Replay Attacks in RPL-Based Internet of Things: Survey and Empirical Comparative Study

Hussah Albinali
King Fahd University of Petroleum and Minerals
Farag Azzedin (fazzedin@kfupm.edu.sa)
King Fahd University of Petroleum and Minerals

Research Article

Keywords: Internet of Things, Routing Protocol for Low-Power and Lossy Networks, RPL, Replay attacks, Routing, Survey

Posted Date: December 5th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3697695/v1

Additional Declarations: No competing interests reported.

Replay Attacks in RPL-Based Internet of Things: Survey and Empirical Comparative Study

Hussah Albinali [1] Farag Azzedin [2]

Abstract– RPL was developed as a routing protocol in low-power and lossy network contexts to connect many applications using IP-based communication. However, RPL has been subjected to several attacks, including replay attacks. Replay attacks pose significant challenges, as any node can initiate the attack by replaying control messages. These messages play a vital role in establishing and sustaining network topology. However, studies that discuss replay attacks are severely limited. To address this issue, we conducted a comprehensive study on replay attack forms and their impact on RPL networks. Our study includes the latest security countermeasures. According to the literature, most RPL replay attacks rely on replaying DIO messages, leading to neighbor, DIO suppression, or copycat attacks. DAO messages are also utilized to launch a replay attack. Nevertheless, the literature lacks any study that analyzes DAO replay attacks. To the best of our knowledge, this is the first study that evaluates DAO replay attacks and includes the route table falsification attack. We conduct a comparative study of these attacks and empirically investigate their impact on networks through extensive evaluation experiments. Our findings verify the harm of replay attacks in terms of packet delivery and latency. The average local delivery ratio dropped to less than 60% under DIO suppression and copycat attacks, and the communication latency increased by 50% under neighbor attacks. These results confirm the threat of replay attacks on RPL networks and the need to design countermeasures to mitigate them.

***Index Terms*—Internet of Things; Routing Protocol for Low-Power and Lossy Networks; RPL; Replay attacks; Routing; Survey**

I. Introduction

The urban development vision to ensure a better quality of life leads to the construction of infrastructure and integration of Internet of Things (IoT) in managing assets, including healthcare centers, vehicles, energy utilities, and other community services. To monitor the evolution and control these assets, a huge investment is promoted using different IoT devices, including sensors. Sensors connected with real-time monitoring systems are projected to open new opportunities for automated services and cutting-edge applications for municipal users where data is gathered from various devices, processed, and evaluated [1]. According to S. OĎea [2], the number of short-range IoT devices is expected to grow to around 22.4 billion by 2027.

However, these sensors have limitations in computing, storage, energy, and communication capacity [3], [4]. Such networks suffer from many vulnerabilities, including low data rates and packet delivery failure [5], [6]. In fact, Low Power and Lossy Network (LLN) is the name given to a network that consists of many constrained nodes with limited processing power and memory. The nature of LLN makes the existing routing protocols not suitable for IoT. As a result, IETF introduced IPv6 routing protocol over LLNs (RPL). RPL became the standard routing protocol in IoT networks since it was designed to efficiently utilize the constraint resources while providing effective routing service [7].

As smart applications collect sensitive data, it is important to consider security and privacy issues at different levels of the architecture while designing and implementing these applications [8]. One of these security concerns is securing data transmission between IoT devices by ensuring confidentiality, authentication, and integrity in data routing [9]–[11].

As such, many researchers have examined IoT security challenges [9], [12]–[15]. According to the literature, Raoof *et al.* [13] and Mayzaud *et al.* in [15] illustrated more than 15 types of RPL attacks. For instance, in [13], authors distinguished Wireless Sensor Networks (WSN) inherited attacks from those attacks that utilize RPL vulnerabilities. WSN-inherited attacks include blackhole; selective forwarding; sinkhole; wormhole; hello flood; sybil; and clone-ID attacks. In comparison, RPL-specific attacks include rank; version; local repair; DIS flooding; neighbor; routing table overload; route table falsification; DODAG inconsistency; and replay attacks [13].

Several of these mentioned attacks misuse control messages by replaying them to accomplish their assaults, including neighbor and route table falsification attacks. In addition, several studies have emphasized that RPL replay attacks can be launched by any control message [9], [13], [14]. Remarkably, the number of studies that tackle replay attacks is limited. Table I shows a summary of the studies that implemented replay attacks in RPL and their contributions. According to the systematic literature review by Al-Amiedy *et al.* [16], replay attacks (including neighbor attacks as a form of replay attack) have only been looked at in seven studies among 127 selected references.

In addition, the reported impacts of replay attacks vary between limited impacts by causing additional delay and utilizing suboptimal routes [15], [25], [26] to severe impacts such as network partitioning [9], [13], [17].

Consequently, no comprehensive study has been performed to analyze and evaluate RPL replay attacks. The lack of investigating RPL replay attack forms and their impacts motivated us to conduct a study that thoroughly evaluates

[1] Information & Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, g201906710@kfupm.edu.sa; Networks and Communication Department, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia, halbinali@iau.edu.sa

[2] **Corresponding Author**. Information & Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, fazzedin@kfupm.edu.sa; Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals

TABLE I: Comparison between references addressing RPL replay attacks.

Reference	# Attacks Examined	Attack Impact				Survey	Impact Analysis	Proposed Taxonomy		Countermeasures
		Delivery	Energy	Delay	Overhead			Attack	Countermeasure	
[17]	1	✓	-	-	-	-	-	-	-	✓
[18]	4	✓	✓	✓	✓	-	-	-	-	✓
[19]	1	✓	✓	-	✓	-	Scalability	-	-	✓
[20]	2	✓	✓	✓	✓	-	-	-	-	✓
[21]	2	✓	✓	✓	✓	-	-	-	-	✓
[22]	1	✓	-	-	-	-	Packet transmission rate	-	-	✓
[23]	2	✓	✓	✓	-	-	-	-	-	✓
[24]	1	✓	-	✓	-	-	-	-	-	✓
Our Paper	5	✓	✓	✓	✓	✓	Network topology, routing selection, control overhead	✓	✓	✓

and analyzes RPL replay attacks and presents state-of-the-art security countermeasures.

In general, replay attacks are one of the threatening challenges to security protocols that have been discussed for quite some time in the literature [11], [13], [27]–[29]. A replay attack deceives the honest participant(s) into believing the protocol run has been successfully completed by replaying messages in a context that differs from the intended context [30]–[32]. In fact, replay attacks resist cryptographic communications and are particularly effective in the absence of data origin authentication schemes [9]. In replay attacks, attackers can capture legitimate messages and then forward them to other nodes as if they came from the original sender nodes [3]. In RPL, replay attacks are critical because of several reasons. First, the vital role of control messages in RPL is constructing and maintaining network topology. Therefore, the consequences of illegitimate replaying of these control messages can be severe in the network [14], [17], [18], [23]. Second, the ease of launching a replay attack in RPL networks increases the threat. A node within the network communication range can launch this attack simply by obtaining and replaying any control message regardless of whether the message is clear or encrypted [9], [18], [33].

Accordingly, this article aims to answer the following research questions: Which type of control messages can be used to launch a replay attack (RQ1)? How to implement replay attacks against RPL protocol (RQ2)? What is the impact of each replay attack on the network (RQ3)? What are the security solutions proposed to mitigate replay attacks (RQ4)?

By answering these questions, we expose the signature/profile of each RPL replay attack. An attack signature is best thought of as “fingerprint” including set of unique data and bits of code that allow an attack to be identified and hence detected. As such, our findings provide the main information resource to design signature-based detection systems. In addition, our empirical evaluation demonstrates the impact of these attacks on network performance. Our evaluation examined network parameters such as packet delivery ratio, end-to-end delay, control messages overhead, and power consumption. The irregularity in these parameters helps to design anomaly-based detection systems and monitor network behavior to identify RPL traffic abnormality.

As such, the contributions of this article are as follows: (a) performing an extensive survey and analysis of RPL replay attacks and countermeasures, (b) proposing taxonomy for RPL replay attacks, (c) analyzing RPL replay attacks’ impact from topology perspective, (d) conducting the first implementation and evaluation of destination advertisement object (DAO) replay attack and route table falsification attack, and (e) providing empirical evaluation to measure the performance of RPL replay attacks considering different topologies.

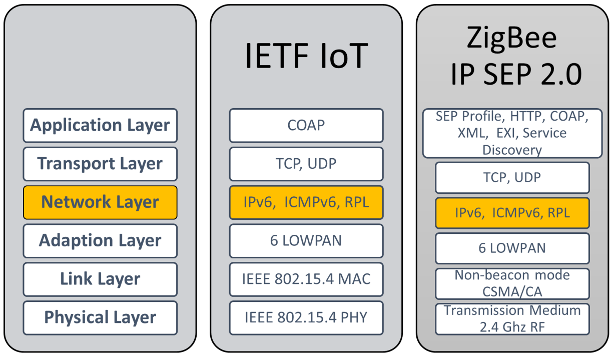
The rest of this article is organized as follows: For clarity and completeness purposes, Section II provides an overview of RPL protocol, including the traffic flow, modes of operations, topology construction, RPL control messages, trickle algorithm, and RPL secure modes. We provide a detailed analysis of RPL replay attacks in Section III, while Section IV introduces our proposed RPL replay attacks taxonomy. Section V surveys and classifies mitigation techniques proposed in the literature to countermeasure replay attacks. Sections VI and VII provide an empirical evaluation to compare the impact of these forms of RPL replay attacks and discuss their performance. Section VIII provides a discussion and suggestions for future research directions. Finally, Section IX concludes the manuscript and envisions new directions.

II. RPL Background

The usage of IoT applications continues to expand rapidly as an effective solution to the challenges faced in smart cities, smart industry, earthquake warning systems, and more [10], [34]. Because IoT encompasses wide range of objects, there is no singular architecture for it. However, researchers have proposed various architectures based on different domains to support its diverse needs, as reported in [35], [36]. According to most researchers, the IoT architecture is composed of three layers, perception, network, and application [35], [36]. The perception layer is the physical layer that consists of environmental information sensors that detect real-time changes in the physical state of connected objects. It includes sensors that measure the physical environment, identify and locate intelligent objects, collect data, and transmit data to the network layer for processing and storage [36]. The network layer plays a crucial role in facilitating network traffic and processing sensor data. Data collected by the perception layer is directly transmitted to the cloud. This layer consists of routers, switches, gateways, and servers [35]. Finally, the application layer is responsible for providing users with specific services such as data collection, analysis, visualization, and security. This layer also defines different IoT applications including smart homes, smart cities, and intelligent healthcare [37].

WSNs play a vital role in integrating with IoT systems for routing and sensing. These networks can be easily deployed to provide a wide range of services and applications [10], [38]. WSN consists of a limited number of sensor nodes (motes) that are managed by a multi-layered protocol organization. The IEEE 802.15.4 protocol is the most widely used communication system for WSNs due to its low power consumption, cost effectiveness, and the ability to defy physical and connecting layers for wireless short-range transmissions [36], [39]. It operates on 800/900 MHz and 2.4 GHz ISM frequency bands and is the foundation for other standards such as ZigBee, Wireless Hart, WIA-PA, and ISA.100.11a [36].

RPL is IPv6 routing protocol for LLN designed by IETF routing over LLN (ROLL) group as a proposed standard. Given the significant overlapping between LLNs and IoT, and the fact that IPv6 is an essential feature in IoT environments, RPL has rapidly become the routing protocol for IoT. As illustrated in Figure 1 (a), IETF places RPL at the network layer [40]. The success of RPL as an IoT standard is also witnessed by companies part of ZigBee Alliance [39], [41]. These companies utilize industry-standard protocols, including IPv6, 6LoWPAN, RPL, and TCP/UDP/IP, to deliver end-to-end IPv6 packets without the requirement of intermediary gateways. As such, ZigBee IP has adopted RPL to easily plug their networks into the IP-based Internet, which was impeding a concrete IoT, as shown in Figure 1 (b).



	IETF IoT	ZigBee IP SEP 2.0
Application Layer	COAP	SEP Profile, HTTP, COAP, XML, EXI, Service Discovery
Transport Layer	TCP, UDP	TCP, UDP
Network Layer	IPv6, ICMPv6, RPL	IPv6, ICMPv6, RPL
Adaption Layer	6 LOWPAN	6 LOWPAN
Link Layer	IEEE 802.15.4 MAC	Non-beacon mode CSMA/CA
Physical Layer	IEEE 802.15.4 PHY	Transmission Medium 2.4 Ghz RF

Fig. 1: Protocol Stacks: (a) IoT IETF, (b) ZigBee IP SEP 2.0.

