applied sciences

Article

A Lightweight Mitigation Approach against a New Inundation Attack in RPL-Based IoT Networks

Mehdi Rouissat [1], Mohammed Belkheir [2], Ibrahim S. Alsukayti [3,*] and Allel Mokaddem [2]

[1] STIC Laboratory, Univeristy Center Nour Bachir El-Bayadh, University Aboubekr Belkaid, Tlemcen 13000, Algeria; m.rouissat@cu-elbayadh.dz
[2] LIMA Laboratory, Univeristy Center Nour Bachir, El-Bayadh 32000, Algeria; m.belkheir@cu-elbayadh.dz (M.B.); a.mokaddem@cu-elbayadh.dz (A.M.)
[3] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
* Correspondence: skiety@qu.edu.sa

Abstract: Internet of Things (IoT) networks are being widely deployed for a broad range of critical applications. Without effective security support, such a trend would open the doors to notable security challenges. Due to their inherent constrained characteristics, IoT networks are highly vulnerable to the adverse impacts of a wide scope of IoT attacks. Among these, flooding attacks would cause great damage given the limited computational and energy capacity of IoT devices. However, IETF-standardized IoT routing protocols, such as the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL), have no relevant security-provision mechanism. Different variants of the flooding attack can be easily initiated in RPL networks to exhaust network resources and degrade overall network performance. In this paper, a novel variant referred to as the Destination Information Object Flooding (DIOF) attack is introduced. The DIOF attack involves an internal malicious node disseminating falsified information to instigate excessive transmissions of DIO control messages. The results of the experimental evaluation demonstrated the significant adverse impact of DIOF attacks on control overhead and energy consumption, which increased by more than 500% and 210%, respectively. A reduction of more than 32% in Packet Delivery Ratio (PDR) and an increase of more than 192% in latency were also experienced. These were more evident in cases in which the malicious node was in close proximity to the sink node. To effectively address the DIOF attack, we propose a new lightweight approach based on a collaborative and distributed security scheme referred to as DIOF-Secure RPL (DSRPL). It provides an effective solution, enhancing RPL network resilience against DIOF attacks with only simple in-protocol modifications. As the experimental results indicated, DSRPL guaranteed responsive detection and mitigation of the DIOF attacks in a matter of a few seconds. Compared to RPL attack scenarios, it also succeeded in reducing network overhead and energy consumption by more than 80% while maintaining QoS performance at satisfactory levels.

Keywords: network security; wireless networks; Internet of Things (IoT); energy efficiency

check for updates

Citation: Rouissat, M.; Belkheir, M.; Alsukayti, I.S.; Mokaddem, A. A Lightweight Mitigation Approach against a New Inundation Attack in RPL-Based IoT Networks. *Appl. Sci.* **2023**, *13*, 10366. https://doi.org/10.3390/app131810366

Academic Editor: Baraq Ghaleb

Received: 19 August 2023
Revised: 9 September 2023
Accepted: 14 September 2023
Published: 16 September 2023

1. Introduction

The provision of full security support is still a pressing challenge for Internet of Things (IoT) networks [1–3]. Current statistics indicate that the number of security threats targeting IoT networks exceeded 112 million in 2022, resulting in an increase of about 87% compared to 2018 [4]. Kaspersky Lab reported an increase of 80% and 50% in the number of DDoS attacks during Q1 of 2020 compared to that in Q1 and Q4, respectively, of 2019 [5]. Earlier, SonicWall reported that more than 34 million IoT malware attacks happened in 2019 and the number rose to 56.9 million attacks in 2020, with an increase of more than 65% [6]. Symantec reported that the average number of monthly attacks on IoT devices was approximately 5200 between 2017 and 2018 [7]. Therefore, the paramount importance of addressing effective IoT security and resilience becomes highly evident, provided that IoT networks are being extensively adopted in critical domains [8–10]. These include industry [11], agriculture [12], and healthcare [13], in which IoT networks emerge as promising solutions for smart monitoring, control, and automation. This has caused the number of IoT-connected objects to exponentially grow in recent years. The growth forecast for IoT devices in 2030 is very high, with an estimation of more than 30 billion devices [14].

The growing and widespread adoption of IoT networks in our daily lives would bring more challenging security threats than in the case of traditional networks. An estimated cost of 400–500 billion was related to cybercrime around the globe in 2015, whereas the figure increased six-fold to 2–3 trillion in 2016 [15]. In addition to serious economic damages, the potential operational damages of such threats would be significant, leading to critical network collapses and complete communication disruption. IoT devices are characterized as Low-Power and Lossy Networks (LLN) devices of constrained resources and limited capabilities. These inherent characteristics make IoT devices highly vulnerable to the adverse impacts of a wide range of IoT attacks such as blackhole, sinkhole, and selective forwarding attacks [16]. Considering the limited computational and energy capacity of IoT devices, another attractive attack would be flooding attacks. Overwhelming the network with a high volume of unnecessary communications would result in effective and rapid Denial of Service (DoS) [17,18].

Flooding attacks present a serious security challenge that can exist in different forms. These include the HTTP flood attack at the application layer [19] as well as the TCP-SYN and UDP flooding attacks at the transport layer [20–22]. The flood attack can also be performed at the network layer by flooding the network with ICMP messages [23,24]. The emergence of such attacks is driven by the lack of full adherence to strict security requirements. Even for a standardized LLN routing protocol such as the IPv6 Routing Protocol for LLNs (RPL), no sufficient security support against a variety of attacks is provisioned. Its potential vulnerability to flooding attacks is still an open security challenge. A flood attack can easily be initiated by any malicious node in an RPL network. To give an example, initiating a hello flood attack only requires a node to broadcast an excessive amount of certain RPL messages [25]. As a result of having no mechanism to mitigate such attacks, RPL would severely struggle with the adverse consequences for the overall network performance and stability.

Different variants of flooding attacks have emerged in RPL networks. Operating at the network layer, they mostly rely on flooding the network with different types of ICMPv6 messages. In this paper, we introduce a new RPL flooding attack that adopts the same approach. It is based on highly frequent transmission of the topology discovery ICMPv6-based messages of RPL, namely the Destination Information Object (DIO) messages. The attack is referred to as the DIO Flooding (DIOF) attack, and it aims at the development of DoS situations over the entire network. The novelty of the DIOF attack lies in the introduction of a new flooding strategy that targets overall network performance and lifetime. It is based on simply tempering the timing configurations of DIO transmissions in a way that ensures high increases in the generation and forwarding rates of control traffic. The ability of the attack to spread its impact across the whole topology makes it superior to common RPL flooding attacks, as indicated by the evaluation results. Addressing such an emerging security threat becomes critical to expediting the widespread deployment of RPL-based IoT networks.

Therefore, the incorporation of effective security support against the DIOF attack into RPL functionality is also addressed in this paper. Although provisioning sufficient security support is overlooked, the protocol design of RPL allows for enough room for improvement and the incorporation of additional security support. Accordingly, we propose a DIOF-Secure scheme for RPL (DSRPL), which enhances RPL functionality with efficient security-oriented procedures. It provides an efficient and lightweight security solution with only simple in-protocol modifications. It is based on incorporating an effective collaborative and distributed scheme to detect and mitigate DIOF attacks by introducing slight modifications to the DIO processing procedure. It ensures no blind involvement in the process of updating DIO timing configurations and allows for applying only the updates that are successfully verified during a specific verification interval. Although it does not eliminate the launch of attacks, DSRPL provides an effective mechanism to contain them and isolate the attacking node in a responsive manner.

This research work offers a three-fold contribution. First, it introduces the DIOF attack, a new variant of the RPL flooding attacks. Second, it presents a novel lightweight collaborative and distributed security scheme, namely DSRPL, to address the DIOF attack. Third, an extensive investigation of the effects of the DIOF attack in addition to an experimental evaluation of DSRPL efficiency, considering varying attack scenarios, is provided.

The following section, Section 2, presents an overview of the operation and security of the standard RPL. In Section 3, a research overview of the related work is provided. Section 4 introduces the DIOF attack and the proposed DSRPL is presented in Section 5. The evaluation methodology adopted in this work is presented in Section 6. The collected results are presented in Section 7 and further discussion is provided in Section 8. Section 9 concludes this paper.

2. RPL Overview

The main characteristics of IoT networks include the deployment of a varying number of small-sized and constrained devices over LLNs. They are typically deployed with limitations in computation, energy, and communication. They can be implemented with CPUs/MCUs of 16 MHz–1 GHz and RAMs of 4 KB–512 MB in addition to being operated using batteries. For example, the Zolertia Z1 device operates at 16 MHz with 92 KB Flash and 8 KB SRAM [26]. The heterogeneity of the IoT devices in terms of sensing, computational, and energy resources is another matter for consideration. IoT networks can also be characterized by ubiquitous deployment on a massive scale.

The interconnectivity among these devices is realized over scarce wireless LLN links without a strict guarantee of communication reliability or high QoS performance. They only provide an effective connectivity solution of low complexity, cost, and energy. A widely adopted communication technology in this regard is IEEE 802.15.4. It operates at the link layer of the LLN architecture and also incorporates an additional LLN-specific layer for realizing effective IP-based communications. This is the IETF-standardized IPv6 over Low Power Wireless Personal Area Networks (6LowPAN) [27,28]. It enables effective integration with the IPv6 infrastructure by facilitating IPv6 adaptation with header compression and fragmentation.

At the network layer, the IETF ROLL working group defined RPL as a primary standardized routing protocol for LLNs. It is specified in RFC 6550 [29], which presents how RPL effectively maintains the routing functionality over scarce LLN links. It is based on distance-vector routing with a proactive mode of operation. The design of RPL completely adheres to the inherent characteristics of LLNs with the full support of three communication schemes. These are point-to-point, multipoint-to-point, and point-to-multipoint.

2.1. RPL Operation

The establishment of an RPL network is based on forming one or more Directed Acyclic Graphs (DAGs). Each DAG contains one or multiple RPL instances constructed with a single or several Destination-Oriented DAGs (DODAG). Each DODAG is formed with one root node representing its data sink, and multiple non-sink nodes interconnected over a multi-hop tree-like topology. The example presented in Figure 1 is for an RPL network with two instances. Two DODAGs exist in Instance_1, whereas Instance_2 shows a single DODAG.

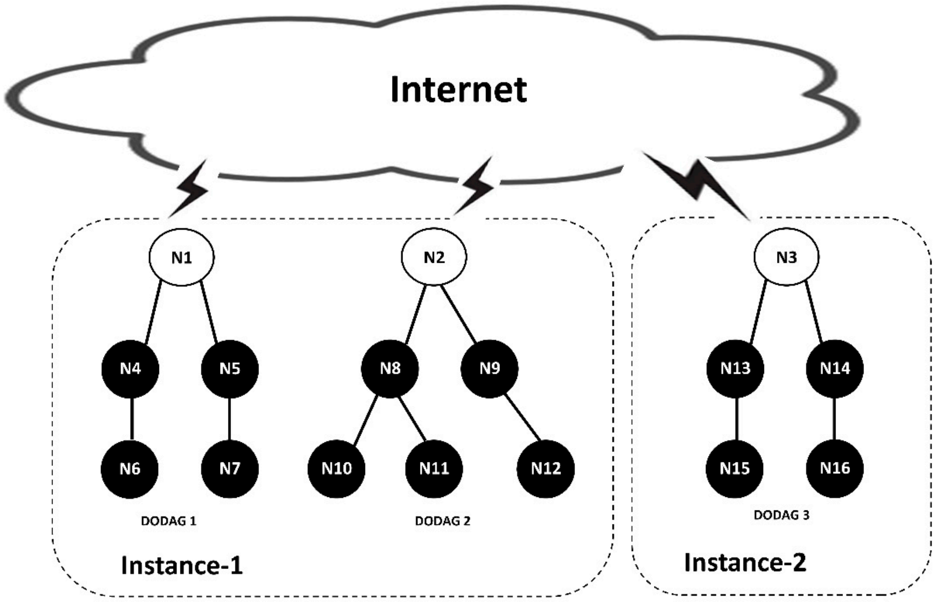

Figure 1. An example of an RPL network.

To effectively facilitate the core routing operation, RPL defines different ICMPv6 control messages. Figure 2 illustrates how RPL operates in an example of a simple RPL network topology. The topology establishment is based on the construction of upward and downward network paths. To initiate this process, the sink node (N1) carries out periodic transmissions of a DODAG Information Object (DIO) message. It contains the necessary routing information to enable successful DODAG discovery and establishment. This includes the Instance ID, DODAG ID, and Version Number (VN), which are used for DODAG identification and topological update tracking.

