IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 September 2022, accepted 8 October 2022, date of publication 19 October 2022, date of current version 26 October 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3215460*

RESEARCH ARTICLE

# A Lightweight Scheme for Mitigating RPL Version Number Attacks in IoT Networks

**IBRAHIM S. ALSUKAYTI[1] AND AMAN SINGH[2,3,4]**

[1]Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2]Higher Polytechnic School, Universidad Europea del Atlántico, 39011 Santander, Spain
[3]Department of Engineering, Universidad Internacional Iberoamericana, Arecibo, PR 00613, USA
[4]Faculty of Engineering, Universidade Internacional do Cuanza, Bairro Kaluapanda, Cuito-Bié, Angola

Corresponding author: Ibrahim S. Alsukayti (skiety@qu.edu.sa)

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

**ABSTRACT** Internet of Things (IoT) systems incorporate a multitude of resource-limited devices typically interconnected over Low Power and Lossy Networks (LLNs). Robust IP-based network routing among such constrained IoT devices can be effectively realized using the IPv6 Routing Protocol for LLN (RPL) which is an IETF-standardized protocol. The RPL design features a topology maintenance mechanism based on a version numbering system. However, such a design property makes it easy to initiate Version Number (VN) attacks targeting the stability, lifetime, and performance of RPL networks. Thus the wide deployment of RPL-based IoT networks would be hindered significantly unless internal routing attacks such as the VN attacks are efficiently addressed. In this research work, a lightweight and effective detection and mitigation solution against RPL VN attacks is introduced. With simple modifications to the RPL functionality, a collaborative and distributed security scheme is incorporated into the protocol design (referred to as CDRPL). As the experimental results indicated, it provides a secure and scalable solution enhancing the resilience of the protocol against simple and composite VN attacks in different experimental setups. CDRPL guaranteed fast and accurate attack detection as well as quick topology convergence upon any attack attempt. It also efficiently maintained network stability, control traffic overhead, QoS performance, and energy consumption during different scenarios of the VN attack. Compared to other similar approaches, CDRPL yields better performance results with lightweight node-local processing, no additional entities, and less communication overhead.

**INDEX TERMS** Internet of Things, wireless sensor networks, RPL, network security.

## I. INTRODUCTION

The emergence of the Internet of Things (IoT) technology brings a revolutionary development of a variety of smart solutions. IoT has gained wide interest in a wide scope of application domains such as industry, agriculture, and healthcare. This has led to varying-scale IoT deployments for a broad range of new IoT applications such as smart city, industrial automation, environmental monitoring, and smart metering. As a result, we witness exponential growth in the number of IoT-connected objects during recent years, with a forecast of more than 30 billion IoT devices for 2030 [1]. The IoT industry is estimated to generate 3.9-11.1 trillion a year in revenue by 2025, according to the McKinsey Global Institute [2].

IoT systems typically incorporate multiple small-sized smart devices with limited hardware resources in terms of computation, storage, and energy. The typical connectivity setup interconnecting these components is based on energy-efficient wireless communication. The adoption of Low power and Lossy Networks (LLNs) [3] provides a viable networking solution for establishing an effective IoT network infrastructure. LLN is designed for networking resource-limited devices over unreliable wireless communication links of low bandwidth and high loss rate. A distinctive feature of LLNs is the support of energy-efficient and low-power communications over a simplistic wireless infrastructure. LLNs facilitate low-cost IoT network

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

deployments interconnecting a multitude of constrained IoT devices.

The establishment of network routing among the IoT devices at the network layer of the LLN architecture is a key to sustaining effective IoT data communication. However, network routing solutions for IoT networks need to strictly adhere to the distinct characteristics and limitations of LLNs. Taking this into consideration, the IETF provides the IPv6 Routing Protocol for Low Power and Lossy Network (RPL). It is an LLN-customized routing protocol designed to extend IPv6 networking for resource-constrained IoT devices. The functionality of RPL enables these devices to establish a structured network topology for effective loop-free routing of IoT data packets. RPL has a flexible design with customizable functionality for easily optimizing its topology to meet the network requirements of any IoT deployment.

However, complete routing security support is still an open challenge for the design of RPL [4]. In fact, such a challenge is common in different IoT network deployments. Without the implementation of strict cyber security support, security attacks become highly possible in IoT networks. Examples of real-life IoT attacks include the "Linux.Darlloz" worm attack that targeted more than 12,000 Linux-powered Intel $\times$86 IoT devices such as industrial control systems and security cameras in November 2013 [5]. In 2016, Mirai-based attacks using different IoT botnets with a set of IoT consumer devices were launched with 600 Gbps to $>$1 Tbps DDoS attacks targeting different internet services [6]. As reported by Kaspersky Lab, the number of DDoS attacks in Q1 of 2020 increased by 80% compared to that in Q1 of 2019 and doubled compared to that in Q4 of 2019 [7]. Between 2017 and 2018, the number of monthly attacks on IoT devices was around 5200 attacks as reported by Symantec [8]. The recent report by SonicWall stated that the number of IoT malware attacks was more than 34 million in 2019 and increased by more than 65% in 2020 with 56.9 million attacks [9].

As the number of IoT devices increases, which is projected to reach approximately 18 billion in 2022 [10], security challenges become more evident in IoT systems. Having such devices in our buildings, vehicles, appliances, and mobile devices brings more severe security threats than traditional networks. The adverse impact of such attacks is significant in a way that causes the network to collapse resulting in communication disruption and complete data loss. It was earlier estimated that cybercrime around the globe cost 400–500 billion in 2015 whereas the figure rose to 2–3 trillion in 2016 [11].

However, the standard RPL still provides no adequate support for securing its communication against different types of attacks [12]. Only limited protection from external attacks is provisioned by RPL [13] whereas no support against internal routing attacks is provided [14], [15]. These include blackhole, sinkhole, selective forwarding, and wormhole attacks in addition to the RPL-specific ones such as rank, version number, and worst parent attacks [16]. This would make RPL networks attractive targets for diverse security attacks. The vulnerability of RPL has been examined and discussed in different research works such as in [4], [17], and [18]. The protocol characteristics of RPL including its limited security support lead to increasing its vulnerability to internal routing attacks. Malicious nodes in an RPL network can easily perform one of these attacks. Taking the version number (VN) attack as an example, it only requires the malicious node to advertise a new VN in the network by incrementing the currently propagated one in the protocol control messages. This would lead to initiating a full reconstruction of the network topology. As a result, network stability would be adversely affected in addition to a considerable increase in processing and traffic overhead.

The VN attack is an RPL-specific attack that exploits the version numbering mechanism of RPL. It can be initiated by simply incrementing the VN being advertised in the RPL control packets. It can lead to an effective Denial of Service (DoS) attack targeting the stability, lifetime, and overall performance of RPL networks as highlighted in different experimental studies [19], [20], [21]. Moreover, such effects can be significantly amplified by having composite VN attacks with multiple attackers. Initiating such attacks with two or more attacking nodes across an RPL network would cause the network to collapse quickly [22].

All these crucial considerations demonstrate the importance of enhancing the resilience of RPL networks against internal routing attacks, particularly the VN attack. Recently, there have been some research efforts toward addressing such a fundamental RPL security challenge of VN attack mitigation [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35]. However, most of the proposed solutions were based on limited design aspects such as the need for additional entities, heavy node-local processing, and further communication establishment. Some proposals also led to other limitations such as adding to protocol convergence time and imposing high detection latency. Another aspect that has been mostly overlooked is addressing composite VN attacks, particularly in large-scale network scenarios.

Therefore, there is an inevitable need for further effective enhancements to RPL security in order to enrich its applicability to real-life IoT applications. This challenge basically constitutes the motivation behind this research work. It is clear that without effectively addressing RPL VN attacks, the wide deployment of RPL-based IoT networks would be hindered. Accordingly, it is still critical to enhance the resilience of RPL networks by developing more efficient solutions against internal routing attacks, particularly the VN attack. However, this should be achieved without adding much to the protocol complexity and network overhead. Moreover, approaching VN attacks mitigation with limited processing overhead and protocol latency is a strict requirement for any new RPL security enhancement. Eliminating the possibilities of composite VN attack scenarios would also increase the innate immunity of RPL and revive its potential for wide-range IoT applications.

In this study, we introduce a novel VN attack detection and mitigation approach that addresses the aforementioned

limitations of prior research works. We propose a Collaborative and Distributed scheme for RPL (CDRPL) which provides a lightweight and effective security enhancement to RPL networks. It aims at achieving effective detection and mitigation of simple and composite VN attacks in varying-scale networks deployed for any IoT application. It provides a lightweight and efficient security solution by extending the RPL functionality without adding much to its complexity. Incorporating a collaborative and distributed scheme into RPL by simple in-protocol modifications for defending simple and composite VN attacks represents the novelty of this work.

This research work provides a four-fold contribution. First, it provides a novel lightweight collaborative and distributed security scheme to enhance RPL resilience against VN attacks. Second, simple modifications to certain RPL operational aspects are only introduced without imposing additional entities, computational complexity, and communication overhead. Third, it addresses more complex composite VN attacks in small and large-scale RPL networks in an effective and scalable manner. Fourth, an extensive evaluation of CDRPL considering varying-scale scenarios is provided.

The rest of the paper is organized as follows. Section II provides an overview of the RPL protocol and the VN attack. Section III presents and discusses the related work. Then, a brief description of the attack model is provided in Section IV. In Section V, the proposed CDRPL solution is detailed. The evaluation methodology and results analysis are presented in Sections VI and VII, respectively. Finally, Section VIII concludes the paper and points out future perspectives.

## II. RPL BACKGROUND

IoT deployments typically incorporate a large number of resource-limited devices interconnected over an LLN. These are constrained devices of small capacity with limited computation, storage, and energy capabilities. The connectivity among the devices is established over scarce wireless links with no guarantee of high performance and reliability. Rather, it enables effective wireless communications with low cost, complexity, and energy. To maintain end-to-end IP networking for LLNs, an additional IP adaptation layer providing header compression and fragmentation for effective integration with IPv6 networks is introduced. That is the IPv6 over Low Power Wireless Personal Area Networks (6LowPAN) specified by The IETF in RFC 4944 [36] and RFC 6282 [37]. On top of this layer, the IPv6 routing functionality is maintained at the network layer by RPL as specified by the IETF in RFC 6550 [12]. RPL is a bespoke networking solution for network layer routing over constrained LLN links. It is based on the distance-vector routing model and operates in a proactive manner. RPL effectively addresses the different routing challenges in LLN environments.

### A. RPL OPERATION OVERVIEW

The architecture of a typical RPL network is formed as a Directed Acyclic Graph (DAG), referred to as an RPL instance. RPL devices in one instance are interconnected into a network topology formed as one or more Destinations-Oriented Directed Acyclic Graph (DODAGs). The formation of a DODAG requires a designated root (RPL sink node) interconnecting multiple RPL nodes over multihop network paths [12]. An example of an RPL network of two RPL instances with varying DODAGs is presented in Figure 1. Instance 1 has two DODAGs whereas only one DODAG exists in Instance 2. All these DODAGs can be interconnected to the Internet infrastructure through their corresponding sink nodes (SN1-3).

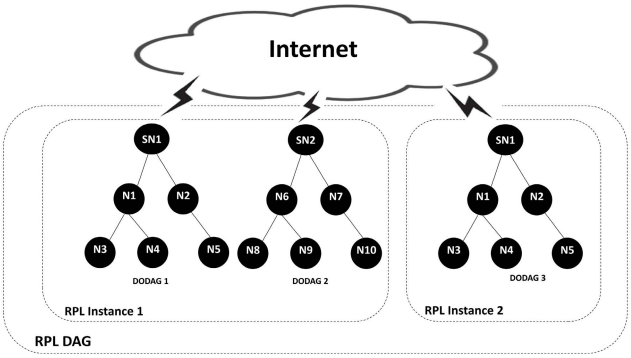


**FIGURE 1.** An example of an RPL network of two RPL instances with varying DODAGs.

The construction of the DODAG topology is based on the establishment of upward and downward network paths among RPL nodes. This operation is initiated by the sink node propagating periodic control messages. These are the DODAG Information Object (DIO) messages which are then forwarded by every RPL node to successfully construct the upward routes. The DIO messages provide the nodes with information that helps in the discovery and maintenance of the DODAG. Each DIO message disseminates the RPL Instance ID, DODAG ID, and VN. Each DODAG is identified by both the instance and DODAG IDs whereas the VN indicates the current version of the DODAG topology. Other information such as the IPv6 address and rank value of the parent node is also included to enable nodes to join the DODAG topology [12].

