Journal of Informatics and Web Engineering
Vol. 5 No. 2 (June 2026) eISSN: 2821-370X

Enhanced Trust-based Security Mechanism to Prevent Temporal DoS Vulnerabilities in IPv6 Link-Local Networks

Iznan Husainy Hasbullah[1], Lokman Mohd Fadzil[2*], Selvakumar Manickam[3], Supriyanto Praptodiyono[4], Mohamad Khairi Ishak[5]**

[1,2,3]Cybersecurity Research Centre (CYRES), Universiti Sains Malaysia, Jalan Universiti, Gelugor, Pulau Pinang, Malaysia
[4]Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jakarta, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta, Indonesia
[5]Department of Electrical and Computer Engineering, Ajman University, University Street - Al jerf 1 - Ajman - United Arab Emirates

**corresponding author: (lokman.mohd.fadzil@usm.my; ORCiD: 0000-0002-0398-0433)*
***corresponding author: (m.ishak@ajman.ac.ae; ORCiD: 0000-0002-3554-0061)*

Abstract - Many computer networks in operation today currently use both IPv4 and IPv6 stacks. On the other hand, there is a transition towards IPv6-only networks as a result of the limited availability of IPv4 addresses. The primary protocol for link-local IPv6 communication is the Neighbor Discovery Protocol (NDP). Regrettably, its insecure design and basic scope-based security mechanisms make the local network susceptible to insider threats. The Internet Engineering Task Force's recommended security mechanism for NDP, which is Secure Neighbor Discovery (SEND), is well documented but complex and unsuitable for resource-constrained devices and networks. Trust-ND was positioned as an alternative to SEND as a lightweight trust-based distributed approach using the NDP extension headers. However, its timestamp design and utilization render it susceptible to temporal DoS vulnerabilities. Therefore, this research proposes eTrustND to improve the Trust-ND mechanism for securing IPv6 link-local networks from insider attacks by addressing the existing vulnerabilities by modifying the timestamp reference, format, precision, and validation rules. This paper documents the methodology, the experimentation, and the resulting outcome that show eTrustND eliminates Trust-ND's temporal DoS vulnerabilities without adding computational and protocol overhead. It also highlights the challenges and best practices of timestamp design and usage in security mechanisms and protocols.

Keywords— IPv6, Denial-Of-Service, Link-Local Network, Network Security, Trust-Based Security Protocol.

Received: 14 June 2025; Accepted: 23 August 2025; Published: 16 June 2026

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2026.5.2.5

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

1. INTRODUCTION

The Internet Engineering Task Force (IETF) proposed Internet Protocol version six (IPv6) in 1998 as the latest iteration of the Internet protocol and standardized it 19 years later with the publication of the RFC 8200 document [1]. It differs from IPv4 in several aspects, such as a significant increase in its address spaces, simplified header format, flow labeling, support for extensions and options, and authentication and privacy features. It aims to solve the problem of IPv4 address exhaustion [2].

Most IP networks today still use both IPv4 and IPv6 simultaneously. However, IPv6-only networks are gaining ground with many government and corporate sector initiatives. In 2020, the US government mandated that by 2025, 80% of IP-enabled assets on federal networks must operate in IPv6-only environments. Similarly, China's Central Cyber Security Committee has prohibited new networks from using IPv4 after 2023, increasing the shift to IPv6-only networks. Besides governments, corporations such as Amazon [3], Google [4] Cisco [5], and Microsoft [6], are also transitioning to IPv6-only environments within their internal networks.

According to GSMA Intelligence, the number of unique mobile subscribers worldwide reached approximately 5.8 billion by the end of 2024, representing 71% of the global population. Of these, around 4.7 billion individuals were active users of mobile internet services [7]. Mobile service providers began integrating IPv6 into their networks in 2009, following a directive from the 3rd Generation Partnership Project (3GPP) standard organization [8] to facilitate IPv6 on their 4G wireless infrastructures, resulting in a notable uptake of IPv6 among mobile device users. For instance, New T-Mobile, the leading mobile service provider in the US, encompassing over 98 million subscribers domestically and 230 million globally, achieved nearly 100% IPv6 adoption in 2020 [9]. Similarly, China Mobile, holding the title of the largest mobile service provider both in China and worldwide, had over 969 million subscribers as of the end of June 2022 [10].

This paper aims to present an improved version of Trust-ND resistant to temporal DoS vulnerabilities. It has three main objectives of (i) To introduce a new Trust-ND timestamp reference with enhanced format and precision; (ii) To incorporate the proposed timestamp into the Trust-ND message without altering the original packet structure; and (iii) To propose a rule-based timestamp verification mechanism to mitigate temporal DoS vulnerabilities on IPv6 link-local networks.

This following subsection provides the brief overview of Neighbor Discovery Protocol (NDP), Secure Neighbor Discovery (SEND), Trust Neighbor Discovery (Trust-ND), and security challenges in link-local networks related to NDP and its derivatives.

1.1 Neighbor Discovery Protocol

IPv6 introduced a new protocol to support additional functionalities, such as the NDP in RFC2461 and later updated by RFC4861 [11]. NDP is a fundamental IPv6 protocol facilitating crucial processes and functionalities within link-local networks. Notably, IPv6 nodes in a link-local network use NDP to locate routers, detect the presence of other nodes, determine their respective link-layer addresses, ensure the address uniqueness, and monitor the accessibility status of neighboring nodes. To carry out its operations, NDP employs five ICMPv6 messages of (i) Router Solicitation; (ii) Router Advertisement; (iii) Neighbor Solicitation; (iv) Neighbor Advertisement; and (v) Redirect messages. Table 1 shows the five ICMPv6 messages used by NDP.

1.1.2 NDP Duplicate Address Detection Process

Duplicate Address Detection (DAD) is a crucial NDP process that prevents conflicting IPv6 addresses within IPv6 link-local networks. Before assigning any unicast IPv6 address, all nodes must perform DAD, whether configuring manually or via SLAAC or DHCPv6, to ensure the address is not already in use. A DHCPv6 server can also perform DAD for the host.

An IPv6 node performs DAD once prior to assigning the address to its network interface, as dictated by RFC4862 [12], by sending an NS message. If a node does not receive an NA message from its neighbor within one second, it assigns the address to the interface. The DAD process reruns only if a new address is assigned to the interface.

Table 1. ICMPv6 Messages Utilized by NDP

Message	Type	Description
Router Solicitation (RS)	133	For IPv6 hosts to inquire about router(s) in a link-local network.
Router Advertisement (RA)	134	Periodically, routers advertise their presence in the link-local network using RA messages, also to respond to hosts' RS, and propagate network parameters, such as prefixes.
Neighbor Solicitation (NS)	135	For IPv6 nodes to request a target node's link-layer address while providing their link-layer address to the target.
Neighbor Advertisement (NA)	136	For IPv6 nodes to respond to NS messages. Also to propagate the latest information to neighbors.
Redirect	137	Routers use Redirect message to update IPv6 hosts with a better first-hop node on the path to a destination or inform a host that the destination is its neighbor.

1.1.3 Secure Neighbor Discovery

The RFC3971 document [13] describes the SEND protocol that improves the IPv6 link-local security by introducing address ownership proof, a message integrity function, and a new router authorization mechanism. Four new Neighbor Discovery options were introduced by SEND to deliver these improvements: the Cryptographically Generated Address (CGA), RSA Signature, Nonce, and Timestamp options. Additionally, it also specifies a new protection mechanism for router discovery operation.

Several studies, such as [14], have thoroughly examined the computational demands and deployment challenges associated with SEND. Both experimental testing and theoretical analysis have established that the processes involved in creating CGAs and generating RSA signatures are the main sources of complexity in SEND [15], [16], resulting in vulnerability to CPU exhaustion attacks, which are a form of Denial-of-Service (DoS) attack [17], [18], [19]. Additionally, adding four extra SEND options increased bandwidth consumption by 368 bytes for each NDP message [16], [19].

1.1.4 Trust-ND

Trust-ND, proposed by Supriyanto [20], is a lightweight mechanism for securing the NDP used by IPv6 nodes in IPv6 link-local networks. Trust-based mechanisms view the existence of intruders in the network as given and attempts to identify them as untrusted parties to prevent malicious activities [21]. Trust-ND utilizes a trust-based soft security approach, specifically the beta reputation function within the model of probabilistic trust model [22] to identify trusted IPv6 nodes within link-local networks.

Hard security approaches, including authentication, encryption, and access control, are effective, widely used, and well researched. However, they tend to be more complex, require additional resources, and may depend on third parties. Trust-ND uses distributed trust management and a cryptographic hash function to ensure data integrity with less computation and overhead than SEND. Network protocols commonly use hash functions to verify message integrity [23].

The subsequent subsections provide more details on Trust-ND, its structure, improvements to SEND, and timestamp verification.

a. Trust-ND Structure

The main component in Trust-ND protocol is the Trust Option, which follows the ICMPv6 Type-Length-Value format, as illustrated in Figure 1.