The DIO message also contains information regarding the applied Objective Function (OF). Each instance is configured with a single OF, which dictates the formation of its DODAGs according to specific routing optimization goals. It enables different routing optimization objectives to be implemented using one or multiple routing metrics. RFC 6551 [30] defines a number of node- and link-routing metrics and constraints that can be utilized to formulate an OF. There are two default OFs defined for RPL. In RFC 6552 [31], Objective Function Zero (OF0) is specified as simply using hop count as a node-routing metric. The other is the Minimum Rank with Hysteresis Objective Function (MRHOF), which is oriented to address network reliability [32]. It incorporates the link-routing metric of Estimated Transmission Count (ETX), which considers the number of transmissions necessary for successfully delivering data packets. Such design flexibility enables customization of the RPL routing functionality to fulfill the network requirements of a specific IoT application.

Once the DIO message of N2 is received by the nearby nodes (N2-N4), the recipients utilize the disseminated information to complete successful attachments to the sink node. After recoding the basic DODAG information, they apply the advertised OF for node ranking calculation and selection of a parent node. Each node performs rank calculation to determine its virtual node-to-sink distance and eliminate routing loops. This results in the rank values increasing as the topological positions of the nodes go deeper into the topology. After that, the node selects a preferred parent (next hop) from the set of neighbor nodes with lower ranks. In this case, each recipient is ranked with *Rank_1* and selects N1 as it is the only available valid candidate. The same process is carried out by N5-N6, which end up being ranked with *Rank_2* and attached to N2 and N3, respectively, as their preferred parent nodes. N7 follows the same procedure and attaches to N3, but after ignoring the DIO advertisement of N6 as being a nearby neighbor with the same rank value.

The ranking position of each node in the network, as illustrated in Figure 2, is a measure of the logical distance between the node and the sink node. Rank calculation is performed based on the following equation:

$$\text{Ranknode} = \text{MHRI} \times (1 + \text{floor}(\text{Rankparent} / \text{MHRI})) \quad (1)$$

where "MHRI" (Minimum Hop Rank Increase) represents the link cost and "Rankparent" is the rank value of a parent node. The objective function, defined by the IETF in RFC 6551 [30], outlines how optimal routing metrics can be utilized to determine the preferred paths toward the sink node. As nodes move further away from the sink node, the "MHRI" value increases, indicating a higher cost or longer distance to reach the sink node.

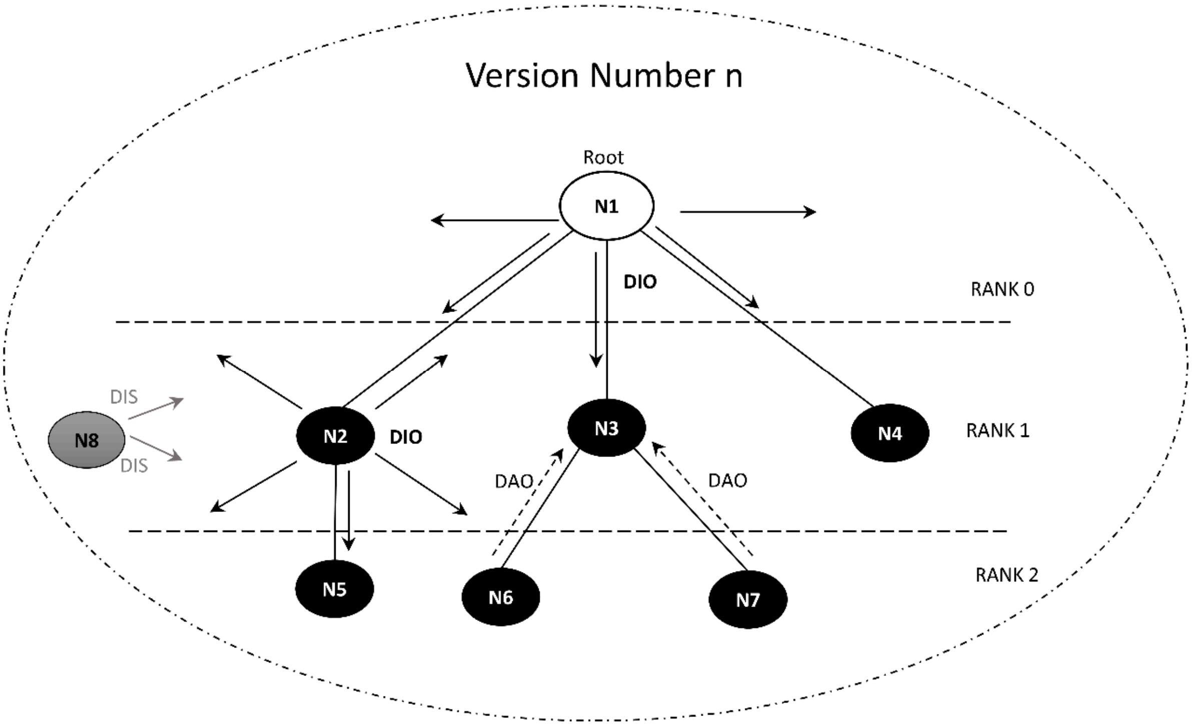

Figure 2. Overview of RPL operation.

In the case of N8, it receives no DIO messages upon running RPL for a while. Thus, it requests to join the network by broadcasting a DODAG Information Solicitation (DIS) message. N2, being in its communication range, then transmits an immediate DIO message in response to the received DIS message. N8 then utilizes the disseminated information to select N2 as a preferred parent and join the DODAG with a rank of *Rank_2*. Note that the reception of the DIS message triggers an immediate DIO transmission without waiting for the next scheduled transmission. That is, the transmission of DIO messages is regulated by the trickle algorithm, as specified in RFC 6206 [33]. A trickle timer is set with a value that is dynamically adjusted whenever a topological change occurs. The algorithm defines the minimum transmission interval value, which is set for the DIO transmissions at the initial stage. If the topology remains stable, the interval of the trickle timer is exponentially increased until reaching a predefined maximum interval value. This helps in minimizing the exchange of DIO messages, thereby optimizing the utilization of node resources. Otherwise, the time interval is reset to start the process over. Such a reset can be triggered by different events, such as the selection of a new preferred parent, reception of new VN updates, and solicitation of DIO broadcasting.

Upon joining the DODAG, each node participates in the establishment of downward network paths. It transmits a Destination Advertisement Object (DAO) message using the already established default route with its parent node. The message carries essential information regarding the parent–child relationship and IPv6 address, in addition to other necessary parameters, to disseminate routing information all the way to the sink node. Two modes were defined by RPL for downward routing. The first is the storing mode with a fully stateful operation that requires the routing information to be stored by each node in its routing table for all the nodes within its sub-DODAG. This would enable the routing of data packets via the common ancestors of the local source and destination. In this mode, each received DAO message is processed and then forwarded all the way to the sink node. In the example presented in Figure 2, the DAO message transmitted from N7 to N3 would allow N3 to use the disseminated information for routing data packets destined from N7 to N6 in the future. The second mode is the non-storing mode, with source routing that allows only the sink node to locally route data traffic across the network. There is no need for any other node in the DODAG to keep a relevant routing state and maintain related routing information in its routing table.

In addition, failures of an RPL node or network link are addressed by two different RPL procedures. The first is local repair, which allows for the immediate switch to an alternative preferred parent node in response to a detected failure. The other procedure is global repair, which is based on addressing failures by initiating full DODAG topology reconstruction. The process is initiated by the sink node updating the current version of the topology. These procedures help in resolving routing problems such as routing loops and inconsistencies.

2.2. RPL Security

The standard RPL design still lacks sufficient security support against different types of attacks. Securing RPL networks is a complex and multifaceted problem requiring solutions capable of balancing multiple objectives, such as rapid detection, attacker isolation, communication overhead, and energy consumption. In this context, multi-objective optimization algorithms can assist in faster and more secure decision-making, allowing the identification of the best security solution that meets the diverse demands imposed on IoT-based RPL networks [34].

However, RPL is developed with only limited protection from external security attacks [35]. Three basic security modes are defined for RPL: authentication, preinstalled, and insecure modes. The authentication process is carried out using a security key collected from an authentication authority in the authentication mode. It prevents any node from attaching to an RPL network and establishing data communications unless it has been successfully authenticated. In the preinstalled mode, secure data communications are established using preinstalled security keys. The insecure mode allows data communications over RPL networks without any security provision [10].

However, no effective security support against internal routing attacks is provisioned in the standard RPL specification [36,37]. It provides no security mechanism to defend against common routing attacks such as blackhole and sinkhole attacks, or against the RPL-specific attacks including version number, rank, and DIS Flooding (DISF). Although malicious RPL network access can be limited using the authentication mode, it is still highly possible to have an RPL node compromised for internally initiating a routing attack. Such potential vulnerability puts it at permanent risk of facing multiple serious security threats. The topology, stability, and overall network performance of RPL networks would be adversely affected as a result of these attacks [38,39].

Most of these routing attacks require the attacking node to join a DODAG before being able to initiate the attack. For example, the rank attack is based on modifying the rank value in the DIO message to deceive the recipients. This can only be possible if the attacking node has already attached to the DODAG and participated in broadcasting DIO messages. The same considerations apply to other examples, such as the version number and worst parent attacks. However, the DISF attack can be initiated by an outsider malicious node that has no complete attachment to the targeted DODAG. It can be regarded as an outside inundation attack that creates a more serious security threat to plain RPL networks.

The DISF attack is based on one of the main operational properties of RPL, which is the solicitation of routing information using DIS messages. This property enables an RPL node to request the information of a nearby DODAG and discover the possible attachment to that DODAG. However, no limitation is imposed by the standard RPL design on the number of DIS messages that can be issued by a node. Such an operational gap can be easily exploited to initiate a high number of DIS transmissions with a DoS-like volume. This would potentially overwhelm the targeted RPL node and adversely affect overall network performance. Having this frequently performed would magnify the effect and degrade network performance noticeably. If this happens in different parts of the network, the network then becomes vulnerable to serious network collapse. As a result, high increases in data loss would be incurred, leading to complete communication disruption.

3. Related Work

The lack of effective security support in RPL functionality has led to a number of basic RPL-specific routing attacks. Examples are VN, rank, DIO suppression, DAO inconsistency, and worst parent attacks [40,41]. These attacks have been reviewed in various research works [42–44]. They can lead to critical deterioration of the overall performance of the targeted RPL networks, as shown in different performance analysis studies. For example, the simulation results in [45] demonstrated how adverse the effect of VN attacks on QoS performance is. It led to a considerable reduction in PDR in addition to considerable increases in network overhead and delay. The VN attack can also incur high power consumption, as shown by the experimental results in [46,47]. The experimental study in [48] also demonstrates RPL vulnerability to high increases in energy consumption and network overhead as a result of the rank attack. In [49], replay attacks caused adverse impacts on energy consumption and QoS performance. The comprehensive study of these attacks in [50] also indicates the severe damages that can be realized if these attacks take more adverse forms.

However, flooding attacks present serious DoS threats to RPL networks. Different evaluation studies have shown the potential damages of flooding attacks. A common example is the DISF attack, which can adversely decrease PDR and increase delay and energy consumption as indicated by the experimental results in [51,52]. The results in [53] demonstrated the ability of the DISF attack to highly degrade energy consumption in addition to nodes' DODAG attachment time. In addition to the high impact on energy consumption, the DISF attack can incur noticeable increases in network overhead as discussed in [54].

Compared to other attacks, the DISF attack would put IoT networks in greater danger of imminent collapse and under a serious risk of complete communication disruption. The experimental study in [55] showed that the DISF attack can have more adverse impacts on QoS performance and network overhead when compared to the VN and worst parent attacks. The DISF attack also led to higher energy consumption compared to the VN and rank attacks, as indicated by the results in [56].

In addition, novel internal routing attacks have emerged in recent years due to the inherent protocol design of RPL. Examples are loophole attacks [57], DODAG partitioning attacks [58], and DAO induction attacks [59]. All of these attacks exploit different security gaps in the RPL design to threaten the stability and QoS performance of RPL networks. In addition, new variants of the flooding attack, such as the multicast-DIS attack [54] and spam-DIS attacks [60], have emerged in RPL networks. These attacks introduced new flooding strategies by targeting specific nodes or segments of a DODAG. A similar strategy was also adopted in [61], but with a more dynamic approach to adaptively select different sets of attackers according to certain network considerations. In [62], the vampire attack is based on dropping data packets to instigate an excessive number of error message transmissions across the network. The Hatchetman attack is presented in [63], which has a similar approach of altering control packets to have them dropped and therefore cause a high increase in the transmission of error messages. In [64], a different flooding strategy based on a hybrid attack referred to as Selective Sub-DODAGs Hiding (SSDH) was presented. The attacker first targets a subset of nodes with a rank attack to attract and then isolates them by running an isolation attack based on dropping their DAO messages. Finally, the attacker initiates a flooding attack to exhaust the resources of the isolated nodes.

In this research work, we present a novel flooding attack that exploits the vulnerability of the RPL DODAG establishment process and effectively floods the entire network. It is based on a simple approach of tampering with the periodic exchange of DIO messages. Compared to other attacks, it only requires the attacker to apply simple adjustments to the trickle timing configurations being disseminated in the DIO messages. This would turn the victim nodes into real players in the attack by propagating the falsified trickle information. The damaging effect of the attack would lead to network inconsistencies and generate a huge amount of control overhead.