Another important information propagated in the DIO messages is the Objective Function (OF). A single OF is applied in all the DODAGs associated with one instance. It dictates the formation of the DODAG topology and enables the achievement of application-specific network requirements. RPL supports customizable OF formulation for flexible implementation of specific optimization objectives such as reliability and energy reservation. Objective-oriented routing among the DODAG nodes is established based on the applied OF. Once a DIO message is received, the recipient node applies the advertised OF for the selection of its preferred parent (next hop) and for joining the DODAG topology. In addition, the OF is utilized for calculating a rank value for each node to represent its distance to the sink node and maintain a loop-free topology [12].

An OF can be formulated with one or multiple routing metrics and constraints as defined in RFC 6551 [38]. The document provides examples of common RPL routing metrics and constraints. These are classified into the node and link routing metrics and constraints. In RFC 6552 [39], the Objective Function Zero (OF0) is specified as a default OF that implements the hop count as a routing metric. Another default OF is the Minimum Rank with Hysteresis Objective Function (MRHOF) which is specified in RFC 6719 [40]. Taking into account network reliability, MRHOF is based on the Estimated Transmission Count (ETX) which is a link metric indicating the number of transmissions necessary for successful packet delivery. Other OFs have been recently considered by the IETF to address other networking aspects. For example, the Common Ancestor (CA) OF [41] provides support for multipath routing to enable effective packet replication and elimination in RPL networks.

To maintain control traffic and network overhead to a minimum level, RPL relies on the Trickle algorithm during topology construction and maintenance. The algorithm enables RPL to control the transmission interval of DIO messages based on the stability of the network. It defines two main configurable values which are the minimum interval and maximum interval. The algorithm enables RPL to start DIO transmission with a low interval that is set to a configured minimum interval value. Then, an exponential increase of the transmission interval is applied as long as the DODAG topology remains stable with no changes. Once an inconsistency is detected in the network (e.g. changes to a preferred parent or version number), the DIO transmission interval is reset to the minimum interval and the process is repeated. Otherwise, it continues increasing the DIO transmission interval until reaching the configured value of the maximum interval [42].

Downward routing is established in response to the topology construction process. Once a node has received a DIO message and joined a DODAG via a parent node, it propagates its routing information upward. A Destination Advertisement Object (DAO) message is sent along the established upward routes to the sink node. Certain routing information including the node's IPv6 address is contained in the DAO message. Two modes of downward routing are supported. The first is the storing mode which is fully stateful and enables a common ancestor to route data packets between an RPL source and destination. In this mode, each node needs to store the routing information advertised in its child nodes' DOA messages. The information is utilized later to enable effective internal DODAG routing across the network. In the other mode, the non-storing mode, source routing is utilized to route data traffic across the network through the root [12].

Moreover, node or link failure is addressed by RPL using two procedures. One is the local repair which enables a node to switch its current preferred parent to an alternative neighbor node. The other one is the global repair which requires the sink node to initiate reconstruction of the DODAG topology by incrementing the DODAG VN. These procedures are instituted in response to any detected failure in the routing process such as routing loops and inconsistency. They result in exchanging several DIS and DIO messages after resetting the trickle timer. Effective failure recovery would be guaranteed by these procedures but at the cost of additional network overhead [12].

### B. RPL VERSION NUMBER ATTACK

RPL defines the version property which specifies the current iteration of a DODAG topology. Each DODAG needs to be assigned a VN which is associated with the current topology of the DODAG. For each version, the nodes may change their rank values and positions within the DODAG topology. Therefore, a new version can result in new changes to the DODAG topology. The version information is advertised in the DIO messages along with other routing information. The VN is advertised by the sink node in the version field of the DIO messages. This field is meant to remain unchanged during the forwarding of the messages by the nodes across the network. Upon the reception of a new VN, a non-sink node updates the state of its current VN and then recalculates its rank and position within the network [12].

As specified in the RPL specification, the sink node is the only eligible node for initiating global repair and changing the value of the current version being advertised. This can be triggered under certain conditions such as the detection of network inconsistencies (e.g. DODAG loops) that cannot be addressed by local repairs. It can also be initiated as a result of detecting a higher VN advertised by a non-sink node. However, the procedure is left implementation-specific, allowing it to be configured following a periodic or event-triggered mechanism [12].

Unfortunately, the standard RPL design provides no guarantee that the initiation of global repairs and changes to VN is only performed by the sink of a DODAG. Therefore, the version property can be utilized to initiate an internal routing attack. Without strict adherence to the RPL specification, one or multiple malicious nodes can advertise a different version than the current one being set by the sink node. This only requires broadcasting fake changes to the current version by changing the current value of the VN being advertised in the DIO messages. As a result, an illegitimate global repair is initiated by a node other than the sink node. Such an attack targets network stability and would drain node resources and degrade overall performance, in particular for large-scale IoT network deployments [19], [20], [21], [22].

## III. RELATED WORK

The standard RPL specification [12] provides minimal support for limited security aspects to address external security attacks. That is, the standard RPL was designed with three basic security modes: the insecure, preinstalled, and authentication modes. In the first one, communications among RPL nodes are carried out without any security support. The preinstalled mode enables RPL nodes to establish secure communications using preinstalled security keys. In the authentication mode, an RPL node cannot attach to an RPL

network and establish data communication unless it obtains a security key from an authentication authority.

Nevertheless, there is still no considerable security support provided by the standard RPL against internal routing attacks. Although the authentication mode can be of use in limiting malicious RPL network access, there is still a great opportunity for an attacker to compromise an RPL node and initiate routing attacks internally. Examples of internal routing attacks are the blackhole, sinkhole, selective forwarding, and wormhole attacks which are already common in traditional networks [15], [18], [43]. In addition, the inherent protocol design of RPL has led to the emergence of internal routing attacks more specific to RPL networks. These include rank, VN, DAO inconsistency, DIO suppression, and worst parent attacks [18], [44]. These attacks lead to considerable degradation of the overall performance of targeted RPL networks. Over recent years, several research efforts have been made to address the different internal routing attacks threatening RPL networks [16], [19], [20], [21], [45], [46].

The VN attack is an RPL-specific attack that can be initiated by any node incrementing the value in the VN field of the DIO message. It can lead to an effective DoS attack with devastating effects. The experimental studies in [19], [20], and [21] highlighted the adverse impact of the VN attack on network overhead, QoS, routing loops, and power consumption. Moreover, performing the VN attack with multiple attackers in a simultaneous manner would result in a more effective attack. As indicated by the simulation results in [22], the impact of the VN attack was amplified as the number of attackers increased.

To overcome such adverse effects of the VN attack, researchers have proposed a variety of security solutions of different approaches. Table 1 provides a summary of these research proposals. In [23], the detection and mitigation of the VN attack were addressed using a passive monitoring strategy based on having non-constrained nodes added to the RPL topology. An independent instance is formed for these nodes to collect and transmit malicious activities to the sink. Upon the reception of all the monitoring data, the sink runs a distributed detection and mitigation process by analyzing the collected information. It is based on cross-matching the reported information from the different monitoring nodes until identifying the attacking source. However, the experimental results showed that strategic placement of the monitoring nodes needs to be effectively achieved to maintain better network scalability and accuracy rate.

The proposed approach in [24] was based on a trust mechanism enabling an RPL node to not trust a single VN update. Each node needs to have a shield table containing the information of its neighbor nodes of lower rank values. When a DIO message with a new VN is received, the VN update takes place only if the message source exists in the shield table and the majority of the nodes in the table share the same VN update. The simulation results indicated the ability of the proposed approach to achieve high QoS performance and an acceptable accuracy rate. A different trust-based solution was also proposed in [25] to address VN attacks in addition to other routing attacks in RPL networks. It is based on the calculation of node trustworthiness using essential trust metrics. By broadcasting this information across the network, malicious nodes are detected based on their estimated trust rating.

The researchers in [26] proposed SRPL-RP, a detection and mitigation solution for both rank and VN attacks. It uses a DIO message timestamp threshold to discard malicious messages exceeding the threshold. Validated DIO senders are added to a monitoring table whereas malicious nodes are added to a blacklist table. The verification process is based on comparing the current and previous advertised information of the nodes and their parents. The VN verification also relies on the majority of the nodes in the neighbor list table to proceed with the VN update, similar to [24]. Once an attacker is detected, a security alert is then broadcasted to all the nodes to disjoin and isolate it. As indicated by the experimental results, SRPL-RP showed better PDR, power consumption, and accuracy rate when compared with similar research proposals.

In [27], a distributed and cooperative verification mechanism was proposed to defend against VN attacks. It requires the recipient of a DIO message with a new VN to establish communications with selected two-hop neighbors and verify the new update. The information received in the cooperative verification reply messages is then collected in a storing table and utilized to determine the attacking source. Another VN attacks detection method which basically relies on received neighbors' information was proposed in [28]. An illegitimate VN update is detected if the root node and the majority of neighbor nodes do not share the same update. However, the simulation results showed that the performance of the proposed solution degraded noticeably as the number of nodes increased. A similar approach was also introduced in [29] to defend against VN attacks using a decentralized solution. It is based on basically allowing a node to only accept the VN updates received from its best parent and ignore any other updates. The experimental results showed that energy consumption and network overhead were minimized using the proposed solution. However, no investigation was provided as to the great possibility that any best parent would originate the VN attack.

Other research efforts addressed VN attack mitigation based on cryptographic mechanisms. The proposed approach in [30] relied on hash chains and Message Authentication Codes (MAC) to secure the rank and VN information in DIO messages. The VN information was also secured using digital signatures [31] and an Identity-based Offline-Online signature-based scheme [32].

Other research works addressed VN attack detection considering machine learning algorithms. The authors in [33] proposed a framework for processing network data at the cloud/edge using machine learning techniques in a centralized manner. Certain network data indicating beacon intervals, power consumptions, and routing metrics

are transmitted to the cloud/edge on a regular basis. The framework relies on several modules implemented in the cloud/edge: input features filtration, feature preprocessing, and ML classification algorithms. Detected malicious activities in the cloud/edge are reported to the sink node and stored in a local database at the edge. However, no further clarification was made on how the VN attack can be stopped using the proposed solution. The experimental evaluation showed that the proposed solution was able to achieve high accuracy and precision results.

The researchers in [34] proposed a blockchain-based framework for securing the RPL routing process against rank and VN attacks. The blockchain network acts as a secure data link between the RPL network and a machine learning-based attack detection module. It implements an eXtreme Gradient Boosting (XGBoost) classifier on a private blockchain network to detect rank and VN attack attempts. Smart contracts are then used to assess these attempts and issue real-time alerts against malicious nodes. The performance results showed that the proposed blockchain-based solution improved the accuracy achieved by the machine learning algorithms in predicting the attack.

Another machine learning model which adopts the Light Gradient Boosting Machine (LGBM) for detecting VN attacks was introduced in [35]. The proposed methodology mainly included the production of a VN attack dataset, feature extraction module, LGBM-based classification algorithm, and parameters optimization. The performance results showed good performance of the proposed model but also indicated that it requires a memory space of at least 347,530 bytes which is far beyond the capabilities of typical IoT devices (e.i. Zolertia Z1 has a memory of less than 100 Kbytes [47]). This indicates that this solution requires additional devices of high memory resources to run the model in a centralized manner.

However, there are still certain limitations that have not been effectively addressed. It can be seen that the existing approaches for detecting and mitigating VN attacks imposed additional entities [23], [33], [34], [35], node-local processing [24], [25], [26], [30], [31], [32], or communication overhead [27], [28], [33], [34], [35]. Therefore, such limitations need further consideration toward enriching the security of the protocol. Moreover, there was no clear indication of how the convergence time and detection delay would be maintained under these limitations. Given their added complexity to the RPL functionality, the convergence time of the RPL topology would increase in the case of a legitimate global repair as well as the detection delay once an attack has been initiated. Furthermore, no adequate consideration has been made to address the challenge of having multiple VN attacks initiated concurrently across the network. In this research work, we provide a novel approach to support the detection and mitigation of single and multiple VN attacks in an effective and lightweight manner. It was based on a slight modification to the VN update procedure of the standard RPL with lightweight node-local processing, no additional entities, and less communication overhead.