The RPL design principle is to construct a tree-like topology called directed acyclic graph (DAG) that has single destination called (DODAG) [42]. This hierarchical design has the advantage of preventing network traffic loops [43]. Each node in the DODAG has a rank that indicating the cost to reach the root; typically, nodes closer to the root have a lower rank than nodes farther away [44]. RPL uses an objective function to calculate the rank of network nodes [44]. To determine this cost, the objective function uses different metrics such as energy consumption, hop count, or quality of the proposed paths [43].

For clarity and completeness purposes, we provide an overview of RPL protocol. The overview covers RPL traffic flow and modes of operation, topology formation, RPL control messages, trickle timers, and, finally, RPL security.

A. Traffic Flows in RPL and Modes of Operations

RPL supports various traffic directions; the first and most vital direction is upward routing or multi-point-to-point (MP2P) [44]. In this direction, the packets are sent by network nodes towards the root [45]. The second supported RPL flow is routing downward or point-to-multi-point (P2MP), which is from the root to any node on the network [44], [45]. The last RPL flow is point-to-point (P2P) routing which refers to sending packets between any nodes in the DODAG. This flow is achieved by sending the packet upward to the nearest common ancestor (or to the root in non-storing mode) and then downward to the destination node. This is done to enable traffic from a point to another point in the DODAG. A root must be able to direct packets to a specific location in the network to facilitate P2P activity. Routing tables to destinations may also exist on network nodes. Until it encounters an ancestor with a known path to the destination, a packet flows in the direction of the root. That common ancestor might be the DODAG root in the most limited scenario (where nodes cannot hold routes) [45].

RPL supports two modes of operation, namely, storing and non-storing modes [45]. In the storing mode, RPL keeps a downward routing table at each node. The routing traffic between two different nodes travels only as far as a common parent [44]. The storing mode has a limitation related to the size of the routing table. Nodes with lower ranks have larger routing tables. In addition, RPL fails when the routing table is full, and a routing entry needs to be appended [46]. In the non-storing mode, all traffic is sent to the root [47]. The root uses source routes to send traffic to leaf nodes [45]. There are many challenges in this mode, including limited memory of DODAG root to handle all network nodes. Another challenge is the overhead, as it requires more compute cycles [44].

B. Topology Construction

The DODAG construction is shown in Figure 2. The root node broadcasts its information using the DODAG information object (DIO) message. Nodes within the communication range of the root will receive the sent DIO message. Typically, when a node receives a DIO message, it evaluates the routing information, such as RPL instance, version number, object function, and mode of operation that represents the network information. DIO messages also carry information about the sender, including node ID and node rank. Therefore, any node should add its routing information including the rank before sending DIO messages [44]–[46].

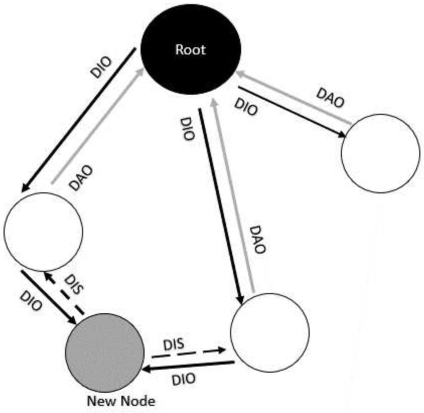


Fig. 2: The DODAG construction mechanism.

To join a DODAG, a new node sends the DAG information solicitation (DIS) message. Nodes within the communication range of the new node reply with a DIO message carrying node ID, objective function, and node rank [44]. Until the new node receives a DIO message from one of its neighbor nodes, it constantly broadcasts DIS messages at a set interval. This interval may vary with different RPL implementations. The new node stops sending DIS messages after receiving DIO messages from neighbor nodes and starts to view all senders as prospective parents. Alternatively, a new node can wait to receive DIO messages from its neighbors without sending a DIS message. The period between two consecutive DIO messages is dynamic, and the trickle timer determines this interval [48].

When a joining node receives a DIO message, it calculates its rank by considering a given objective function that aims to optimize energy consumption, hop count, or quality of the proposed paths [44]. The main purpose of the objective function is to determine the rank of each node within the DODAG. Therefore, the root node is the sink with the minimum rank [45]. Additionally, the joining node prioritizes the nearby nodes as prospective parents in an ordered list [47]. In DODAG, each node selects the preferred parent, which is the node that offers the lowest cost or the minimum rank for this node [44], [45], [47]. The IETF has officially defined two objective functions: Objective Function Zero (OF0), which considers hop count as the routing metric [49], and the Minimum Rank with Hysteresis Objective Function (MRHOF), which uses paths that minimize Expected transmission count (ETX) as a metric [50]. Based on the objective function and the rank of the sending node, nodes decide whether to join this DODAG [43].

Depending on RPL's mode of operation, the joining node may participate in downward routing by sending a destination advertisement object (DAO) message to its preferred parent or to the DODAG root [44]. In storing mode, each node maintains a routing table that maps all reachable destinations in its sub-DODAG to their corresponding next-hop nodes, as discovered when receiving DAOs. In non-storing mode, the DAO is delivered directly to the root. When the root receives the DAO, it adds the node to its routing table and stores the parent-child relationship, which is later utilized for source routing. The sender of the DAO message may optionally request their destination to send an acknowledgment. The destination advertisement object-acknowledgment (DAO-ACK) message is sent back to the DAO sender.

C. RPL Control Messages

RPL provides the following four primary types of control messages.

- *DODAG information solicitation* (DIS): This message is employed to request DIO from an RPL node. Usually, a node probes neighbor nodes in nearby DODAGs using a DIS packet. The structure of the DIS message is shown in Figure 3.
- *DODAG information object* (DIO): This message contains data that enables a node to locate an RPL instance, learn about its configuration settings, choose a DODAG parent set, and keep the DODAG up to date. Specifically, DIO packets are used to construct MP2P routing paths as well as assist new nodes in finding neighboring DODAG [44], [51]. Figure 4 illustrates the structure of DIO messages. The version number denotes the DODAG version number and keeps all nodes in sync. This number normally increases upon each network information update. The rank details the node's rank that sent the DIO message. The MOP determines the RPL's mode of operation. The DODAG-ID is a unique value for each DODAG identified specifically by a DODAG root.

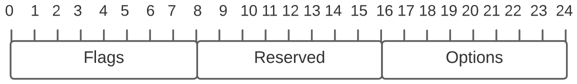


Fig. 3: The structure of DIS message.

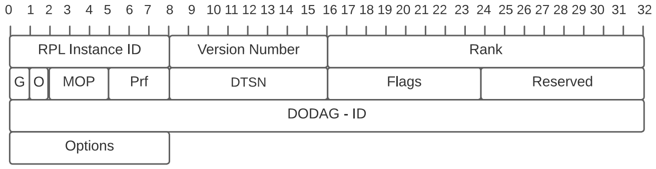


Fig. 4: The structure of DIO message.

- *Destination advertising object* (DAO): This message is utilized to spread destination information upwards. In storing mode, the DAO is sent as a unicast message to specific parents while DAO is sent as a unicast message to only the DODAG root in non-storing mode. RPL facilitates P2MP traffic by relaying on DAO packets [44], [51]. As modifications are made to the underlying DODAG topology, destination advertisements may update routing tables [44]. When the root sends messages to a descendant node, it will utilize the routing table based on the received DAO messages [43]. The structure of DAO message is shown in Figure 5.

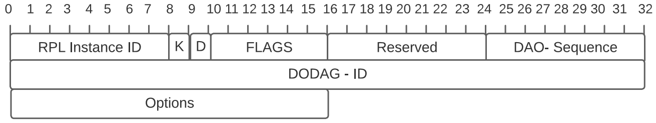


Fig. 5: The structure of DAO message.

- *Destination Advertisement Object-Acknowledgement* (DAO-ACK): In response to a unicast DAO message, a DAO recipient (a DAO parent or DODAG root) sends the DAO-ACK message as a unicast packet [44], [45]. The structure of DAO-ACK message is shown in Figure 6.

D. Trickle Algorithm

In RPL, network nodes frequently announce their routing information using DIO messages. The frequency between

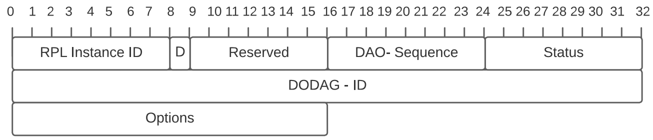


Fig. 6: The structure of DAO-ACK message.

these messages is dynamic, and it is determined by the trickle algorithm [48]. The trickle algorithm is designed to minimize sending redundant messages, which is achieved by checking the consistency in received DIO messages. As a result, the trickle timer allows nodes to exchange routing information only if an inconsistency is detected. In order to send minimum redundant messages when DODAG exhibits inconsistency, the trickle algorithm sends DIO messages in short intervals. However, when nodes receive consistent information about their DODAG, the trickle algorithm increases the interval between transmissions until a preset maximum interval is reached. This mechanism is implanted using three parameters [48]: minimum and maximum interval times and a consistency threshold. The minimum and maximum intervals define the shortest and longest time between sending consecutive DIO messages. The consistency threshold determines the number of consistent messages before forcing the node to suppress its DIO information. DODAG root determines the values of all these parameters [48], [52]. Accordingly, when a node detects DODAG inconsistency, such as a change in a neighbor rank, the trickle algorithm resets to the minimum interval between two DIO messages. In comparison, when the network is stable, the time between two DIO messages exponentially grows until it hits the maximum set interval [52].

E. *RPL Secure Modes*

To ensure the security of RPL messages in terms of confidentiality and integrity, IETF proposed secure modes for RPL. The proposed secure modes are the pre-installed secure mode and the authenticated mode [44], [45]. The pre-installed mode allows nodes to join the DODAG, process, create, and exchange RPL messages only if they have a shared key. Similar to the pre-installed mode, nodes in the authenticated mode can join the DODAG as leaf nodes by utilizing the pre-installed shared keys. However, nodes must get a key from a centralized authentication authority to take part in the forwarding process [44], [45].

Secure modes provide optional services such as encryption using AES128 to ensure confidentiality for RPL messages. Another optional service is replay protection using a consistency check control message [44], [45]. This message is a secure control message with a non-repetitive nonce value [18], [45]. The main limitations regarding these secure modes are that there are no official implementations yet for them [18], [53], [54]. In addition, symmetric encryption algorithms are unable to provide an authenticated mode where RPL only supports symmetric encryption [45].

III. Analysis of RPL Replay Attacks

Multiple RPL replay attacks can be launched based on the type of control message being replayed, the modification performed before replaying the message, and the frequency of replaying the message. Figure 7 summarizes the mechanism of these various attacks and answers the first research question regarding the types of control messages used to launch replay attacks. First, the DIS message, this message is sent by an external node probing the network, and it does not contain information about the sender. Therefore, replaying this message by any other node will not impact the network. The second message is a DIO message which can be used to launch multiple RPL replay attacks. When the attacker node replays the exact received DIO message without adding its IP address or routing information, such as the rank value, it performs a neighbor attack [14], [25], [55]. However, when the attacker frequently sends the received message, the attack becomes a DIO suppression attack [14], [17]. On the other hand, if the attacker changes the IP address of the received message to include its IP address while keeping the routing information of the received node and sending the message frequently, this attack is called a copycat attack [20], [56]. The DAO message is another control packet used to launch a replay attack. In a DAO replay attack, the attacker replays the received DAO message to a neighbor node in the DODAG [13], [57]. Another attack based on replaying DAO messages is a route table falsification attack. Here, the attacker modifies the DAO message by changing the source's IP address and then replays it to its parent [13], [15].

In addition, we determine the impact of replay attacks by first considering used control messages and listing different attacks that might occur. Based on the mentioned consequences of these attacks according to the literature, we analyze and classify the impact level as shown in Table II. The attack has low impact when it only leads to additional exchanges of control messages without impacting network topology. For moderate impact, the attacker node destructs the optimal structure of DODAG by driving network nodes to use sub-optimized paths or deceiving them into considering an out-of-range node as a neighbor. When the attack has a critical impact, it falsifies the network by either announcing a false or out-of-range node to add non-existing paths or by isolating some nodes from the DODAG.

TABLE II: Impact levels description table.