Recently, different research proposals have been made toward addressing the different RPL routing attacks. Different security provisioning approaches have been adopted in this effort. Some of the proposed solutions incorporated mitigation mechanisms based on machine learning [65,66], whereas others relied on cryptographic mechanisms [67]. Blockchain-based approaches were also proposed for securing RPL networks [68]. In other proposals, additional architectural entities were introduced to the RPL network to provide security support [69]. Although they would provide effective security solutions, all of these approaches would come at the cost of additional design complexity and computational overhead. Therefore, different approaches have been further introduced with simple in-protocol modifications [70–73]. In this work, a very simple approach is adopted to specifically address the newly-introduced attack without adding much to the functionality of RPL.

4. The DIOF Attack

The reception of a DIO message causes the recipient to carry out different operations. These include DODAG information retrieval, rank calculation, and parent selection. The recipient also needs to forward the message further up to its parent node. During stable and steady network conditions, DIO messages would convey no important routing updates. These operations then become unnecessary and would only overuse node's resources. Therefore, there is no need for a frequent exchange of DIO messages across the network in such cases.

Accordingly, RPL provisions an important operational feature that keeps DIO transmissions at the necessary level. DIO transmissions are managed based on a trickle-timed policy without affecting DODAG topology maintenance. Only specific events trigger the frequent DIO transmissions for a limited duration. These include the initiation of a new version number for the DODAG, changes to the preferred parent, and reception of a DIS message. Otherwise, DIO messages are transmitted with an increasing transmission interval. Such a strategy is critical to maintain communication, keep processing overhead to a minimum, and maximize protocol efficiency.

However, an intruder can simply violate such a protocol policy to launch a new form of flooding attack. This is the DIOF attack, which is a newly investigated RPL-specific attack in this paper. It can be regarded as a form of a DoS-oriented flooding attack that targets the overall performance of RPL networks. Since no restrictions are imposed by RPL to guarantee trickle-timed DIO transmissions, the attack can be easily realized by any node in an RPL DODAG. With no special requirements and restrictions, the attacking node only needs to join the DODAG and participate in DIO broadcasting prior to launching the attack.

The DIOF attack is launched by transmitting frequent DIO messages without adhering to the applied trickle timing configurations. This additionally requires modifying the disseminated trickle timing parameters in the DIO messages. These parameters are included in the DODAG configuration option, which is added to the DIO messages as an additional standard ICMPv6 option. The information is initially set by the sink node and then disseminated across the DODAG without being changed. It must be maintained static and kept unchanged during the propagation of the DIO messages as per the standard RPL specification. However, the attacking node can easily falsify the trickle timing parameters in the messages to deceive the recipients. The falsified information causes illegitimate trickle timing configurations to be disseminated across the network.

These messages are then received and processed by the attacker's neighbors as legitimate ones, since RPL has no means to detect such protocol violations. Further frequent transmissions of the falsified DIO messages are then carried out by the recipients. This makes the victim nodes act similarly to the attacking nodes without being aware of the situation. As a result, expansion of the frequent DIO transmissions would take place across the network. The network is then flooded with a high number of unnecessary DIO messages.

The trickle algorithm defines different main configuration parameters. Two important ones are "*I-min*" and "*I-max*", which limit the minimum and maximum time interval of DIO transmissions. These also include a counter, which controls the increase in the transmission interval by doubling its value. To implement the DIOF attack, the doubling procedure needs to be deactivated and the maximum interval between two successive DIO transmissions is set to be the same as the minimum. This is defined as follows:

$$I\text{-}max = I\text{-}min \tag{2}$$

In the example DODAG presented in Figure 2, a DIOF attack can be initiated by any node in the DODAG. Considering N3 as the attacking node, it applies the following process to initiate the attack:

- Set its local *I-max* value to the current value of *Imin*.
- Set the *DIOIntervalDoublings* field in the DODAG configuration option of the DIO message to the same value as the *DIOIntervalMin* field.
- Start transmitting the DIO messages with the modified option.

Once these messages have been received by N2, N6, and N7, the nodes apply the disseminated updates to their local trickle timing configurations. As a result, frequent DIO transmissions are carried out by the attacking node in addition to the victim nodes without the system being aware of the ongoing attack.

Compared to the DISF attack, this attack would have a wider effect on the network. It causes the frequent DIO transmissions to spread beyond the neighboring area of the attacking node. Having the neighbor nodes participating in the process magnifies the effectiveness of the attack. However, the DISF attack triggers the frequent transmission of DIO messages by a single node. Only the neighboring area of the attacking node would then be highly affected. Another consideration that differentiates between these attacks is where the attack can be performed. The DIOF attack is a form of an inside inundation attack that requires the attacking node to join the DODAG in advance. In the case of the DISF attack, the attacking node can perform it without prior DODAG attachment. Moreover, note that the attacks are initiated using different types of RPL control messages. The DISF attack utilizes the DIS message, which is typically of a smaller size than the DIO message being utilized by the DIOF attack.

The major objective of the DIOF attack is to introduce a high volume of unnecessary control traffic to the network. It targets RPL networks by exhausting available resources and causing high degradation to overall network performance. This would also drain node energy and reduce network lifetime, particularly in large-scale network deployments. Moreover, the attack can lead to DODAG inconsistency and disruption, resulting in serious network damages and communication collapses. In addition, having the DIO messages frequently transmitted would open the doors for further security threats in RPL networks. Making effective use of such a situation by incorporating other routing attacks would amplify their adverse effects. Taking the version number attack as an example, this can be easily realized by keeping the version number value incremented for every newly transmitted DIO message. The network would then experience frequent initiations of the global repair process in addition to the DIO flooding situation. This would result in a hybrid attack with a wider and more adverse effect on the overall performance of the network.

5. Solution

As specified in the original RPL specification [29], the trickle timing parameters are exclusively set by the sink node and disseminated in DIO messages. Receiving new trickle timing information requires RPL nodes to blindly accept and apply the updates without further verification. Such behavior allows the initiation of the DIOF attack in a more effortless and less complicated manner. This makes it critical to verify the legitimacy of any trickle timing updates before any further action. Therefore, improving RPL immunity against DIOF attacks requires extending RPL functionality to a further operational level. However, certain considerations need to be taken into account before introducing any modification to the design of RPL. These include the constrained computational and energy resources of typical RPL devices. In addition, it is critical to maintain communication and processing overhead at very low levels in RPL networks.

Accordingly, DSRPL is introduced in this paper to incorporate simple, yet effective, verification and mitigation against the DIOF attack. It provides efficient protection against DIOF attacks based on slight modifications to the DIO processing procedure without adding much to the design complexity of RPL. The DSRPL design incurs no extra hardware requirements or additional architectural entities. It basically extends the standard RPL functionality with a simple distributed and collaborative scheme.

DSRPL is developed based on the original RPL principle that the dissemination of trickle timing information must be preserved from top to bottom. The process must be initiated at the sink node and then continued following a downward direction. Accordingly, any trickle timing update is propagated across a standard RPL network one level after another. Making effective use of such a behavior enables DSRPL to establish a distributed and collaborative scheme within the nodes located at different segments of the DODAG. This is simply based on not trusting any update received by a node until it has been verified that a relevant update is being advertised in a different DODAG segment. Therefore, DSRPL protects the node from blind involvement in applying trickle timing updates and ensures that only legitimate updates are processed. Although this does not stop the initiation of the attack, DSRPL can effectively detect and isolate the attackers in addition to eliminating the adverse effect of the attack on the overall network performance.

Accordingly, the RPL operation is modified to enable effective verification of trickle timing updates in a distributed manner. Once an update is received, DSRPL moves to a legitimacy check stage to decide how to react to the advertised update. This is based on simply collecting and inspecting the relevant information being exchanged in different parts of the network during a specific time interval. Upon the detection of an attack, it moves to the stage of attack mitigation. The modifications introduced to the standard RPL operations can be summarized as follows:

- Preventing blind acceptance of and participation in the process of updating the trickle timing configurations.
- The updates from parent nodes are only applied after successful verification with another separate node during a specific verification interval.
- Isolating and blacklisting malicious parent nodes.

A detailed explanation of the DSRPL operation is provided in the following subsections in addition to the presentation of an illustrative example.

5.1. Preliminary Considerations

This subsection briefly describes the major protocol characteristics and network assumptions considered for the proposed work. The topology of RPL networks is considered to consist of multiple non-sink RPL nodes interconnected with a single sink node. The initiation of the DODAG is carried out by the sink node in the storing mode. During a DIOF attack scenario, one of the non-sink nodes maliciously initiates the attack. The attacking node joins the network legitimately and establishes normal communications with the legitimate nodes. Each node runs an original RPL implementation [29] that is DSRPL-modified.

All the nodes are assumed to be homogeneous and stationary devices that are of small size and limited computing resources. They are also battery powered, which makes it critical to take into consideration energy efficiency and resource utilization. Examples of off-the-shelf RPL-enabled devices in the market are Tmote [74] and MicaZ [75] motes. The deployment of the nodes can be of varying scale and varying density. It can also take different forms, considering random or uniform positioning. Multi-hop wireless connectivity is established among the nodes using scarce wireless links.

The deployment of the nodes is assumed to be for a specific IoT application. Example applications would be smart buildings, industrial automation, and smart cities. Accordingly, IoT devices are equipped with the necessary hardware components, such as sensors and actuators, to allow for the collection of application-specific IoT data. The data are collected and then transmitted by the nodes periodically. The transmission/reception of the data packets is carried out at a predefined time interval over the established RPL paths. The sink node performs the role of a network gateway, over which the data traffic is forwarded to/from the Internet.

The sink node is assumed to be under no exposure to any kind of attack. In the initial stages, the network runs without an ongoing DIOF attack until it reaches a good level of topological stability. The attack is only initiated after the network topology comes to complete convergence. The major objective is to launch a DoS-oriented flooding attack to flood the network with unnecessary communications. The attack is targeted towards imposing critical disruptions to network stability and reliability, with a high volume of control traffic. Consequently, the system would incur a considerable drop in network performance and lifetime.

5.2. DSRPL Operation

Figure 3 presents a flow diagram that provides an overview of the main procedure of processing DIO messages at a non-sink node. It defines two stages, which are indicated by the *must_check* parameter. The first involves the detection of the DIOF attack once the node receives a DIO message with new updates to the trickle timing configurations. In this case, the node proceeds with processing the update if, and only if, the DIO source is its current parent. The update is then accepted and applied if the sender is a sink parent node. In case the sender is a non-sink parent node, it is held up until being verified. The node then moves to the verification stage by setting the *must_check* parameter. The stage is based on verifying the legitimacy of the advertised trickle timer updates through reliable sources. Any other node residing at a different DODAG segment and sharing no parent or child with the node is considered a reliable node that is not affected by the same attacker. The stage starts with running a verification timer and with recording the identity of the current parent as the source of the update (just in case the node makes a parent change during the verification stage).

In the case of receiving a DIO message with a non-parent source, it is discarded to prevent the updates from taking directions other than the downward routing paths. This would help in limiting the impact of any DIOF attack while ensuring the ability to communicate legitimate updates. Note that the message is also discarded directly if the source is already suspicious and added to the node's blacklist, as explained later.