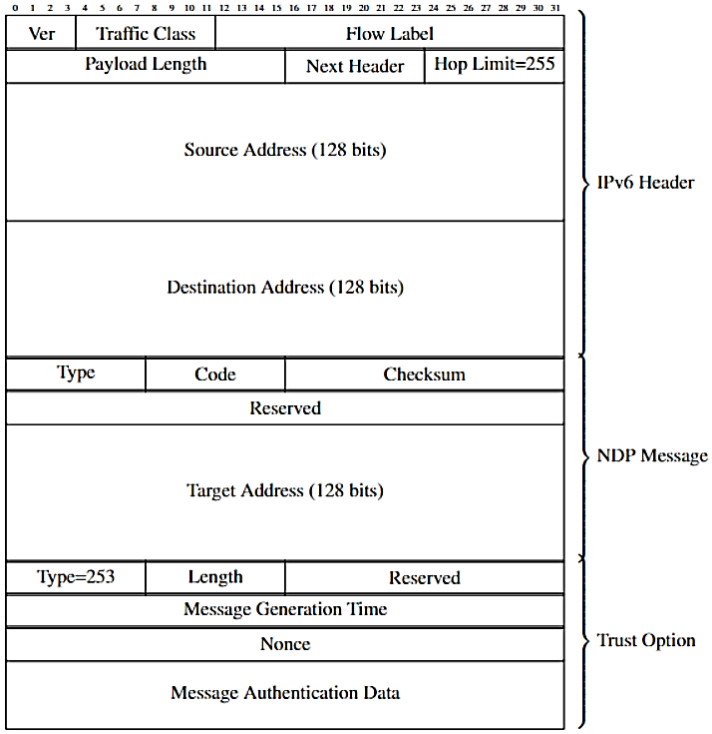


Figure 1. Trust-ND Packet structure: IPv6 Header, NDP Message, and Trust Option

All Trust-ND messages must incorporate Trust Option, and its fields are as follows.

1. The TYPE field is a 1-octet identifier that specifies the type of content carried by the NDP message. Trust-ND uses the value 253, which has been officially allocated by the Internet Assigned Numbers Authority (IANA) for experimental use (IANA, 2021).

2. The LENGTH field is a one-octet parameter that specifies the total size of the Trust Option, inclusive of both the TYPE and LENGTH fields measured in octets. For instance, if the Trust Option's size is 32 bytes, the LENGTH field will hold a value of four (since 4 × 8 = 32).

3. MESSAGE GENERATION TIME is a 4-octet (32-bit) timestamp field that records the time the sender generates each message. It is a hex-formatted time field, including hour, minute, second, and millisecond [20] to mitigate DoS flooding and replay attacks by ensuring timely message delivery and freshness.

4. The NONCE is a 4-octet (32-bit) random number generated by the sender of the Trust-ND message to confirm the uniqueness of each NDP message, preventing replay attacks. It ensures the freshness of advertisement messages and corresponds to solicitation messages.

5. MESSAGE AUTHENTICATION DATA (MAD) is a 20-octet (160-bit) field comprising the Secure Hash Algorithm 1 (SHA-1) hash function output on the sender's message header and the data to ensure data integrity within the message, allowing the receiver to detect any error or modification after generation.

b) Trust-ND Improvement to SEND

The authors of SEND admitted its complexity due to the addition of CGA and RSA Signature options [13]. Therefore, the author of Trust-ND eliminated the source of SEND's complexity by removing the RSA Signature option, which is computationally expensive to generate and verify, replacing it with the SHA-1, an unkeyed hash function. Furthermore, hiding or encrypting the node's address is not required for NDP since the content of NDP message should be visible to neighbors to enable NDP processes to work correctly, especially for discovering neighbors and

routers, and detecting duplicate addresses.

Trust-ND retains the Timestamp and Nonce options from SEND but in a different form as a field within the Trust Option. Trust-ND replaces four SEND options (368 bytes) with one Trust option (32 bytes), reducing 336 bytes from the NDP packet size, which significantly reduces bandwidth consumption for Trust-ND compared to SEND [24].

c) Trust-ND Timestamp Verification

All Trust-ND messages received must be validated by the receiver without exception. After confirming the correctness of all standard NDP fields' values and ensuring that the Trust Option exists, the receiver checks the Nonce and Message Generation Time fields, then validates the message integrity using the hash value in the MAD field. Nonce must be unique, except for operations that utilize a pair of Trust-ND messages in a request-respond manner. Additionally, the message arrival time must be greater than the message generation time (Equation (1)). In other words, logically, messages must arrive at the receiver after the sender sends them, not before.

$$T_r > T_s \tag{1}$$

where,
T_r is the time at the receiver when the message arrives, and
T_s is the message generation time or timestamp embedded within the message's Trust Option.

Most Trust-ND processes, including DAD, involve a pair of Trust-ND messages such as Trust-NS and Trust-NA, in a request-respond behaviour. These types of message exchanges have two extra verification criteria. First, the Nonce of the solicited advertisement message must be identical to the solicited message's Nonce value. Second, the generation time of the solicited message must fall within a certain window relative to the sender, bound by the following criteria (Equation (2)).

$$T_s < TS < T_{r,max} \tag{2}$$

where,
TS is the timestamp of the solicited Trust-NA, T_s is the timestamp or message generation time embedded within the Trust Option of the soliciting message, and
$T_{r,max}$ is the cutoff time for the sender to receive a corresponding reply for its solicitation message, as defined by Equation 2.

The solicited message's time must be after T_s. The solicited Trust-NA's timestamp must be greater than the soliciting message's generation time, T_s. In addition, the solicited message's time must not exceed $T_{r,max}$ (Equation (3)).

$$T_{r,max} = T_s + t \tag{3}$$

where,
T_s is the timestamp embedded in the Trust Option of the solicitation message, and
t is the time limit that defines the maximum duration the sender will wait for a response.

The sender initiating the message exchange must cache the message generation time (T_s) as the starting point to keep track of the time limit (i.e., $T_{r,max}$) on receiving a reply to the solicitation message.

If the value of TS is less than or equal to T_s ($TS \ll T_s$), then the received packet is assumed to be invalid since it indicates that the reply message is generated either before or simultaneously with the soliciting message.

Trust-ND specifies that if the receiving time of the reply message T_r is less than or equal to T_s ($T_r \leq T_s$), then the value of T_s is invalid, and the corresponding packet must be discarded. Therefore, any message received after T_s and before $T_{r,max}$ will be accepted for further processing. Otherwise, the receiving interface must discard any reply

message outside the ranges.

1.1.5 Common Attacks Against the NDP Process

IPv6 link-local networks a susceptible to security threats due to lack of a robust default security mechanism for NDP, including DoS, DDoS, replay, spoofing, redirect, masquerading, redirect, and Man-in-the-Middle (MitM) attacks (Nikander et al., 2004; Supriyanto, 2012; Najjar et al., 2015; A. K. Al-Ani, Anbar, Manickam, Al-Ani, et al., 2019). In addition, misconfiguration also contributes to security threats to IPv6 link-local networks, such as rogue router incidents described in RFC6104 [25].

DoS attacks are the second most common type of incident listed in the Common Vulnerabilities and Exposures (CVE) database since 1999, following code execution, with 28,960 incidents compared to 44,266. The CVE database is a publicly accessible repository that catalogs standardized information about security vulnerabilities and exposures found in software and hardware systems. In addition, more than half of the reported software vulnerabilities in MITRE's 2022 CWE™ Top 25 [26] could result in DoS, affecting the availability of applications and services. The CWE™ Top 25 identifies common, exploitable software weaknesses that can lead to security breaches, data loss, and system downtimes.

Adversaries could exploit temporal DoS vulnerabilities, whether due to poorly designed network protocol [27], the inability of network devices to handle an extreme load [28], or malicious activities. Temporal DoS vulnerabilities allow adversaries to interrupt network operations or computer systems by altering event timing or protocol behavior. Some examples of such exploits include temporal lensing DoS [29], low- and high-rate DoS [30], and replay [31] attacks.

2. LITERATURE REVIEW

The related works reviewed comprise literature that not only highlights Trust-ND's advantages and criticizes its weaknesses and disadvantages but also those that utilize or derive inspiration from it in their respective works.

Thulasiraman and Wang [32] proposed a lightweight trust-based security architecture to secure routing in a mobile IoT network against Sybil and DoS attacks. The proposed approach relies on node trust values based on Nonce and timestamp to determine the node's trustworthiness, like Trust-ND. They modified the RPL algorithm by adopting some ideas from Trust-ND and using the UTC as the timestamp's reference time per recommendation by Hasbullah et al. [33]. However, they added a network "whitelist" to calculate the node trust values. They also used the average received Signal Strength Indicator (ARSSI) to determine a routing path over a mobile IoT wireless network.

Al-Ani et al. [34] devised a secure technique called Match-Prevention to protect the NDP's duplicate address resolution and address resolution processes from DoS attacks. The proposed technique hides the target IP addresses during address resolution and DAD using a cryptographic mechanism. The authors argued that Trust-ND is vulnerable to hash collision due to its dependency on SHA-1, therefore replacing it with a newer and more secure SHA-3. Like Trust-ND, it also utilizes the Neighbor Discovery option from RFC4861 [11] to avoid changing the original messaging structure of the NDP, calling it a "match-option" to replace the Trust Option in the Trust-ND protocol. The newly introduced "match-option" is 24-byte with five fields: Type (1 byte), Length (1 byte), Nonce (2 bytes), RIN (4 bytes), and IPhash (16 bytes). RIN holds a random integer between 0 and 32 used by the receiver for verification purposes, and IPhash comprises the Interface ID and RIN hash for NS and NA verification. Missing from the "match-option" is a timestamp.