Impact Magnitude	Impact Description
Low	Performing the attack leads to additional exchanges of control messages without impacting network topology
Moderate	Performing the attack leads to using sub-optimized routing paths or adding an out-of-range node as neighbor
Critical	Performing the attack leads to adding non-existing routing path to DODAG or isolating node(s) from DODAG

A. *Attacks Resulting from Replaying DIO*

The DIO control message is the packet that has the most information about the network and the sender. In addition, it is the message responsible for constructing the upward path [44],

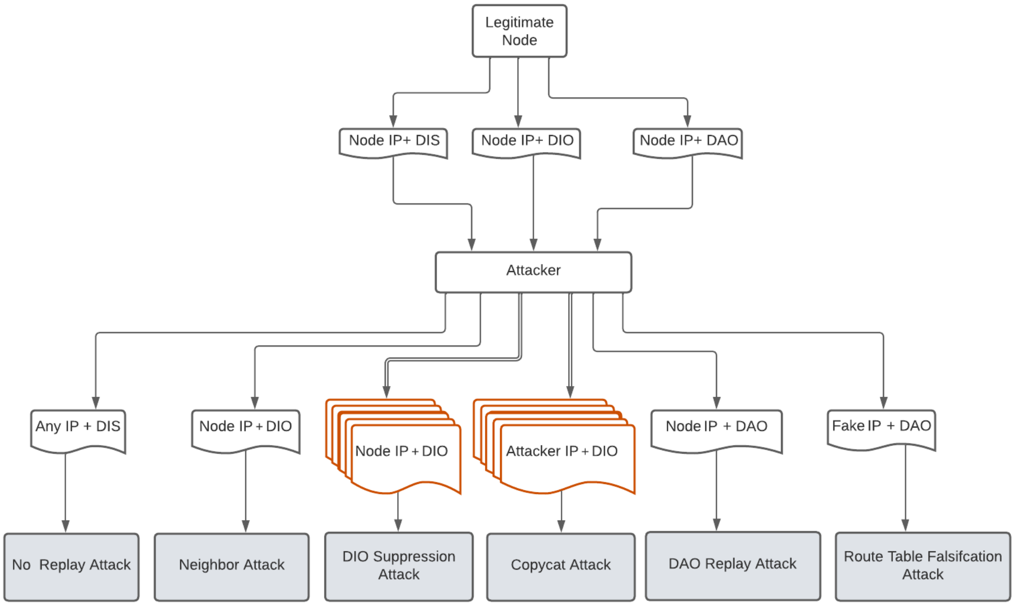


Fig. 7: RPL replay attacks mechanism.

[45]. Typically, when the joining node receives DIO messages for the first time, it performs the following steps. First, it adds the DIO sender address to its parent list and computes its rank according to the OF. After that, it sends its DIO message with the calculated rank and its information. The node chooses the preferred parent among the list of its parents, which offers the minimum rank for this node. When a node connected to a DODAG receives a DIO message from a neighbor in the same DODAG, the node must handle this DIO message [45]. The node either maintains its rank in an existing DODAG or enhances its rank by getting a lower rank value in the DODAG based on computing the path cost indicated by the OF [44], [45]. As a result, the upward route is constructed by the node's preferred parent [58]. Therefore, this control message is exposed to multiple replay attacks. Specifically, three types of replay attacks have been introduced based on replaying DIO messages: neighbor attack [13], [55], [59], DIO suppression attack [14], [17], and copycat attack [20], [56]. We will study these forms of DIO replay attacks in terms of the attackers' mechanisms and their impact on the routing service. We will also highlight the DODAG topology under these attacks, whether by constructing sub-optimized routing paths or by isolating some nodes from the DODAG.

1) DIO Replay Attack

When an attacker node replays the received DIO message without modification, this attack is called a neighbor attack. To perform this attack, the attacker node has to replay an exact copy of the received DIO message without adding its information (neither its rank nor the IP address). As a result, this replayed DIO deceives the recipients by considering the original sender of this DIO message in their communication range. The intent of this attack is to compel nodes to pursue sub-optimized or nonexistent paths [26], [59]. The neighbor attack has been studied by many researchers. For instance, Sharma *et al.* [25] considered the neighbor attack as a type of sub-optimization attack in their classification and comparison of RPL routing attacks. The impact of this attack, according to [25], is the delay in network performance as packets use non-optimal paths. Raoof *et al.* [13] classified neighbor attacks as a type of RPL-specific attacks, emphasizing this attack as a result of RPL vulnerabilities. In addition, [13] pointed out that this adds non-significant network delay. However, Mangelkar *et al.* [59] considered the neighbor attack as a type of confidentiality threat attack that happens as a result of a failure to keep routing information confidential, which leads to adapting sub-optimal or non-existing routes and increasing resource consumption. Furthermore, several studies have demonstrated that this attack increases end-to-end delay, changes network topology, adds additional control overhead, and loses some data packets [18], [60]–[62].

To illustrate the impact of this attack and how it negatively affects the routing service, we need to consider the position of the adversary node, victim nodes, and the node that has been exploited in the DODAG, as the position of the exploited node plays a vital role in amplifying the impact of the neighbor attack [13], [59]. Figure 8 illustrates various positions of the attacker node, the exploited node, and the victim node. The critical impact of the neighbor attack happens when the exploited node has a lower rank than the victim's parent, as mentioned in [13], [19], [26]. This case is shown in Figure 8(c). Specifically, the victim receives the replayed DIO message. This message contains information about a new neighbor node from the victim's perspective. Therefore, the exploited node will be added as a candidate parent in its parent set. Also, it will be added to the victim's routing table if the DODAG uses a storing mode [45]. In addition, the victim node will calculate its new rank based on the rank of the exploited node and will choose it as a preferred parent

[45]. Accordingly, the victim node will also send a DAO message to the exploited node as a new parent. However, as this path does not exist, it leads to high exchange of control messages and loses some data packets [13]. In Figure 8 (b), the exploited node is out of the victim's range. The victim receives the replayed DIO message sent by the attacker. As a result, the victim node will add the exploited node to the parent list as well as the routing table, similar to the previous case. However, the replayed DIO message contains a higher rank than the victim's parent. Therefore, the exploited node will not be considered as a preferred parent, and the upward route will not be impacted. In this case, the impact of the neighbor attack is moderate by adding an out-of-range node in the victim's potential parents and the routing table if the victim is in the storing mode [45]. In addition, this attack leads to a further exchange of control messages that causes some delay.

In Figure 8 (a), both the exploited node and victim nodes are within each other's communication range. Therefore, the replayed DIO message, previously received by the victim, will only be a repeated consistent DIO message. The only impact, in this case, is to increase the control message exchanges by triggering the trickle timer of the victim to send its own DIO message. This scenario has a low impact on routing because it does not directly affect network topology nor lead to the consideration of an out-of-range neighbor as a potential parent. It is worth noting that the neighbor attack can be launched by any internal or external node, as the adversary uses a DIO message multicast by each node in the network [25].

2) *Frequent DIO Replay Attacks*

The second and third types of DIO replay attacks fall under this category. The second DIO replay attack is a DIO suppression attack, where the attacker replays a received legitimate DIO message many times within a fixed frequency [17]. The DIO suppression attack has the same mechanism as the neighbor attack in terms of replaying the received DIO message without adding any information about the sender node. The difference in this attack appears in replaying the received DIO message within a fixed frequency [17], [63]. This attack aims to hide the existence of some network nodes that could lead to partitioning the network by misusing the trickle timer [13], [63]. According to the trickle algorithm, when a node receives a specific consistent number of DIO messages, the trickling method prevents this node from transmitting its own DIO messages to minimize the congestion by reducing the exchanged control messages in the network [48]. As such, the adversary replays a specific DIO message several times to trigger the suppression feature in the victim node. The suppressed nodes may remain hidden, and some routes may go undiscovered as a result of the ongoing suppression [14], [64].

Several researchers have tackled the consequences of the DIO suppression attack. For instance, Avalia *et al.* [65] included the DIO suppression attack as a type of attack that threatens the network in terms of eliminating some control messages and destructing network topology. Other consequences of this attack appeared in the form of degrading the quality of routing services by reducing PDR and network path stretch [17], [66]. In addition, other studies have highlighted the DIO suppression attack as an attack that forces nodes to update their routing tables with outdated and wrong data [13], [17], [22], [26], [67].

To illustrate the impact of DIO suppression attacks, we evaluate only the suppressing feature and its impact. To achieve that, the malicious node is positioned close to the replay source node to focus only on the frequent replay effect. By doing this, every node that receives replayed DIOs can reach the replay source node [17]. However, the attacker node may replay a DIO message from an out-of-range node in frequent replay DIO attacks [17], [20]. The combined impact of suppressing and attempting to reach an out-of-range node is not assessed in this study and is left for future work. Figure 9 demonstrates different positions that attacker and victim nodes can take under a DIO suppression attack. Even though Perazzo *et al.* [17] and Raoof *et al.* [13] mentioned that the worst impact of this attack leads to network partitioning and isolating some nodes, they have not mentioned the position that causes such impact. To attain this level of harm in the network, we choose the setup shown in Figure 9 (c). In this position, the victim node has only the exploited node and the attacker node as potential parents. Besides, the sub-DODAG of the victim node is connected to the DODAG only through the victim node. According to this position, when the victim node receives consistent DIO messages from the exploited and attacker nodes, it suppresses its DIO. As a result, the sub-DODAG of the victim node that relies only on DIO messages from the victim to have the routing information about the network will not receive any DIO message. Therefore, suppressing the victim node will lead to network partitioning and sub-DODAG isolation from the network. In this case, the DIO suppression attack has a critical impact on the network.

However, Verma *et al.* [68] emphasized the impact of this attack on the routing performance is small when the network has a small size. Figure 9 (a) illustrates this limited effect. In this scenario, the victim node has multiple potential parents. Therefore, when the victim node receives the replayed consistent DIO messages, it suppresses its DIO according to the trickle algorithm. However, the victim node also receives other DIO messages from the other potential parent (node 3). Accordingly, the victim node reacts by transmitting its DIO. This scenario has low impact on network topology because it only leads to additional exchanges of control messages owing to the victim node not suppressing its DIO where the suppression feature is disabled by other parents.

Another setup is introduced in Figure 9 (b) to illustrate the impact of creating sub-optimized paths as mentioned by [17] and [26]. In this setup, the victim node has only the exploited and the attacker nodes as potential parents. In addition, the sub-DODAG of the victim node can connect to the DODAG using another node. In this case, the impact of this attack is moderate on the network topology, as even if the attacker can successfully suppress the victim node, its impact will be limited by hiding only the optimal route through the victim node and forcing the sub-DODAG to use a sub-optimal routing

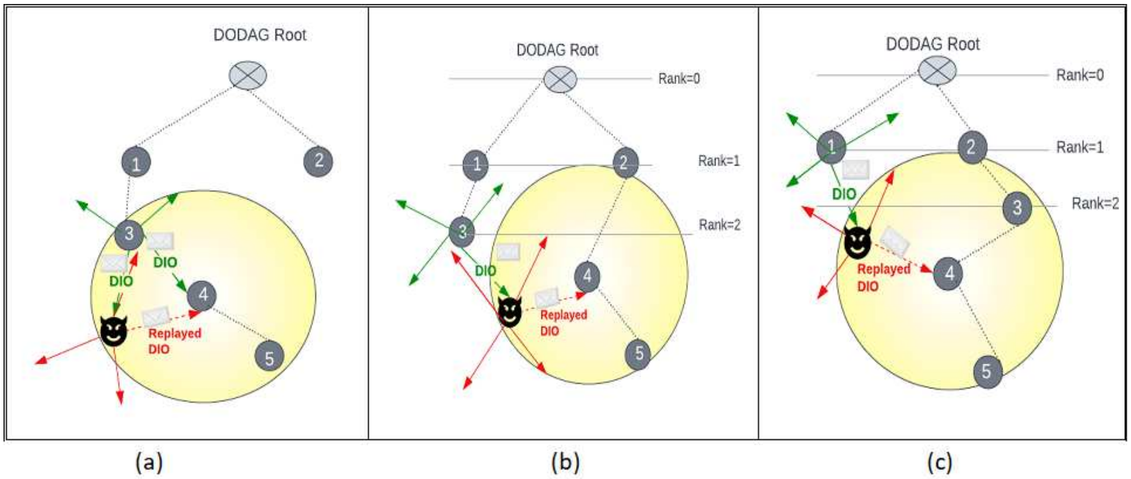


Fig. 8: DIO replay attack analysis based on relative location: (a) low impact, (b) moderate impact, and (c) critical impact.

path without partitioning the network.