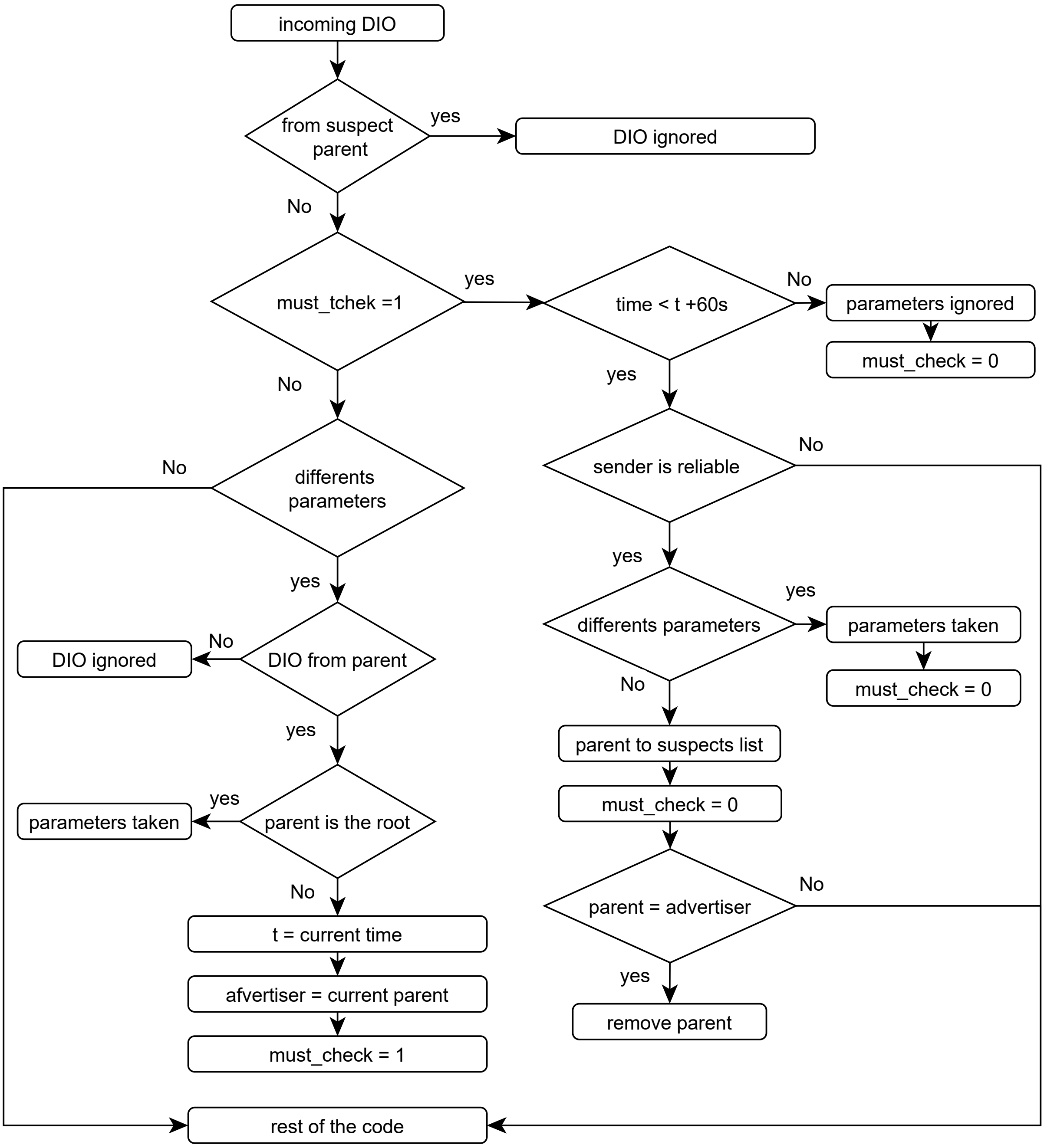

Figure 3. An operational overview of DSRPL.

The verification stage is activated when a new update to the trickle timing configurations is advertised by a non-sink parent node. The DODAG would then be flooded with a high number of DIO messages over different DODAG segments. This can be effectively exploited to quickly verify the ongoing update and identify any possible attack. During a verification interval of 60 s, the node collects the disseminated information to discover whether the situation is suspicious. If the same update information is received from a source located in another segment of the DODAG, it is then considered a sufficient indication of a legitimate update. Since this source shares no parent or child node and DSRPL allows only processing DIO messages from parent nodes, it would indicate that another ongoing relevant update separately exists. This would be sufficient to rest assured that the ongoing update is legitimate to a high degree. In this case, the update is applied, and the verification stage is deactivated by unsetting the *must_check* parameter and stopping the verification timer.

If the node learns from the collected information that no update is being advertised in another segment during the verification interval, it considers the situation fatal and runs the mitigation process. Since the only advertiser of the update in the DODAG is then its parent node, it blacklists the advertising parent node and then deactivates the verification stage. It then moves on to perform a standard local repair and change its suspicious parent node after removing it from its parent set. It runs the parent selection process to select a new parent node. However, there is a possibility that topological changes happened during the verification stage and the node has already attached to a different parent node. In this case, no further actions need to be taken other than blacklisting it, as the node has already detached from the suspicious parent node.

Note that the node moves to the verification stage for a specific duration. A timer of only 60 s is set to ensure a reasonable trade-off of the verification interval. Since DIO messages would be sent more frequently when the network is under a DIOF attack, such an interval is large enough for a verifying node to collect and examine DIO messages from neighbor nodes. It is also short enough not to delay legitimate updates being applied across the network. Nevertheless, it is highly possible that a decision is made quickly as a result of the high DIO transmission rate during the attack. As discussed later in Section 8, DSRPL requires only a few seconds to successfully detect and act upon DIOF attack situations. However, once the verification timer is over without any decision having been made, it means that the node did not find a reliable node for verifying the legitimacy of the received updates. Then, the update is ignored, and the verification stage is deactivated.

5.3. Example Scenario

Further elaboration on how DSRPL operates during a DIOF attack is provided in this subsection, considering the simple example scenario shown in Figure 4. N2 presents the DIOF attacker, which initiates the attack after having the DODAG topology converged for a considerable amount of time. It sends multiple DIO messages (A) containing new trickle timing configurations to propagate falsified updates and flood the network with a high number of falsified DIO messages. Upon reception of the messages, N4 and N5 move into the attack detection stage. Each node runs the DSRPL algorithm, which moves the nodes into the verification stage, since the source of the messages is a non-sink parent node (N2). The *must_check* parameter is set, and the verification timer of 60 s goes off (B). Afterward, any DIO message originating from the same sub-DODAG of N2 will be discarded by the recipients (C).

During the verification stage, N5 receives a DIO message from N6, which is considered a reliable neighbor node located outside its current sub-DODAG (D). It learns from the message that no update to the trickle timing configuration is being advertised across the network. It concludes that there is a very high possibility that the received update is illegitimate and a DIOF attack is being performed by N2. N5 than moves out of the verification stage by unsetting the *must_check* parameter after blacklisting and removing N2 from its parent set (E). Then, it starts the local repair process to detach from N2 and run the parent selection process. As a result, N3 is selected as the new preferred parent to which N5 is now attached, as shown in Figure 4b.

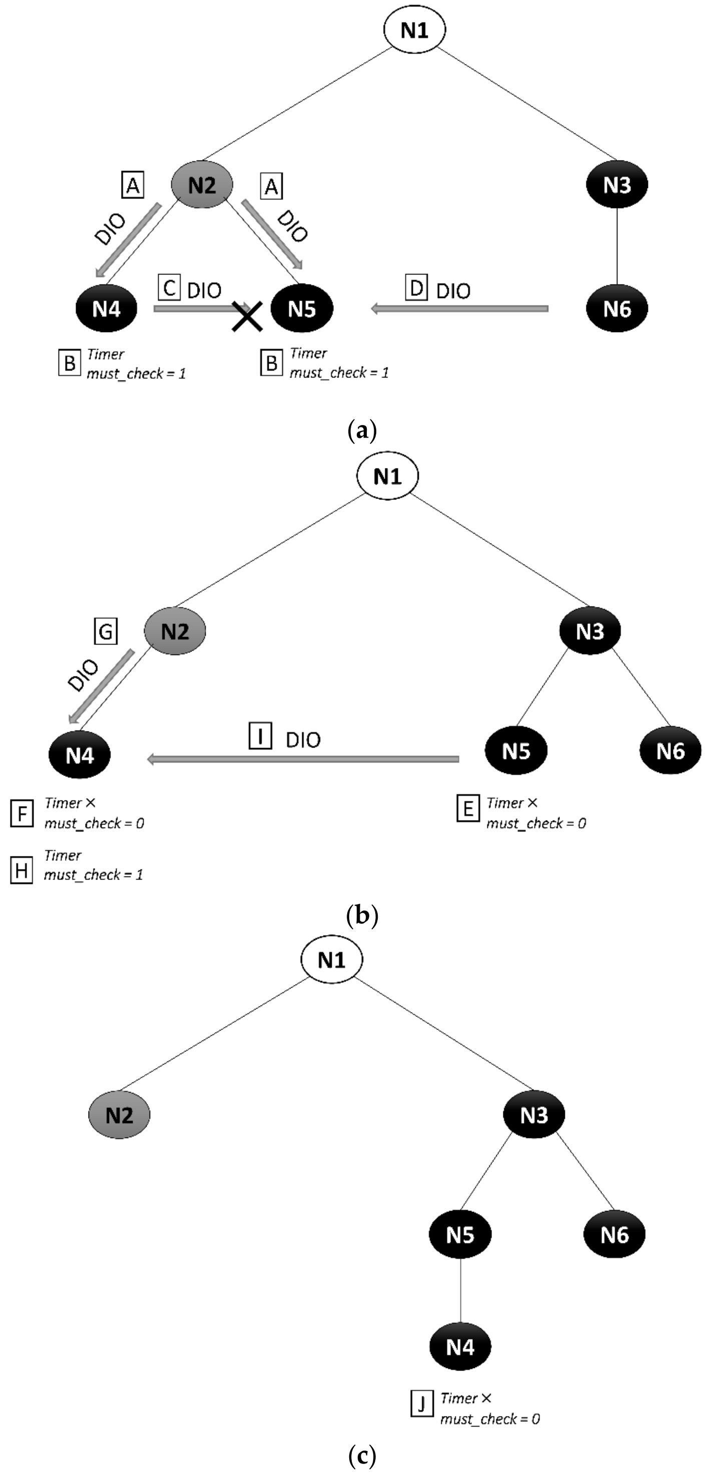

Figure 4. An example of how DSRPL operates during a DIOF attack: (**a**–**c**) demonstrate the different stages of the attack detection and mitigation.

However, the timer expires at N4 without deciding on the legitimacy of the advertised update as it has no valid neighbors to verify the updates with. N4 then ignores the advertised update and moves back to the normal state (F). N2 would then keep transmitting the malicious DIO messages, which would keep N4 under the ongoing attack (G). However, N4 now has a reliable neighbor node, which is N5, after being attached to a different parent at a different segment of the DODAG. This enables N4 to verify the advertised updates after moving to the verification stage again (H) and receiving N5′s DIO messages (I). It then learns that no relevant update is being propagated outside its sub-DODAG and detects the potential DIOF attack. As a result, N4 moves out of the verification stage (J) and proceeds to the mitigation process to firstly blacklist N2. Then, it detaches from the attacker and attaches to N5 after running the parent selection process as shown in Figure 4c. As a result, DSRPL succeeds in isolating N2 and mitigates the adverse effect of the DIOF attack.

6. Evaluation

Due to their constrained capabilities and limited resources, IoT devices are commonly implemented using customized operating systems (OSs). Popular examples are Contiki OS and TinyOS, which provide open-source implementations. They typically come with 6LowPAN- and RPL-enabled network stacks. In addition, Contiki OS provides an additional software component, the Cooja network simulator [76]. It facilitates effective simulation and evaluation of RPL-based IoT setups with varying configurations. It enables the running of simulated RPL networks with various virtual IoT motes operating the real implementation of Contiki OS. This work was implemented and experimented with using the Cooja simulator of Contiki 3.0.

To realize a complete implementation of the DISF attack, DIOF attack, and DSRPL, modifications were introduced to the code base of the RPL implementation in the Contiki OS. These are the header and source files that were modified:

- "rpl-conf.h": the RPL_DIO_INTERVAL_MIN was set to 10 instead of 12 to modify the minimum trickle time interval to approximately 1 s.
- "rpl-private.h": the value of "#define RPL_DIS_START_DELAY" was set to the value of 0 (seconds), which allows the node to start sending a DIS message without waiting 5 s (default value). In addition, the value of (#define RPL_DIS_INTERVAL) was set to the value of 1 (second).
- "rpl-timers.c": the RPL_DIO_INTERVAL_MIN two functions were modified:
  - ○ "handle_periodic_timer (void *ptr)": adding (next_dis++; dis_output(NULL);) in order to make the node send DIS messages without stopping.
  - ○ "handle_dio_timer(void *ptr)": adding "dio_output(instance, NULL);" in order to push the node to keep sending DIOs without stopping. In addition, we deactivated the need to double the DIO interval in (instance->dio_intcurrent++;).
- "rpl-icmp6.c": the function "dio_output(rpl_instance_t *instance, uip_ipaddr_t *uc_addr)" was modified with the value of minimum DIO interval (N1) fixed to 10 (buffer[pos++] = instance->dio_intmin);) instead of the default value, which was 12. These modifications made the malicious node send a DIO message every 1 s and advertise these values to its neighbors.
- "rpl-dag.c": mainly, the DIO processing was modified to implement the introduced functionality of DSRPL.

To investigate and analyze the varying behaviors of the protocol and give a fair, detailed analysis, we used a network topology with 13 stationary nodes, as shown in Figure 5. The node in green is the sink node (Node 1) whereas the nodes in yellow are the non-sink RPL nodes (Nodes 2–13). All of the nodes were configured to run as Zolertia Z1 motes with 16-MHz MCUs, 8KB RAMs, 92KB flash memories, and CC2420 transceivers. In all the experiments, the adopted OF was MRHOF with a routing metric of ETX. Table 1 provides a summary of the main configurations considered in the simulation.

Every RPL node was configured to carry out periodic transmissions of IoT data traffic over UDP. The frequent transmission of the data packet was regulated with a communication interval of ±60 s. All of the data packets were received by a UDP server running at the sink node. Each node was also configured with different Cooja plugins, namely the collect-view and powertrace modules, for the collection of additional experimental simulation data. Other important settings were the communication range and interference range, which were set to 50 and 100 m, respectively.