## IV. ATTACK MODEL

This section briefly discusses the basic network assumptions and characteristics considered in this work. The RPL network topology is considered to comprise a single sink node and a set of normal RPL nodes. During attack scenarios, the topology also comprises one or more malicious nodes performing VN attacks. Multiple VN attacks can be individually or cooperatively performed by different nodes of different positions at the same time. The malicious nodes join as legitimate ones and communicate directly with the legitimate nodes. The nodes run an RPL implementation based on the RPL specification provided in RFC 6550 [12] in addition to the CDRPL implementation.

All the nodes are assumed to be battery-powered and small-sized devices that are homogeneous and stationary. So energy efficiency and resource utilization are critical considerations. Examples of real RPL devices are the Zolertia Z1 [47], Tmote [48], MicaZ [49], and TelosB [50] motes which can be implemented to form RPL networks. The nodes can be densely deployed with any form of deployment considering random or uniform positioning. The nodes are fully connected with multi-hop wireless connectivity. The communications take place over multi-hop paths up to the sink node.

The nodes are assumed to be deployed for certain IoT applications. The collection/reception of the application-specific IoT data is done in a periodical manner. The data is transmitted/received in RPL packets over the established RPL paths at a certain time interval. The sink node is the only point at which the data is forwarded to/from the Internet.

It is assumed that the sink node cannot be exposed to any kind of attack. The sink node is specified to be the node that legitimately initiates a global repair and modifies the VN advertised in the DIO messages. To initiate a VN attack, however, an attacking node carries out the same operation during a trickle time interval. It starts forwarding the DIO messages after incrementing the advertised VN. The recipients consider that this advertisement indicates a running legitimate global repair as it comes from a node joining the network. They have no option but to accept the fake global repair and participate in the process according to the RPL specification. Each node updates its DODAG version record, resets its trickle time, and then forwards the DIO messages with the newly advertised VN. The VN attack is initiated after the initial setup of the RPL network. The objective of the attacker is to disrupt the network stability with unnecessary global repairs and flood the network with control packets. The main target is to cause a considerable drop in network performance and a reduction in network lifetime [19], [20], [21].

## V. SOLUTION

The original RPL specification [12] specifies that a legitimate global repair is initiated by the sink node sending a DIO

**TABLE 1. Summary and comparison of RPL VN attack detection and mitigation approaches.**

| Ref. | Methodology | Evaluation | Results | Limitations | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Additional Entities | Computational Complexity | Communication Overhead | Detection Delay | Only Single Attacks |
| [23] | Passive monitoring strategy using non-constrained nodes | Simulation (Cooja) | strategic placement of the monitoring nodes is critical | √ | × | × | × | √ |
| [24] | A trust-based mechanism for approving new VN updates from only the majority of neighbor nodes | Simulation (Cooja) | high QoS performance and an acceptable accuracy rate | × | √ | × | × | √ |
| [25] | Trust-based solution based on the calculation of node trustworthiness using essential trust metrics | Analytical (comparative) | It considers node mobility and sink movements compared to other approaches | × | √ | × | × | √ |
| [26] | VN update verification based on comparing the current and previously advertised VN information and DIO message timestamp threshold | Simulation (Cooja) | Improved PDR, power consumption, and accuracy rate | × | √ | × | × | √ |
| [27] | VN updates verification with selected two-hop neighbors | Simulation (Cooja) | High PDR and Low control overhead | × | × | √ | √ | × |
| [28] | VN updates verification with the root and the majority of neighbor nodes | Simulation (Cooja) | The performance is degraded as number of attacks increased | × | × | √ | √ | √ |
| [29] | A decentralized solution based on basically accepting VN updates only from the best parent | Simulation (Cooja) | energy consumption and network overhead were minimized | × | × | × | √ | √ |
| [30] | Cryptographically centered protection scheme using Hash Chains and Message Authentication Codes (MAC) to secure rank and VN information | Analytical (Quantitative) | Low time overhead | × | √ | × | × | √ |
| [31] | Securing VN information using digital signatures | Physical Testbed | Low resource consumption and convergence time | × | √ | × | × | √ |
| [32] | Securing VN information using an Identity-based Offline-Online signature-based scheme | Analytical (Quantitative) | Low computational time and energy consumption | × | √ | × | × | √ |
| [33] | A security framework based on processing network data at the cloud/edge using machine learning techniques | Simulation (Cooja) | high accuracy and precision | √ | × | √ | × | × |
| [34] | A blockchain-based framework implementing the XGBoost classifier on a private blockchain network with a smart contract to detect rank and VN attacks | Simulation (Cooja) | It helped improving the accuracy achieved by the machine learning algorithms | √ | × | √ | × | √ |
| [35] | Implementation of a machine learning model based on the Light Gradient Boosting Machine (LGBM) for the detection of VN attacks | Simulation (Cooja) | High accuracy and precision | √ | × | √ | × | × |
| Prop. Work | Simple in-protocol modifications incorporating a lightweight collaborative and distributed security scheme | Simulation (Cooja) | Maintained high network stability, control overhead, QoS performance, and energy consumption | × | × | × | × | × |

message with a new VN (donated as $DIO_{nv}$). Upon the reception of this message, RPL nodes participate in the process by processing and then forwarding the message. However, blindly accepting and acting upon such messages without further verification would allow attackers to initiate harmful VN attacks in the network. It is critical to examine the legitimacy of a running global repair before any further processing. Therefore, enhancing the immunity of RPL against VN attacks requires extending the DODAG maintenance process to a further level. However, taking into account certain considerations is important before designing any solution for RPL networks. These include the scarce resources of typical

RPL devices and the need for maintaining low processing and traffic overhead in RPL networks.

Accordingly, the proposed CDRPL in this research work is based on effective collaboration among the nodes to enable a simple, yet efficient, verification scheme. It applies such an approach in a distributed manner to enhance RPL functionality to a further security limit. Effective protection against VN attacks is enabled by only introducing a slight modification to the DODAG maintenance process without adding to the complexity of the RPL design. No additional hardware resources nor network entities are imposed by the design of CDRPL. It completely relies on incorporating an efficient collaborative and distributed scheme into RPL functionality.

CDRPL is built based on the principle that RPL global repair takes the downward direction of the DODAG. In the standard RPL, the sink node firstly increments the current VN and then resets the trickle timer to start immediate broadcasting of $DIO_{nv}$. The nodes at the upper level of the DODAG are the first recipients of these messages. They then carry on the global repair process and forward the messages down to their sub-DODAGs. The process continues one level after another following a downward direction. Accordingly, the global repair process in the standard RPL takes place at the same level in the different sub-DODAGs simultaneously [12].

Making effective use of such behavior helps CDRPL in establishing a collaborative and distributed scheme among the nodes within different DODAG segments. This is based on enabling RPL nodes to not trust any advertisement of a new VN until verifying that a relevant legitimate global repair is taking place in another sub-DODAG of the network. Therefore, CDRPL prevents the node from blindly participating in any global repair process to only allow legitimate VN updates. Although this does not stop the initiation of VN attacks, CDRPL is highly effective in isolating the attackers and ensuring protection against the adverse effect of such attacks.

Accordingly, CDRPL modifies RPL operation to enable effective cooperation among RPL nodes for verifying global repairs. During any global repair initiated in the network, CDRPL goes into a legitimacy check stage to decide whether to participate in the global repair. This is based on simple in-protocol communications enabled among the nodes to detect any VN attack. Once an attack is detected, CDRPL then moves to the stage of attack mitigation which is managed by the sink node. The following subsections explain CDRPL operation during legitimate and non-legitimate global repair in addition to presenting a thorough example.

### A. CDRPL OPERATION

Upon the reception of a $DIO_{nv}$, CDRPL enables the recipient to verify the legitimacy of such a new version advertisement before participating in the indicated global repair. There are here two possibilities to consider. It could be that the message reached the recipient after being legitimately initiated by the sink node or as a result of an attack initiated by a node. In both cases, we need to ensure that only legitimate messages initiated by the sink are processed. Suspicious messages need to be discarded after being reported to the sink node for further action.

**TABLE 2. The cases indicating legitimate VN updates.**

| Case | Description |
|---|---|
| C1 | A node received a $DIO_{nv}$ from the sink |
| C2 | A node received a $DIO_{nv}$ from its parent and then receives an $S\text{-}DIO_{nv}$ from a neighbor joining a different sub-DODAG of the network |
| C3 | A node received a $DIO_{nv}$/$S\text{-}DIO_{nv}$ from a neighbor joining a different sub-DODAG of the network and then receives a $DIO_{nv}$ from its parent |
| C4 | A node received a $DIO_{nv}$/$S\text{-}DIO_{nv}$ from its parent and then receives a DIOnv from any neighbor/child node |
| C5 | A node, having a sub-DODAG of a single node with no other neighbors, received $S\text{-}DIO_{nv}$ from its parent and later receives a $DIO_{nv}$ from it |
| C6 | A leaf node having no other neighbors than its parent receives a $DIO_{nv}$ from it |

Any $DIO_{\text{nv}}$ is inspected before taking any global repair action. This is achieved with a collaborative verification mechanism established among the nodes. This requires each node to announce the reception of a $DIO_{nv}$ using a special $DIO_{nv}$ (S-$DIO_{nv}$). It is a copy of the $DIO_{nv}$ that has the newly advertised VN and is tagged with a special flag, called the $S\text{-}DIO_{nv}$ flag, by using a single bit of the flags field of the DIO Base Object. It indicates that this message should not be normally processed, but rather handled by the collaborative verification process run by each node in the DODAG. Additionally, $S\text{-}DIO_{nv}$ carries the node's current parent information in an additional option attached to the DIO base object to help in identifying the source of the message.

The flow diagram presented in Figure 2 provides an overview of the main procedure for processing $DIO_{nv}$ at each node. It describes several cases that affirm the legitimacy of the advertisement. Each case approves the node to safely participate in the running global repair. By this, the node updates its current recorded VN of the DODAG, resets its trickle time, and then starts broadcasting $DIO_{nv}$ advertisements. The cases are described in Table 2.

As indicated in Figure 2, the process starts when a $DIO_{nv}$ is received at an RPL node. If the sender is the sink node, this is exactly C1 thus the node carries on the global repair and starts broadcasting the $DIO_{nv}$ to its neighbors. In the case of the sender being the recipient's parent, the recipient records the $DIO_{nv}$ information and then sends $S\text{-}DIO_{nv}$ to its neighbors. Note that a single transmission of $S\text{-}DIO_{nv}$ is performed without complying with the trickle time rules. Rather, it is sent every time a $DIO_{nv}$ is received from a parent node which normally resets the trickle time. This ensures sufficient announcements while maintaining overhead at a low level.