The third DIO replay attack is the copycat attack, which was first studied by Verma *et al.* [20]. In this attack, the attacker receives broadcast DIO messages from a legitimate node, and sends previously received DIO messages many times with a fixed replay interval [21], [69]. There are two types of copycat attacks. The first type is called a spoofed copycat attack, where the attacker replaces the source IP address in the IPv6 packet with the IP address of the original DIO sender. That is, the attacker replays the received DIO message frequently without modification. As a result, the receiver will think the DIO sender is in its communication range. This attack has more impact when victim nodes try to add an attacker that is out of range as a parent, assuming it is a part of the optimal path to the root [20]. In this case, this attack is identical to the DIO suppression attack [17]. The second type is called a non-spoofed copycat attack, where the attacker replays DIO messages with its own IP address. As a result, neighbor nodes will receive false information about the rank of the attacker. This makes these neighbor nodes update their rank values and the selection of their parent using incorrect network information [56]. This attack aims to create sub-optimal routing paths in the attack region [21] and increase the control message overhead due to the frequency of the replaying packets [56]. It should be noted that the non-spoofed copycat attack will be studied as a copycat attack in this research. Because the mechanism of this attack is similar to the DIO suppression attack (except that it adds the attacker IP), the analysis of the impact of this attack in terms of the attacker and victim nodes' position is identical to the previously explained DIO suppression attack, as shown in Figure 9.

Few researchers have tackled the copycat attack. Simoglou *et al.* [70] classified this attack as a combination of flooding and replay attacks. Pasikhani *et al.* [71] illustrated copycat attacks are a form of the DIO suppression attack considered to be a topology attack. According to Verma and Ranga [26], the impact of this attack on the network lies in its decreasing PDR and increasing end-to-end network delay [20], [21], [56].

B. Attacks Resulting from Replaying DAO

The DAO message plays a vital role in downward traffic. The destination node must send a DAO control message to announce reverse route information along the upward route. A full downward path is constructed from the DODAG root to the destination node after the DAO message has been transmitted [72]. The parent node is responsible for forwarding this message to the root node to add the sender routing information in the root's routing table, whether the parent node is in storing or non-storing mode. In addition, if the parent is in storing mode, it adds this route to its own routing table for the sender node. Accordingly, each intermediate router transmits a DAO message to the root associated with its address of the DAO message in the reverse routing path. This allows the source to execute source routing to reach each node [44], [58]. Replay attacks launched by a DAO message have not been investigated in the literature. However, Mayzaud *et al.* and Sharma *et al.* described attacks that involve replaying the routing information of path sequence in DAO messages by storing legitimate control messages from other nodes and later transmitting them around the network [15], [25]. We study two types of replay attacks based on replay DAO messages. The first type is a DAO replay attack, and the second type is a route table falsification attack. According to Mangelkar *et al.* [59], all replay attacks are classified as a failure to secure the integrity of the network. In replay attacks using DAO, a malicious node intends to announce fake routes. The difference appears in the mechanism; in a DAO replay attack, the adversary replays legitimate DAO message not in the victims' sub-DODAG. While in a route table falsification attack, the replayed DAO message is modified to advertise a different source node. It is worth mentioning that launching a replay attack using a DAO is harder than a DIO replay attack because the DAO message is typically unicast from the child node to the preferred parent only [44], [45]. Therefore, the attacker node can be either an internal attacker node or a node that sniffs a DAO packet sent by a child to its parent. We will emphasize the mechanism of each of them, the intent of these attacks as well as their impact on the routing service.

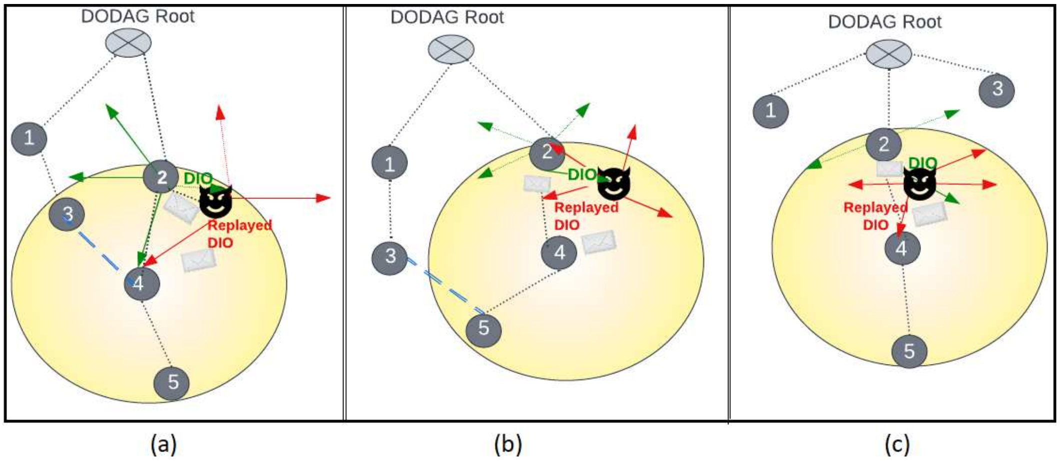


Fig. 9: Frequent DIO replay attacks analysis based on relative location: (a) low impact, (b) moderate impact, and (c) critical impact.

We will also highlight the resulting DODAG topology under these attacks in terms of blocking the downward traffic and dropping the packet delivery.

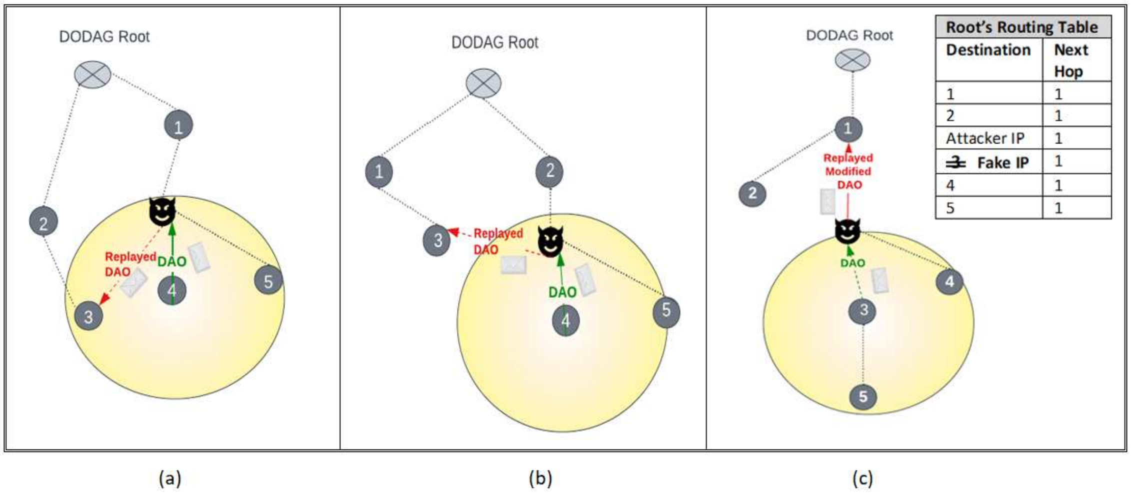
1) DAO Replay Attack

To launch the DAO replay attack, the attacker node receives a DAO message from the exploited node. Then, it replays the DAO message and unicasts it to the victim node. As a result, the victim node adds the exploited node to its neighbor table (in the case of storing mode) and forwards this DAO to its ancestors. The ancestors add this fake entry in their routing tables (in the storing mode) in addition to the root's routing table. Figure 10 highlights various scenarios of the attacker, exploited, and victim nodes. To illustrate the impact of having longer routing paths that lead to more delays (according to Kamble *et al.* [57] and Raoof *et al.* [13]), Figure 10 (a) represents the scenario of the exploited and victim nodes, where nodes are completely reachable. In this case, the attacker node replays the DAO message to a neighbor node. Accordingly, the victim node considers the exploited node as a child, and it forwards this DAO message to all its ancestors. When the root wants to send a message to the exploited node, it sends it through the victim node, and then the victim node delivers it to the exploited node because it is a neighbor node. Therefore, this position in a DAO replay attack has a moderate impact on the routing in terms of using a sub-optimized downward path for the exploited node. Also, in the storing mode, it fills the routing tables of the exploited node and all its ancestors with a false entry. Conversely, Figure 10 (b) reveals the case where the DAO replay attack has a critical and harmful impact that includes packet drops and an increase in the number of RPL messages [13] and [57]. In this case, the victim node is out of the exploited node range. When the root tries to reach the exploited node, the traffic will flow toward the victim node. Because the victim node does not reach the exploited node, it drops packets directed toward the exploited node. As a result of failing to deliver the required data to the destination, the route purging process will be invoked to reconstruct the downward paths, which requires additional exchanges of control messages and leads to losing multiple data packets.

2) DAO Replay Attack with Source Forging

According to the literature, this attack is known as a route table falsification attack, and it requires enabling the storing mode. This attack happens when the attacker modifies received DAO messages to build fake downward routes [13], [15]. We define this attack as follows: the route table falsification attack occurs when a malicious node replays the received DAO message after modifying this message by replacing the original source with a fake source. As a result, the replayed DAO message informs other nodes of false routes. The nodes that are advertised in the bogus route might be (a) genuine but not present in the adversary's sub-DODAG or (b) completely fictitious [9]. A thorough examination of how these attacks impact RPL networks has not yet been revealed [9], [13]. Figure 10 (c) illustrates the mechanism of this attack that causes false routing entries, as mentioned by Raoof *et al.* [13] and Kamble *et al.* [57]. When the attacker node receives a legitimate DAO message from its child, it replays the received DAO message after changing the source IP address to a fake one. The impact of this attack is to hide the downward route of the exploited node because the root cannot reach this node. The effect of this mechanism is critical as the exploited node is isolated from the downward traffic, which leads to lose multiple data packets. This mechanism also announces a false path and adds a fake entry in the routing tables of the root and all victim's ancestors with a storing mode.

Few researchers have studied the route table falsification attack or DAO falsification attack. This attack was first explained by Mayzaud *et al.* [15]. According to [15], a route table falsification attack was considered a topology attack that aimed to impact the optimal paths in DODAG topology. Moreover, Raoof *et al.* [33] argued that the route table falsification attack is a type of RPL storing mode attack because this attack only happens when RPL runs in storing mode of operation. The aim of this attack, according to [33], is to harm routing tables. Verma and Ranga [26] also studied the route table falsification attack and presented this attack as a DAO-related attack, mentioning the impact of forcing legitimate nodes to create non-existing routes. Moreover, [26] discussed a case when a victim node might be filled with wrong routing entries as a



Root's Routing Table	
Destination	Next Hop
1	1
2	1
Attacker IP	1
~~3~~ Fake IP	1
4	1
5	1

Fig. 10: DAO replay attacks analysis based on relative location: (a) moderate impact, (b) critical impact, and (c) DAO replay attack with source forging.

result of these fake routes. In addition, Mangelkar *et al.* [59] discussed this attack as a failure to protect network integrity, arguing it creates inconsistent information in the network and impacts the optimality of routing paths [59]. Moreover, the longer routing path can lead to more delays, packet drops, or an increase in the number of RPL messages [57] and [13]. Table III summarizes the impact of various replay attacks in RPL based on the attacker's impact on the DODAG and answers RQ3, which is about the impact of each replay attack on network topology.

IV. Proposed RPL Replay Attacks Taxonomy

Several works have introduced taxonomies to classify RPL attacks [13], [15], [59], [73]. The first RPL attacks taxonomy was proposed by Mayzaud *et al.* [15]. Based on the attack's goals and effects on the network, this taxonomy categorized attacks into three subcategories: network resource attacks, traffic eavesdropping attacks, and topology modification attacks. RPL attacks, however, had been divided into two groups by Raoof *et al.* [13], namely, WSN-inherited attacks and RPL-specific attacks. The first category included routing attacks inherited from the structure and operation of WSNs. The second category listed the attacks that exploit RPL protocol's properties and functionalities, such as rank and version attacks.

As such, RPL has been exposed to many attacks, including several replay attacks. We classify an RPL attack as a replay attack if the attacker replays (with or without modification) any received DIO or DAO messages. Thus, deceiving the victim nodes into thinking the original sender is within their range. As illustrated in Figure 11, we classify RPL replay attacks into two types: replaying DIO or DAO messages. Replaying DIO messages with modification results in a copycat attack, whereas replaying without modification results in either a neighbor or a DIO suppression attack. Moreover, the DAO replay attack results in replaying DAO packets without modification, whereas the route table falsification attack replays DAO messages with modification. Figure 11 also answers RQ2, which is about the implementation of replay attacks against RPL protocol.