Rehman and Manickam [35] proposed Secure-DAD to provide security to the DAD process from DoS attacks. The authors attempted to improve Trust-ND's DAD process by introducing a Secure-tag field to NS and NA messages exchanged between hosts. Furthermore, instead of the problematic SHA-1, as highlighted by A. K. Al-Ani et al. [34], they utilized Universal Hashing (UMAC). The authors claimed that Secure-DAD achieved better processing time than Trust-ND.

NDPSec is a security mechanism for NDP, using Ed25519 digital signatures to prevent unauthorized access [36]. It secures RA messages with a public-private key pair, distributing the router's public key via Public Key Infrastructure (PKI) or manual pre-configuration. The authors reported NDPSec achieved a 144% reduction in processing time and 50% less traffic overhead than SEND, and it outperformed Trust-ND and Match-Prevention in resisting NDP-based attacks. However, its processing time and protocol overhead were higher compared to Trust-ND and Match-

Prevention Technique due to differing approaches.

2.1 Critical Review

The transmission of ICMPv6 messages, including NDP, is not encrypted [37]. Moreover, NDP only offers a very basic security measure that relies on rudimentary address scoping strategy [38] to secure link-local networks from ingress threats, leaving the IPv6 link-local network exposed to many internal threats and vulnerabilities [39], [40], malicious attacks by adversaries or agents, and unintentional misconfigurations [25]. Consequently, the IETF recommended two security mechanisms, SEND [13] and IP Security (IPsec), to secure the IPv6 link-local network from insider attacks. However, the bootstrapping problem renders the IPsec ineffective in providing security to IPv6 link-local networks [41].

The ability of Trust-ND to provide comprehensive security to NDP without using complex mechanisms is laudable, provided it can resolve and withstand attacks that target its trust component. The attacks that target trust-based security mechanisms differ from those that target hard security-based ones. Some examples of attacks targeting the trust component include conman, Sybil, bad-mouthing or slander, on-off, and newcomer attacks [42]. In addition, many have proven that SHA-1, the cryptographic hash function utilized by Trust-ND, is susceptible to a collision attack and its more potent variant, a chosen-prefix collision attack [43], [44].

Although Secure-DAD has a better processing time than Trust-ND [45], it has higher Bandwidth Utilization (BU) because the NS message must be sent to all nodes in the same network. In addition, there is considerable processing overhead during verification [39]. Unfortunately, in its attempt to improve the security of Trust-ND, it overlooked investigating the timestamp utilization, resulting in its susceptibility to temporal-based vulnerabilities, such as temporal DoS and replay attacks.

It is clear from the literature [32] that the reliance on ARSSI in determining the routing path meant that the proposed trust-based security architecture is exclusively meant for IoT devices and does not apply to general computing devices. In addition, using UTC as the reference time for the timestamp without a proper verification process does not guarantee a hundred percent success in timestamp verification while ignoring the behavior of heterogeneous computer systems' clocks on the network, as shown empirically by [46]. In addition, it uses a timestamp field format and precision like Trust-ND, which subjected this protocol to the same temporal DoS vulnerabilities.

The Match-Prevention Technique solved the problem of SHA-1 collision by using SHA-3. However, it only secures DAD, leaving other processes susceptible to attacks. Critically, without a timestamp to ensure freshness of NS-match and NS-match messages, it is vulnerable to attacks such as Replay, session hijacking, data tampering, and DoS attacks.

The NDPSec has a higher protocol overhead than the Trust-ND and Match-Prevention Technique due to the need to deploy the digital signature to neighbors. It also has a longer processing time than Trust-ND since it uses a signature-based mechanism, which is slower than the hashing approach, such as those employed by Trust-ND and the Match-Prevention Technique.

However, the most significant weakness of NDPSec is the dependency on a private-public key pair to secure the RA message. This approach has four potential problems: performance, distribution, bootstrapping, and scalability. First, the performance of key pair authentication approach is much slower than a hash-based approach. Second, distributing the router's public key to all hosts in the network typically requires a third-party key distribution mechanism, such as via PKI. Involving a third party goes against the distributed nature of the NDP. Third, a bootstrapping problem occurs when a newly connected host cannot generate its IPv6 address because it cannot reach a key distribution server without an IPv6 address. Lastly, without a distribution mechanism, the keys must be manually pre-configured on all hosts, leading to scalability problems especially for large networks.

The summary of related works and critical reviews are in Table 2, listing the proposed mechanism or technique with its author(s) and the year it was proposed as well as their disadvantages or limitations.

Table 2. Summary of Related Works and Disadvantages or Limitations

Mechanism or Technique, Author(s) (Year)	Proposed Mechanism or Technique	Disadvantages or Limitations
Standard NDP, Narten T. et al. (2007) [11]	• The original standard for NDP, first defined in 1999 in RFC2461 and updated in RFC48 in 2007. • One of the core protocols in IPv6 protocol for link-local communications between local IPv6 nodes.	• Lack of built-in security mechanism, exposing it to address and router spoofing, MitM, and DoS attacks. • It is vulnerable to temporal-based attacks, such as Replay, due to lack of timestamp and/or Nonce to ensure message freshness and uniqueness, respectively.
Trust-ND, Supriyanto (2015) [20]	• Uses a distributed trust-based approach to secure the standard NDP. • Utilize IPv6 header extension feature to add Trust Option to standard NDP packets.	• Reliance on local system time for timestamp exposes it to temporal DoS. • It has not been evaluated against attacks that target the trust component. • SHA-1 is known to be susceptible to collision.
Secure-DAD, Rehman and Manickam (2016) [35]	• Add Secure-tag field to NS and NA messages to mitigate DoS in DAD. • Replaces SHA-1 in Trust-ND with UMAC due to vulnerability to collision.	• It has a higher BU because NS messages must be sent to all nodes in the same network. • It has a considerable processing overhead during verification. • It is also susceptible to temporal-based DoS due to its timestamp.
,Thulasiraman and Wang (2019) [32]	• A trust-based security architecture to secure routing in a mobile IoT network against Sybil and DoS attacks by adapting Trust-ND approach. • Introduce a network "whitelist" to calculate the node trust values.	• The architecture is exclusively meant for IoT devices and does not apply to general computing devices. • Using the same reference time subjected this protocol to the same temporal DoS vulnerabilities at Trust-ND.
Match-Prevention, Al-Ani et al. (2020) [34]	• It hides the target IP addresses during address resolution and DAD using a cryptographic mechanism. • Uses SHA-3 instead of SHA-1.	• It only secures DAD, leaving other processes susceptible to attacks. • Without a timestamp to ensure freshness of NS-match and NS-match messages, it is vulnerable to attacks such as Replay, session hijacking, data tampering, and DoS attacks.
NDPSec, A. K. Al-Ani et al. (2022) [36]	• Uses Ed25519 digital signatures. • Use Public-Key Infrastructure (PKI) architecture.	• It has a higher protocol overhead due to the need to deploy digital signatures to neighbours. • Key pair authentication is slower than hash-based ones. • The use of PKI breaks the distributed nature of NDP.

2.2 Proposed eTrustND

The proposed mechanism aims to secure the IPv6-only link-local networks by preventing temporal DoS vulnerabilities due to Trust-ND's weakness in its design.

The assumptions made in this research are as follows.

- All nodes in the network are using Trust-ND or eTrustND via agents. Operating system kernels typically have built-in support for the IPv6 network stack and interfering with or modifying it is not recommended.

- IPv6 addresses are manually assigned to all hosts' network interfaces since there are no routers and DHCPv6 servers in the testbed.
- The testbed operates as an IPv6-only network. IPv6 network operates in either one of two modes: dual-stack or IPv6-only. A dual-stack network allows communication using IPv4 or IPv6, enabling a host to reach a network time server to set and synchronize its clock. In addition, modern operating systems can automatically set a computer's time zone provided they have Internet access.

The proposed enhancement comprises three stages of (i) Timestamp Formulation; (ii) Enhanced Trust-ND; and (iii) Rule-based Temporal DoS Prevention Mechanism, as shown in Figure 2.

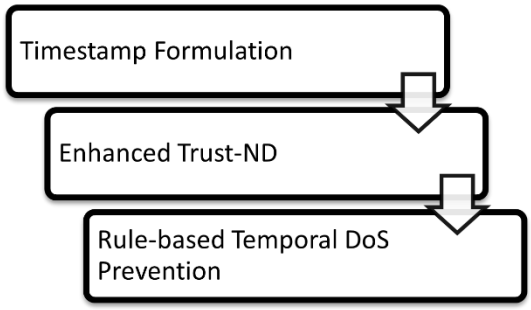


Figure 2. Overview of the Research Stages

2.2.1 Timestamp Formulation Stage

The Timestamp Formulation stage improves the design and use of the existing Trust-ND timestamp to prevent temporal DoS vulnerabilities by refining reference time, time representation, and precision. The outcome of this stage serves as the foundation for improving the Trust-ND messages in the subsequent stage.