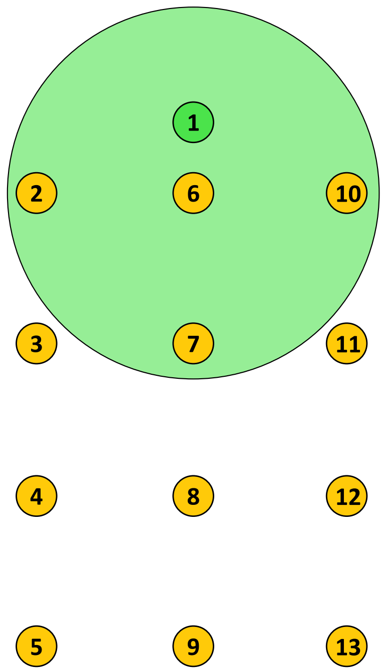

Figure 5. The experimental setup in Cooja.

Table 1. A summary of the simulation parameters.

Simulation Parameter		**Value**
Number of Nodes		13
Area Size		200 × 200 m
Mote		Zolertia Z1
Mote Current Consumption	CPU Mode	0.5 mA at 3 V
	LPM Mode	0.0005 mA at 3 V
	Tx Mode	17.4 mA at 3 V
	Rx Mode	18.8 mA at 3 V
Mote RAM Size		8 kB
Mote ROM Size		92 kB
RTIMER		32,768 ticks /s
Operating System		Contiki 3.0
Radio Medium Model		UDGM: Distance Loss
Simulator		Contiki Cooja
RPL Routing Mode		Storing Mode
RPL Objective Function		MRHOF (ETX)
MAC Layer		ContikiMAC
Interference Range		100 m
Communication Range		50 m
Traffic Type		CBR
Data Packet Size		40 bytes
Control Message Size		4 bytes
Data Transmission Interval		±60 s
Simulation Duration		10 mins

The evaluation methodology was based on four main stages. The first stage was based on running the implementation of the original RPL in the experimental setup with no attacks. This enabled the establishment of the performance baseline required to realize effective comparison with the experimental results obtained in the next stages. In the second and third evaluation stages, different scenarios were carried out by running the DISF and DIOF attacks, separately, over the experimental setup. These helped in investigating and comparing the effectiveness of both attacks. The fourth stage was considered to examine the performance of DSRPL under the DIOF attack. The obtained evaluation results during all of these stages were then processed and analyzed to provide an overall insight into the adversity of the DIOF attack and the efficiency of DSRPL.

Different nodes with varying positions were considered in the attack scenarios. These can be presented as follows:

- Scenario 1: the malicious node is one hop away from the sink (Node 6 is the malicious node).
- Scenario 2: the malicious node is two hops away from the sink (Node 7 is the malicious node).
- Scenario 3: the malicious node is three hops away from the sink (Node 8 is the malicious node).
- Scenario 4: the malicious node is four hops away from the sink (Node 9 is the malicious node).

All the attack scenarios were considered during the second and third evaluation stages whereas only Scenario 1 was considered for the fourth stage, as it was the most challenging scenario. A simulation duration of 10 min was set for each simulation, which was run 10 times. Then, the performance results were collected and averaged. Figure 6 provides an overview of the adopted evaluation methodology.

For effective analysis of the overall network performance, the evaluation was based on different network measurement parameters. These can be categorized as follows:

- QoS-oriented performance: Packet Delivery Ratio (PDR) and latency.
- Network overhead: number of control packets (DIO, DAO, and DIS).
- Energy efficiency: energy consumption.
- Memory occupancy: memory footprint of DSRPL.

The calculation of these performance parameters was based on a well-defined measurement. This can be simply explained, as follows:

- PDR: the proportion of the total number of data packet transmissions to the total number of data packets successfully received at the sink node.
- Latency: the average amount of time taken by the transmitted data packets to successfully reach the sink node without considering dropped and lost packets.
- Control overhead: the total number of control packets being exchanged over the network. This was calculated as follows:

$$Control\_Overhead = \sum_{1}^{n} DIS + \sum_{1}^{n} DIO + \sum_{1}^{n} TOTAL\_DAO \tag{3}$$

where TOTAL_DAO refers to the total DAO Packets being exchanged, including the Regular-DAO and the No-Path DAO.

- Energy consumption: the energy consumption data provided by the powertrace module were used to perform the following calculation:

$$Consumed\_energy = \frac{Energest_{value} * current * voltage}{Rtimer\_\sec ond} \tag{4}$$

where $Energest_{value}$ is the total number of ticks during a given energy mode, and RTIMER, Current, and voltage for the Z1 motes are as given in Table 1.

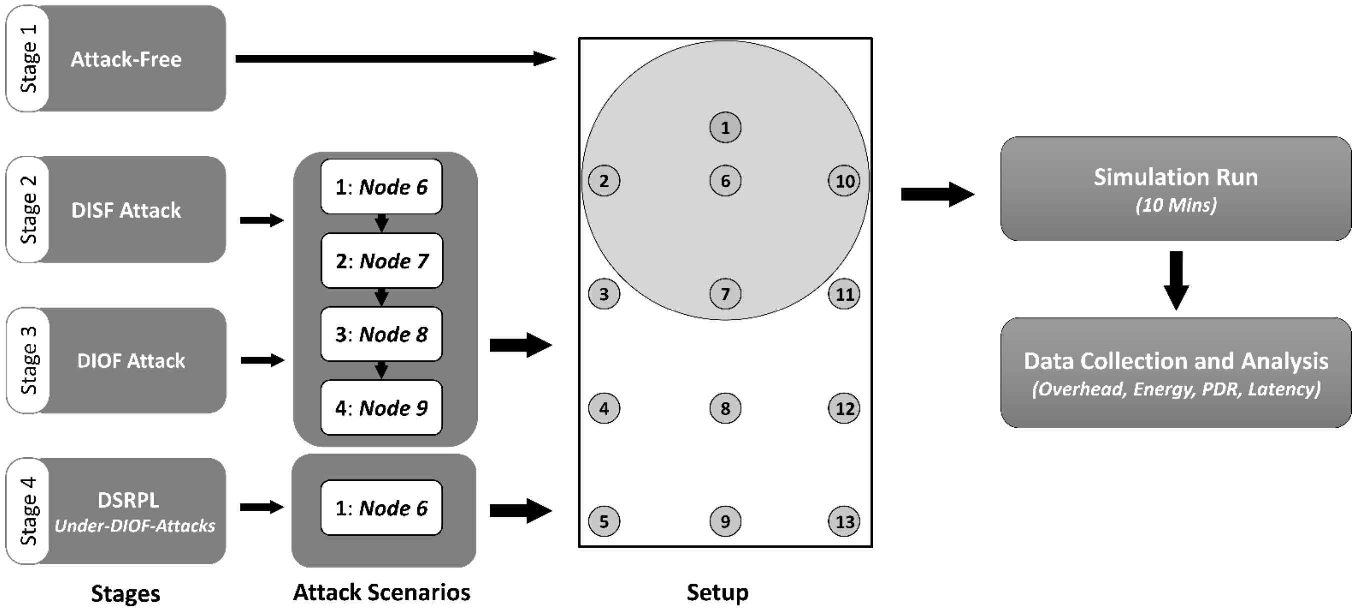

Figure 6. An overview of the adopted evaluation methodology.

7. Results

7.1. DISF Attack Results

Table 2 shows that the DIS transmission rate increased to the same number for all of the DISF attack scenarios. As a result of such increases, the number of control messages being generated and forwarded across the network increased noticeably. Regarding the DIO transmissions, an increase of more than 200% was experienced as a result of the attack. The numbers of generated and forwarded DAO messages were also higher by more than 60% and 150%, respectively, in all of the attack scenarios. It can also be seen that the number of the No-Path DAO messages, which are used to invalidate downward routes, was also increased. This indicates how unstable the network became due to the routing inconsistency incurred by the attack.

The results in Table 2 illustrate the impact of the attacker's position in the DODAG. The closer the attacker is to the sink node, the higher the DIO and DAO generation rates. For example, the attacker in Scenario 1 is connected directly to the sink node, which resulted in more than 90% higher DAO generation rates compared to the other attack scenarios. However, the number of DAO messages being forwarded in Scenario 1 decreased by more than 7%. That is, the DAO forwarding rate increased as the attack was initiated away from the sink node. This is reasonable, as the DAO messages propagated from the bottom all the way to the top of the DODAG were forwarded level after level.

Another important consideration is the number of neighbor nodes around the attacker. This had a noticeable effect on the number of DIO and DAO transmissions in Scenario 4. Compared to the other attack scenarios, fewer neighbor nodes (only three nodes) were positioned close to the attacker, which is also considered a leaf node with no child nodes. As a result, lower message generation and forwarding rates by more than 23% and 11%, respectively, were experienced.

Figure 7 shows that the DISF attack led to an overall increase of more than 100% in energy consumption. As discussed above, this increase varied as the attacker became closer to the sink node and had more neighbor nodes. Therefore, the attacks in Scenario 1 and Scenario 2 incurred similar increases in energy consumption, whereas it was higher by more than 19% in Scenario 3. The least energy was consumed in Scenario 4, as the attacker is involved with the least number of neighbor nodes.

It can also be seen that the most energy-consuming mode is RX in the attack-free scenario, and less time was spent in transmitting control messages in the TX mode. A difference of more than 90% was experienced since the messages were wirelessly received from all the neighbor nodes, irrespective of being relevant. However, the attack pushed the nodes to use the TX mode more frequently to transmit the attack messages. As a result, Figure 7 shows comparable results considering the RX and the TX modes. The nodes were involved in both receiving and transmitting DIO and DAO messages at similar rates.

Table 2. Network overhead results: DISF attack.

Scenario	Generated Messages				Forwarded Messages		Total
	DIS	DIO	DAO	No-Path DAO	DAO	No-Path DAO	
Normal	12	145	95	6	142	5	405
Scenario 1	608	603	543	9	409	25	2194
Scenario 2	608	580	278	8	439	17	1930
Scenario 3	608	585	324	7	787	24	2335
Scenario 4	608	463	158	3	361	8	1601

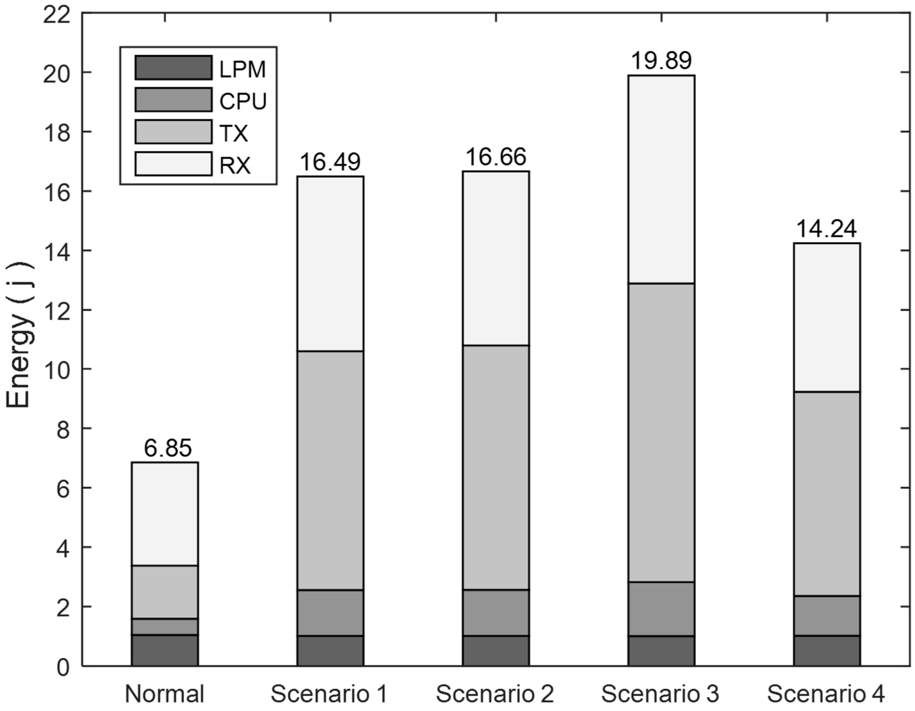

Figure 7. Energy consumption results: DISF attack.

7.2. DIOF Attack Results

Table 3 demonstrates the impact of the DIOF attack on increasing the transmission rate of different control messages. Compared to the attack-free scenario, a noticeable increase of more than 700% in the number of transmitted DIO messages was experienced. The DAO generation and forwarding rates were also increased by more than 160% and 300%, respectively. Notice that the DIOF attack led to higher overall DIO and DAO transmissions of more than 100% compared to the DISF attack. For example, the DIOF attack in Scenario 3 incurred an additional 2868 DIO messages more than the DISF attack in the same scenario. This was due to the adverse effect of the DIOF attack, reducing the transmission interval and increasing the generation rate of DIO messages across a wider area of the network.