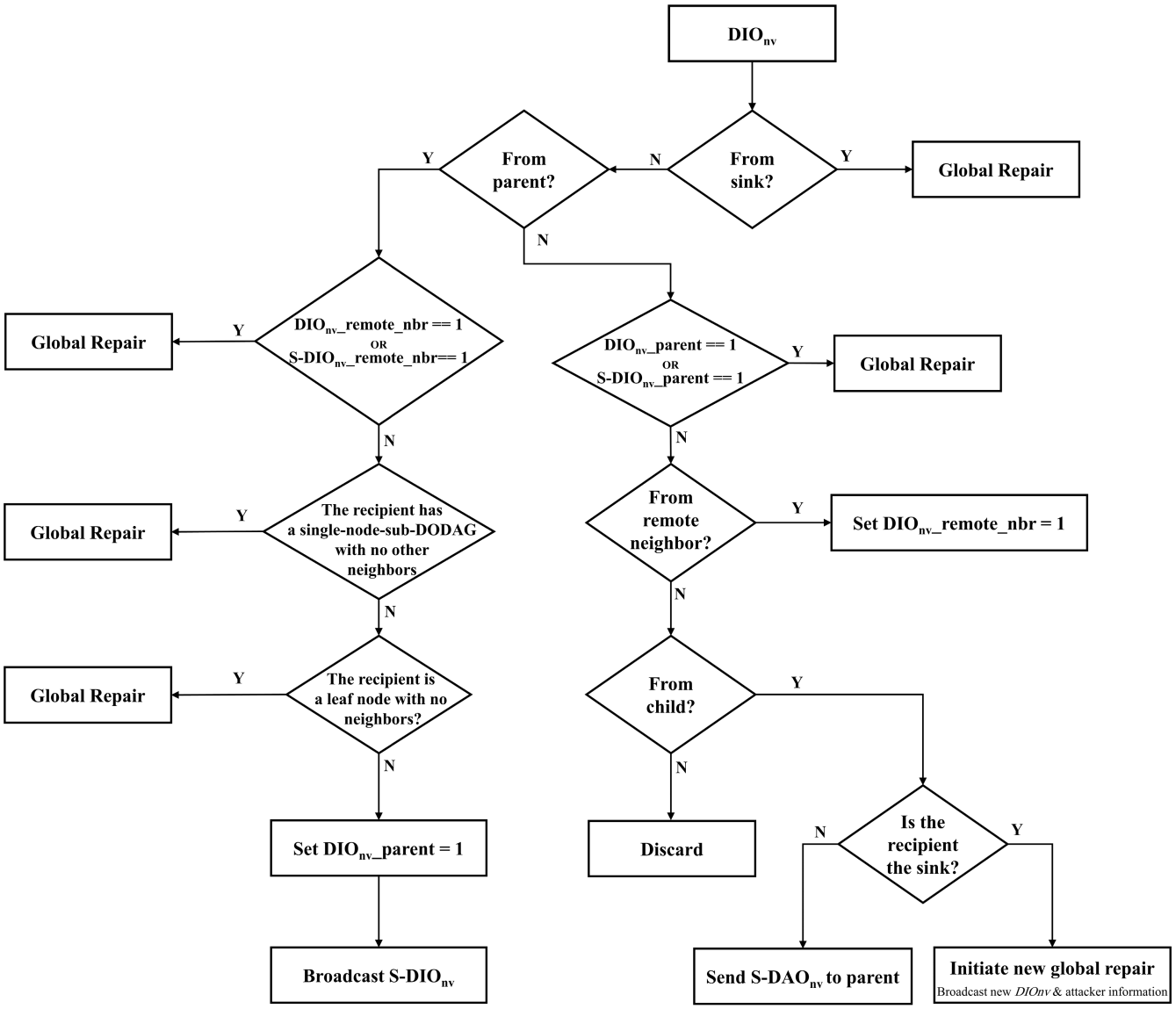


**FIGURE 2.** An overview of the $DIO_{nv}$ processing procedure.

As a result, a node can receive *S-DIO*$_{nv}$ from its parent, child, and other neighbor nodes. If it comes from its parent, the node only records the message information and forwards it. Otherwise, we need to differentiate between neighbors within the node's sub-DODAG and those outside its sub-DODAG in a different segment of the network. That is, having a node receiving a *DIO*$_{nv}$ or *S-DIO*$_{nv}$ from a neighbor located outside its sub-DODAG indicates that the global repair is taking place in a different network segment. It is a confirmation that the new version is advertised by other parents in addition to the node's parent. Accordingly, the node can then rest assured to participate in the global repair if it has already received a *DIO*$_{nv}$ from its parent. This case fits in C2 and is also indicated in C4. However, there is no further action to be taken if an *S-DIO*$_{nv}$ is received from a child or neighbor within the same sub-DODAG. This prevents any attacker at the top level of the sub-DODAG from influencing the nodes' decisions and reduces the false positive rate. The flow diagram in Figure 3 presents an overview of the *S-DIO*$_{nv}$ processing procedure.

The DODAG topology can also include secluded nodes with only one single child node while having no other neighbors. In this case, there is no harm in accepting their parents' advertisements as they have no means to participate in the collaborative verification process with other nodes. As indicated in C5, it then needs to first receive *S-DIO*$_{nv}$ from its parent before accepting any received *DIO*$_{nv}$ from its parent. A similar consideration can be taken for the leaf nodes having no other neighbors as indicated in C6.

On the other hand, a node can receive a $DIO_{nv}$ from other nodes than its parent node. That is, the sender can be a neighbor or child node. In both cases, the recipient node participates in the global repair if it has already received a $DIO_{nv}$ or $S\text{-}DIO_{nv}$ from its parent, as indicated in C4. These messages assure that a legitimate global repair is running in different segments of the RPL network. Otherwise, the received $DIO_{nv}$ message is discarded and if it is from a neighbor located outside the recipient's sub-DODAG, its information is recorded.

### B. CDRPL DURING A VN ATTACK

The case of non-legitimate global repair occurs once a node receives a $DIO_{nv}$ from a child node without any prior $DIO_{nv}$ or $S\text{-}DIO_{nv}$ advertised by the recipient's parent node. This indicates malicious activity as it violates the normal flow of a legitimate global repair process. That is, it comes from one direction which is the upward direction without any prior relevant advertisements from the downward direction of the topology. Then, we need to report the information of the malicious $DIO_{nv}$ to the sink by sending a special DAO ($S\text{-}DAO_{nv}$) message. The formation of $S\text{-}DAO_{nv}$ is based on using a normal DAO message tagged with a special flag by setting a single bit of the flags field of the DAO base object. It also includes the $DIO_{nv}$ sender address and the newly advertised VN using an additional option attached to the DAO base object. Similar to $S\text{-}DIO_{nv}$, only a single $S\text{-}DAO_{nv}$ is sent every time the malicious $DIO_{nv}$ is received to maintain a low frequency of such messages without adding much to the processing and traffic overhead. It can also be noted that the malicious $DIO_{nv}$ would also be received by other nodes which could be other child and neighbor nodes. However, this is handled by the explained procedures above which mitigate any possible attack effectively.

Upon the reception of an $S\text{-}DAO_{nv}$ reporting a malicious VN increment, the recipient directly forwards it to its parent with no further processing. Once it reaches the top of the DODAG, the sink identifies the running VN attack and initiates a new legitimate global repair process with a new $DIO_{nv}$ of a different VN. Along with that, it informs the DODAG nodes of the malicious activity and tagged the malicious node as an untrusted node that must not be joined anymore. This is achieved by attaching a new option to the DIO base object of the new $DIO_{nv}$ to include the attacker's information. Receiving the new option instructs the recipient to blacklist the indicated attacker. It must break any current attachment to and discard any future communication with the attacker. This would enable the protocol with the capability to provide effective isolation of the attacker.

Moreover, CDRPL is highly effective in protecting the network from being affected by broadcasting false $S\text{-}DIO_{vn}$ messages. Although it does not prevent these messages from being transmitted, CDRPL eliminates the possibility of utilizing them to initiate a new attack and degrade the security and overall performance of the network. If a false $S\text{-}DIO_{vn}$ message is received, it is mostly going to be discarded by

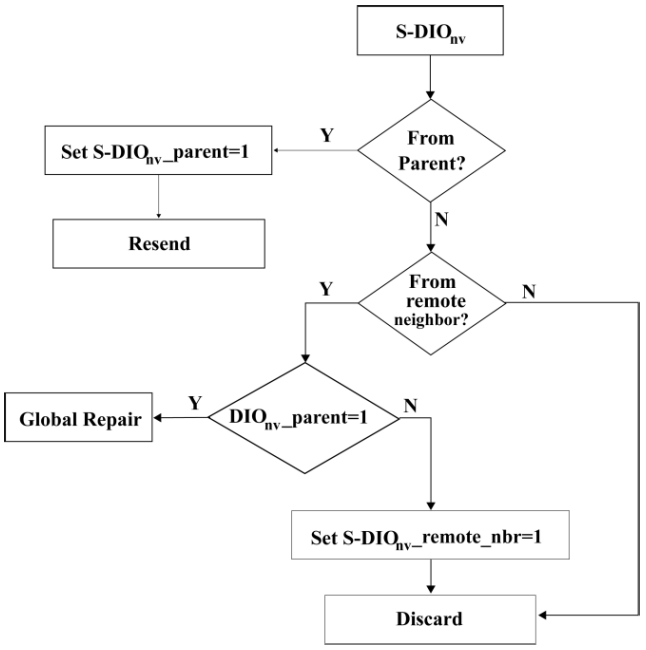


**FIGURE 3.** An overview of the $S\text{-}DIO_{nv}$ processing procedure.

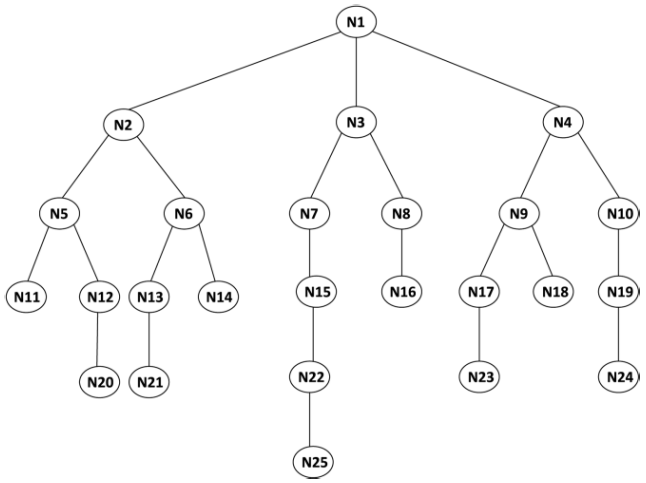


**FIGURE 4.** An example RPL DODAG.

CDRPL as illustrated in Figure 3. The message is only processed in cases when the sender is the recipient's parent or a remote neighbor. In both cases, CDRPL allows the recipients to only record the event with no further actions. Therefore, no malicious node can make use of the $S\text{-}DIO_{nv}$ procedure of CDRPL to initiate a new attack and cause network volatility.

### C. EXAMPLE OF CDRPL UNDER LEGITIMATE GLOBAL REPAIR

Let's elaborate further on how CDRPL operates during a legitimate global repair in the example DODAG presented in Figure 4 and the operational overview provided in Figure 5. We used a filled circle in Figure 5 to indicate that a node has verified the local repair process and started forwarding

$DIO_{nv}$ while using a non-filled circle to indicate otherwise. As shown in Figure 5-a, once the N2-N4 receives a $DIO_{nv}$ from N1 (the sink node), they participate in the running global repair (C1). They update the DODAG VN in their records, reset their trickle time, and forward the received message further. N5-N10 then receive the forwarded $DIO_{nv}$ and record the information of the message. Each one of these nodes then individually sends an $S\text{-}DIO_{nv}$ announcing the reception of the $DIO_{nv}$ messages from their parents. Note that these nodes have not participated in the global repair and forwarded the $DIO_{nv}$ messages yet. The $S\text{-}DIO_{nv}$ will then be received, recorded, and forwarded by N11-N19 to their neighbor nodes. This process is repeated by each recipient receiving an $S\text{-}DIO_{nv}$ from its parent. As a result, all the nodes across the DODAG become informed of the running global repair.

Meanwhile, N5 and N6 will receive each other's S-$DIO_{nv}$ as neighbors, and the same happens for N7 and N8 on one side as well as N9 and N10 on the other side. However, no action will be taken by the nodes as each of these pair nodes belongs to the same sub-DODAG. N6 and N7, neighbors joining a different sub-DODAG, also receives each other's $S\text{-}DIO_{nv}$ after already having $DIO_{nv}$ received from their parents (C2). So the nodes participate in the global repair and start forwarding $DIO_{nv}$, as shown in Figure 5-b. N5 and N8, which have already received their parents' $DIO_{nv}$, receive the forwarded $DIO_{nv}$ and then participate in the process (C4), as shown in Figure 5-c. $DIO_{nv}$ of N6 and N7 also reach their child, N13 and N15, respectively. Since they are neighbors joining different sub-DODAGs and have already received each other's $S\text{-}DIO_{nv}$, Figure 5-d shows that they participate in the global repair (C3). Then, they start forwarding the $DIO_{nv}$ which is then received by N14 and N22. Having already received the advertisement from its parent, N14 participates in the global repair (C4). For N22, it also participates in the process as it already recorded an $S\text{-}DIO_{nv}$ from its parent while having a single node sub-DODAG with no other neighbors (C5). They then start forwarding the $DIO_{nv}$ which is then received by N21 and N25. As these are leaf nodes with no other neighbors, they carry on the global repair (C6). The same also goes for N16 once it receives the $DIO_{nv}$ from its parent as shown in Figure 5-e.