Trust-ND's author assumed that all local hosts have the same time zone configured since NDP only operates within a local area network, leading to the selection of the host's local system clock as the timestamp reference. This assumption overlooked cases involving organizations that practice the Bring-Your-Own-Device (BYOD) policy and even users that connect their portable devices from overseas to their IPv6-only home network. Consequently, it exposes Trust-ND to temporal DoS vulnerabilities when the time difference between hosts is substantial.

There are several scenarios where the time difference between hosts is substantial, resulting in temporal DoS vulnerabilities as (i) when users coming from different geographical time zones connect their IPv6 devices to the network; (ii) misconfiguration of computing device's time or time zone; (iii) successful attack on the time synchronization system; (iv) faulty clock subsystem; and (v) clock drift.

It is worth noting that while these issues may not be caused by adversaries directly, they are still exploitable by attackers to cause a DoS by intentionally triggering or amplifying the vulnerability. Therefore, addressing and remediating these vulnerabilities is vital to prevent accidental and intentional DoS attacks.

This stage has three steps that involve changes to the timestamp's reference, format, and precision.

The first step involves replacing the timestamp reference from the device's system clock with Coordinated Universal Time (UTC). The experiment and observation revealed that using the system clock as the timestamp's reference time creates temporal DoS vulnerabilities when there is a substantial difference between the system clocks of Trust-ND nodes. The difference could be due to different time zones or daylight-saving time settings, which is possible when connecting devices from overseas to the link-local network. By using UTC, timestamps remain consistent across all IPv6 nodes, independent of their time zone settings. Furthermore, UTC is unaffected by Daylight Saving Time (DST), which many countries implement to extend daylight hours during summer by advancing clocks an hour in spring and resetting them an hour back in fall. DST adjustments can vary frequently and sometimes occur with little notice [47].

The RFC3339 standard, titled "Date and Time on the Internet: Timestamps" [47], is the basis for using UTC to

replace the system clock as the proposed mechanism's timestamp reference. It states that using UTC as the reference time zone is the best way to achieve true interoperability for date and time formats on the Internet. Most network or Internet protocols use UTC as their timestamp reference, including but not limited to NTP, ICMP/IP, and SEND.

The second step changes the timestamp format to better suit machine operation compared to the existing format. Instead of using a 24-hour time format, which is prone to reset or rollover every 12 or 24 hours, eTrustND's timestamp employs epoch second and sub-second. In computing, an epoch is a specific time from which a computer measures system time. It is usually used as the starting point to measure relative time. Examples of protocols that use epoch time in their timestamp include the NTP, PTP, and SEND. Along with the use of epoch second and sub-second, the timestamp field data type is also changed from Byte to IntField, making the process more efficient.

The third step increases the precision of the timestamp since the current Trust-ND's use of a hexadecimal format to represent the timestamp severely restricts the precision. To represent 24-hour time (e.g., HHMMSSss) in hexadecimal format within a packet structure, a string or char data type is necessary, especially to handle hex digits 0xA through 0xF. Depending on the programming language, a string or char data type may require one or two bytes. When using a language where each char is one byte (8 bits), only the hour and minute components of the timestamp (e.g., 0x17 3B for 23:59, without a colon) can fit within a 32-bit field, providing a precision of just one second. The proposed enhancement extends the timestamp field size from 32 bits to 48 bits, thus increases the precision which eliminates temporal DoS vulnerabilities due to inadequate Trust-ND precision.

Figure 3 and Figure 4 illustrate the original TrustNA packet structure and the proposed eTrustNA packet structure, respectively. TrustNS and eTrustNS packets have similar Trust or eTrust Option structures.

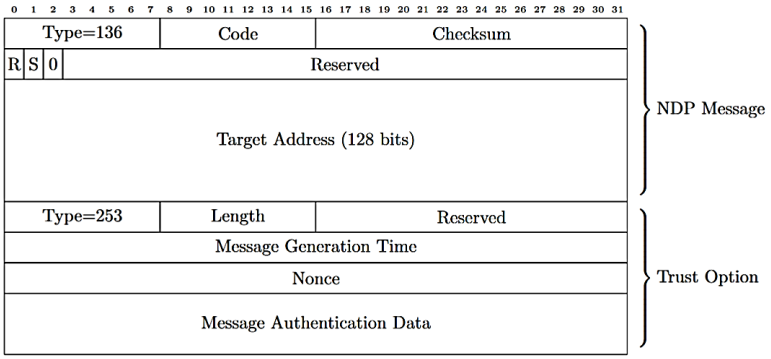


Figure 3. The packet structure of TrustNA

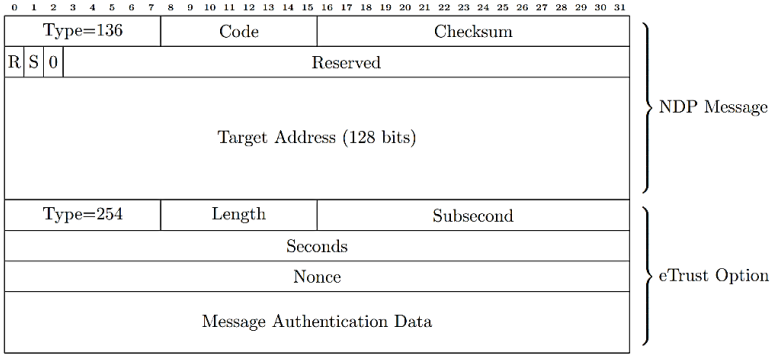


Figure 4. The packet structure of eTrustNA

2.2.2 Enhanced Trust-ND Stage

The Enhanced Trust-ND stage generates eTrustNA and eTrustNS messages, which are the enhanced vanilla Trust-

NA and Trust-NS versions. Enhancement involves utilizing the existing but unused field in the Trust-ND's structure to increase the precision, accommodates the timestamp formulated in the first stage, and changes the Message Generation Time field's data type.

The Message Generation Time field in Trust-ND's Trust Option is four bytes, as shown in Figure 3 for Trust-NA, which is similar for Trust-NS. Unfortunately, it is insufficient to solve the temporal DoS vulnerability due to the lack of precision, as elaborated in the previous section. Increasing the timestamp precision from 32 bits to 48 bits, as formulated previously, requires additional space in the packet. The requirement of preserving Trust-ND packet structure is fulfilled by making use of the existing 16-bit Reserved field for the sub-second component.

2.2.3 Rule-based Temporal DoS Prevention Stage

The Rule-based Temporal DoS Prevention Mechanism stage aims to remediate the vulnerabilities caused by the improper design of the Trust-ND's timestamp and its verification rule. The preventive mechanism is based on several proposed rules to validate the timestamp and handle unsynchronized clocks. The rule-based mechanism is employed on the receiver's side.

Three rules are defined to prevent temporal DoS vulnerabilities in Trust-ND. The receiver uses Rules I and II to ensure the received packets are eTrustNA or eTrustNS messages, respectively. This checking is crucial to ensure the existing Trust-ND messages are not mistaken as eTrustND message since only eTrustND message contains the necessary information to prevent temporal DoS vulnerabilities. The notation of Rule I is as follows.

Rule I
If *ICMPv6 Type = 136 AND eTrust Option Type = 254,* ***then***
    Process *eTrustNA,*
else
    Drop *message,*

Rule I verify the validity of an eTrustNA message based on the ICMPv6's Type and the eTrust Option's Type. The message is an eTrustNA message and processed if the ICMPv6 Type = 136 and eTrust Option Type = 254; otherwise, drop the message. This rule ensures that all NA messages received are eTrustNA messages, which is possible only if the eTrust Option is present.

Meanwhile, the validity of an eTrustNS message can be checked using Rule II, as follows.

Rule II
If *ICMPv6 Type = 135 AND eTrust Option Type = 254,* ***then***
    Process *eTrustNS,*
else
    Drop *message*

Rule II verifies the validity of an eTrustNS message based on the ICMPv6's Type and the eTrust Option's Type. The message is an eTrustNS message and processed if the ICMPv6 Type = 135 and eTrust Option Type = 254; otherwise, drop the message. This rule ensures that all NS messages received are eTrustNS messages, which is possible only if the eTrust Option is present.

After performing the standard verification per RFC 4861 and Rules I and II to ensure the packet's correctness, the mechanism checks Rule III, which aims to tackle the problem of out-of-sync hosts that causes temporal DoS vulnerability in Trust-ND.

Let $T_{diff} = T_r - TS$
Where T_{diff} *is the time difference,* T_r *is Received Time, and TS is the Seconds + Subsecond field values in the eTrustND message.*

Rule III
If $T_{diff} < -\delta$ ***OR*** $Tdiff > +\delta$ ***then***
Drop *eTrustND packet*

else
Process *eTrustND packet*

Where $\pm\delta$ denotes the lower and upper limit of an acceptable time window to receive eTrustND messages. The value of δ is configurable, and this research employs a similar value used by SEND, which is 500 seconds or 3 minutes.

