap of security threats between IPv4/IPv6,'' in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS)*, Apr. 2021, pp. 1–6.

[11] J. L. Shah and H. F. Bhat, ''Towards a secure IPv6 autoconfiguration,'' *Inf. Secur. J., Global Perspective*, vol. 29, no. 1, pp. 14–29, Jan. 2020.

[12] M. Tayyab, B. Belaton, and M. Anbar, ''ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: A review,'' *IEEE Access*, vol. 8, pp. 170529–170547, 2020.

[13] S. Supriyanto, I. Hasbullah, R. Murugesan, and A. Osman, ''Risk analysis of the implementation of IPv6 neighbor discovery in public network,'' in *Proc. 1st Int. Conf. Elect. Eng., Comput. Sci. Inform.* Institute of Advanced Engineering and Science, 2014.

[14] A. S. A. M. S. 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, doi: 10.1109/ACCESS.2017.2737524.

[15] A. Cooper, F. Gont, and D. Thaler, *Security and Privacy Considerations for IPv6 Address Generation Mechanisms*, document RFC 7721, 2016, pp. 1–18.

[16] T. Narten, R. Draves, and S. Krishnan, *Privacy Extensions for Stateless Address Auto Configuration in IPv6*, RFC 3041, Jan. 2001.

[17] 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.

[18] 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, 2013.

[19] A. Herrera, ''How secure is the next-generation internet? An examination of IPv6,'' Defence Sci. Technol. Org. Edinburgh (Australia) Cyber Electron. Warfare Division, 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, doi: 10.1371/journal.pone.0151612.

[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 (NETAPPS)*, 2015, pp. 967–978.

[22] J. Wu, J. Bi, X. Li, G. Ren, K. Xu, and M. Williams. (2007). *Source Address Validation Architecture (SAVA) Framework: Draft-Wu-Sava-Framework00. Txt*. Accessed: Mar. 31, 2022. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.182.8500

[23] 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. Accessed: Jun. 30, 2022. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/02564602.2016.1192964?casa_token=ZVZBjnHwBJQAAAAA:vomsootFc2dJSQXnO5wRtlfTja2debLWfWfxu4RoNfw8aGH2jbVB0DS4kqvZ0pN9Z2XwK5Kl-bh62WA

[24] J. Arkko, J. Kempf, B. Zill, and P. Nikander, *Secure Neighbor Discovery (SEND)*, document RFC 3971, 2005.

[25] T. Aura and C. G. Addresses, *Cryptographically Generated Addresses (CGA)*, document RFC 3972, 2005.

[26] 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.

[27] 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, doi: 10.1007/s13369-018-3643-y.

[28] 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.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 6917, 2011, pp. 312–325, doi: 10.1007/978-3-642-23951-9_21.

[29] J. L. Shah, ''A novel approach for securing IPv6 link local communication,'' *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 1–3, Apr. 2016, doi: 10.1080/19393555.2016.1180568.

[30] E. Andreeva, B. Mennink, and B. Preneel, ''Open problems in hash function security,'' *Des., Codes, Cryptogr.*, vol. 77, nos. 2–3, pp. 611–631, 2015, doi: 10.1007/s10623-015-0096-0.

[31] A. K. Al-Ani, M. Anbar, A. Al-Ani, and D. R. Ibrahim, ''Match-prevention technique against Denial-of-Service attack on address resolution and duplicate address detection processes in IPv6 link-local network,'' *IEEE Access*, vol. 8, pp. 27122–27138, 2020. Accessed: Mar. 31, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8977518/

[32] Y. Romailler and S. Pelissier, ''Practical fault attack against the Ed25519 and EdDSA signature schemes,'' in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC)*, Sep. 2017, pp. 17–24, doi: 10.1109/FDTC.2017.12.

[33] S. W. B. Tan, P. K. Naraharisetti, S. K. Chin, and L. Y. Lee, ''Simple visual-aided automated titration using the Python programming language,'' *J. Chem. Educ.*, vol. 97, no. 3, pp. 850–854, Mar. 2020.