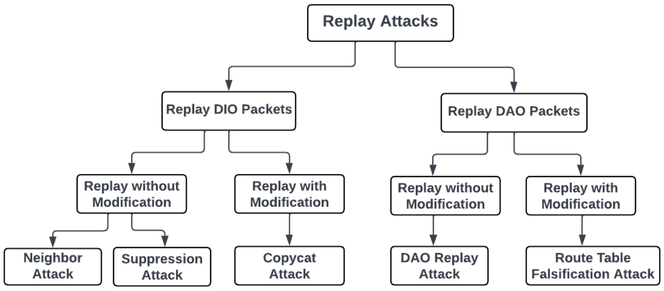


Fig. 11: Proposed taxonomy of RPL replay attacks.

V. Solutions For Replay Attacks

To countermeasure replay attacks, researchers have proposed several solutions that either mitigate or only detect their existence. To answer RQ4, we illustrate in this section different security solutions.

Researchers utilized different methods and various features to identify and detect replay attacks. Figure 12 summarizes features commonly used in replay attacks detection. However, most studies only focused on detecting one type of replay attack. i.e., neighbor or DIO suppression, etc.

To detect neighbor attack, multiple methods have been proposed. For instance, Le *et al.* [74] presented monitoring system, where the monitored values were the number of DIS messages, sequence of DIO messages, stability of the node in the assigned topology, and difference in the rank value between the parent and child. Meanwhile, [55] used location information and received signal strength information in their detecting solution. Moving to another detection system approach proposed in [75], which utilized a trust-based mechanism to detect and isolate neighbor attackers. Another detection method was developed by Farzaneh *et al.* [76] by considering the total received DIO messages and the number of neighbors. In addition, many detection systems have been proposed to detect neighbor attacks employing machine-learning and deep-learning techniques. For instance, Wei *et al.* [77] applied a K-nearest neighbor to identify malicious

TABLE III: RPL replay attacks impact.

Attack	Level of Impact	Positions of Attacker and Exploited Node	Impact
DIO Replay	Low	Deceived node is neighbor of exploited node	Exchanging additional control messages
	Moderate	Deceived node is out-of-range of exploited node with higher rank than deceived node's parent	Exchanging additional control messages; adding out-of-range neighbor as potential parent
	Critical	Deceived node is out-of-range of exploited node with lower rank than deceived node's parent	Exchanging additional control messages; adding out-of-range neighbor as potential parents; attempting to use non-existing upward path
Frequent DIO Replay	Low	Deceived node has multiple potential parents	Exchanging additional control messages
	Moderate	Decedents of deceived node have multiple potential parents	Exchanging additional control messages; suppressing deceived node; using sub-optimized routing path
	Critical	Deceived node and its sub-DODAG are connected to the DODAG through attacker	Exchanging additional control messages; suppressing deceived node; partitioning the DODAG
DAO Replay	Moderate	Deceived node is neighbor of exploited node	Exchanging additional control messages
	Critical	Deceived node is out-of-range of exploited node	Exchanging additional control messages; adding non-existing downward path; isolating exploited node from downward traffic
DAO Replay With source forging	Critical	Deceived node forges exploited node's address	Exchanging additional control messages; adding non-existing downward path; isolating exploited node from downward traffic

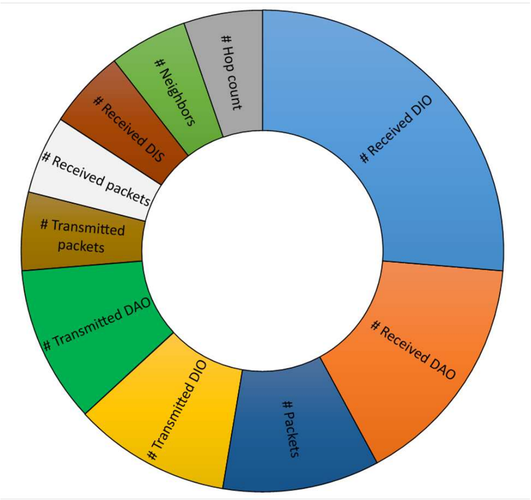


Fig. 12: Common used features to detect RPL replay attacks.

nodes. The features used in their identifier were a combination of application layer features, RPL features, transaction-based features, and others. Another detection method was proposed in [78] that utilized a convolutional neural network to detect different attacks, including replay attacks.

Several methods have been proposed to detect DIO suppression attacks. For instance, a heuristic-based detection scheme was proposed by Pu and Zhou [22]. This method is built on the threshold value of an acceptable sequence number increment. Another study [79] utilized a Gini Index-based countermeasure that measures the disparity of the identities in the received messages to detect DIO suppression attacks. In addition, Pu and Groves [80] introduced a detection method based on the number of received packets per child to detect DIO suppression attacks. Yadav and Bhatt [81] utilized hop count value and link reliability obtained from forwarding and reversing mechanisms. To identify copycat attacks, a few studies were proposed [20], [21], [56]. All these studies suggested using outlier detection to find attacks based on abnormal behavior. It is worth noting that no study has been proposed to detect DAO replay attacks.

Beyond detecting replay attacks, some studies suggest how to mitigate these attacks in RPL. Generally, the mitigation solutions for replay attacks are divided into two main categories, as illustrated in Figure 13, which are sender authentication and RPL enhancements, considering that replay attacks can be launched by internal or external nodes, as mentioned in [60] and [23].

Sender authentication includes challenge handshake using consistency check RPL control message. According to Perazzo *et al.* [53] and Raoof *et al.* [18] in their evaluation of the pre-installed secure mode, when the joining node verifies the consistency check of the sender node, it approves its DIO messages which eliminate the risk of neighbor attacks. Another suggested authentication technique is the chained secure mode using secret chaining fields proposed by Raoof *et al.* [23]. In this approach, secret chaining fields are included in the RPL message, where the entire RPL message is encrypted using the latest secret chaining value. This method successfully mitigated neighbor attacks. Finally, the digital signature is also proposed by Pu and Carpenter [82] as an authentication approach. Specifically, SHA-256 is the suggested algorithm to verify the sender's identity. According to [82], when a node in the forwarding path receives a modified packet or a packet that cannot verify the sender's digital signature, the packet will be discarded.

On the other hand, several RPL enhancements are recommended to address replay attacks. One of these enhancements is mandating new nodes to send a DAO-ACK packet during the joining process as indicated by Sahay *et al.* [63]. According to this enhancement, the new node transmits a DAO message to the root node and waits for its acknowledgment before being registered. This measure prevents exploiting any received

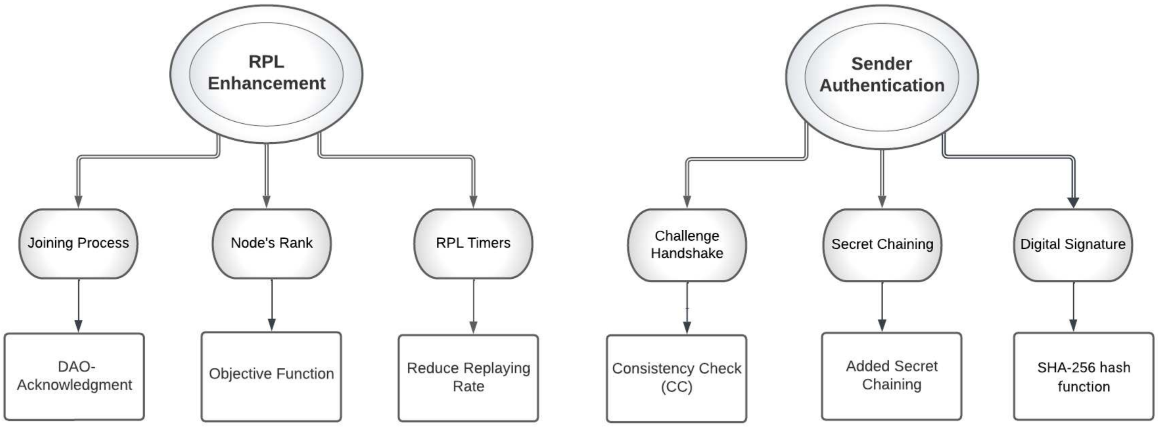


Fig. 13: Mitigation techniques for RPL replay attacks.

DIO messages to launch a replay attack. Additionally, using different objective functions such as OF0 [66], congestion awareness [83], and echelon metric [84] objective functions can help combat replay attacks. According to [66], using OF0 provided more robust performance against various attacks compared to MRHOF, which has frequent parent changes and impacts network topology. Bhandari *et al.* [83] proposed a congestion-aware routing protocol to avoid congestion and avoid nodes with frequent emission of packets which avoid suppression and copycat attackers. For the echelon metric, objective function, Bang *et al.* [84] proposed this new objective function that calculates the rank by utilizing the value of the echelon metric to consider the edges values of the DODAG metric and the selected best parent. As a result, this objective function is robust against replayed messages [84]. Finally, reducing the replay rate is a crucial modification that helps eliminate replayed messages and minimizes the need for extra control messages [64].

VI. Empirical Evaluation

To evaluate different RPL replay attacks, we conducted several experiments to test the impact of these attacks on network performance. We simulated the following RPL replay attacks: neighbor attack, DIO suppression attack, copycat attack, DAO replay attack, and route table falsification attack. This section also provides a comparative study of their performance.

A. Experimental Setup and Design

The simulation environment uses Contiki-NG because it is an open-source operating system focusing on low-power IoT devices. In addition, Contiki-NG is also lightweight and has flexible resource management modules [85]. RPL protocol is already implemented in Contiki-NG. Cooja is the simulation environment in Contiki-NG that reflects the actual compiled IoT networks. The experiment has been set to evaluate a network with single root. The complexity of the observed metrics was decreased by determining a single attacker node and the minimum neighbors needed for each attack. The simulation parameters are summarized in Table IV.

TABLE IV: Simulation parameters.

Parameter	Value
Sensor nodes type	Zolertia Z1 mote
PHY and MAC layer	IEEE 802.15.4 with CSMA and ContikiMAC
Objective function	MRHOF OF
Mode of operation	Storing
No. of replay attacks	5
Simulation time	5 min, 20 min, 60 min
Node positioning	Illustrated in Figures 8, 9, and 10
Deployment area	100m W × 100m L
Number of nodes (adversary included)	7

Table V shows the different replay attack configurations. Under replay attack scenarios, control messages are replayed once for neighbor, DAO replay, and route table falsification attacks [15], [23]. In comparison, DIO suppression and copycat attacks replay the received DIO messages 10 times within a five-second interval. The attacker node keeps performing the replay attack whenever it receives a control message from the exploited node. In our experiment, the nodes are set to compute their rank based on MRHOF objective function [43].

TABLE V: Configuration parameters in different RPL replay attacks.

Replay Attack	Control Message	Modification	# Replay	Interval
Neighbor	DIO	-	1	-
DIO suppression	DIO	-	10	5 sec.
Copycat	DIO	Sender's IP	10	5 sec.
DAO replay	DAO	-	1	-
Route table falsification	DAO	Sender's IP	1	-

B. Evaluation Metrics

To analyze replay attacks' effects on the network, four evaluation metrics are used, as defined below. The evaluation metrics are implemented in the simulation script editor.

- **Packet delivery ratio (PDR):** the packet delivery ratio is calculated by dividing the number of sent data packets by the number of received packets at the destination node.

$$PDR = (PacketsReceived)/(PacketsSent) \quad (1)$$

- **End-to-end delay:** E2E delay is measured as the consumed time to transmit packets from one node to another through the network, including both processing and queuing time [58]. Considering n to be the number of received packets, the delay is calculated as follows:

$$E2E\_D = \sum_{1}^{n} (ReceiveTime - SentTime)/n \quad (2)$$

- **Control messages overhead:** this measure calculates the average number of control messages exchanged in the network to maintain the DODAG topology, including DIS, DIO, DAO, and DAO-ACK control messages [58].
- **Power consumption:** the power consumption is computed by recording the total energy needed to transmit all the packets, which is obtained using the Contiki-NG (energest.h) library. The power (in milliwatts) is calculated by multiplying the current of each state by the period of time the sensor node spends operating in each of the four states, CPU, low power mode [LPM], transition [Rx], and receive [Tx] together by the voltage [86]. The voltage and current values are taken from the Z1 sensor datasheet [87], as shown in Table VI. In general, the energest time is the number of ticks spent in a specific state in each interval, which needs to be divided by Contiki's RTIMER_ARCH_SECOND =32768 [88] to obtain the time in seconds. The power consumption for all states is computed using Equation 3 [89]. To compute the average power consumption of impacted nodes in the network, the power consumed by all nodes, excluding the attackers, is summed. This value is averaged by the number of neighbors to find the local impact of average power consumption.

$$PC = \frac{EnergestValue \times Current \times Voltage}{RTIMER\_ARCH\_SECOND \times Period} \quad (3)$$

TABLE VI: Z1 mote power specification.