Other neighbors joining different sub-DODAGs are N12 and N18 which have received each other $S\text{-}DIO_{nv}$ as shown in Figure 5-f. As a result, the former participates in the global repair and then starts forwarding the $DIO_{nv}$ since it has already received the $DIO_{nv}$ from its parent, N5 (C2). This is presented in Figure 5-g which also shows that N11 receives the forwarded advertisement and then participates in the process since it received an earlier $DIO_{nv}$ from its parent (C4). N20 also participates in the process after receiving the forwarded message as being a leaf node with no other neighbors (C6). Note that N11 is a leaf node but with a neighbor, so it does not fit in C6 but works for C4.

Going back to N18, Figure 5-g shows that it participates in the global repair after receiving N12's $DIO_{nv}$ (C4). It then starts forwarding the advertisements which reach its parent, N9, and lead it to join the global repair. This is a result of

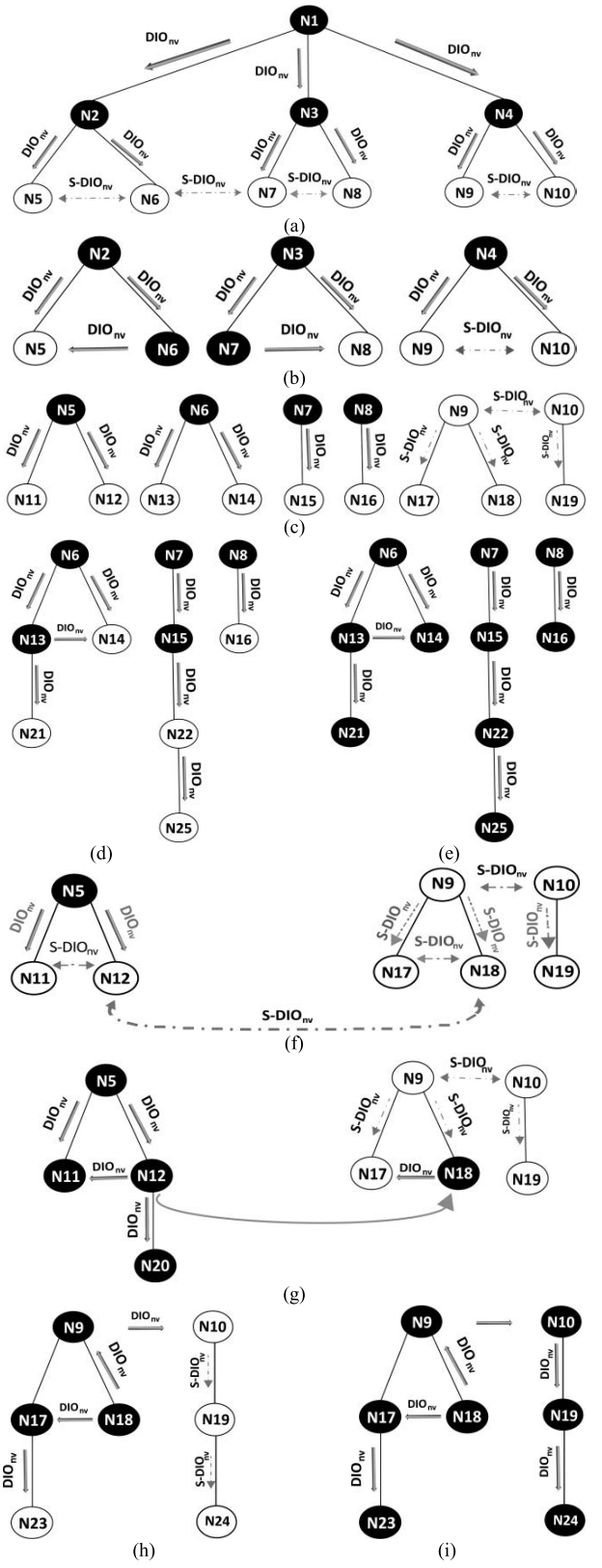


**FIGURE 5.** Example of the flow of the global repair process. (a) – (i) demonstrate the different stages of the process.

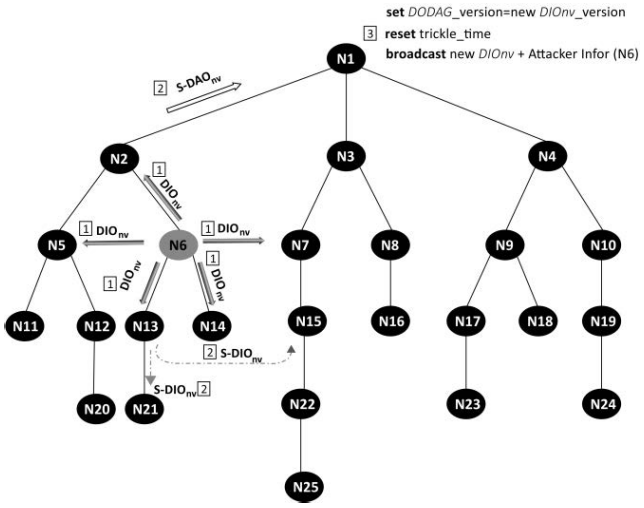


**FIGURE 6.** Example of a simple VN attack scenario with a single attacker (N6).

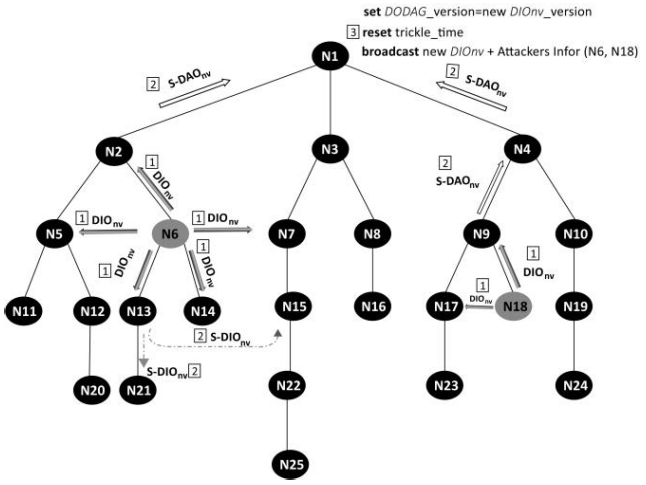


**FIGURE 7.** Example of a composite VN attack scenario with two attackers (N6 and N18).

having one of N9's child nodes able to verify the legitimacy of the process while it has already received the advertisement from its parent, N4 (C4). N18's advertisement is also received by N17 which then decides to participate in the process. This is because it has already received its parent's $DIO_{nv}$ (C4), as shown in Figure 5-h. This then enables N10 to participate in the global repair after receiving N9's $DIO_{nv}$ (C4). Having a sub-DODAG of a single node and an $S\text{-}DIO_{nv}$ already received from its parent, N19 also participates in the global repair (C5). Finally, the process converged at N23 and N24 which participate in the process as being leaf nodes with no other neighbors (C6), as shown in Figure 5-i.

### D. EXAMPLE OF CDRPL UNDER NON-LEGITIMATE GLOBAL REPAIR

The example DODAG in Figure 4 can also be used to illustrate how CDRPL operates during a VN attack. Assuming that N6 initiates the attack as shown in Figure 6, a new $DIO_{nv}$ is then received by its neighbors: N2, N5, N7, N13, and N14. For each, this is a new version advertisement with no prior relevant $DIO_{nv}$/ $S\text{-}DIO_{nv}$ received from its parent or any other node. Therefore, N2, being the parent node, considers this a malicious activity and sends an $S\text{-}DAO_{nv}$ via its parent. It contains information on the new VN and the attacking node. The message reaches N1 (the sink) which then identifies the attack attempt as it has no running global repair with the reported VN. It then initiates a new legitimate global repair process with a new $DIO_{nv}$ of a different VN. In addition, it adds a new option to the message to include N6's ID and address. This indicates that N6 is an attacking node and instructs other nodes to not trust and join it anymore.

Meanwhile, N6's $DIO_{nv}$ reaches the other nodes but with no impact on the overall node operation and network security. It is discarded by N5 since it is a new $DIO_{nv}$ coming from a neighbor joining the same sub-DODAG. N7, being relevantly a remote neighbor, receives the message and then records its information before discarding it. N13 and N14 receive the message from their parent, so they record its information and then broadcast $S\text{-}DIO_{nv}$. Once received by N21, the $S\text{-}DIO_{nv}$ is resent after recording its information. Being a neighbor joining a different segment of the network, N15 records the message information and then discards it. This is the extent to which the messages reach whereas the other parts of the network stay unaware of the running attack. The mechanism keeps the attack harmless within the sub-DODAG of the attacker while isolating most of the nodes in the other segments of the network.

The example presented in Figure 7 also illustrates the ability of CDRPL to considerably limit the impact of the attack while having a composite VN attack across the network. When N6 and N18 initiated the attacks, the same procedure illustrated in Figure 6 was performed against the N6 attack. Regarding the other attack, the $DIO_{nv}$ of N18 is processed by N9 as a malicious message since it is a new VN advertisement coming from its child node. It is then reported to the sink by sending an $S\text{-}DAO_{nv}$ via its parent. The message is received and forwarded by N4 to the sink. The malicious $DIO_{nv}$ is also received and discarded by N17 since it comes from a neighbor joining the same sub-DODAG.

Upon the reception of the $S\text{-}DAO_{nv}$ messages from N2 and N4, the sink initiates a new global repair process with a new $DIO_{nv}$ of a different VN. For each attacker, a new option is added to the message to include the attacker's ID and address information. It can be seen that CDRPL succeeded in containing the attack even with multiple attacks running at the same time and in different positions. Notice that the attacks in this example are mitigated while almost half of the nodes remain completely unaware of the attempts.

## VI. EXPERIMENT

IoT devices require customized operating systems (OSs) that work well under constrained resources and limited

capabilities. Commonly adopted OSs in this context are Contiki OS and TinyOS. These provide open-source implementations of network stacks operating 6LowPAN and RPL. The 6LowPAN implementation provides IP adaptation whereas the RPL implementation enables IPv6-based network routing. Contiki OS [51] incorporates the Cooja network simulator which enables effective configuration and evaluation of a variety of emulated RPL-based IoT setups. It facilitates the emulation of a wide range of IoT scenarios with different types of virtual IoT motes running the real Contiki OS implementation. For the implementation and experimentation of this work, the Cooja simulator under the latest Contiki OS version (Contiki 3.0) was utilized.

To implement CDRPL, the code base of the RPL implementation of the Contiki OS was modified. The source files, rpl-dag.c and rpl-icmp6.c, were the ones that incorporate most of the code modifications. The rpl-dag.c file was modified to implement the proposed processing of the $DIO_{nv}$, $S\text{-}DIO_{nv}$, and $S\text{-}DAO_{nv}$ messages. The implementation of the handling of $S\text{-}DIO_{nv}$ and $S\text{-}DAO_{nv}$ transmission was integrated into the rpl-icmp6.c file.

For the effective evaluation of CDRPL, the Cooja simulator was utilized to build three experimental setups of varying-scale and varying-complexity RPL network topologies. In each setup, a single RPL instance containing a DODAG of one sink node and a different number of RPL nodes was considered. The setups are donated S1, S2, and S3 which have a total of 25, 40, and 65 nodes, respectively. The example DODAG presented in Figure 4 also represents the network topology of S1. The placement of the nodes was done randomly in the simulated deployment area. Each of the RPL sink and regular nodes was configured as a Zolertia Z1 mote that runs an MSP430 16-MHz MCU with an 8KB RAM, 92KB flash memory, and CC2420 transceiver. The adopted OF in all the experiments was the MRHOF with its basic routing metric, ETX. A summary of the main simulation parameters is provided in Table 3.

A UDP client was configured to run for every RPL node for the transmission of IoT data traffic. The packets are frequently transmitted by every node at a communication interval of ±5 seconds. The data traffic of all the nodes is received by a UDP server that runs at the sink node. In addition, every node is configured to run different Cooja plugins such as the collect-view and powertrace modules which provide additional experimental simulation data. Other important configurations were the communication range which was set to 25 meters and the interference range which was fixed to 50 meters for all the nodes.

The design of the evaluation methodology was based on a set of different stages. At the initial evaluation stage, each one of the three experimental setups was run using only the original RPL implementation and considering attack-free scenarios. This helped in establishing a performance baseline for effective comparison against the network performance in the following stages. The CDRPL implantation was then tested in the next stage for each experimental setup without employing any attack scenario. The collected results at these stages were utilized to gain an initial insight into the efficiency and performance of CDRPL.