As the DIOF attack focuses on creating DIO floods, the DIO messages were the main contributor to this huge increase in network overhead. The proportion of the number of DIO messages to the total number of control messages was up to 83% of the total, whereas it was up to 36% in the case of the DISF attack, considering all the scenarios. In addition, notice that the DISF attack imposed almost the same amount of DIS messages, whereas the number of DIS messages remained at the same low amount irrespective of the DIOF attack. This is evident, as the DIOF attack relied only on the DIO messages to flood the network with larger-sized messages.

The results also show that the impact of the attack is inversely proportional to the position of the attacker. As the attacker was closer to the sink node, the increases in the DIO and DAO transmission rates were more noticeable. For example, the attack by node 6 in Scenario 1 caused an increase of more than 16% in the network overhead compared to the attack by Node 7 in Scenario 2. This is also more evident for Scenario 3, in which the attack incurred more than a 90% increase in the control message transmissions than the attack in Scenario 4. In addition to the attacker position, this case also shows the effect of the number of neighbor nodes surrounding the attacker, as discussed before.

Flooding the network with mostly DIO messages instead of DIS messages in the case of the DIOF attack led to a high impact on energy consumption. That is, a DIO message has a larger size than a DIS message, thus taking up more reception and transmission time. As a result, the DIOF attack consumed high energy levels as shown in Figure 8. It resulted in an increase of more than 800% in energy consumption, whereas the DISF attack incurred up to 190%. The adverse effect on energy consumption was amplified as the attacker came closer to the sink node. For example, more than 7000 mj was consumed as a result of the attack in Scenario 1, compared to the effect of the attack in Scenario 2.

Table 4 presents comparisons of the PDR and latency results of the DISF and DIOF attacks. It is clear that the DIOF attack had an adverse impact on PDR, with a high reduction of up to 38%, whereas the DISF attack resulted in at most a 3% reduction in PDR. The latency results also show that the network experienced higher latency during the DIOF attack than in the case of the DISF attack. For example, the DIOF attack increased the latency by more than two seconds in Scenario 2, whereas the highest increase incurred by the DISF attack was almost 300 ms in Scenario 3. It can be noticed that the attacks in Scenario 1 and Scenario 2 yielded the most effects on PDR and latency. This was due to the attackers having closer positions to the sink node and a higher number of neighbor nodes.

Table 3. Network overhead results: DIOF attack.

Scenario	Generated Messages				Forwarded Messages		Total
	DIS	DIO	DAO	No-Path DAO	DAO	No-Path DAO	
Normal	12	145	95	6	142	5	405
Scenario 1	12	5698	525	35	596	15	6881
Scenario 2	12	4555	516	38	744	25	5890
Scenario 3	12	3453	348	19	862	6	4700
Scenario 4	12	1235	250	12	894	25	2428

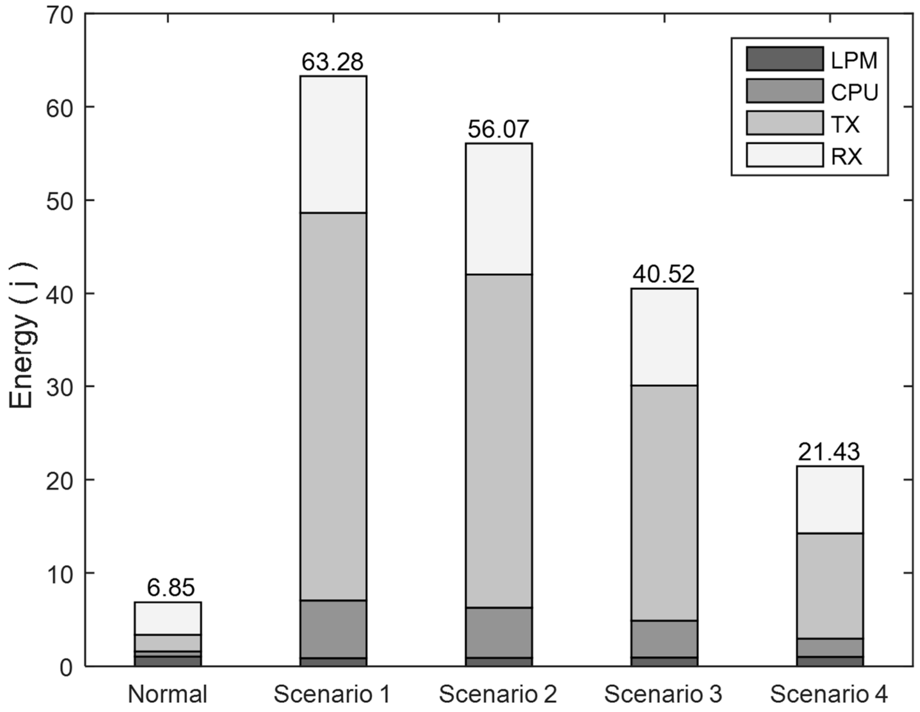

Figure 8. Energy consumption results: DIOF attack.

Table 4. PDR and latency results.

Scenario	**DISF Attack**		**DIOF Attack**	
	PDR (%)	**Latency (s)**	**PDR (%)**	**Latency (s)**
Normal	100	0.139	100	0.139
Scenario 1	100	0.386	62	1.876
Scenario 2	97	0.375	68	2.231
Scenario 3	100	0.434	95	0.974
Scenario 4	99	0.224	100	0.406

7.3. DSRPL Evaluation Results

The adverse effect of the DIOF attack on increasing the DIO and DAO transmission rates as shown in Table 5 is challenging for the standard RPL networks. More than 5500 and 400 additional DIO and DAO messages, respectively, in total were exchanged across the network as a result of the attack. It was very hard for the standard RPL to mitigate the DIOF attack and maintain the network overhead to a normal level. Rather, it made it easy for the attack to spread out, since no mechanism was provisioned to stop forwarding malicious DIO messages and propagating falsified trickle timing information.

Table 5. Network overhead results: DSRPL.

Scenario	**Generated Messages**				**Forwarded Messages**		**Total**
	DIS	**DIO**	**DAO**	**No-Path DAO**	**DAO**	**No-Path DAO**	
RPL	12	145	95	6	142	5	405
DIOF	12	5698	525	35	596	15	6881
DSRPL	12	619	86	4	161	1	865

To address such a critical security gap, DSRPL provides a mechanism that prevents any malicious activity and allows for only the propagation of legitimate DIO messages. It succeeded in addressing the DIOF attack without much of an increase in the network overhead. Less than 500 and 50 additional DIO and DAO messages, respectively, in total were transmitted during the attack. Note that most of these DIO messages were initiated by the attacker before the attack was detected and contained. However, DSRPL managed to reduce the DIO generation rate by more than 89% compared to the case of the attacked standard RPL network. The overall DAO transmission rate was also reduced by more than 72%. It can be noticed that DSRPL maintained high routing stability and kept the No-Path DAO transmission rate to a minimum.

The results in Figure 9 illustrate the ability of DSRPL to reduce energy consumption by more than 80% compared to the standard RPL under the DIOF attack. It maintained a close level of energy consumption to the attack-free scenario. Only about 5 joules were added to the total consumed energy, whereas the standard RPL failed to provide satisfactory results.

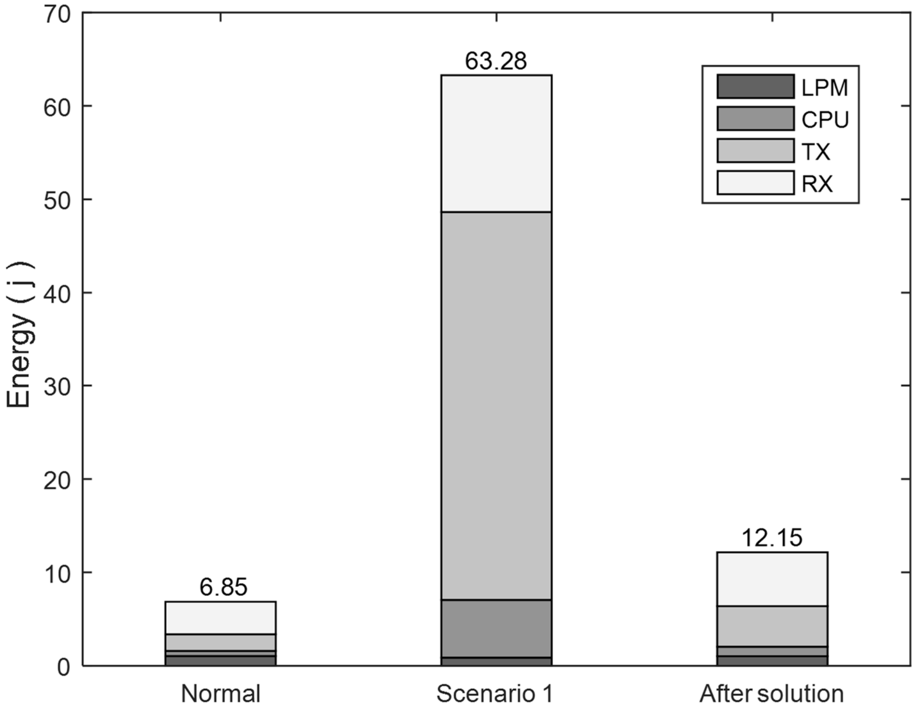

Figure 9. Energy consumption results: DSRPL.

DSRPL allowed the nodes to be relatively less involved in the control message transmission activities during the attack. It can be noticed in Figure 9 that DSRPL enabled the maintenance of the same behaviors observed in the attack-free RPL network. Most of the energy was consumed during the RX mode, whereas less energy consumption was experienced during the other modes. This helped in delivering similar results to the attack-free RPL scenarios. However, the standard RPL showed a high increase in energy consumption under the TX mode during the attack. That is, DSRPL allowed the nodes to spend much less time transmitting than receiving control messages. This is important to keep the energy consumption to a minimum, since the TX mode is the most energy-consuming.

Table 6 demonstrates the effectiveness of DSRPL in addressing the DIOF attack while maintaining high QoS performance. It was able to sustain high PDR while the standard RPL network experienced a reduction of 38% during the attack. In addition, DSRPL succeeded in reducing the latency experienced by the attacked RPL network close to the level of the attack-free results. It managed to add much less to network latency and achieve a reduction of more than 84% compared to the standard RPL under the attack. The slight increase of 150 ms in latency was mostly caused by the congestion at the forwarder nodes, since the routes through the malicious node were discarded. However, this lasted for a very short time thanks to the immediate verification mitigation actions taken by DSRPL.

Nodes are typically deployed in RPL networks with built-in limitations. They mostly come with constrained Flash Memory (ROM) and Random Access Memory (RAM). Regarding the Z1 nodes in our simulations, the RAM is limited to 20 kilobytes and the ROM is limited to 100 kilobytes. Therefore, adopting solutions with a minimal footprint is crucial, owing to the restricted capacity of the available memory in such nodes.

Table 7 compares the memory occupancy of the standard RPL and the modified RPL after implementing the proposed mitigation algorithm. It shows that DSRPL introduces only 392 bytes of extra ROM, presenting an increase of only 0.84%. Regarding the usage of the RAM, DSRPL only occupied an additional RAM footprint of 0.76%. These findings demonstrate that DSRPL is a lightweight, efficient solution and very well-suited to constrained LLN networks.

Moreover, it is important to understand how responsive DSRPL was to the ongoing attack. When the attack was initiated by Node 6 in Scenario 1, Node 7 was the first node to receive the malicious DIO messages. This made Node 7 move into the verification stage, during which it discovered the ongoing attack and switched its preferred parent to Node 2. It learned from the messages received from Node 3 that no relevant update was being propagated across the network. These actions took Node 7 only less than 4 s, which indicates the ability of DSRPL to adapt responsively to DIOF attack situations.

Table 6. PDR and latency results: DSRPL.

Scenario	PDR (%)	Latency (s)
RPL	100	0.139
DIOF	62	1.876
DSRPL	100	0.290

Table 7. Memory occupancy results: DSRPL.

Scenario	RAM (Bytes)		ROM (Bytes)	Total (Bytes)
	Data	Bss	Text	
RPL	328	4960	46,704	51,992
DSRPL	328	4998	47,096	52,422

8. Discussion

The IETF-standardized RPL provides a basic routing solution for IoT networks. No sufficient security support is provisioned in its original protocol design. Rather, the inherent characteristics and design properties of RPL make it easy to launch different types of routing attacks. The basic processes of RPL topology establishment and maintenance allow for easy yet effective flooding attacks to be initiated across the network. The DISF attack only requires excessive transmissions of DIS messages to flood the network and target the overall network performance. The intrinsic RPL vulnerability to emerging and more severe routing attacks is evident, as demonstrated by the DIOF attack introduced in this paper.