[34] Y. Acar, M. Backes, S. Fahl, S. Garfinkel, D. Kim, M. L. Mazurek, and C. Stransky, ''Comparing the usability of cryptographic Apis,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 154–171.

[35] H. Häffner, C. Roos, and R. Blatt, ''Quantum computing with trapped ions,'' *Phys. Rep.*, vol. 469, no. 4, pp. 155–203, Dec. 2008, doi: 10.1016/j.physrep.2008.09.003.

[36] A. Al-Ani, M. Anbar, A. K. Al-Ani, and I. H. Hasbullah, ''DHCPv6Auth: A mechanism to improve DHCPv6 authentication and privacy,'' *Sādhanā, Acad. Proc. Eng. Sci.*, vol. 45, no. 1, Dec. 2020, doi: 10.1007/S12046-019-1244-4.

[37] 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. Accessed: Jun. 30, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8726291/

AYMAN AL-ANI received the Ph.D. degree in advance computer network from Universiti Sains Malaysia (USM). He is currently a Senior Lecturer with the Faculty of Computing and Informatics, Universiti Malaysia Sabah (UMS). His current research interests include malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, the Internet of Things (IoT), IPv6 security, artificial intelligence, machine learning, data mining, and optimization algorithms.

AHMED K. AL-ANI received the bachelor's degree in computer technique engineering from the University of Al-Ma'mun, in 2013, the M.Sc. degree in information technology from Universiti Utara Malaysia, in 2016, and the Ph.D. degree in internet infrastructure security from Universiti Sains Malaysia (USM), in 2020. He currently works as an Assistant Professor at the School of Computing and Data Science, Xiamen University Malaysia. His current research interests include software security, intrusion detection system (IDS), intrusion prevention system (IPS), network monitoring, the Internet of Things (IoT), cloud computing, databases, and IPv4/IPv6 security.

SHAMS A. LAGHARI received the B.Sc. (Hons.) and M.Sc. degrees in computer science from the University of Sindh, Jamshoro, Pakistan, and the M.S. degree in computer science from PAF-KIET Karachi, Pakistan. He is currently pursuing the Ph.D. degree in network security with the National Advanced IPv6 Centre, Universiti Sains Malaysia. His research interests include cybersecurity, industry 4.0, distributed systems, cloud computing, and mobile cloud computing.

SELVAKUMAR MANICKAM is currently an Associate Professor working in cybersecurity, the Internet of Things, industry 4.0, and machine learning. He has authored or coauthored more than 160 papers in journals, conference proceedings, and book reviews, and graduated 13 Ph.D. degree students. He has ten years of industrial experience prior to joining academia. He is a member of technical forums at national and international levels. He has also experience in building IoT, embedded servers, and mobile- and web-based applications.

KHIN WEE LAI (Senior Member, IEEE) received the B.Eng. degree (Hons.) from Universiti Teknologi Malaysia, and the dual Ph.D. degree in biomedical engineering from Technische Universitat Ilmenau, Germany, and Universiti Teknologi Malaysia, Malaysia, through the DAAD Ph.D. Sandwich Program. He is currently the Program Head of the M.E. degree (biomedical) with the Faculty of Engineering, Universiti Malaya. His research interests include computer vision, machine learning, medical image processing, and healthcare analytics. He is a Registered Professional Engineer with Practicing Certificate (PEPC) at the Board of Engineers Malaysia (BEM), a fellow of the Engineers Australia (FIEAust), an APEC Engineer at IntPE, Australia, and a Chartered Professional Engineer (CPEng.) at NER, Australia. He is a fellow of the Institute of Engineers Malaysia (IEM), and a member of the Institution of Engineering and Technology (IET), and a U.K. Chartered Engineer (C.Eng.). He currently serves as the Associate Editor for IEEE Access.

KHAIRUNNISA HASIKIN received the B.Eng. degree in electrical engineering and the M.Sc. (Eng.) degree in biomedical engineering from the University of Malaya, Malaysia, and the Ph.D. degree from Universiti Sains Malaysia. She is currently a Senior Lecturer with the Department of Biomedical Engineering, Faculty of Engineering, Universiti Malaya. Her research interests include medical image processing and analysis, expert systems, medical informatics, and sustainability management.