3. RESEARCH METHODOLOGY

NDP is critical for IPv6 network operation, as many vital link-local processes, such as the DAD process, depend on its messages. Therefore, this research uses the DAD process to demonstrate and evaluate the performance of the proposed enhancement to Trust-ND regarding its impact on the computation and bandwidth overheads.

The performance of eTrustND under Normal and DoS scenarios was evaluated in terms of processing time, bandwidth consumption, and DoS prevention success rate, as shown in Figure 5.

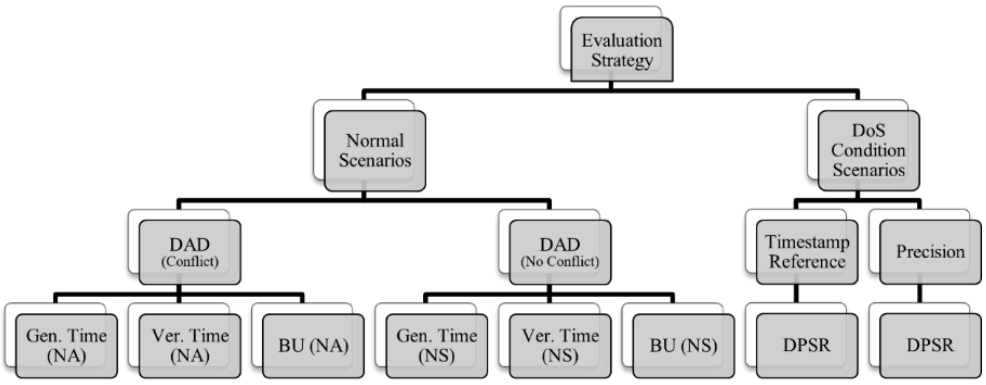


Figure 5. Evaluation Strategy and Experimental Scenarios

The Normal scenario experiments measure the processing time for generating and validating NA and NS packets and calculate the bandwidth consumption of the NA and NS packets. The results will show the impact of the proposed changes to Trust-ND on the host's performance and network BU. Meanwhile, the DoS condition scenarios measure the susceptibility of the mechanism to DoS vulnerabilities.

The evaluation followed the rule of thumb [48] and the recommendation by [49] to ensure adequate coverage and reduce prediction uncertainty due to insufficient sampling by running each scenario 20 times for the Normal scenarios. However, the number of experimental runs under the DoS Condition scenarios is only five since the experiment is deterministic, which will give the same values for all input variables regardless of the number of repetitions [50]. Specifically, the experiments under DoS Condition scenarios are deterministic since they are (i) highly consistent and stable, with little to no variability; (ii) the process is well understood and not subject to unexpected changes; and (iii) the experiment design is well controlled and does not have external variability sources.

3.1 Testbed Topology

The testbed topology used for the experiment comprises two workstations connected to a gigabit switch via Cat-6 UTP cables, as depicted in Figure 6. The testbed does not require a router nor connectivity to the Internet since NDP only involves link-local networks. Furthermore, this research only involves NA and NS messages, not RA and RS.

The two workstations used in the testbed have similar specifications, as shown in Table 3, except for the memory and hard disks' sizes, which should not affect the experiment's outcome.

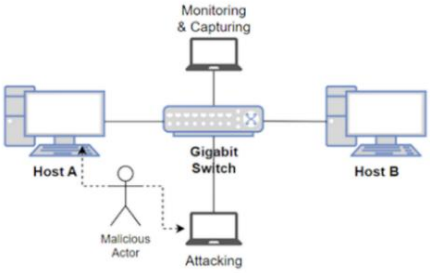


Figure 6. The Topology of the Experimental Testbed

Table 3. Hardware Specification for Testbed

Hardware	Specifications
Workstation (Host A)	Intel® Core™ i5-9400 2.90 GHz, 32GB DDR4 RAM, 128GB NVMe M.2 SSD
Workstation (Host B)	Intel Core i5-9400 2.90 GHz, 16GB DDR4 RAM, 500GB NVMe M.2 SSD
Laptop (Monitor & Capture)	Intel Core i7-8665U 4.8 GHz, 16GB DDR4 RAM, 1TB PCIe NVMe
Laptop (Attacking)	Intel Core i7 16GB DDR4 RAM, 1TB PCIe NVMe
Gigabit Switch	HP® ProCurve™ 2910al-48G-PoE+

3.2 Evaluation Metrics

The outcomes of this study are quantified using three distinct metrics: DAD Processing Time (DPT), BU, and the success rate of DoS prevention (DPSR). The effectiveness of the proposed enhancements to the Trust-ND protocol in thwarting DoS attacks is attainable through the measurement and analysis of these metrics.

a) Total Processing time

The DPT metric measures the time it takes to generate and verify e-TrustNA and e-TrustNS messages for the sender and receiver in an IPv6 link-local network. Comparing the total processing time to complete a DAD process confirms the proposed mechanism's adherence to the research requirement not to pose additional overhead (computation) and its fulfillment of one of the objectives. We expect it will take more time to generate and verify eTrustNA and eTrustNS messages than the standard NS and NA due to the addition of the Trust Option. However, the DPT of eTrustNA and eTrustNS messages should not differ significantly from TrustNA and TrustNS since the structure of both protocols remains the same. If anything, changes to the eTrustND's timestamp field format could slightly improve the validation time of eTrustNA and eTrustNS messages.

Only two processes occur during the DAD process if there is no IP conflict in the link-local network: NS message generation and NS message verification. Therefore, the total processing time for the DAD process without a duplicate IP is the sum of the processing time to generate and verify an NS message.

$$DPT_{NoConf} = t_{NS,g} + t_{NS,v} \quad (4)$$

where,
DPT_{NoConf} is the total processing time for the DAD process without conflict,
$t_{NS,g}$ is the processing time of NS packet generation, and
$t_{NS,v}$ is the processing time of NS packet verification.

However, if there is an IP address conflict, four processes occur between the sender and receiver during the DAD process, involving generating and verifying NS and NA messages. Therefore, the total processing time for the DAD process with a duplicate IP address is the sum of the processing time to generate and verify the NA and NS messages, as follows.

$$DPT_{Conf} = t_{NS,g} + t_{NS,v} + t_{NA,g} + t_{NA,v} \quad (5)$$

where,
DPT_{Conf} is the total processing time for the DAD process with conflict,
$t_{NS,g}$ and $t_{NS,v}$ are the processing times of NS packet generation and verification, respectively, and
$t_{NA,g}$ and $t_{NA,v}$ are the processing times of NA packet generation and verification, respectively.

b) Bandwidth Utilization

The BU, calculated using Equation 6, indicates how much the protocol's messages consume the network's bandwidth.

$$BU = \frac{\sum M_z}{LC} \quad (6)$$

where,
BU is the bandwidth utilization,
$\sum M_z$ is the total ICMPv6 message size, and
LC is the maximum link capacity between Host A and Host B measured using iPerf3, which is 910 Mbit/s in this testbed setup.

c) DoS prevention success rate

The DoS prevention success rate (DPSR), calculated using Equation 7, measures the ability of a mechanism to prevent a DoS attack. It allows us to measure the protocol's performance in detecting and preventing malicious traffic.

$$DPSR = 1 - \frac{F}{N} \quad (7)$$

where,
$DPSR$ is the DoS Prevention Success Rate,
N is the number of messages sent, and
F is the number of failed messages.

Based on the definition of DPSR, a mechanism successfully survives an attack if it is equal to 1. However, a $DPSR$ equal to 0 indicates it failed to prevent attacks. Any value between 0 and 1 indicates partial success in preventing attacks. Therefore, the ability of a security mechanism to prevent DoS can use $DPSR$ in the form of a ratio or percentage.

4. RESULTS AND DISCUSSIONS

This section presents and discusses the experiment results under the Normal and DoS Condition scenarios, as detailed in the previous section. The following subsections present and discuss the resulting processing time for generating and verifying NA and NS messages, DPT, BU, and DPSR.

4.1 Total Processing Time

Figure 7 and Figure 8 illustrate the DAD's total processing time (DPT) for eTrustND, both with and without a duplicate IP address, compared to the standard NDP and Trust-ND protocols.

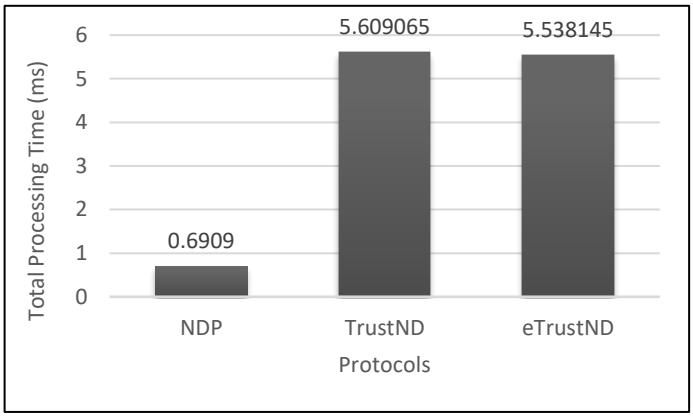


Figure 7. DPT without IP Conflict for the Standard NDP, Trust-ND, and eTrustND Protocols in Milliseconds

It is clear from Figure 7 that the standard NDP has the lowest DPT (0.69 ms) without an IP conflict, followed by eTrustND (5.53 ms) and Trust-ND (5.61 ms). The significant processing time difference between standard NDP and TrustND or eTrustND is due to the hash function operation, which also explains why TrustND and eTrustND have nearly identical processing times.