The DIOF attack presents a serious threat to the stability and overall performance of RPL networks. It can effectively introduce an increase of more than 500% in network overhead to RPL networks even in relatively small-scale setups, as presented in Table 3. It can also be adversely used to incur very high energy consumption with an increase of at least 210%, as shown in Figure 8. Compared to the DISF attack, it incurred an increase of 50–283% in energy consumption, considering all the attack scenarios. That is, the DISF attack targets only the neighbor nodes of the attacker, whereas the DIOF attack extends it to all of the descendants of the attacker and their neighbor nodes as well. This makes the DIOF attack involve more victim nodes, especially if the malicious node is close to the sink.

These nodes participate with the attacker in flooding the network during a DIOF attack, whereas only the attacker is involved in the case of a DISF attack. As a result, resource utilization and network lifetime can be easily and adversely targeted during the DIOF attack without any resistance from the RPL functionality.

In addition, experiencing high QoS performance degradation is another major challenge caused by the attack. Table 4 indicates that PDR and latency can be adversely affected by a reduction of more than 32% and an increase of more than 192%, respectively. Notice that these severe effects were evident even in small-scale RPL setups and would be more significant in large-scale deployments. These were also preserved irrespective of the position of the attacker in the network and regardless of the number of neighbor nodes in the vicinity of the attacker.

A further comparison of the DIOF attack with different variants of the flooding attack in RPL networks is provided in Table 8. These are the Multicast-DIS [54], Spam-DIS [60], and SSDH [64] attacks, discussed already in Section 3. The presented results were calculated relative to attack-free RPL scenarios. It is apparent that the DIOF attacks present the most challenging attack with the highest adverse impacts on network overhead and energy consumption. It flooded the network with almost 300% more additional DIO messages than in the case of the Multicast-DIS attack, whereas it increased energy consumption by more than 50% compared to the Spam-DIS attack. Although it experienced a similar PDR reduction with the SSDH [64], the DIOF attack resulted in a higher latency of approximately 165%. It is evident that the DIOF attack presents a serious security threat to RPL networks and can introduce more damaging DOS-oriented attacks than the other common variants of the flooding attack.

Without additional security support, fostering wide RPL network deployments, particularly for demanding and sensitive IoT applications, would become a serious challenge. RPL networks would be at permanent risk of easy-to-initiate DIOF attacks, with serious damage to overall network performance. Having this critical consideration in mind, DSRPL provides the solution to address such an inevitable security issue with comparable overall performance to the standard RPL. While attacking the standard RPL network resulted in very high network overhead, DSRPL allows for a reduction of more than 87% in the total transmissions of the DIO and DAO messages, as presented in Table 5. It also provides a promising solution that can maintain energy consumption at a very low level with a relatively slight increase of less than 55%, as shown in Figure 9. It was able to effectively minimize the time spent by the nodes in the demanding TX mode, with very similar behavior to the standard RPL. DSRPL demonstrated the ability to sustain high QoS performance when defending the attack. Table 6 indicates that only negligible impact on latency was experienced, whereas there was no impact at all on PDR.

Table 9 provides a comparison of DSRPL with different flooding attack countermeasures, namely RPL-MRC [64], Secure-RPL [77], Sec-RPL [78], and SecRPL1 [79]. The presented results were calculated relative to the corresponding under-flooding-attack RPL scenarios. They indicate how much reductions in network overhead, energy consumption, and latency, and an increase in PDR, were achieved by each solution. It is apparent that a promising security solution is provided by DSRPL against DIOF attacks. Compared to the presented countermeasures, DSRPL not only ensures a competitive reduction in network overhead, but also a higher reduction in energy consumption of more than 12%. It can also effectively maintain PDR at higher levels and noticeably rectify adverse latency situations. DSRPL outperforms Sec-RPL [78] and SecRPL1 [79] in increasing PDR by 7% and reducing latency by 40%. The efficiency of DSRPL enables highly secure and well-performing RPL networks without adding much to RPL complexity.

Table 8. Comparison of DIOF attack with other variants of the flooding attack.

Attack	Overhead Increase	Energy Consumption Increase	PDR Reduction	Latency Increase
Multicast-DIS [54]	700%	130%	-	-
Spam-DIS [60]	300%	160%	-	-
SSDH [64]	250%	150%	37%	27%
DIOF	1000%	212%	38%	192%

Table 9. Comparison of DSRPL with other relevant solutions.

Solution	Overhead Reduction	Energy Consumption Reduction	PDR Increase	Latency Reduction
RPL-MRC [64]	83%	51%	-	-
Secure-RPL [77]	81%	68%	-	-
Sec-RPL [78]	-	44%	54%	19%
SecRPL1 [79]	72%	39%	17%	44%
DSRPL	87%	80%	61%	84%

9. Conclusions

No sufficient security support is provisioned in the design of RPL against the DoS-oriented flooding attacks. To advance security in RPL-based IoT networks, this work provides practical exposure to a new variant of the flooding attack in addition to a simple countermeasure solution. It reveals the potential vulnerability of RPL to emerging network security attacks, such as the DIOF attack, that can be easily initiated by any compromised node. It also makes it clear to what degree such an adverse attack can be of serious damage to the network. Compared to the DISF attack, it has a wider effect on the network beyond the neighboring area of the attacking node. The experimental evaluation results presented in this paper indicate the adversity of the DIOF attack, which incurred a higher increase in energy consumption of more than 50% compared to the DISF attack. It also increased network overhead to higher figures of more than 100% value due to the frequent DIO transmissions across the entire network. Its severe impact on QoS performance is also evident even in simple attack scenarios. A reduction of 38% in PDR and an increase of 490% in latency can be effectively achieved by the DIOF attack instead of the DISF attack.

Considering these implications, the development of DSRPL, providing an effective verification and mitigation solution against the DIOF attack, is presented in this paper. It introduced simple and light modifications to RPL functionality to incorporate an effective collaborative and distributed security scheme. The presented analysis of DSRPL showed its efficiency and highlighted its robustness against the DIOF attack, considering different experimental scenarios. It guarantees responsive detection and mitigation actions in a matter of a few seconds. It also succeeded in maintaining high QoS performance by increasing PDR and reducing latency by 38% and 84%, respectively, compared to the attack-free scenario. In addition, DSRPL decreased network overhead and energy consumption by more than 80%. Such effective security support would contribute toward enriching the RPL resilience and reviving its potential for a broader range of RPL-based IoT applications.

However, DSRPL is still limited to addressing the specific flooding attack of DIOF. Although focusing on such an adverse attack is a feasible consideration, this can also be considered as the first step toward a more generalized mitigation solution against a wide scope of RPL flooding attacks. That is, DSRPL has great potential to be effectively extended as an integrated solution for addressing other variants of the flooding attack. The adopted approach of simply considering the updates from only the parent node while using the information of remote sources for effective verification is flexible enough to be optimized for more comprehensive security support.

In future work, the focus will be on investigating how the current work can be extended, with additional security modules to develop an integrated RPL security architecture against different types of RPL flooding attacks. The objective will be to enhance RPL functionality with comprehensive and effective security support to prevent the use of the different types of RPL control messages for launching any flooding attack. Another consideration will be investigating the adversity of combining DIOF attacks with other routing attacks, such as the rank and VN attacks.

Author Contributions: Conceptualization, M.R. and M.B.; methodology, A.M.; software, I.S.A. and M.R.; validation, M.R. and I.S.A.; formal analysis, M.R. and M.B.; investigation, I.S.A.; resources, A.M.; data curation, M.R, M.B. and I.S.A.; writing—original draft preparation, M.R. and M.B.; writing—review and editing, I.S.A.; visualization, A.M.; supervision, A.M.; project administration, M.R.; funding acquisition, I.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bagchi, S.; Abdelzaher, T.; Govindan, R.; Shenoy, P.; Atrey, A.; Ghosh, P.; Xu, R. New Frontiers in IoT: Networking, Systems, Reliability and Security Challenges. *IEEE Internet Things J.* **2020**, *7*, 11330–11346. [CrossRef]
2. Mohanty, J.; Mishra, S.; Patra, S.; Pati, B.; Panigrahi, C.R. IoT Security, Challenges, and Solutions: A Review. In *Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing*; Panigrahi, C.R., Pati, B., Mohapatra, P., Buyya, R., Li, K.C., Eds.; Springer: Singapore, 2021; Volume 1199.
3. Rekha, S.; Thirupathi, L.; Renikunta, S.; Gangula, R. Study of security issues and solutions in Internet of Things (IoT). *Mater. Today Proc.* **2021**, *80*, 3554–3559. [CrossRef]
4. Petrosyan, A. Global Annual Number of IoT Cyber Attacks 2018–2022. 2023. Available online: https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/ (accessed on 17 July 2023).
5. Kupreev, O.; Badovskaya, E.; Gutnikov, A. DDoS Attacks in Q1 2020. Kaspersky, DDOS Reports, 2020. [Online]. Available online: https://securelist.com/ddos-attacks-in-q1-2020/96837/ (accessed on 17 July 2023).
6. "Sonicwall Cyber Threat Report—Cyber Threat Intelligence for Navigating the New Business Reality," Sonicwall. 2021. Available online: https://www.sonicwall.com/medialibrary/en/white-paper/2021-cyber-threat-report.pdf (accessed on 17 July 2023).
7. "Symantec Corporation Internet Security Threat Report 2019", Symantec, Vol. 24, Feb. 2019. Available online: https://docs.broadcom.com/doc/istr-24-2019-en (accessed on 17 July 2023).
8. Majid, A. Security and Privacy Concerns over IoT Devices Attacks in Smart Cities. *J. Comput. Commun.* **2023**, *11*, 26–42. [CrossRef]
9. Rahmani, A.M.; Bayramov, S.; Kalejahi, B.K. Internet of Things Applications: Opportunities and Threats. *Wirel. Pers. Commun.* **2022**, *122*, 451–476. [CrossRef] [PubMed]
10. Razmjoo, A.; Gandomi, A.; Mahlooji, M.; Astiaso Garcia, D.; Mirjalili, S.; Rezvani, A.; Ahmadzadeh, S.; Memon, S. An Investigation of the Policies and Crucial Sectors of Smart Cities Based on IoT Application. *Appl. Sci.* **2022**, *12*, 2672. [CrossRef]
11. Fantana, N.L.; Riedel, T.; Schlick, J.; Ferber, S.; Hupp, J.; Miles, S.; Michahelles, F.; Svensson, S. IoT Applications—Value Cre-ation for Industry. In *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*; Vermesan, O., Friess, P., Eds.; River Publishers: Aalborg, Denmark, 2022; pp. 153–206. [CrossRef]
12. Farooq, M.S.; Sohail, O.O.; Abid, A.; Rasheed, S. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Livestock Environment. *IEEE Access* **2022**, *10*, 9483–9505. [CrossRef]
13. Al-rawashdeh, M.; Keikhosrokiani, P.; Belaton, B.; Alawida, M.; Zwiri, A. IoT Adoption and Application for Smart Healthcare: A Systematic Review. *Sensors* **2022**, *22*, 5377. [CrossRef] [PubMed]
14. Sujey, L. Number of Internet of Things (IoT) Connected Devices Worldwide in 2018, 2025 and 2030. Available online: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/ (accessed on 17 July 2023).