**TABLE 3. Simulation parameters.**

| Simulation Parameter | | Value |
|---|---|---|
| Area Size | | 300×300 m |
| Number of Nodes | S1 | 25 |
| | S2 | 40 |
| | S3 | 65 |
| Topology | | Random |
| Mote Type | | Zolertia Z1 |
| Mote Current Consumption in CPU Mode | | 0.5 mA at 3V |
| Mote Current Consumption in LPM Mode | | 0.0005 mA at 3V |
| Mote Current Consumption in Tx Mode | | 17.4 mA at 3V |
| Mote Current Consumption in Rx Mode | | 18.8 mA at 3V |
| RaDIO Medium Model | | UDGM: Distance Loss |
| Operating System | | Contiki 3.0 |
| MAC Layer | | ContikiMAC |
| RPL Objective Function | | MRHOF (ETX ) |
| RPL Routing Mode | | Storing Mode |
| Communication Range | | 25 m |
| Interference Range | | 50 m |
| Traffic Type | | CBR |
| Data Transmission Interval | | ±5 seconds |
| Control Message Size | | 4 Bytes |
| Data Packet Size | | 40 Bytes |
| Confidence Level | | 95% |
| Simulation Duration | | 50 Mins |

The next evaluation stage was based on examining the original RPL implementation in each experimental setup considering different attack scenarios of a single attacker. To enable this, the application file of the attacker node was modified to increment the VN being added to the advertised DIO message. Then, in the next evaluation stage, the same attack scenarios were repeated while having the nodes running the CDRPL implementation. Furthermore, the composite VN attacks were considered in the final evaluation stage which incorporated several attack scenarios with multiple attackers. For this, we only considered the S2 setup for the implementations of both the original RPL and CDRPL.

A different selected node is configured in each attack scenario to run the VN attack after 10 minutes of the simulation start time. We made sure that varying nodes of different properties such as DODAG position and neighbor count were selected. Each simulation run lasted for a simulation time of 50 minutes and was repeated 10 times to take the average of the collected results.

The performance of CDRPL was compared with the performance of the standard RPL under different attack scenarios. Furthermore, a comparison with other existing VN attack countermeasures, namely RPL+Shield [24], SRPL-RP [26], VeNADet [28], and LRPL [29], was also provided. The

evaluation was based on a number of network measurement parameters providing an effective indication of the overall network performance. These can be classified as follows:

- QoS-oriented performance: Packet Delivery Ratio (PDR), throughput, end-to-end delay, and ETX
- Network overhead: DIO transmission rate, DAO transmission rate, overall control packets transmission rate.
- Network stability: Preferred Parent Change (PPC) rate.
- Energy efficiency: energy consumption.
- Protocol response latency: convergence time and VN attack detection delay.
- Protocol Accuracy rate: True Negative (TN) and False Negative (FN) rates.

The PDR was calculated as the ratio of all the data packets received by the UDP server to all the data packets transmitted by the UDP clients. For throughput, the average of the total number of successfully transmitted bits of the data packets per second was calculated. The calculation of the end-to-end delay was based on the average time needed by the transmitted data packets to arrive at the UDP server. The ETX was calculated as the average of the total number of transmissions required to successfully deliver data packets.

The calculation of the network overhead (OH) rate was based on the average number of control packets (DIO, DAO, DIS, $DIO_{nv}$, $S\text{-}DIO_{nv}$, and $S\text{-}DAO_{nv}$ messages) being exchanged across the network per minute. To calculate the PPC rate, the total number of changes made by all the nodes to their preferred parent during the simulation time. This was divided by the number of nodes and then the average was calculated.

The powertrace module provided by the Cooja simulator was utilized to collect the data required for the calculation of the energy consumption metric. It provides the time spent during different mote states (CPU, Low power mode, Transmit, and Listen states). These were multiplied by their corresponding current consumption levels and the power supply voltage as specified in the Zolertia Z1 datasheet [48].

Other important evaluation measures taken into consideration included the convergence time and VN attack detection delay. The convergence time is the time required to complete a legitimate global repair process. It is the difference between two specific events. The first is when the sink node initiates the process by transmitting the $DIO_{nv}$ message. The second is when the message is received by all the nodes across the network. This was calculated for each global repair process to work out and then average the total for each simulation run. On the other hand, the attack detection time refers to the time needed by the sink to detect malicious activity and start acting upon it. It is calculated by taking the average of the total differences between two specific events during each simulation run. The first is when an $S\text{-}DAO_{nv}$ message is initially sent by a node. the second is when the message is received by the sink node.

We also measured the detection accuracy which indicates the efficiency of identifying legitimate and malicious global repair. If there is a legitimate global repair running across the network, it is considered True Positive (TP) once a node updates its current VN accordingly. Otherwise, it is considered False Negative (FN) as it does not participate in a legitimate process. In the case of having a malicious global repair, it is considered True Negative (TN) if a node does not participate in the process and refrains from updating its current VN. Otherwise, it is considered False Positive (FP) once participating in a malicious global repair.

## VII. RESULTS AND DISCUSSION

Given the inherent functionality of CDRPL, it is important to understand how quickly it responds to the global repair process. Figure 8 presents a comparison of the convergence times required by the standard RPL and CDRPL in the different experimental setups with and without a running attack. It can be seen that CDRPL added to the time required by the new topology to converge in all the considered scenarios when compared with the standard RPL. Considering attack-free scenarios, the convergence time increased by approximately 8 seconds when running CDRPL, resulting in 45-55% additional delay. It adds such a delay as a result of the collaborative security mechanism of CDRPL. That is, additional time is taken during topology convergence at the initial stages to verify the running global repair process. Comparing such delay increases, however, CDRPL incurred less convergence delay when it became under a VN attack compared with non-attack scenarios. In the S2 setup, for example, the convergence time increased by less than 5% in the case of CDRPL whereas the standard RPL required approximately 25% additional time to converge. That is, CDRPL imposed similar performance in all the scenarios, compared to the standard RPL which became less robust with a convergence time increase of more than 20% once being attacked.

Considering the time required by the sink node to detect the attack and initiate a legitimate global repair, CDRPL significantly outperforms the standard RPL. As presented in Figure 9, the sink response delay was considerably higher in the case of the standard RPL. It experienced a delay of more than 10 seconds in the S1 setup and reached almost 18 seconds as the network scaled up in the S3 setup. This was due to the absence of attack detection and alerting mechanisms in the standard RPL. It rather relies on the sink to detect the VNs inconsistency after being widely advertised across the RPL network. CDRPL improved the protocol response to the attack and achieved a reduction of more than 90% of the attack detection time. It only took approximately a second for the sink to detect the attack and act upon it. Notice that the detection delay in the case of CDRPL increased by no more than 2% as the network scaled up.

Tables 4, 5, and 6 present the network overhead results considering the different scenarios in S1, S2, and S3 setups, respectively. In attack-free scenarios, the results show that CDRPL experienced a slight increase in the overall control packet rate. This is due to the security-oriented functionality added by CDRPL, introducing new lightweight DIO and DAO-based communications ($S\text{-}DIO_{nv}$ and $S\text{-}DAO_{nv}$). As a

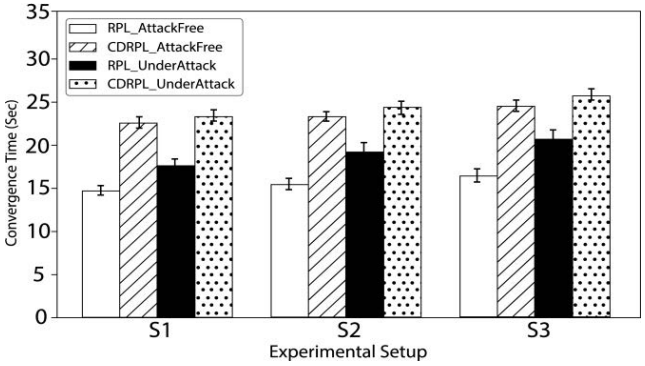


**FIGURE 8.** Topology convergence time results.

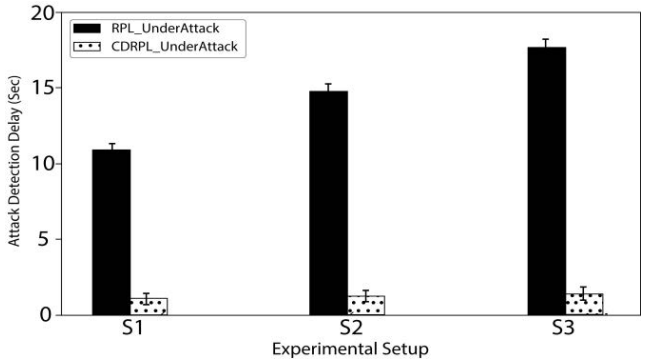


**FIGURE 9.** VN attack detection delay results.

**TABLE 4.** Network overhead results – S1 setup.

| Protocol | DIO_rate | DAO_rate | OH_rate |
|---|---|---|---|
| RPL_AttackFree | 6 | 6 | 12 |
| CDRPL_AttackFree | 9 | 7 | 16 |
| RPL_UnderAttack | 55 | 48 | 104 |
| CDRPL_UnderAttack | 10 | 9 | 19 |

**TABLE 5.** Network overhead results – S2 setup.

| Protocol | DIO_rate | DAO_rate | OH_rate |
|---|---|---|---|
| RPL_AttackFree | 20 | 20 | 40 |
| CDRPL_AttackFree | 26 | 23 | 50 |
| RPL_UnderAttack | 203 | 147 | 352 |
| CDRPL_UnderAttack | 36 | 32 | 69 |

result, the overall DIO and DAO transmission rates increased for all the setups. The *S-DIO*$_{nv}$ and *S-DAO*$_{nv}$ communications incurred additional network overhead of approximately 30% and 25%, respectively. It can also be noticed in all the scenarios that CDRPL exhibited similar behavior to the standard RPL in terms of having the rate increase as the network scaled up.

However, the standard RPL suffered the adverse effect of the NV attack on the network overhead considering all

**TABLE 6.** Network overhead results – S3 setup.

| Protocol | DIO_rate | DAO_rate | OH_rate |
|---|---|---|---|
| RPL_AttackFree | 38 | 35 | 74 |
| CDRPL_AttackFree | 45 | 40 | 87 |
| RPL_UnderAttack | 377 | 258 | 637 |
| CDRPL_UnderAttack | 66 | 52 | 120 |

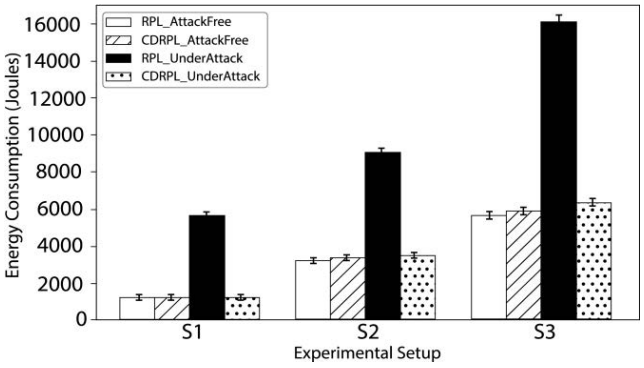


**FIGURE 10.** Energy consumption results.

three experimental setups. In S1, 104 control packets were transmitted per minute by the standard RPL. The situation became even worse as the network scaled up, resulting in the transmission of more than 300 and 600 control packets per minute in S2 and S3, respectively. This was due to the inability of the standard RPL to mitigate the NV attack in an efficient manner. Rather, it allows the attack to spread across the network as there is no mechanism to prevent forwarding malicious *DIO*$_{nv}$ messages. To rectify such critical situations, CDRPL provides a mechanism that ensures only legitimate *DIO*$_{nv}$ messages are forwarded. Although this incurred additional network overhead in the attack-free scenarios,

CDRPL comes with the benefits of significantly reducing the adverse effect of NV attacks on network overhead. When the CDRPL network became under attack, the results show that CDRPL experienced a transmission of additional 33 control packets per minute at most in all the experimental setups. In S2, for example, it exhibited an increase of only 10 and 9 packets per minute for the DIO and DAO transmission rates, respectively. This means that the *S-DIO*$_{nv}$ and *S-DAO*$_{nv}$ communications resulted in only 19 control packets being additionally transmitted per minute during the VN attack scenarios. Compared to the standard RPL, CDRPL managed to mitigate the VN attack with at least a 75% reduction in network overhead.