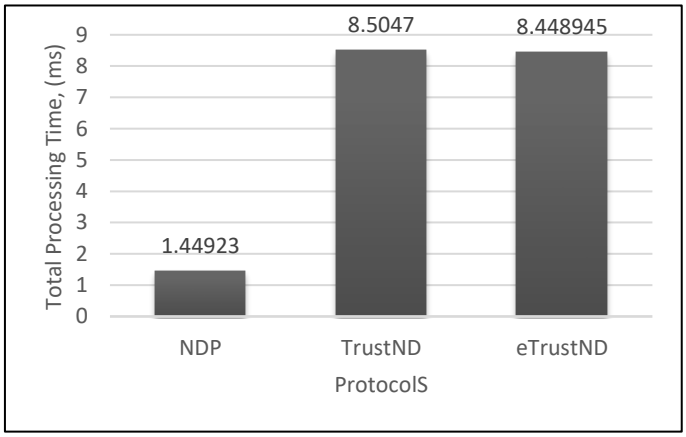


Figure 8. DPT with IP Conflict for the Standard NDP, Trust-ND, and eTrustND Protocols in Milliseconds

Similarly, Figure 8 shows the same trend for DAD when there is an IP conflict. The standard NDP has the lowest DPT (1.45 ms), followed by eTrustND (8.45 ms) and Trust-ND (8.51 ms), as expected.

It is apparent from Figure 7 and Figure 8 that the proposed eTrustND is marginally faster than Trust-ND in executing DAD without and with an IP conflict between 0.071 ms and 0.056 ms, respectively. This improvement is due to the change in timestamp reference, allowing the use of integer epoch second format instead of a 24-hour time in hexadecimal form. Furthermore, using the proposed rules speeds up the verification process due to less logical operation and computation, unlike Trust-ND, which requires additional routines for string conversions. In addition, the Trust-ND verification rules slow down due to the need for more logical operations and computations.

The eTrustND process improves upon Trust-ND more in total processing time without IP conflict (0.071 ms) than with IP conflict (0.056 ms). The difference is due to the fewer processes involved since the DAD (No Conflict) scenario only involves two operations, generating and verifying an eTrustNS message. However, a DAD process in the DAD (Conflict) scenario has four, introducing more computation overhead.

4.2 Bandwidth Utilization

The DAD process, when there is no IP address conflict, consists of a solitary NS message. Consequently, the total

bandwidth used for the DAD process equates to that consumed by a single NS message: 86 bytes per DAD process for the standard NDP and 118 bytes for both Trust-ND and eTrustND, as indicated in Table 4.

Table 4. Traffic Overhead of Standard NDP, Trust-ND, and eTrustND for DAD Without and With IP Address Conflict in Bytes

Mechanism	Without Conflict		With Conflict	
	Total Message Size (bytes)	**Traffic Overhead (bytes)**	**Total Message Size (bytes)**	**Traffic Overhead (bytes)**
Standard NDP	86	0	172	0
Trust-ND	118	32	236	64
eTrustND	118	32	236	64

Table 4 shows the bandwidth used by standard NDP, Trust-ND, and eTrustND during a DAD process with an IP address conflict, which involves exchanging a pair of NS-NA messages. Therefore, the bandwidth consumed for the DAD process is the total NS message and NA message sizes, which is 172 bytes per DAD process for the standard NDP and 236 bytes per DAD process for both Trust-ND and eTrustND, as shown in Table 4. Trust-ND and eTrustND have a 64-byte traffic overhead compared to the standard NDP, which is the size of the Trust Option in each NS and NA message.

Once we have established that eTrustND has the same packet size as Trust-ND, we can calculate its protocol overhead without and with conflict using Equations 4 and 5, respectively, and Equation 6 for the BU, yielding the result in Table 5.

Table 5. Protocol Overhead and BU for a DAD Process with and without Conflict for eTrustND

Number of Hosts	DAD Without Conflict			DAD With Conflict		
	(byte)	**(Mbit)**	**BU (%)**	**(byte)**	**Mbit**	**BU (%)**
1	118	0.000944	1.04E-06	236	0.001888	2.07E-06
2	236	0.001888	2.07E-06	472	0.003776	4.15E-06
3	354	0.002832	3.11E-06	708	0.005664	6.22E-06
4	472	0.003776	4.15E-06	944	0.007552	8.30E-06
5	590	0.004720	5.12E-06	1180	0.009440	1.04E-05

Table 5 shows that the protocol overhead for a single DAD process without conflict is 118 and 236 bytes for a DAD process with conflict. The BUs for a single DAD process without and with conflict are just 1.04×10-6 % and 2.07×10-6 %, respectively, which are minuscule to the point of being negligible. It is apparent from Table 4 that there is a linear correlation between the number of hosts and packet size, which means the increased packet size leads to higher BU.

4.3 Temporal DoS Prevention Success Rate

This section discusses the experiment results for the DoS condition scenarios, comprising the resulting Temporal DoS Prevention Success Rate related to the timestamp reference and precision.

4.3.1 Timestamp Reference

This section presents the result of the ground truth experiment to prove the susceptibility of Trust-ND to temporal

DoS vulnerability due to the simple rule (Equation (2)) used for timestamp verification. The result substantiates the claim that the default timestamp verification rule in Trust-ND resulted in temporal DoS vulnerabilities. Additionally, the result proves the proposed rules' robustness in dealing with the hosts' time zone differences that could cause temporal DoS vulnerabilities in Trust-ND. Tables 6 and 7 show the experiment results measuring Trust-ND's susceptibility to temporal DoS vulnerability in verifying Trust-NA and Trust-NS timestamps, respectively.

Table 6 shows that the local system time of the sender (Host A) with MYT time zone and receiver (Host B) with JST time zone differs by one hour, resulting in Trust-NA message verification failure. The results show that the Trust-NA timestamp verification failed when the hosts' local time differed.

Table 6. Local Time of Sender in JST Time Zone (UTC+9) and Receiver in MYT Time Zone (UTC+8) for Trust-NA Messages

Runs	Sender (Host A)		Receiver (Host B)		Verification ($T_r > T_s$)
	Local Time	Timestamp (T_s)	Local Time (T_r)	Diff ($T_r - T_s$)	
1	16:49:49.314	16:49	15:49:49.830	-59:10.170	Fail
2	16:50:12.947	16:50	15:50:13.461	-59:46.539	Fail
3	16:51:01.570	16:51	15:51:02.084	-59:57.916	Fail
4	16:51:57.198	16:51	15:51:57.712	-59:02.288	Fail
5	16:52:36.834	16:52	15:52:37.350	-59:22.650	Fail

Similarly, Table 7 shows that the local system time of the sender (Host A) and receiver (Host B) differ by one hour, resulting in Trust-NS message verification failure. The results show that the Trust-ND timestamp verification rule for Trust-NS messages failed when the hosts had different time zones configured.

Table 7. Local Time of Sender in JST Time Zone (UTC+9) and Receiver in MYT Time Zone (UTC+8) for Trust-NS Messages

Runs	Sender (Host A)		Receiver (Host B)		Verification ($T_r > T_s$)
	Local Time	Timestamp (T_s)	Local Time (T_r)	Diff ($T_r - T_s$)	
1	16:53:59.245	16:53	15:53:59.764	-59:00.236	Fail
2	16:54:37.867	16:54	15:54:38.382	-59:21.618	Fail
3	16:54:59.488	16:54	15:55:00.003	-58:59.997	Fail
4	16:55:37.118	16:55	15:55:37.633	-59:22.367	Fail
5	16:56:57.753	16:56	15:56:58.267	-59:01.744	Fail

* δ = 500 seconds

Meanwhile, Tables 8 and 9 show eTrustND's robustness in dealing with temporal DoS vulnerabilities caused by different hosts' time zones while verifying eTrustNA and eTrustNS messages, respectively.

Table 8 shows that the system's local time of the sender (Host A) and receiver (Host B) differs roughly one hour due to the different time zones. However, the time difference does not affect the eTrustND's timestamp values for the sender and receiver using UTC as the reference. All five runs passed the eTrustNA verification rule even though the hosts' time zones and system times differed substantially.

Like the previous runs with eTrustNA, all five runs passed the eTrustNS verification rule even though the hosts' time zones and local times differed substantially. As shown in 8, the local system time of the sender (Host A) and receiver (Host B) differ by roughly one hour due to the different time zones. However, it does not affect the eTrustND's

timestamp values for the sender and receiver using UTC as the reference.