15. Morgan, S. Global Cybersecurity Spending Predicted to Exceed $1 Trillion from 2017–2021. 2019. Available online: https://cybersecurityventures.com/cybersecurity-market-report/ (accessed on 17 July 2023).
16. Avila, K.; Jabba, D.; Gomez, J. Security Aspects for RPL-Based Protocols: A Systematic Review in IoT. *Appl. Sci.* **2020**, *10*, 6472. [CrossRef]
17. Kamaldeep, M.M.; Dutta, M. Feature engineering and machine learning framework for DDoS attack detection in the standardized internet of things. *IEEE Internet Things J.* **2023**, *10*, 8658–8669. [CrossRef]
18. Al-Shareeda, M.A.; Manickam, S.; Saare, M.A. DDoS attacks detection using machine learning and deep learning techniques: Analysis and comparison. *Bull. Electr. Eng. Inform.* **2023**, *12*, 930–939. [CrossRef]
19. Ankam, S.; Reddy, N.S. A mechanism to detecting flooding attacks in quantum enabled cloud-based lowpower and lossy networks. *Theor. Comput. Sci.* **2023**, *941*, 29–38. [CrossRef]
20. Bahashwan, A.A.; Anbar, M.; Manickam, S.; Al-Amiedy, T.A.; Aladaileh, M.A.; Hasbullah, I.H. A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking. *Sensors* **2023**, *23*, 4441. [CrossRef]
21. Shah, S.Q.A.; Khan, F.Z. Mitigating TCP SYN flooding based EDOS attack in cloud computing environment binomial distribution in SDN. *Comput. Commun.* **2022**, *182*, 198–211. [CrossRef]
22. Alabsi, B.A.; Anbar, M.; Rihan, S.D.A. Conditional Tabular Generative Adversarial Based Intrusion Detection System for Detecting Ddos and Dos Attacks on the Internet of Things Networks. *Sensors* **2023**, *23*, 5644. [CrossRef] [PubMed]
23. Adedeji, K.B.; Abu-Mahfouz, A.M.; Kurien, A.M. DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges. *J. Sens. Actuator Netw.* **2023**, *12*, 51. [CrossRef]
24. Przybocki, P.; Vassilakis, V.G. An Analysis into Physical and Virtual Power Draw Characteristics of Embedded Wireless Sensor Network Devices under DoS and RPL-Based Attacks. *Sensors* **2023**, *23*, 2605. [CrossRef]
25. Alansari, Z.; Anuar, N.; Kamsin, A.; Belgaum, M. A systematic review of routing attacks detection in wireless sensor networks. *PeerJ. Comput. Sci.* **2022**, *8*, e1135. [CrossRef]
26. Zolertia, "Z1 Datasheet", Zolertia Advancare. 2010. Available online: http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf (accessed on 2 August 2023).
27. Kushalnagar, N.; Montenegro, G.; Hui, J.; Culler, D. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*; IETF: Fremont, CA, USA, 2007. [CrossRef]
28. Hui, J.; Thubert, P. *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*; IETF: Fremont, CA, USA, 2011. [CrossRef]
29. Winter, T.; Thubert, P.; Brandt, A.; Hui, J.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, J.; Alexander, R. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*; IETF: Fremont, CA, USA, 2012. [CrossRef]
30. Vasseur, J.P.; Kim, M.; Pister, K.; Dejean, N.; Barthel, D. *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*; IETF: Fremont, CA, USA, 2012. [CrossRef]
31. Thubert, P. *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*; IETF: Fremont, CA, USA, 2012. [CrossRef]
32. Gnawali, O.; Levis, P. *The Minimum Rank with Hysteresis Objective Function*; IETF: Fremont, CA, USA, 2012. [CrossRef]
33. Levis, P.; Clausen, T.; Hui, J.; Gnawali, O.; Ko, J. *The Trickle Algorithm*; IETF: Fremont, CA, USA, 2011. [CrossRef]
34. Lovatto, J.; Santos, R.C.; Souza, C.; Zucca, R.; Lovatto, F.; Geisenhoff, L.O. Use of linear programming for decision making: An analysis of cost, time and comfort of rural housing dwellings. *Rev. Bras. Eng. Agrícola E Ambient.* **2020**, *24*, 622–629. [CrossRef]
35. Tsao, T.; Alexander, R.; Dohler, M.; Daza, V.; Lozano, A.; Richardson, M. *A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)*; IETF: Fremont, CA, USA, 2015. [CrossRef]
36. Perazzo, P.; Vallati, C.; Arena, A.; Anastasi, G.; Dini, G. An Implementation and Evaluation of the Security Features of RPL. In Proceedings of the 16th International Conference Ad-Hoc Networks and Wireless, Messina, Italy, 20–22 September 2017; pp. 63–76. [CrossRef]
37. Raoof, A.; Matrawy, A.; Lung, C.H. Enhancing Routing Security in IoT: Performance Evaluation of RPL's Secure Mode Under Attacks. *IEEE Internet Things J.* **2020**, *7*, 11536–11546. [CrossRef]
38. Bang, A.O.; Rao, U.P.; Kaliyar, P.; Conti, M. Assessment of Routing Attacks and Mitigation Techniques with RPL Control Messages: A Survey. *ACM Comput. Surv.* **2023**, *55*, 44. [CrossRef]
39. Altulaihan, E.; Almaiah, M.A.; Aljughaiman, A. Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions. *Electronics* **2022**, *11*, 3330. [CrossRef]
40. Al-Amiedy, T.A.; Anbar, M.; Belaton, B.; Bahashwan, A.A.; Hasbullah, I.H.; Aladaileh, M.A.; Mukhaini, G.A. A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things. *Internet Things* **2023**, *22*, 100741. [CrossRef]
41. Raoof, A.; Matrawy, A.; Lung, C.H. Routing Attacks and Mitigation Methods for RPL-Based Internet of Things. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1582–1606. [CrossRef]
42. Mayzaud, A.; Badonnel, R.; Chrisment, I. A Taxonomy of Attacks in RPL-based Internet of Things. *Int. J. Netw. Secur.* **2016**, *18*, 459–473. [CrossRef]
43. Al-Hadhrami, Y.; Hussain, F.K. DDoS Attacks in IoT Networks: A Comprehensive Systematic Literature Review. *World Wide Web* **2021**, *24*, 971–1001. [CrossRef]

44. Pongle, P.; Chavan, G. A survey: Attacks on RPL and 6LoWPAN in IoT. In Proceedings of the International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015; pp. 1–6. [CrossRef]
45. Mayzaud, A.; Sehgal, A.; Badonnel, R.; Chrisment, I.; Schönwälder, J. A Study of RPL DODAG Version Attacks. In *Monitoring and Securing Virtualized Networks and Services*; Sperotto, A., Doyen, G., Latré, S., Charalambides, M., Stiller, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8508, pp. 92–104. [CrossRef]
46. Aris, A.; Oktug, S.F.; Berna Ors Yalcin, S. RPL Version Number Attacks: In-depth Study. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 776–779. [CrossRef]
47. Ambarkar, S.S.; Shekokar, N. Critical and Comparative Analysis of DoS and Ver-sion Number Attack in Healthcare IoT Sys-tem. In Proceedings of the First Doctoral Sym-posium of Natural Computing Research, Pune, India, 8 August 2020; pp. 301–312. [CrossRef]
48. Bang, A.; Rao, U.P. Impact Analysis of Rank Attack on RPL-Based 6LoWPAN Networks in Internet of Things and After-maths. *Arab. J. Sci. Eng.* **2022**, *48*, 2489–2505. [CrossRef]
49. Verma, A.; Ranga, V. The Impact of Copycat Attack on RPL based 6LoWPAN Networks in Internet of Things. *Computing* **2021**, *103*, 1479–1500. [CrossRef]
50. Alsukayti, I.S.; Alreshoodi, M. RPL-Based IoT Networks under Simple and Complex Routing Security Attacks: An Experimental Study. *Appl. Sci.* **2023**, *13*, 4878. [CrossRef]
51. Rajasekar, V.R.; Rajkumar, S. A Study on Impact of DIS flooding Attack on RPL-based 6LowPAN Network. *Microprocess. Microsyst.* **2022**, *94*, 104675. [CrossRef]
52. Nguyen, T.; Ngo, T.; Nguyen, T.; Tran, D.; Tran, H.A.; Bui, T. The Flooding Attack in Low Power and Lossy Networks: A Case Study. In Proceedings of the International Conference on Smart Communications in Network Technologies (SaCoNeT), El Oued, Algeria, 27–31 October 2018; pp. 183–187. [CrossRef]
53. Kalita, A.; Brighente, A.; Khatua, M.; Conti, M. Effect of DIS Attack on 6TiSCH Network Formation. *IEEE Commun. Lett.* **2022**, *26*, 1190–1193. [CrossRef]
54. Medjek, F.; Tandjaoui, D.; Djedjig, N.; Romdhani, I. Multicast DIS attack mitigation in RPL-based IoT-LLNs. *J. Inf. Secur. Appl.* **2021**, *61*, 102939. [CrossRef]
55. Dogan, C.; Yilmaz, S.; Sen, S. Analysis of RPL Objective Functions with Security Perspective. In Proceedings of the 11th In-ternational Conference on Sensor Networks (SENSORNETS), Vienna, Austria, 7–8 February 2022; pp. 71–80. [CrossRef]
56. Sharma, S.; Verma, V.K. Security Explorations for Routing Attacks in Low Power Networks on Internet of Things. *J. Supercomput.* **2021**, *77*, 4778–4812. [CrossRef]
57. Chowdhury, M.; Ray, B.; Chowdhury, S.; Rajasegarar, S. A Novel Insider Attack and Machine Learning Based Detection for the Internet of Things. *ACM Trans. Internet Things* **2021**, *2*, 26. [CrossRef]
58. Sahay, R.; Geethakumari, G.; Mitra, B. A novel Network Partitioning Attack against Routing Protocol in Internet of Things. *Ad Hoc Netw.* **2021**, *121*, 102583. [CrossRef]
59. Baghani, A.S.; Rahimpour, S.; Khabbazian, M. The DAO Induction Attack: Analysis and Countermeasure. *IEEE Internet Things J.* **2022**, *9*, 4875–4887. [CrossRef]
60. Pu, C. Spam DIS Attack against Routing Protocol in the Internet of Things. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Istanbul, Turkey, 25–29 April 2019; pp. 73–77.
61. Biswas, R.; Wu, J.; Li, X. A capacity-aware distributed denial-of-service attack in low-power and lossy networks. In Proceedings of the 2019 IEEE 40th Sarnoff Symposium, Newark, NJ, USA, 23–24 September 2019; pp. 1–6.
62. Pu, C.; Brown, J.; Carpenter, L. A Theil Index-Based Countermeasure Against Advanced Vampire Attack in Internet of Things. In Proceedings of the 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), Newark, NJ, USA, 11–14 May 2020; pp. 1–6.
63. Sharma, G.; Grover, J.; Verma, A.; Kumar, R.; Lahre, R. Analysis of hatchetman attack in RPL based IoT networks. In Proceedings of the International Conference on Emerging Technologies in Computer Engineering, Jaipur, India, 4–5 February 2022; pp. 666–678.
64. Belkhira, S.A.H.; Rouissat, M.; Belkheir, M.; Bouziani, M. Selective Sub-DODAGs Hiding "SSDH" a new Attack in IoT RPL-Based Networks. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2022**, *13*, 633–648.
65. Al-Amiedy, T.A.; Anbar, M.; Belaton, B.; Kabla, A.H.H.; Hasbullah, I.H.; Alashhab, Z.R. A Systematic Literature Review on Machine and Deep Learning Approaches for De-tecting Attacks in RPL-Based 6LoWPAN of Internet of Things. *Sensors* **2022**, *22*, 3400. [CrossRef]
66. Zahra, F.; Jhanjhi, N.Z.; Khan, N.A.; Brohi, S.N.; Masud, M.; Aljahdali, S. Protocol-Specific and Sensor Network-Inherited Attack Detection in IoT Using Machine Learning. *Appl. Sci.* **2022**, *12*, 11598. [CrossRef]
67. Nikravan, M.; Movaghar, A.; Hosseinzadeh, M. A lightweight defense approach to mitigate version number and rank attacks in low-power and lossy networks. *Wirel. Pers. Commun.* **2018**, *99*, 1035–1059. [CrossRef]
68. Ali, S.E.; Tariq, N.; Khan, F.A.; Ashraf, M.; Abdul, W.; Saleem, K. BFT-IoMT: A Blockchain-Based Trust Mechanism to Mitigate Sybil Attack Using Fuzzy Logic in the In-ternet of Medical Things. *Sensors* **2023**, *23*, 4265. [CrossRef] [PubMed]
69. Mayzaud, A.; Badonnel, R.; Chrisment, I. A Distributed Monitoring Strategy for Detecting Version Number Attacks in RPL-Based Networks. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 472–486. [CrossRef]
70. Alsukayti, I.S.; Singh, A. A Lightweight Scheme for Mitigating RPL Version Number Attacks in IoT Networks. *IEEE Access* **2022**, *10*, 111115–111133. [CrossRef]

71. Bang, A.O.; Rao, U.P. EMBOF-RPL: Improved RPL for early detection and isola-tion of rank attack in RPL-based Internet of Things. *Peer-Peer Netw. Appl.* **2022**, *15*, 642–665. [CrossRef]
72. Sahay, R.; Geethakumari, G.; Mitra, B. Mitigating the worst parent attack in RPL based internet of things. *Clust. Comput.* **2022**, *25*, 1303–1320. [CrossRef]
73. Rouissat, M.; Belkheir, M.; Belkhira, S.A.H.; Hacene, S.B.; Lorenz, P.; Bouziani, M. A new lightweight decentralized mitigation solution against Version Number Attacks for IoT Networks. *JUCS J. Univers. Comput. Sci.* **2023**, *29*, 118–151. [CrossRef]
74. Tmote Sky Datasheet, Moteiv Corporation. Available online: https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf (accessed on 2 August 2023).
75. *MICAz, Wireless Measurement System Datasheet, Document Part Number: 6020-0060-04 Rev A*; Crossbow Technology Inc.: San Jose, CA, USA, 2008; Available online: http://courses.ece.ubc.ca/494/files/MICAz_Datasheet.pdf (accessed on 2 August 2023).
76. Dunkels, A.; Gronvall, B.; Voigt, T. Contiki- a Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 16–18 November 2004; pp. 455–462. [CrossRef]
77. Verma, A.; Ranga, V. Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, E3802. [CrossRef]
78. Guo, G. A Lightweight countermeasure to DIS attack in RPL routing protocol. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; pp. 753–758.
79. Wadhaj, I.; Ghaleb, B.; Thomson, C.; Al-Dubai, A.; Buchanan, W.J. Mitigation mechanisms against the DAO attack on the routing protocol for low power and lossy networks (RPL). *IEEE Access* **2020**, *8*, 43665–43675. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.