The energy consumption results are presented in Figure 10. The results illustrate how CDRPL was able to reduce the consumed energy by more than 60% compared to the standard RPL in the VN attack scenarios for all the experimental setups. It maintained similar levels of energy consumption in the attack-free and VN attack scenarios whereas the standard RPL failed to show such a maintained behavior. In S3, for

**TABLE 7.** Detection accuracy rate results.

| Accuracy | Protocol | S1 | S2 | S3 |
|---|---|---|---|---|
| TN | RPL_UnderAttack | 20.83% | 23.08% | 18.75% |
| | CDRPL_UnderAttack | 99.01% | 99.16% | 99.32% |
| FN | RPL_UnderAttack | 0.00% | 0.00% | 0.00% |
| | CDRPL_UnderAttack | 1.25% | 1.41% | 1.48% |

example, an increase of more than 150% in the level of the consumed energy was experienced once the standard RPL became under attack. For CDRPL, the increase that resulted from a VN attack was up to 6% considering all the experimental setups. That is, CDRPL requires the nodes to be relatively less active during all the different scenarios hence consuming lower energy levels in any VN attack scenarios.

Table 7 presents the attack detection accuracy results which indicate the effectiveness of CDRPL in identifying VN attack attempts. CDRPL achieved a low FN rate and high TN rate in all the experimental setups. It was able to lower the FN rates to a level close to the zero rates achieved by the standard RPL. The highest FN rate experienced by CDRPL was in S3 and did not exceed 1.5%. On the other hand, CDRPL was able to achieve a rate of more than 99.3% in the accuracy of detecting malicious VN updates and successfully enabling RPL nodes to refrain from participating in malicious global repairs. Although it performs well in terms of the FN rate, the standard RPL failed in detecting VN attacks by more than 75%. In S2, the standard RPL was able to reach a TN rate of only 23% which was the best achievement in all the experimental setups. CDRPL provides significant improvements to the VN attack detection accuracy and makes RPL more accurate by more than 70%. Moreover, it can be noticed that CDRPL maintained accurate VN attack detection despite the scale and density of the network.

The results presented in Figure 11 indicate how frequently the nodes changed their preferred parents during the different scenarios in all the experimental setups. The results demonstrate the ability of CDRPL to effectively maintain route stability under VN attacks regardless of the scale of the network topology. That is, the PPC rate showed a slight increase when the CDRPL network became under attack. The highest increase in the rate was in the S3 setup showing an increase of less than one change/node on average.

In contrast, network stability was adversely affected by VN attacks in the case of the standard RPL. The PPC rate significantly increased by more than 4 changes/node on average considering all the setups. In S3, the attack caused the topology to be noticeably unstable with the increase in the rate reaching almost 9 changes/node on average. Compared to the standard RPL, a reduction of more than 80% in the PPC rate was achieved by CDRPL in the VN attack scenarios. Such an improvement in network stability helps RPL networks to preserve network lifetime and maintain QoS performance.

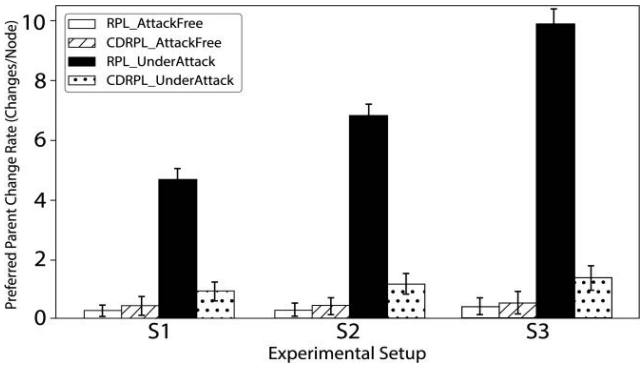


**FIGURE 11.** Preferred parent change (PPC) rate results.

**TABLE 8.** QoS performance results – S1 setup.

| Protocol | PDR | Throughput | Delay | ETX |
|---|---|---|---|---|
| RPL_AttackFree | 98.62 | 230.98 | 131.04 | 199.50 |
| CDRPL_AttackFree | 98.24 | 221.15 | 131.45 | 205.37 |
| RPL_UnderAttack | 87.07 | 200.26 | 217.62 | 269.42 |
| CDRPL_UnderAttack | 97.98 | 214.40 | 132.76 | 209.58 |

**TABLE 9.** QoS performance results – S2 setup.

| Protocol | PDR | Throughput | Delay | ETX |
|---|---|---|---|---|
| RPL_AttackFree | 94.28 | 341.75 | 136.35 | 248.05 |
| CDRPL_AttackFree | 94.07 | 333.42 | 137.16 | 263.43 |
| RPL_UnderAttack | 82.02 | 290.76 | 234.96 | 421.47 |
| CDRPL_UnderAttack | 93.68 | 322.50 | 139.13 | 272.80 |

**TABLE 10.** QoS performance results – S3 setup.

| Protocol | PDR | Throughput | Delay | ETX |
|---|---|---|---|---|
| RPL_AttackFree | 91.60 | 546.35 | 142.70 | 345.40 |
| CDRPL_AttackFree | 90.48 | 531.88 | 143.65 | 371.48 |
| RPL_UnderAttack | 74.03 | 439.07 | 263.36 | 598.62 |
| CDRPL_UnderAttack | 89.72 | 512.45 | 146.28 | 389.45 |

Upon the initiation of the VN attack, network overhead and topology stability were adversely affected as aforementioned in the case of standard RPL. This resulted in considerable degradation of the QoS performance of the under-attack RPL network as presented in Tables 8, 9, and 10. It can be seen that the standard RPL in VN attack scenarios experienced a high drop in the achieved PDR and throughput. This is more evident in relatively large-scale setups such as S3 in which drops of almost 20% in the PDR and throughput measurements were experienced. In the case of having CDRPL under attack, it showed no considerable reduction in such measurements. Rather, it was able to maintain the achieved QoS performance with a minimal drop of up to 4%.

**TABLE 11.** Multi-attack scenarios results.

| Protocol | Setup | PDR | Throughput | Delay | ETX | OH rate | EC | PPC Rate |
|---|---|---|---|---|---|---|---|---|
| RPL_UnderAttack | S1 | 83.1934 | 192.08571 | 226.8372 | 281.5271 | 126 | 6208 | 19 |
| | S2 | 76.9403 | 274.7394 | 248.4401 | 435.3469 | 422 | 9821 | 28 |
| | S3 | 66.7901 | 398.6093 | 282.0936 | 622.0745 | 742 | 17983 | 34 |
| CDRPL_UnderAttack | S1 | 97.0846 | 210.3028 | 134.0992 | 212.0358 | 28 | 2196 | 1.02 |
| | S2 | 92.8913 | 315.9002 | 142.5833 | 275.783 | 73 | 3589 | 1.3 |
| | S3 | 88.9016 | 501.2655 | 149.849 | 397.1163 | 129 | 6525 | 1.55 |

**TABLE 12.** Comparison of CDRPL with different VN attack countermeasures.

| Protocol | PDR | Control Message Overhead | Energy Consumption | Accuracy Rate | Multi-VN-Attack Support | Fast Detection Mechanism |
|---|---|---|---|---|---|---|
| RPL+Shield [24] | 96.24% | 5045 | 1314.9 | 96.7% | × | × |
| SRPL-RP [26] | 97.95% | 1095 | 1244.8 | 98.3% | × | × |
| VeNADet [28] | 95.47% | 2068 | - | 95.8% | × | × |
| LRPL [29] | - | 1337 | 1658 | - | × | × |
| CDRPL_UnderAttack | 97.98% | 950 | 1247.90 | 99.0% | √ | √ |

Moreover, the delay results indicate that the standard RPL experienced an increase of more than 66% once it became under VN attack in all the experimental setups. It also experienced an increase in the ETX measurements by up to 73% in the same scenarios. For CDRPL, the increase in the delay and ETX was less than 2% and 5%, respectively, in the VN attack scenarios. Compared to the standard RPL, CDRPL was able to improve the delay and ETX measurements in VN attack scenarios by more than 38% and 22%, respectively.

On the other hand, the results of the multi-attack scenarios are presented in Table 11. It can be seen that CDRPL still provides a promising performance irrespective of the number of VN attacks initiated across the network. While the standard RPL severely suffered the consequences of the composite attack, CDRPL succeeded in defending the multiple VN attacks and maintaining the overall performance of the network. Comparing the collected results of the single and composite attack scenarios, CDRPL was able to mitigate the multiple VN attacks while maintaining the QoS performance. Only a decrease of less than 2.2% in PDR and throughput in addition to an increase of less than 2.5% in delay and ETX were experienced considering all the experimental setups. This was not the case for the standard RPL which was noticeably affected by the additional attacker.

Considering network stability and overhead, an increase that reached 52% in the PPC rate and 21% in the number of control packets was experienced by the standard RPL after initiating the composite attack. For CDRPL, little increase in the PPC rate of less than 1 change/node on average was experienced considering all the setups. In addition, no more than 8 control packets per minute were added by CDRPL to the network overhead as a result of the multiple attacks. Another important consideration is the wasted energy after initiating an additional attack. This caused an increase of more than 10% in energy consumption considering the case of the standard RPL. However, CDRPL was able to improve that by only allowing an increase of less than 2.5% in the consumed energy considering all the setups. Accordingly, the ability of CDRPL in enhancing the security and performance of the original protocol specification is evident irrespective of the number of VN attacks being performed.

Table 12 provides a comparison of CDRPL with different VN attack countermeasures, namely RPL+Shield [24], SRPL-RP [26], VeNADet [28], and LRPL [29]. It is apparent that CDRPL provides a promising security solution for RPL networks against VN attacks. It effectively ensures competitive overall network performance, better detection accuracy, and improved functionality. CDRPL and SRPL-RP yielded almost similar PDR and energy consumption results which were better than that of RPL+Shield and VeNADet. However, such a high performance was achieved by CDRPL with the least control traffic compared to the other solutions. It reduced the control packets by more than 10%.

CDRPL also provided a fast mechanism for detecting VN attacks with improved detection accuracy. It relies on a simple attack detection approach whereas the other solutions incorporate detection mechanisms with high processing overhead. In addition, CDRPL provides an effective security solution for not only single VN attacks but also composite ones with multiple and concurrent attacks. Although SRPL-RP supports multi-attack mitigation, it only considers concurrent attacks of different types, particularly rank and VN attacks. While this is an important security consideration, CDRPL also ensures that critical security support is available to mitigate complex and massive VN attacks.

## VIII. CONCLUSION

VN attacks have an adverse impact on RPL networks, particularly in large-scale deployments and also under composite VN attack scenarios. To address such a security challenge in RPL networks, this paper presented CDRPL which introduces a simple modification to the RPL functionality. It incorporates a collaborative and distributed mitigation scheme against VN attacks. The extensive analysis and evaluation of CDRPL proved its high robustness and performance considering different VN attack scenarios in varying-scale setups. It guaranteed accurate and fast detection of simple and composite VN attacks while maintaining the stability and QoS performance of the network. Increasing the number of attacking nodes in the networks had no noticeable impact on the performance and accuracy of CDRPL. Compared to other similar approaches, CDRPL provides a simple and scalable security solution to RPL networks without adding much to the complexity of the RPL design. This would help in enriching the security of the protocol and reviving its potential for a wider scope of RPL-based IoT applications. Our plan for future work is to implement and experiment with CDRPL in a physical testbed with different types of IoT devices running real data traffic. Another direction that will be considered is the incorporation of other mitigation modules for addressing additional attacks. The target is to build a comprehensive and effective integrated RPL architecture against the different potential internal routing attacks. This also includes addressing effective mitigation of hybrid attacks combining VN attacks with other relevant internal routing attacks such as rank and replay attacks.

## REFERENCES

[1] L. Sujey. *Number of Internet of Things (IoT) Connected Devices Worldwide in 2018, 2025 and 2030*. Accessed: Nov. 25, 2021. [Online]. Available: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[2] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon, "The Internet of Things: Mapping the value beyond the hype," McKinsey Global Inst., USA, Tech. Rep., Jun. 2015.

[3] J. P. Vasseur, *Terms Used in Routing for Low-Power and Lossy Networks*, document IETF RFC 7102, Jan. 2014, doi: 10.17487/rfc7102.