Attribute	Value
Voltage	3 V
Current transmit (Tx)	17.4 mA
Current receive (Rx)	18.8 mA
Current CPU	10 mA
Current idle	23µ A

VII. Results and Discussion

A. Packet Delivery Ratio

To evaluate PDR, Figure 14 shows the local effect of various forms of replay attacks on PDR on nodes that are neighbors of the attacker. Figure 14 reveals neighbor attacks in all its positions have no impact on data delivery. Despite advertising for an out-of-range node, the victim node finds a legitimate path to transmit data packets successfully. Nevertheless, DAO replay and route table falsification attacks lead to losing 15% of the data packets when the attacker has a critical impact. We found the lost packets in the DAO replay attack are the data sent by the root to the node that has been exploited where no data packets can be delivered to them. In this case, when the root sends a data packet to the exploited node, the data is sent to the node that has received the replayed DAO packet. When this node is unable to reach the exploited node, the data packet is dropped. For route table falsification attacks, even though the data is sent through its legitimate route toward the destination, the data packet is directed to a forged IP. Therefore, it does not reach the target node.

For DIO suppression and copycat, both attacks have identical impacts on whether the DIO packet is modified before replaying or not. The results reveal the attacker's position has the largest impact on data delivery. When the attacker node locates in the critical position, the PDR is dropped to 50%. Also, other positions of the attacker lead to losing around 20% of the data packets. As such, the most harmful replay attacks on the PDR are DIO suppression and copycat attacks. However, the attack with the lowest impact on PDR is neighbor attack.

B. End-to-End Delay

Figure 15 shows how each replay attack impacts E2E delay on the attacker's neighbors. The most impact in terms of delay is from neighbor attacks, particularly when an attacker is placed in a low-impact position and replays DIO messages of a node that is a neighbor to the deceived node. In this case, the delay reaches up to 50% higher than the no-attack scenario. We found the delay occurs when sending and receiving data packets belonging to the exploited node whose DIO message is used to launch replay attacks. For DAO replay and route table falsification attacks, the delay increases by around 23% in the case of critical impact. Nevertheless, when the attacker performs a moderate-impact DAO replay attack, the latency is negligible. DIO suppression and copycat attacks have similar behavior in terms of end-to-end delay. The most affected case in the latency is the moderate impact scenario when the victim's child can join DODAG through another parent. In particular, the delay occurs at the child of the deceived node, where this node frequently cannot access its parent and needs to consider a sub-optimized routing path. It is worth mentioning that when the attacker node has a critical impact, the data packets that have been successfully delivered cause less delay than when a network is not under attack. The reason behind this behavior is that data packets of the exploited node and the deceived node and its child are not delivered when the attacker is launching the attack by frequently replaying DIO packets. Most of the delivered packets belong to nodes close to the root and out of the attacker's range. In summary, replay attacks that cause high delay are neighbor and route table falsification attacks, whereas the attacks with the lowest impact on latency are DIO suppression and copycat attacks.

C. Control Message Overhead

Figure 16 illustrates the average exchanged control messages. In general, attacks resulting in replaying DIO lead to more exchanges in RPL messages, as DIO messages are usually broadcast to all neighbors. Also, according to RPL, all

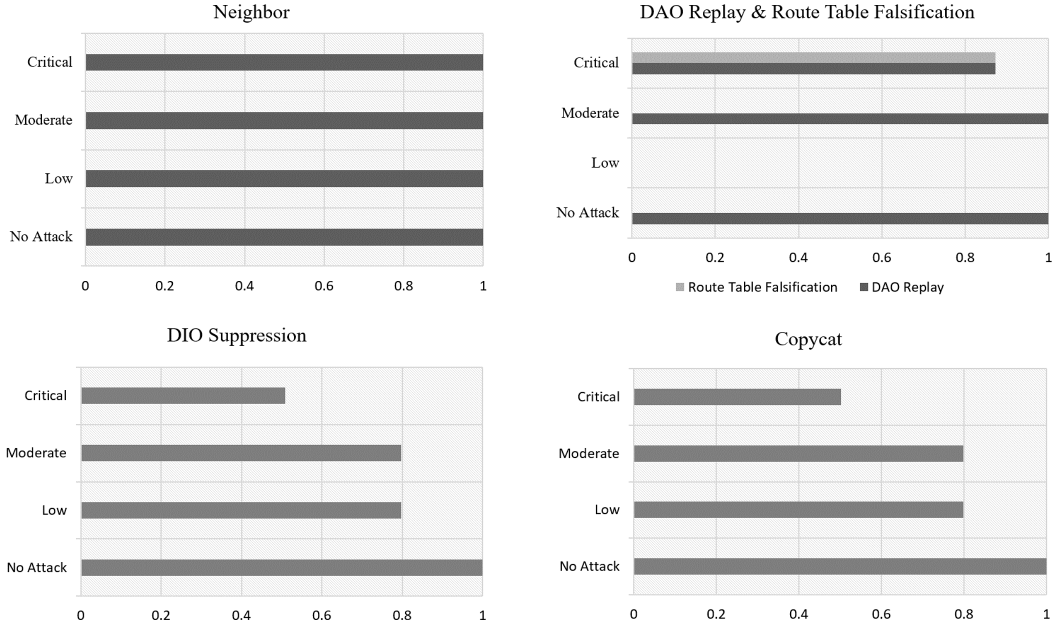


Fig. 14: Local impact of replay attacks on PDR.

receiver nodes must process each incoming DIO to track the changes in DODAG and transmit their own DIO message. The attacks that cause the highest exchange of control messages are copycat and DIO suppression attacks, where the increase in control messages is between 500% and 800%. In specific, the copycat attack, when an attacker has a low impact, shows the highest control overhead where the increase percentage reaches about 800%. This increase is due to the exploited node in this position having more neighbors than in other cases where the exploited node reacts to every received DIO packet by generating its DIO message according to the trickle timer specification that leads to replay more DIO messages by the attacker. Even though copycat and DIO suppression attacks have a similar setup, in the copycat attack, the attacker node adds its IP to the replayed messages, which makes receiver nodes consider these messages as different DIO messages. Whereas DAO replay and route table falsification attacks lead to a slight increase in exchanged control messages. This result is justified by the nature of DAO messages, which are unicast only to the parent node. This finding also highlights the difficulty of detecting this type of replay attack.

D. Power Consumption

When the network is under replay attacks, the control messages overhead emphasize the wasted energy extinguished in exchanging futile control messages. Because the additional overhead consumes nodes' batteries, these attacks impact network availability. Figure 17 shows the impact of the replay attacks on average power consumption for neighbor nodes of the attacker. In general, replay attacks have a noticeable impact on the nodes' power as a result of the additional exchange of control messages. Figure 17 confirms that power consumption is correlated with messages overhead. Accordingly, DIO suppression and copycat attacks affect the average power consumption of the network the most. In comparison, DAO replay and route table falsification attacks affect it the least.

As a whole, the performance evaluation illustrates the impact of replay attacks on PDR, E2E delay, control message overhead, and power consumption and answers RQ3, which is related to the impact of each replay attack on the network from a performance perspective.

VIII. Discussion and Research Directions

According to IETF, for securing IoT devices, any device connected to the Internet must safeguard itself against attacks that may cause malfunctions or unauthorized use for unintended purposes [90]. Thus, launching various replay attacks using different control messages can be done effortlessly. RPL operates on sensors that gather data from different systems so that they may have been installed in non-traditional locations. Therefore, physical security is difficult or impossible to achieve, and attackers may have direct physical access to IoT devices. In RPL, control messages are used for constructing and maintaining the topology to take the optimal path for packets. When a node is programmed to replay a received control message illegitimately, it reposts fake routing information, violating the service's integrity. For replaying DIO messages, any node, whether or not it belongs to the network, can receive this broadcast message and replay it. Moreover, it can frequently send this message to exploit the trickle algorithm that manages the time of emitting the control messages. On the other hand, replaying DAO control messages requires additional efforts. The malicious node needs first to join the network to receive an authentic DAO packet and then replay it to launch various replay attacks.

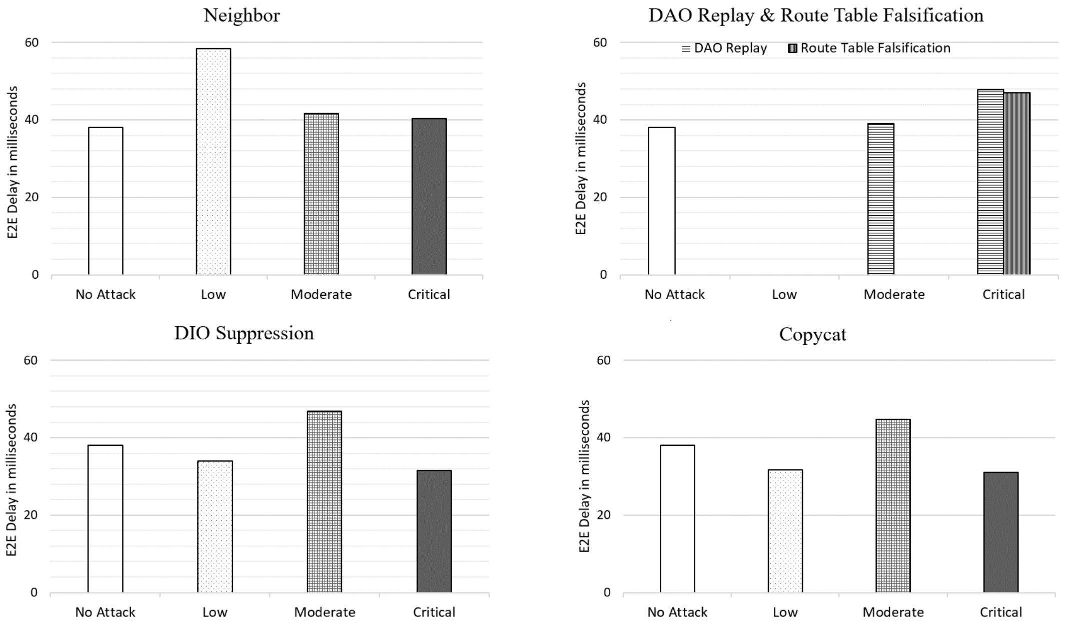


Fig. 15: Local impact of replay attacks on E2E delay.

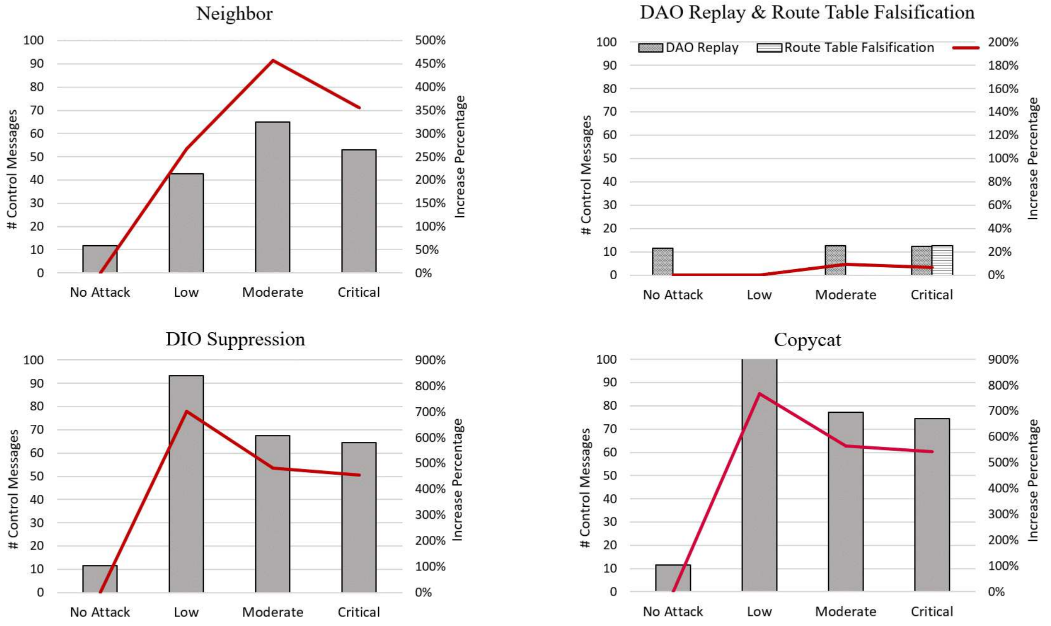


Fig. 16: Local impact of replay attacks on average control message overhead.