Table 8. Local Time of Sender in JST Time Zone (UTC+9) and Receiver in MYT Time Zone (UTC+8) for eTrustNA Messages

Runs	Sender (Host A)		Receiver (Host B)		T_{Diff} ($T_r - T_s$)	Verification $-\delta < T_{Diff} < \delta$ *
	Local Time	Timestamp (T_s), second	Local Time	Receive Time (T_r), second		
1	16:58:31.825	1658131111.8234	15:58:32.340	1658131112.3396	0.5162	Pass
2	17:00:01.489	1658131201.4877	16:00:02.004	1658131202.0036	0.5159	Pass
3	17:01:08.111	1658131268.1095	16:01:08.626	1658131268.6256	0.5161	Pass
4	17:02:56.753	1658131376.7520	16:02:57.269	1658131377.2681	0.5161	Pass
5	17:03:26.386	1658131406.3841	16:03:26.901	1658131406.9004	0.5163	Pass

* δ = 500 seconds

Employing Equation (7) on the results from Table 5 to Table 8 yields the DPSR values for Trust-ND and eTrustND protocols, as shown in Equation 8 and 9, respectively. For example, since Trust-ND failed to verify the Trust-NA and Trust-NS timestamps (F = 5) in all five attempts (N = 5), then the DPSR is 0.0 in ratio or 0 % in percentage based on the following calculation as in (8).

$$DSPR = 1 - \frac{F}{N} = 1 - \frac{5}{5} = 1 - 1 = 0 \quad (8)$$

A similar calculation for eTrustND with F = 0 and N = 5 resulted in a DPSR value of 1.0 in ratio or 100 % in percentage.

$$DSPR = 1 - \frac{F}{N} = 1 - \frac{0}{5} = 1 - 0 = 1 \quad (9)$$

Table 9 presents the DPSR for the Trust-ND and eTrustND protocols, illustrating their vulnerability or resistance to temporal DoS threats stemming from hosts' system time discrepancies due to the protocol's timestamp reference and precision.

Table 9. The DPSR of Trust-ND vs. eTrustND under DoS Condition

Protocol	Message	DPSR (%)	
		Timestamp Reference	**Precision**
Trust-ND	Trust-NA	0	0
	Trust-NS	0	0
eTrustND	eTrustNA	100	100
	eTrustNS	100	100

As shown in Table 9, the Trust-ND protocol has 0 % DPSR, indicating it is vulnerable to temporal DoS. Meanwhile, the eTrustND has 100 % DPSR, proving it successfully prevents temporal DoS in this scenario.

4.3.2 Precision

The Trust-ND timestamp precision is primarily bound to the size and format of the Message Generation Time field. Using Python – or other programming languages with 8-bit char data type – to implement the Trust-ND protocol as per specification means a 32-bit field can only accommodate the hour and minute in hexadecimal form, resulting in a precision of a minute.

5. CONCLUSION

In summary, the mechanism proposed in eTrustND enhances Trust-ND to address temporal DoS vulnerabilities in IPv6 link local networks caused by the design issues with the Trust-ND's Message Generation Time field. Future potential research includes studying the causes of DoS vulnerabilities or QoS issues of the Trust-ND protocol other than the Timestamp field, such as the MAD field. Also possible is to investigate the susceptibility of the Trust-ND protocol to attacks that target its soft security aspect or trust component that are different from the typical adversarial security attacks, such as Sybil, bad-mouthing, on-off, newcomer, and conflicting behavior attacks. The exponential rise in the number of low-power and low-computation devices needing lightweight security mechanisms warrants a serious look at an alternative security mechanism to SEND. Since Trust-ND (and eTrustND) has shown promising results in securing NDP messages without extensive computation and high bandwidth overhead, future research into the viability of eTrustND, or its variation, in resource-restricted IoT devices is warranted.

ACKNOWLEDGEMENT

The authors would like to acknowledge all Universiti Sains Malaysia (USM) staff and students, especially Cybersecurity Research Centre (CYRES), Research Creativity and Management Office (RCMO) and University Community Engagement (BJIM) staff, and those working under Intelligent Connected Streetlights (ICS) research project for their full support, resulting in the publication of this paper, as well as the anonymous reviewers for the meticulous review of this manuscript.

FUNDING STATEMENT

This work is funded by Renesas-Universiti Sains Malaysia (USM) industry matching grant as per MoA#A2021098 agreement with grant account no [7304.PNAV.6501256.R128], and [No: 1001/PNAV/811298].

AUTHOR CONTRIBUTIONS

Iznan Husainy Hasbullah: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft, Revisions;
Lokman Mohd Fadzil: Methodology, Graphics and Tables, Writing – Final Draft;
Selvakumar Manickam: Proofreading and Structuring;
Supriyanto Praptodiyono: Evaluation, Concluding, Supervision – Review & Editing;
Mohamad Khairi Bin Ishak: Article Processing Charge (APC) Payment.

CONFLICT OF INTERESTS

No conflict of interest was disclosed.

ETHICS STATEMENTS

No statements to be disclosed.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," Jul. 2017. doi: 10.17487/RFC8200.

[2] C. Zhiruo Liu *et al.*, "IPv6 Best Practices, Benefits, Transition Challenges and the Way Forward," 2020, *ESTI, Sophia Antipolis CEDEX*. Accessed: Sep. 19, 2021. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_WP35_IPv6_Best_Practices_Benefits_Transition_Challenges_and_the_Way_Forward.pdf

[3] A. Santhanam and R. Aswani, "Introducing IPv6-only subnets and EC2 instances," Networking & Content Delivery. Accessed: Aug. 04, 2022. [Online]. Available: https://aws.amazon.com/blogs/networking-and-content-delivery/introducing-ipv6-only-subnets-and-ec2-instances/

[4] H. Babiker, I. Nikolova, and K. K. Chittimaneni, "Deploying IPv6 in the Google Enterprise Network Lessons Learned," in *Proceedings of the 25th International Conference on Large Installation System Administration*, in LISA'11. USA: USENIX Association, 2011, p. 10.

[5] A. Oswal, "An IPv6 Campus of the Future - Cisco Blogs." Accessed: Sep. 21, 2021. [Online]. Available: https://blogs.cisco.com/networking/an-ipv6-campus-of-the-future

[6] V. McKillop, "Microsoft Works Toward IPv6-only Single Stack Network - Team ARIN." Accessed: Sep. 19, 2021. [Online]. Available: https://teamarin.net/2019/04/03/microsoft-works-toward-ipv6-only-single-stack-network/

[7] GSMA Intelligence, "The Mobile Economy 2025," London, 2025. Accessed: Oct. 18, 2025. [Online]. Available: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2025/04/030325-The-Mobile-Economy-2025.pdf

[8] 3GPP, "3GPP TR 21.101: 'Technical Specifications and Technical Reports for a UTRAN-based 3GPP system (Release 8),'" Valbonne, France, Mar. 2009.

[9] HexaBuild Inc, "IPv6 Adoption Report 2020," Pheonix, AZ, 2020. Accessed: Sep. 24, 2021. [Online]. Available: https://hexabuild.io/assets/files/HexaBuild-IPv6-Adoption-Report-2020.pdf

[10] China Mobile Limited, "Investor Relations > Monthly Customer Data." Accessed: Jul. 26, 2022. [Online]. Available: https://www.chinamobileltd.com/en/ir/operation_m.php

[11] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," Sep. 2007, *RFC Editor*. doi: 10.17487/RFC4861.

[12] S. Thomson Narten T. and T. Jinmei, "RFC 4862 IPv6 Stateless Address Autoconfiguration," 2007, [Online]. Available: http://www.rfc-editor.org/info/rfc4862

[13] J. Arkko, J. Kempf, B. Zill, and P. Nikander, "RFC 3971 - Secure neighbor discovery (SEND)," 2005. [Online]. Available: http://www.hjp.at/doc/rfc/rfc3971.html

[14] Y. E. Gelogo, R. D. Caytiles, and B. Park, "Threats and security analysis for enhanced secure neighbor discovery protocol (SEND) of IPv6 NDP security," *International Journal of Control and Automation*, vol. 4, no. 4, pp. 179–184, 2011.

[15] A. Alsa'deh and C. Meinel, "Secure neighbor discovery: Review, challenges, perspectives, and recommendations," *IEEE Security and Privacy*, no. July/August, pp. 26–34, 2012. doi: 10.1109/MSP.2012.27.

[16] G. An, K. Kim, J. Jang, and Y. Jeon, "Analysis of SEND protocol through implementation and simulation," in *2007 International Conference on Convergence Information Technology, ICCIT 2007*, 2007, pp. 670–676. doi:

10.1109/ICCIT.2007.4420336.

[17] O. E. Elejla, M. Anbar, and B. Belaton, "ICMPv6-Based DoS and DDoS Attacks and Defense Mechanisms: Review," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 34, no. 4, pp. 390–407, Jul. 2017, doi: 10.1080/02564602.2016.1192964.

[18] M. Pohl, "Experimentation and evaluation of IPv6 Secure Neighbor Discovery Protocol," Naval Postgraduate School, Monterey, CA, 2007. Accessed: Oct. 03, 2021. [Online]. Available: http://hdl.handle.net/10945/3222

[19] 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*, 2020, doi: 10.1109/access.2020.3022963.

[20] Supriyanto, "Trust-ND: Lightweight And Secured IPv6 Neighbor Discovery Using A Distributed Trust Mechanism," Unpublished PhD thesis, Universiti Sains Malaysia, 2015.