[4] A. Verma and V. Ranga, "Security of RPL based 6LoWPAN networks in the Internet of Things: A review," *IEEE Sensors J.*, vol. 20, no. 11, pp. 5666–5690, Jun. 2020, doi: 10.1109/JSEN.2020.2973677.

[5] K. Hayashi. (2014). *IoT Worm Used to Mine Cryptocurrency*. Accessed: Nov. 25, 2021. [Online]. Available: https://www.symantec.com/connect/blogs/iot-worm-used-mine-cryptocurrency

[6] (Oct. 2016). *Heightened DDoS Threat Posed by Mirai and Other Botnets*. Accessed: Nov. 25, 2021. [Online]. Available: https://www.cisa.gov/uscert/ncas/alerts/TA16-288A

[7] O. Kupreev, E. Badovskaya, and A. Gutnikov. (2020). *DDoS attacks in Q1 2020*. Kaspersky. DDOS REPORTS. Accessed: Nov. 25, 2021. [Online]. Available: https://securelist.com/ddos-attacks-in-q1-2020/96837/

[8] (Feb. 24, 2019). *Symantec Corporation Internet Security Threat Report 2019*. Symantec. Accessed: Nov. 25, 2021. [Online]. Available: https://docs.broadcom.com/doc/istr-24-2019-en

[9] (2021). *Sonicwall Cyber Threat Report-Cyber Threat Intelligence for Navigating the New Business Reality*. Sonicwall. Accessed: Nov. 25, 2021. [Online]. Available: https://www.sonicwall.com/medialibrary/en/white-paper/2021-cyber-threat-report.pdf

[10] (2016). *Ericsson—The Connected Future*. Ericsson. Accessed: Nov. 26, 2021. [Online]. Available: https://www.ericsson.com/en/mobility-report/internetof-things-forecast

[11] S. Morgan. (Jun. 2019). *Global Cybersecurity Spending Predicted to Exceed $1 Trillion From 2017–2021*. Cybercrime Magazine. Accessed: Nov. 26, 2021. [Online]. Available: https://cybersecurityventures.com/cybersecurity-market-report/

[12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document IETF RFC 6550, Mar. 2012, doi: 10.17487/rfc6550.

[13] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, *A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)*, document IETF RFC 7416, Jan. 2015, doi: 10.17487/rfc7416.

[14] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, and G. Dini, "An implementation and evaluation of the security features of RPL," in *Proc. 16th Int. Conf. Ad-Hoc Netw. Wireless*, vol. 2017, pp. 63–76, doi: 10.1007/978-3-319-67910-5_6.

[15] A. Raoof, A. Matrawy, and C.-H. Lung, "Enhancing routing security in IoT: Performance evaluation of RPL's secure mode under attacks," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11536–11546, Dec. 2020, doi: 10.1109/JIOT.2020.3022276.

[16] A. Le, J. Loo, Y. Luo, and A. Lasebae, "The impacts of internal threats towards routing protocol for low power and lossy network performance," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2013, pp. 789–794, doi: 10.1109/ISCC.2013.6755045.

[17] A. Mayzaud, R. Badonnel, I. Chrisment, and I. G. Est-Nancy, "A taxonomy of attacks in RPL-based Internet of Things," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016, doi: 10.6633/IJNS.201605.18(3).07.

[18] K. Avila, D. Jabba, and J. Gomez, "Security aspects for RPL-based protocols: A systematic review in IoT," *Appl. Sci.*, vol. 10, no. 18, p. 6472, Sep. 2020, doi: 10.3390/app10186472.

[19] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "A study of RPL DODAG version attacks," in *Proc. 8th IFIP Int. Conf. Auton. Infrastruct., Manag. Secur.*, Brno, Czech Republic, Jun. 2014, doi: 10.1007/978-3-662-43862-6_12.

[20] A. Aris, S. F. Oktug, and S. Berna Ors Yalcin, "RPL version number attacks: In-depth study," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Istanbul, Turkey, Apr. 2016, pp. 776–779, doi: 10.1109/NOMS.2016.7502897.

[21] S. Sharma and V. K. Verma, "Security explorations for routing attacks in low power networks on Internet of Things," *J. Supercomput.*, vol. 77, no. 5, pp. 4778–4812, May 2021, doi: 10.1007/s11227-020-03471-z.

[22] A. Aris and S. F. Oktug, "Analysis of the RPL version number attack with multiple attackers," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Dublin, Ireland, Jun. 2020, pp. 1–8, doi: 10.1109/CyberSA49311.2020.9139695.

[23] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in RPL-based networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 472–486, Jun. 2017, doi: 10.1109/TNSM.2017.2705290.

[24] A. Arış, S. B. Ö. Yalçın, and S. F. Oktuğ, "New lightweight mitigation techniques for RPL version number attacks," *Ad Hoc Netw.*, vol. 85, pp. 81–91, Mar. 2019, doi: 10.1016/j.adhoc.2018.10.022.

[25] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, and L. T. Jung, "SMTrust: Proposing trust-based secure routing protocol for RPL attacks for IoT applications," in *Proc. Int. Conf. Comput. Intell. (ICCI)*, Bandar Seri Iskandar, Malaysia, Oct. 2020, pp. 305–310, doi: 10.1109/ICCI51257.2020.9247818.

[26] Z. A. Almusaylim, N. Jhanjhi, and A. Alhumam, "Detection and mitigation of RPL rank and version number attacks in the Internet of Things: SRPL-RP," *Sensors*, vol. 20, no. 21, p. 5997, Oct. 2020, doi: 10.3390/s20215997.

[27] F. Ahmed and Y.-B. Ko, "A distributed and cooperative verification mechanism to defend against DODAG version number attack in RPL," in *Proc. 6th Int. Joint Conf. Pervasive Embedded Comput. Commun. Syst.*, Lisbon, Portugal, 2016, pp. 55–62, doi: 10.5220/0005930000550062.

[28] A. A. Anitha and L. Arockiam, "VeNADet: Version number attack detection for RPL based Internet of Things," *Solid State Technol.*, vol. 64, no. 2, pp. 2225–2237, 2021.

[29] M. Belkheir, M. Rouissat, M. Achraf Boukhobza, A. Mokaddem, and M. Bouziani, "A new lightweight solution against the version number attack in RPL-based IoT networks," in *Proc. 7th Int. Conf. Image Signal Process. Their Appl. (ISPA)*, Mostaganem, Algeria, May 2022, pp. 1–6, doi: 10.1109/ISPA54004.2022.9786370.

[30] A. Dvir, T. Holczer, and L. Buttyan, "VeRA–version number and rank authentication in RPL," in *Proc. IEEE 8th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Valencia, Spain, Oct. 2011, pp. 709–714, doi: 10.1109/MASS.2011.76.
[31] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch, and T. C. Schmidt, "TRAIL: Topology authentication in RPL," in *Proc. Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*, Graz, Austria, Feb. 2016, pp. 59–64.
[32] M. Nikravan, A. Movaghar, and M. Hosseinzadeh, "A lightweight defense approach to mitigate version number and rank attacks in low-power and lossy networks," *Wireless Pers. Commun.*, vol. 99, no. 2, pp. 1035–1059, Mar. 2018, doi: 10.1007/s11277-017-5165-4.
[33] R. Sahay, G. Geethakumari, B. Mitra, and I. Sahoo, "Efficient framework for detection of version number attack in Internet of Things," in *Proc. 20th Int. Conf. Intell. Syst. Design Appl.*, Dec. 2020, pp. 480–492, doi: 10.1007/978-3-030-16660-1_47.
[34] R. Sahay, G. Geethakumari, and B. Mitra, "A novel blockchain based framework to secure IoT-LLNs against routing attacks," *Computing*, vol. 102, pp. 2445–2470, Nov. 2020, doi: 10.1007/s00607-020-00823-8.
[35] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, "ML-LGBM: A machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks," *IEEE Access*, vol. 9, pp. 83654–83665, 2021, doi: 10.1109/ACCESS.2021.3087175.
[36] N. Kushalnagar, G. Montenegro, J. Hui, and D. Culler, *Transmission of IPv6 Packets Over IEEE 802.15.4 Networks*, document IETF RFC 4944, Sep. 2007, doi: 10.17487/rfc2464.
[37] J. Hui and P. Thubert, *Compression Format for IPv6 Datagrams Over IEEE 802.15.4-Based Networks*, document IETF RFC 6282, Sep. 2011, doi: 10.17487/rfc6282.
[38] J. P. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*, document IETF RFC 6551, Mar. 2012, doi: 10.17487/rfc6551.
[39] P. Thubert, *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*, document IETF RFC 6552, Mar. 2012, doi: 10.17487/rfc6552.
[40] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, document IETF RFC 6719, Sep. 2012, doi: 10.17487/rfc6719.
[41] R. A. Koutsiamanis, G. Z. Papadopoulos, N. Montavont, and P. Thubert, "Common ancestor objective function and parent set DAG metric container extension," IETF Internet Draft, Tech. Rep. draft-ietf-roll-nsa-extension-09, Oct. 2020.
[42] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, *The Trickle Algorithm*, document IETF RFC 6206, Mar. 2011, doi: 10.17487/rfc6206.
[43] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020, doi: 10.1109/COMST.2019.2953364.
[44] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing attacks and mitigation methods for RPL-based Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1582–1606, 2nd Quart., 2019, doi: 10.1109/COMST.2018.2885894.
[45] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The impact of rank attack on network topology of routing protocol for low-power and lossy networks," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3685–3692, Oct. 2013, doi: 10.1109/JSEN.2013.2266399.
[46] R. Sahay, G. Geethakumari, and K. Modugu, "Attack graph—Based vulnerability assessment of rank property in RPL-6LOWPAN in IoT," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Singapore, Feb. 2018, pp. 308–313, doi: 10.1109/WF-IoT.2018.8355171.
[47] Zolertia. (Mar. 2010). *Z1 Datasheet*. Zolertia Advancare. Accessed: Nov. 26, 2021. [Online]. Available: http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf
[48] *Tmote Sky Datasheet*. Moteiv Corporation. Accessed: Sep. 3, 2022. [Online]. Available: https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf
[49] MICAz. *Wireless Measurement System Datasheet, Document Part Number: 6020-0060-04 Rev A*. Crossbow Technology. San Jose, CA, USA, Accessed: Sep. 3, 2022. [Online]. Available: http://courses.ece.ubc.ca/494/files/MICAz_Datasheet.pdf
[50] *TelosB Datasheet, Document Part Number: 6020-0094-01 Rev B*. Crossbow Technology, San Jose, CA, USA. Accessed: Sep. 3, 2022. [Online]. Available: https://www.willow.co.uk/TelosB_Datasheet.pdf
[51] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Tampa, FL, USA, Nov. 2004, pp. 455–462, doi: 10.1109/LCN.2004.38.

**IBRAHIM S. ALSUKAYTI** received the B.S. degree in computer science from Qassim University, Buraydah, Saudi Arabia, in 2006, the M.S. degree in computer and information networks from the University of Essex, Colchester, U.K., in 2010, and the Ph.D. degree in computer networks from Lancaster University, Lancaster, U.K., in 2014. He is currently an Associate Professor with the Computer Science Department, College of Computer, Qassim University. He is also the Director of a Research Group targeting Internet of Things technologies & applications. His research interests include network routing, wireless sensor networks, networking protocols, network security, and the IoT.

**AMAN SINGH** received the Ph.D. degree in computer science and engineering from Lovely Professional University, India. He is currently with the Universidad Europea del Atlántico, Spain. He is also the Scientific Advisor of the Faculty of Engineering, Universidade Internacional do Cuanza, Angola. His research interests include education 4.0, artificial intelligence, deep learning, machine learning, and mathematical modeling. He has published more than 60 refereed papers, including journals and international conferences. Most of his papers appeared in very selective and reputable conferences and journals, such as *IEEE Wireless Communication Magazine* and IEEE Internet of Things Journal. He has been invited to give keynote talks, lectures, and tutorials on artificial intelligence and mathematical modeling in international conferences and summer schools. He also has strong collaboration with industry, including projects, consultancy, and corporate social work. He has been the guest editor of several special issues for reputable international journals. He is a member of the editorial board of numerous international journals. He is also a Regular Reviewer for prominent journals in the field, including IEEE Transactions on Industrial Informatics, IEEE Journal of Biomedical and Health Informatics, *ACM Transactions on Sensors*, IEEE Internet of Things Journal, *AI*, *CMPB*, and *CBM*.