Our study effectively exposes the distinct characteristics of every type of replay attack present in RPL. This proves to be an invaluable resource for the research community in developing detection systems based on these signatures. The results we attained verified that replay attacks are a serious concern in IoT devices because the availability of neighbor nodes is threatened under these attacks. Specifically, the DIO suppression and copycat attacks have the greatest potential effect on a network owing to their affecting data delivery and network resources as they contain various malicious activities. These activities include illegitimate replaying of control messages, faking the routing properties of yet another node (rank), and constantly repeating this behavior. Also, DAO replay and route table falsification attacks can isolate the exploited node from receiving root node data. Additionally, the delay is significant with neighbor attacks. This assessment provides clear path for developing anomaly-based detection systems to monitor network behavior and detect irregularities

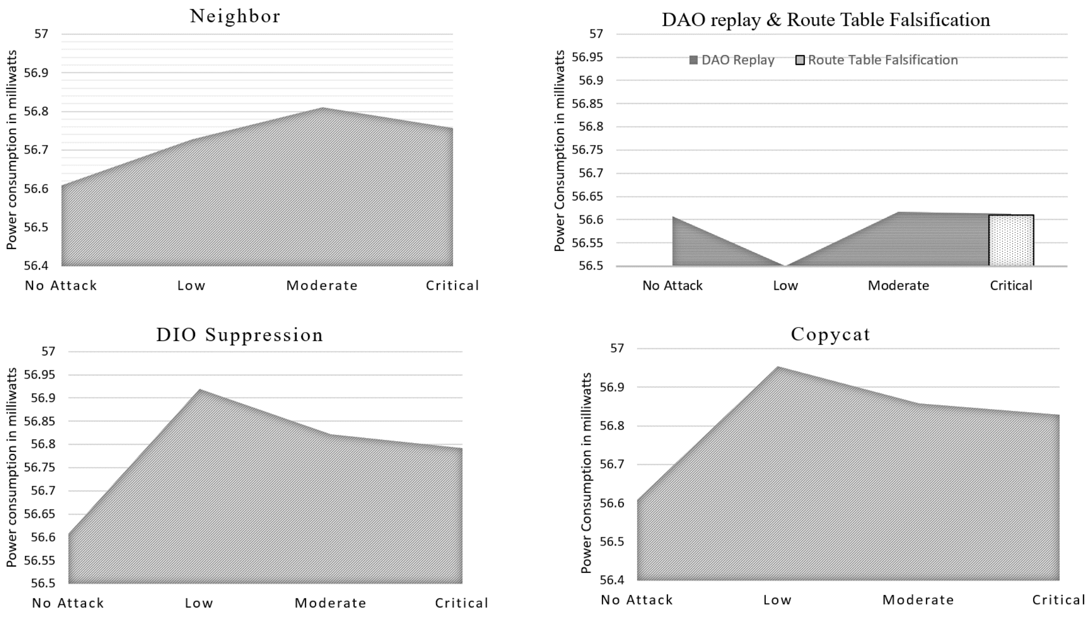


Fig. 17: Local impact of replay attacks on average power consumption.

related to replay attacks.

Simultaneously, the proposed countermeasures against replay attacks are considerably fewer compared to solutions proposed to mitigate other attacks, such as blackhole or version attacks. Regarding the proposed mitigation solutions, we generally divide these solutions into RPL enhancement and sender authentication. For sender authentication, the main drawback of various authenticating algorithms revolves around memory, computing, and energy resources, which limits the expansion of studying these techniques. This limitation leads to adapting few lightweight algorithms. Another drawback of authentication techniques is the usage of shared keys, where internal nodes can misbehave and launch multiple attacks utilizing their key. An internal node, for instance, can use a replay attack to fake some of the properties of other nodes. Therefore, there is a need to have a mechanism for message-sender authentication. In addition, researchers have to consider when to invoke the authentication algorithm carefully. Because node authentication requires an additional process for the sender verification, it is preferred to impose this algorithm on the minimum occasions that guarantee authenticity while simultaneously saving the nodes' energy to balance the security requirement and reduce authentication cost.

For the RPL enhancement technique, most of the suggested solutions relied on defining the objective function based on trust and node behavior to avoid the misbehaving node. As the trust-based objective function is widely used to mitigate various attacks, it is crucial to establish the replay process behavior and also define this behavior in a given function. This has been identified as a significant research gap in the literature. In the proposed solutions, the trust level is considered based on the time interval between replay messages. Besides, the trust can also consider the data packet loss, which results from DAO replay and route table falsification attacks. Although these measures were designed to address other attacks, their efficiency in the face of replay attacks has yet to be thoroughly assessed. Furthermore, other objective functions need empirical evaluations under replay attacks. Another RPL enhancement method is based on modifying the joining process to reduce the risk of DAO replay and route table falsification attacks. By applying this mitigation, the node should wait for the acknowledgment from the root, which eliminates the main impact of isolating the exploited node from receiving root packets. However, this mechanism has not been evaluated under these attacks.

In summary, RPL is at risk of replay attacks that can easily be initiated through control messages. The extent of damage caused by these attacks is determined by the type of control message employed and the attacker's location and exploited nodes. Security solutions must be meticulously assessed and applied with varied approaches that cater to different scenarios to counter these challenges.

Acknowledgments

The authors would like to acknowledge the support provided by the Interdisciplinary Research Center for Intelligent Secure System (IRC-ISS) and the Deanship of Scientific Research at King Fahd University of Petroleum & Minerals (KFUPM). This project is funded by IRC-ISS Grant #INSS2202.

IX. Conclusions and Future work

This article investigates the problem of RPL replay attacks in IoT environments where attackers record and send control messages to disrupt the routing topology. The article also reveals multiple RPL replay attacks based on the type of replayed control packet, the modification performed before replaying, and the frequency of replaying the packet. Specifically, we study and analyze five forms of replay attacks. Three types have been introduced based on replaying DIO messages namely, neighbor, DIO suppression, and copycat attacks. We also analyze the impact on the DODAG topology under these attacks, whether by constructing sub-optimized routing paths or isolating some nodes from the DODAG. In addition, we introduce the first study of two replay attacks launched by DAO messages. The first is the DAO replay attack, and the second is the route table falsification attack. We also investigated the DODAG topology under these attacks in terms of blocking the downward traffic and dropping packet delivery. This article also proposes a taxonomy for RPL replay attacks to distinguish their various forms. Furthermore, we also highlight the existing countermeasures to mitigate and detect RPL replay attacks. Mitigation solutions include sender authentication and RPL enhancements, while detection solutions include threshold, trust, location, and machine learning.

Extensive evaluation experiments, to measure network performance under these attacks, reveal that replay attacks can severely impact a network. Specifically, DIO suppression and copycat attacks have the greatest potential effect on a network owing to their affecting data delivery (PDR has dropped to 50%) and the network resources (the increase in control messages is between 500% and 800%). In addition, we found that DAO replay and route table falsification attacks isolate the downward traffic toward the exploited node. Our results also confirm that the neighbor attack leads to considerable delays that reach up to 50% higher than the no-attack scenario. Looking ahead, we are working on proposing and developing solutions to mitigate RPL replay attacks with different numbers of attacker nodes, network sizes, and topologies.

References

[1] P. Bellini, P. Nesi, and G. Pantaleo, “IoT-enabled smart cities: A review of concepts, frameworks and key technologies,” *Applied Sciences (Switzerland)*, vol. 12, 2 2022.

[2] S. O’Dea, “Wide-area and short-range IoT device installed base worldwide 2014-2027,” https://www.statista.com/statistics/1016276/wide-area-and-short-range-iot-device-installed-base-worldwide/, 2022.

[3] John R. Vacca, *Computer and Information Security Handbook*, 2017.

[4] F. Azzedin and M. Ghaleb, “Internet-of-things and information fusion: Trust perspective survey,” *Sensors*, vol. 19, no. 8, p. 1929, 2019.

[5] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” *Future Generation Computer Systems*, vol. 82, pp. 395–411, May 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X17315765

[6] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, “A survey on routing protocols supported by the contiki internet of things operating system,” *Future Generation Computer Systems*, vol. 82, pp. 200–219, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2017.12.045

[7] J. Tripathi, J. de Oliveira, and JP. Vasseur, “Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC Editor, Tech. Rep. RFC6687, Oct. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6687

[8] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, “Cyber threats to industrial IoT: A survey on attacks and countermeasures,” *IoT*, vol. 2, pp. 163–186, 3 2021.

[9] A. O. Bang, U. P. Rao, P. Kaliyar, and M. Conti, “Assessment of routing attacks and mitigation techniques with RPL control messages: A survey,” *ACM Computing Surveys*, vol. 55, pp. 1–36, 2023.

[10] J. Neeli and S. Patil, “Insight to security paradigm , research trend & statistics in internet of things(IoT),” *Global Transitions Proceedings*, vol. 2, no. 1, pp. 84–90, 2021. [Online]. Available: https://doi.org/10.1016/j.gltp.2021.01.012

[11] D. Swessi and H. Idoudi, *A Survey on Internet-of-Things Security: Threats and Emerging Countermeasures*. Springer US, 2022, no. 0123456789. [Online]. Available: https://doi.org/10.1007/s11277-021-09420-0

[12] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, “A security threat analysis for the routing protocol for low-power and lossy networks (RPLs),” *RFC 7416*, p. 131, 2015.

[13] A. Raoof, A. Matrawy, and C. H. Lung, “Routing Attacks and Mitigation Methods for RPL-Based Internet of Things,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1582–1606, 2019.

[14] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, “A Comprehensive Review on Secure Routing in Internet of Things: Mitigation Methods and Trust-based Approaches,” *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–1, 2020.

[15] A. Mayzaud, R. Badonnel, and I. Chrisment, “A Taxonomy of Attacks in RPL-based Internet of Things,” *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016. [Online]. Available: https://hal.inria.fr/hal-01207859

[16] T. A. Al-Amiedy, M. Anbar, B. Belaton, A. A. Bahashwan, I. H. Hasbullah, M. A. Aladaileh, and G. A. Mukhaini, “A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things,” *Internet of Things*, vol. 22, p. 100741, Jul. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2542660523000641

[17] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, “DIO suppression attack against routing in the internet of things,” *IEEE Communications Letters*, vol. 21, no. 11, pp. 2524–2527, 2017.

[18] A. Raoof, A. Matrawy, and C.-H. Lung, “Enhancing routing security in IoT: Performance evaluation of RPL secure mode under attacks,” *arXiv preprint arXiv:2004.07815*, 2020.

[19] A. Arena, P. Perazzo, C. Vallati, G. Dini, and G. Anastasi, “Evaluating and improving the scalability of RPL security in the Internet of Things,” *Computer Communications*, vol. 151, pp. 119–132, Feb. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0140366419307479

[20] A. Verma and V. Ranga, “The impact of copycat attack on RPL based 6LoWPAN networks in Internet of Things,” *Computing*, 2020. [Online]. Available: https://doi.org/10.1007/s00607-020-00862-1

[21] A. Verma and V. Ranga, “Addressing copycat attacks in ipv6-based low power and lossy networks,” in *Science and Information Conference*. Springer, 2020, pp. 415–426.

[22] C. Pu and X. Zhou, “Suppression attack against multicast protocol in low power and lossy networks: Analysis and defenses,” *Sensors (Switzerland)*, vol. 18, no. 10, pp. 1–20, 2018.

[23] A. Raoof, C.-H. Lung, and A. Matrawy, “Securing RPL using network coding: The chained secure mode (CSM),” *IEEE Internet of Things Journal*, 2021.

[24] A. Thomas, T. Gireesh Kumar, and A. K. Mohan, “Neighbor Attack Detection in Internet of Things,” in *Advanced Computational and Communication Paradigms*, S. Bhattacharyya, N. Chaki, D. Konar, U. K. Chakraborty, and C. T. Singh, Eds. Singapore: Springer Singapore, 2018, vol. 706, pp. 187–196, series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-981-10-8237-5-18

[25] D. Sharma, I. Mishra, and S. Jain, “A Detailed Classification of Routing Attacks against RPL in Internet of Things,” *International Journal of Advance Research*, vol. 3, pp. 692–703, 2017. [Online]. Available: www.ijariit.com

[26] A. Verma and V. Ranga, “Security of RPL Based 6LoWPAN Networks in the Internet of Things: A Review,” *IEEE Sensors Journal*, vol. 20, no. 11, pp. 5666–5690, 2020.