[21] L. Rasmusson and S. Jansson, "Simulated Social Control for Secure Internet Commerce," in *Proceedings of the 1996 Workshop on New Security Paradigms*, in NSPW '96. New York, NY, USA: Association for Computing Machinery, 1996, pp. 18–25. doi: 10.1145/304851.304857.

[22] A. Josang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bled electronic commerce conference*, 2002, pp. 2502–2511.

[23] C. J. Mitchell, "Timestamps and authentication protocols," Surrey, Feb. 2005.

[24] S. Chiu and E. Gamess, "Easy-SEND: A Didactic Implementation of the Secure Neighbor Discovery Protocol for IPv6," in *Proceedings of the World Congress on Engineering and Computer Science 2009*, 2009.

[25] T. Chown and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement," Feb. 2011. doi: 10.17487/RFC6104.

[26] MITRE, "CWE - 2022 CWE Top 25 Most Dangerous Software Weaknesses." Accessed: Nov. 01, 2022. [Online]. Available: https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html#cwe_top_25

[27] M. Handley, E. Rescorla, and IAB, "Internet Denial-of-Service Considerations," Dec. 2006, doi: 10.17487/RFC4732.

[28] S. Ramanauskaite and A. Cenys, "Taxonomy of DoS attacks and their countermeasures," *Open Computer Science*, vol. 1, no. 3, pp. 355–366, Sep. 2011, doi: 10.2478/s13537-011-0024-y.

[29] R. Rasti, M. Murthy, N. Weaver, and V. Paxson, "Temporal lensing and its application in pulsing denial-of-service attacks," *Proc IEEE Symp Secur Priv*, vol. 2015-July, pp. 187–198, Jul. 2015, doi: 10.1109/SP.2015.19.

[30] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," *Pattern Recognit Lett*, vol. 51, pp. 1–7, Jan. 2015, doi: 10.1016/J.PATREC.2014.07.019.

[31] 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," *Arab J Sci Eng*, vol. 44, no. 4, pp. 3745–3763, Apr. 2019, doi: 10.1007/s13369-018-3643-y.

[32] P. Thulasiraman and Y. Wang, "A Lightweight Trust-Based Security Architecture for RPL in Mobile IoT Networks," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2019, pp. 1–6. doi: 10.1109/CCNC.2019.8651846.

[33] I. H. Hasbullah, M. M. Kadhum, Y.-W. Chong, K. Alieyan, A. Osman, and Supriyanto, "Timestamp utilization in Trust-ND mechanism for securing Neighbor Discovery Protocol.," in *14th Annual Conference Privacy, Security & Trust*, Auckland, New Zealand, 2016, pp. 275–281. doi: 10.1109/PST.2016.7906974.

[34] 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, doi: 10.1109/ACCESS.2020.2970787.

[35] S. U. Rehman and S. Manickam, "Novel mechanism to prevent denial of service (DoS) attacks in IPv6 duplicate address detection process," *International Journal of Security and Its Applications*, vol. 10, no. 4, pp. 143–154, 2016.

[36] A. Al-Ani, A. K. Al-Ani, S. A. Laghari, S. Manickam, K. W. Lai, and K. Hasikin, "NDPsec: Neighbor Discovery Protocol Security Mechanism," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3196028.

[37] 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, doi: 10.1166/asl.2017.10272.

[38] B. Haberman, B. Zill, E. Nordmark, T. Jinmei, and Dr. S. E. Deering, "IPv6 Scoped Address Architecture," RFC Editor, Mar. 2005. doi: 10.17487/RFC4007.

[39] 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, doi: 10.1371/JOURNAL.PONE.0214518.

[40] 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, doi: 10.1109/ACCESS.2017.2737524.

[41] Supriyanto, I. H. Hasbullah, R. K. Murugesan, and S. Ramadass, "Survey of Internet Protocol Version 6 Link Local Communication Security Vulnerability and Mitigation Methods," *IETE Technical Review*, vol. 30, no. 1, pp. 64–71, 2013, doi: 10.4103/0256-4602.107341.

[42] W. Fang, W. Zhang, W. Chen, T. Pan, Y. Ni, and Y. Yang, "Trust-Based Attack and Defense in Wireless Sensor Networks: A Survey," *Wirel Commun Mob Comput*, vol. 2020, 2020, doi: 10.1155/2020/2643546.

[43] G. Leurent and T. Peyrin, "From collisions to chosen-prefix collisions application to full SHA-1," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11478 LNCS, pp. 527–555, 2019, doi: 10.1007/978-3-030-17659-4_18/COVER.

[44] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10401 LNCS, pp. 570–596, 2017, doi: 10.1007/978-3-319-63688-7_19/FIGURES/6.

[45] S. U. Rehman and S. Manickam, "Improved Mechanism to Prevent Denial of Service Attack in IPv6 Duplicate Address Detection Process," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, 2017, doi: 10.14569/IJACSA.2017.080209.

[46] F. Buchholz and B. Tjaden, "A Brief Study of Time," in *The Digital Forensic Research Conference*, Pittsburgh, PA: Elsevier Ltd, 2007, pp. 31–42. doi: 10.1016/j.diin.2007.06.004.

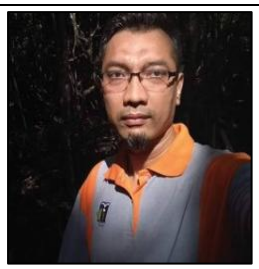
[47] G. Klyne and C. Newman, "Date and Time on the Internet: Timestamps," Jul. 2002. doi: 10.17487/rfc3339.

[48] J. L. Loeppky, J. Sacks, and W. J. Welch, "Choosing the sample size of a computer experiment: A practical guide," *Technometrics*, vol. 51, no. 4, pp. 366–376, Nov. 2009, doi: 10.1198/TECH.2009.08040.

[49] F. M. Hemez and S. Atamturktur, "The dangers of sparse sampling for the quantification of margin and uncertainty," *Reliab Eng Syst Saf*, vol. 96, no. 9, pp. 1220–1231, Sep. 2011, doi: 10.1016/J.RESS.2011.02.015.

[50] D. Bingham, P. Ranjan, and W. J. Welch, "Design of Computer Experiments for Optimization, Estimation of Function Contours, and Related Objectives," in *Statistics in Action: A Canadian Outlook*, 1st ed., Chapman and Hall/CRC, 2014, pp. 109–124.

BIOGRAPHIES OF AUTHORS

	**Iznan Husainy Hasbullah** is a Research Officer at CYRES, Universiti Sains Malaysia, with degrees from Rensselaer Polytechnic Institute in Electrical Engineering and USM in Advanced Computer Networks. He brings extensive industry experience as a developer, CTO, and security auditor. A Certified Network Professional for IPv6 (CNP6) trainer, his research focuses on network security, intrusion detection, and IPv6 protocols. He can be contacted at email: iznan@usm.my.
	**Lokman Mohd Fadzil** is a Senior Lecturer at Cybersecurity Research Centre (CYRES), Universiti Sains Malaysia, with over 20 years of ICT industry experience. He holds a bachelor's degree from University of Wisconsin-Milwaukee, USA, master's degree from Universiti Sains Malaysia (USM), and doctoral degree from Universiti Kuala Lumpur (UniKL). He is a member of IEEE, a certified Professional Engineer (Ir.), a Technologist (Ts.), and an HRDF Trainer (CIT). His research focuses on smart systems integration, computer vision, IoT automation, and cybersecurity in industrial control systems. He can be contacted at email: lokman.mohd.fadzil@usm.my.
	**Selvakumar Manickam** is the Director of the Cybersecurity Research Centre (CYRES) and a Professor at Universiti Sains Malaysia. He specializes in cybersecurity, IoT, cloud computing, and machine learning, with over 220 publications and 18 Ph.D. graduates. A frequent keynote speaker and trainer, he bridges academia and industry through projects in robotic automation, embedded systems, and data analytics. His work spans software development, IPv6, and industrial communication protocols. He can be contacted at email: selva@usm.my.
	**Supriyanto Praptodiyono** is a distinguished academic and the current Dean of the Faculty of Computer Science at Universitas Pembangunan Nasional 'Veteran' Jakarta (UPNVJ). He obtained his bachelor's degree in electrical engineering from Brawijaya University, Indonesia, and master's and doctoral degrees in Advanced Computer Networks from Universiti Sains Malaysia (USM). His research interests include IPv6 protocol and mitigation methods, network security, and computer networks. He can be contacted at email: supriyanto.fik@upnvj.ac.id.
	**Mohamad Khairi Ishak** holds a B.Eng. in Electrical and Electronics Engineering from the International Islamic He can be contacted at email University Malaysia (IIUM), an MSc in Embedded Systems from the University of Essex, and a Ph.D. from the University of Bristol. He is a member of IEEE and a registered graduate engineer with the Board of Engineers Malaysia (BEM). Currently, Dr. Ishak serves as an Associate Professor and Lecturer in Computer Engineering at the Department of Electrical and Computer Engineering, Ajman University, United Arab Emirates. His research interests include embedded systems, real-time control communications, Artificial Intelligence, and the Internet of Things (IoT). His work focuses on both the development of theoretical frameworks and the application of practical methodologies, with a strong emphasis on real-world impact. He can be contacted at email: mishak@sharjah.ac.ae