[27] M. Z. Hussain and Z. M. Hanapi, “Efficient Secure Routing Mechanisms for the Low-Powered IoT Network: A Literature Review,” *Electronics*, vol. 12, no. 3, p. 482, Jan. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/3/482

[28] A. Gerodimos, L. Maglaras, M. A. Ferrag, N. Ayres, and I. Kantzavelou, “IoT: Communication protocols and security threats,” *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 1–13, 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2667345222000293

[29] G. Sharma, S. Vidalis, N. Anand, C. Menon, and S. Kumar, “A Survey on Layer-Wise Security Attacks in IoT: Attacks, Countermeasures, and Open-Issues,” *Electronics*, vol. 10, no. 19, p. 2365, Sep. 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/19/2365

[30] S. Malladi, J. Alves-Foss, and R. B. Heckendorn, “On Preventing Replay Attacks on Security Protocols:,” Defense Technical Information Center, Fort Belvoir, VA, Tech. Rep., Jan. 2002. [Online]. Available: http://www.dtic.mil/docs/citations/ADA462295

[31] M. E. Whitman and H. J. Mattord, *Principles of information security*. Cengage learning, 2021.

[32] M. Ciampa, *CompTIA security+ guide to network security fundamentals*. Cengage Learning, 2021.

[33] A. Raoof, "Secure Routing and Forwarding in RPL-based Internet of Things: Challenges and Solutions," no. May, 2021. [Online]. Available: https://curve.carleton.ca/208970b8-5740-437a-bea2-f7e1a55718c7

[34] N. Sultana and M. Tamanna, "Exploring the benefits and challenges of Internet of Things (IoT) during Covid-19: a case study of Bangladesh," *Discover Internet of Things*, vol. 1, no. 1, 2021. [Online]. Available: https://doi.org/10.1007/s43926-021-00020-9

[35] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–25, 2017. [Online]. Available: https://www.hindawi.com/journals/jece/2017/9324035/

[36] L. Kakkar, D. Gupta, S. Saxena, and S. Tanwar, "IoT architectures and its security: a review," in *Proceedings of the Second International Conference on Information Management and Machine Intelligence: ICIMMI 2020*. Springer, 2021, pp. 87–94.

[37] M. Saad and T. R. Soomro, "CYBER SECURITY AND INTERNET OF THINGS," *Pakistan Journal of Engineering, Technology & Science*, vol. 7, no. 1, Apr. 2018. [Online]. Available: http://journals.iobmresearch.com/index.php/PJETS/article/view/2084

[38] M. Ghaleb and F. Azzedin, "Towards scalable and efficient architecture for modeling trust in IoT environments," *Sensors*, vol. 21, no. 9, 2021.

[39] I. Alaoui Ismaili, A. Azyat, N. Raissouni, N. Ben Achhab, A. Chahboun, and M. Lahraoua, "Comparative Study of ZigBee and 6LoWPAN Protocols: Review," in *Proceedings of the Third International Conference on Computing and Wireless Communication Systems, ICCWCS 2019, April 24-25, 2019, Faculty of Sciences, Ibn Tofaïl University -Kénitra- Morocco*. Kenitra, Morocco: EAI, 2019. [Online]. Available: http://eudl.eu/doi/10.4108/eai.24-4-2019.2284215

[40] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[41] N. Accettura and G. Piro, "Optimal and secure protocols in the IETF 6TiSCH communication stack," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. Istanbul, Turkey: IEEE, Jun. 2014, pp. 1469–1474. [Online]. Available: http://ieeexplore.ieee.org/document/6864831/

[42] K. Avila, D. Jabba, and J. Gomez, "Security aspects for RPL-based protocols: A systematic review in IoT," *Applied Sciences*, vol. 10, no. 18, p. 6472, 2020.

[43] B. Mohamed and F. Mohamed, "QoS Routing RPL for Low Power and Lossy Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[44] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2012.06.016

[45] E. C. S. T. Winter, E. C. S. P. Thubert, A. B. S. Designs), J. H. A. R. Corporation), R. K. E. Corporation), P. L. S. University), K. P. D. Networks), R. S. S. S. Consultancy), J. V. C. Systems), and R. A. C. P. Systems), "RPL: IPv6 routing protocol for low-power and lossy networks abstract low-power," *Internet Engineering Task Force (IETF)*, vol. 6550, pp. 1689–1699, 2012.

[46] O. Iova, P. Picco, T. Istomin, and C. Kiraly, "RPL: The Routing Standard for the Internet of Things... or Is It?" *IEEE Communications Magazine*, vol. 54, no. 11, pp. 16–22, 2016.

[47] A. Parasuram, D. Culler, and R. Katz, "An Analysis of the RPL Routing Standard for Low Power and Lossy Networks," *Technical Report No. UCB/EECS-2016-106*, p. 98, 2016.

[48] P. Levis; T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," *Internet Engineering Task Force (IETF)*, 2011. [Online]. Available: http://www.rfc-editor.org/info/rfc6206.

[49] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC Editor, Tech. Rep. RFC6552, Mar. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6552

[50] O. Gnawali and P. Levis, "Rfc 6719: The minimum rank with hysteresis objective function," *Internet Engineering Task Force (IETF) Request For Comments*, 2012.

[51] C. Pu and K. K. R. Choo, "Lightweight sybil attack detection in IoT based on bloom filter and physical unclonable function," *Computers and Security*, vol. 113, 2 2022.

[52] R. Masadeh, B. AlSaaidah, E. Masadeh, M. R. Al-Hadidi, and O. Almomani, "Elastic Hop Count Trickle Timer Algorithm in Internet of Things," *Sustainability*, vol. 14, no. 19, p. 12417, Sep. 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/19/12417

[53] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, and G. Dini, "An implementation and evaluation of the security features of RPL," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10517 LNCS, no. November, pp. 63–76, 2017.

[54] A. Raoof, A. Matrawy, and C.-H. Lung, "Secure routing in IoT: Evaluation of RPL's secure mode under attacks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[55] A. Thomas, T. Gireesh Kumar, and A. K. Mohan, "Neighbor attack detection in internet of things," in *Advanced Computational and Communication Paradigms*. Springer, 2018, pp. 187–196.

[56] A. Verma and V. Ranga, "CoSec-RPL: detection of copycat attacks in RPL based 6LoWPANs using outlier analysis," *Telecommun Syst*, vol. 75, no. 1, pp. 43–61, Sep. 2020. [Online]. Available: https://link.springer.com/10.1007/s11235-020-00674-w

[57] A. Kamble, V. S. Malemath, and D. Patil, "Security attacks and secure routing protocols in RPL-based Internet of Things: Survey," *2017 International Conference on Emerging Trends and Innovation in ICT, ICEI 2017*, pp. 33–39, 2017.

[58] H. Ali, S. Duquennoy, and M. Boldt, "A Performance evaluation of RPL in Contiki," *CLOSER 2015 - 5th International Conference on Cloud Computing and Services Science, Proceedings*, pp. 233–240, 2015. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84969793934{&}partnerID=40{&}md5=e4087628bf9bdf2fce89089ddeb8b1e1

[59] S. Mangelkar, S. N. Dhage, and A. V. Nimkar, "A comparative study on RPL attacks and security solutions," *Proceedings of 2017 International Conference on Intelligent Computing and Control, I2C2 2017*, vol. 2018-January, pp. 1–6, 2018.

[60] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[61] S. Lata, S. Mehfuz, and S. Urooj, "Secure and Reliable WSN for Internet of Things: Challenges and Enabling Technologies," *IEEE Access*, vol. 9, pp. 161 103–161 128, 2021.

[62] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in Internet of Things," *International Journal of Computer Applications*, vol. 121, no. 9, 2015.

[63] R. Sahay, G. Geethakumari, and B. Mitra, "A novel Network Partitioning Attack against Routing Protocol in Internet of Things," *Ad Hoc Networks*, vol. 121, no. January, p. 102583, 2021. [Online]. Available: https://doi.org/10.1016/j.adhoc.2021.102583

[64] B. Groves and C. Pu, "A Gini index-based countermeasure against sybil attack in the Internet of Things," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.

[65] K. Avila, D. Jabba, and J. Gomez, "Security Aspects for RPL-Based Protocols: A Systematic Review in IoT," *Applied Sciences*, vol. 10, no. 18, p. 6472, 2020.

[66] C. Dogan, S. Yilmaz, and S. Sen, "Analysis of RPL Objective Functions with Security Perspective," no. February, pp. 71–80, 2022.

[67] M. A. Boudouaia, A. Abouaissa, A. Benayache, and P. Lorenz, "Divide and conquer-based attack against RPL routing protocol," *2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings*, 2020.

[68] A. Verma and V. Ranga, "Comment on "DIO Suppression Attack Against Routing in the Internet of Things"," *TechRxiv*, 2023.

[69] I. S. Alsukayti and M. Alreshoodi, "RPL-Based IoT networks under simple and complex routing security attacks: An experimental study," *Applied Sciences*, vol. 13, no. 8, p. 4878, 2023.

[70] G. Simoglou, G. Violettas, S. Petridou, and L. Mamatas, "Intrusion detection systems for RPL security: A comparative analysis," *Computers and Security*, vol. 104, pp. 1–56, 2021.

[71] A. M. Pasikhani, J. A. Clark, P. Gope, and A. Alshahrani, "Intrusion Detection Systems in RPL-Based 6LoWPAN: A Systematic Literature Review," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12 940–12 968, 2021.

[72] C. Pu, "Mitigating DAO inconsistency attack in RPL-based low power and lossy networks," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC 2018*, vol. 2018-Janua, pp. 570–574, 2018.

[73] F. Azzedin and H. Albinali, "Security in Internet of Things: RPL attacks taxonomy," in *The 5th International Conference on Future Networks & Distributed Systems*, 2021, pp. 820–825.

[74] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," *Information (Switzerland)*, vol. 7, no. 2, 2016.

[75] P. P. Ioulianou and V. G. Vassilakis, "A Trust-Based Intrusion Detection System for RPL Networks : Detecting a Combination of Rank and Blackhole Attacks," pp. 124–153, 2022.

[76] F. Behnam, M. Mohammad Ali, and S. Jamali, "An Anomaly-Based IDS for Detecting Attacks in RPL-Based Internet of Things," in *2019 5th International Conference on Web Research (ICWR)*. IEEE, 2019, pp. 61—-66.

[77] M. Wei, C. Rong, E. Liang, and Y. Zhuang, "An intrusion detection mechanism for IPv6-based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 18, no. 3, p. 155013292210779, 2022. [Online]. Available: https://doi.org/10.1177/15501329221077922

[78] K.-H. Le, M.-H. Nguyen, T.-D. Tran, and N.-D. Tran, "IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT," *Electronics*, vol. 11, no. 4, p. 524, 2022.

[79] C. Pu, "Sybil attack in RPL-based Internet of Things: Analysis and defenses," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4937–4949, 2020.

[80] C. Pu and B. Groves, "Energy Depletion Attack in Low Power and Lossy Networks: Analysis and Defenses," *Proceedings - 2019 2nd International Conference on Data Intelligence and Security, ICDIS 2019*, pp. 14–21, 2019.

[81] N. Yadav and A. Bhatt, "A Secure IoT Communication against Reactive Jamming Attack in CRN Subtitle," vol. 5, pp. 1543–1549, 2019.

[82] C. Pu and L. Carpenter, "Digital Signature Based Countermeasure Against Puppet Attack in the Internet of Things," *2019 IEEE 18th International Symposium on Network Computing and Applications, NCA 2019*, pp. 1–4, 2019.

[83] K. S. Bhandari, A. S. Hosen, and G. H. Cho, "CoAR: Congestion-aware routing protocol for low power and lossy networks for IoT applications," *Sensors (Switzerland)*, vol. 18, no. 11, 2018.

[84] A. O. Bang and U. P. Rao, "EMBOF-RPL: Improved RPL for early detection and isolation of rank attack in RPL-based internet of things," *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 642–665, 2022.

[85] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, p. 101089, 2022.

[86] J. Schandy, L. Steinfeld, and F. Silveira, "Average power consumption breakdown of wireless sensor network nodes using IPv6 over LLNs," in *2015 International Conference on Distributed Computing in Sensor Systems*. IEEE, 2015, pp. 242–247.

[87] "Z1 datasheet," https://zolertia.sourceforge.net/wiki/index.php/Z1, 2013.

[88] A. Velinov and A. Mileva, "Running and Testing Applications for Contiki OS Using Cooja Simulator," *International Conference on Information Technology and Development of Education*, no. August, pp. 279–285, 2016. [Online]. Available: http://eprints.ugd.edu.mk/16096/1/Zbornik-ITRO-2016-283-289.pdf

[89] N. S. Han, "Semantic service provisioning for 6lowpan: powering internet of things applications on web," Ph.D. dissertation, Institut National des Télécommunications, 2015.

[90] K. Moore, R. Barnes, and H. Tschofenig, "Best current practices for securing Internet of Things (IoT) devices," *IETF Draft, Oct*, 2016.