
EXPLORING PLACEMENT OF INTRUSION DETECTION SYSTEMS IN RPL-BASED INTERNET OF THINGS

Selim Yilmaz

WISE Lab., Department of Computer Engineering
Hacettepe University
Ankara

Department of Software Engineering
Muğla Sıtkı Koçman University
Muğla
selimyilmaz@mu.edu.tr

Emre Aydoğan

WISE Lab., Department of Computer Engineering
Hacettepe University
Ankara

emreaydogan@cs.hacettepe.edu.tr

Sevil Sen

WISE Lab., Department of Computer Engineering
Hacettepe University
Ankara

ssen@cs.hacettepe.edu.tr

ABSTRACT

Intrusion detection is an indispensable part of RPL security due to its nature opening to attacks from insider attackers. While there are a good deal of studies that analyze different types of attack and propose intrusion detection systems based on various techniques that are proposed in the literature, how to place such intrusion detection systems on RPL topology is not investigated. This is the main contribution of this study, and three intrusion detection architectures based on central and distributed placement of intrusion detection nodes are analyzed rigorously against different types of attacks and attackers at various locations in the RPL topology and evaluated from different aspects including their effectiveness, cost, and security.

Keywords IoT · security · RPL · intrusion detection

1 Introduction

Internet of Things becomes pervasive in many applications from industrial IoT to the Internet of medical things, the Internet of drones. While there are 13.1 billion IoT connected devices worldwide as in 2022, this number is predicted to reach up to 29.4 billion devices by 2030 [1]. The low power and lossy networks (LLN) are a special type of IoT that takes place in various application areas such as industry, smart homes. In such networks, resource constrained devices are connected over lossy links, which results in high packet loss. Therefore, new routing protocols are proposed for providing communication in such lossy networks in the literature. Among them, routing protocol for low power and lossy networks (RPL) has become the standard routing protocol for LLNs. However, RPL is still open to improvements in many research areas such as load balancing, mobility, security.

In the last years, a lot of approaches are proposed for securing RPL. While some specifications are defined for external attacks against RPL in its RFC document [2], it is still vulnerable to insider attacks. Therefore, researchers have been working on analyzing RPL-specific attacks and developing intrusion detection systems for RPL in order to respond to such attacks in a timely manner. However, the position of attackers and the placement of intrusion detection systems (IDSs) could dramatically affect the performance of the proposed IDSs. Therefore, in this study, the placement of IDSs is thoroughly explored for different types of attack carried out by attackers located at various locations.

In this study, three different intrusion detection (ID) architectures are analyzed. In *Central ID with Local information (CIDwL)*, IDS is placed at one node which relies on its local information in order to reach decisions. Many studies in the literature rely on a central IDS placed at the root node due to being a data collector point and a more powerful device in terms of computation capabilities. In the second architecture called *Central ID with Global information (CIDwG)*, the central IDS collects information from other nodes in order to have enough data for reaching to a decision. Lastly, *Distributed and Collaborative ID (DCID)*, in which every node has an IDS agents and decides upon collaboratively by using a voting mechanism, is analyzed. To the best of our knowledge, this is the first study that analyze the placement of intrusion detection system in RPL topology comprehensively from different aspects that are accuracy, cost, and security. A rigorous analysis is carried out for answering the following research questions:

RQ1: Is one central IDS enough for effectively detecting all types of attacks that are performed at different locations?

RQ2: What is the minimum number of IDS required for effective detection?

RQ3: How do IDSs make decision together efficiently from the communication cost perspective?

The results show that, depending on the attack type and attacker location, the position of intrusion detection is of a great impact for the CIDwL architecture. Moreover, one central ID node, for example that is placed in the root node is not very effective against attacks from various positions. Therefore, a distributed IDS is more suitable, especially where at least one ID is placed in each level, since each IDS could effectively detect attacks in its proximity. Moreover, if these ID nodes send their local features for global intrusion detection periodically rather sending their local alarms, a better accuracy is obtained. However, from the communication cost of view, a voting based intrusion detection could be preferred.

The paper is organized as follows. Section II covers background information including RPL and internal attacks against RPL. Section III discusses related studies in the field of intrusion detection on RPL. The ID architectures are introduced in detail in Section IV. Experimental settings and results are evaluated in Section V. Section VI discusses the results comparatively for each architecture and, the possible future studies. Finally, Section VII concludes the findings of this study.

2 Background

2.1 Protocol Overview

RPL is a distance vector routing protocol in which nodes communicate over a special topology called Destination-Oriented Directed Acyclic Graph (DODAG). The formation of DODAG is initiated by the root node only by broadcasting control packets called DODAG Information Object (DIO). The DIO package carries the information needed for the nodes to join the DODAG. Each node that receives the DIO packet adds the sender address in the incoming DIO packet to its parent list. In addition, each node calculates ‘rank’ value according to the Objective Function (OF) defined in DIO to determine its position in DODAG with respect to the root node. This guarantees the acyclic nature of the graph. After the DIO packet is updated with its own address and rank value, the node forwards it to its neighbors. In this way, an *upward route* is established from the leaf nodes to the root node.

A protocol-specific algorithm called ‘trickle timer’ controls the transmission interval of DIO packets. DIO packets are initially broadcast more often for the fast attachment of nodes to the graph, and the interval increases as long as the network is stable. A new node does not necessarily wait for a DIO packet to join DODAG; instead, it sends a control package called DODAG Information Solicitation (DIS) to its neighbors. The neighboring node sends the DIO packet immediately after taking the DIS packet. Therefore, the new node can join the DODAG. The *downward route* in RPL, however, is realized with control packages named Destination Advertisement Object (DAO). The downward routes are built in two modes: *storage mode* and *non-storing mode*. In the storage mode, each node keeps a routing table, and, instead of sending them towards the root node, it forwards packets to the next hop that routes to the destination address. In non-storage mode, however, the routing table is kept by the root node only, and the packets must be forwarded to the root node which operates the downward routing.

2.2 RPL Attacks

RPL control messages can be exploited to change the topology of a network. Furthermore, IoT nodes are more susceptible to various attacks due to mobility and their resource constraints. In the literature, RPL attacks are split into three categories [3]; attacks on topology, attacks on resources, and attacks on traffic. The first category consists of two subcategories; sub-optimization and isolation. The former creates non-optimal routes that cause poor performance in the network, and the latter deals with separation of a node or nodes from the network so that they cannot communicate with other nodes in the network. Attacks on resources cause an increase in computing, communication, energy usage

in nodes and can affect local or whole network. Attacks on traffic, however, aims to deceive the legitimate nodes by claiming other node's identity, as well as passive listening them by focusing on eavesdropping activities. In this study, we focus on the following seven RPL attacks belonging to different categories.

- *Decreased Rank (DR)*: The attacker nodes decrease the value of its rank and send it to other nodes in the network. Due to lower ranks denote higher position in DODAG tree and being closer to the root node, attacker nodes are preferred as parent nodes so that a large portion of traffic goes through these attacker nodes.
- *Increased Version (IV)*: Only the root node in DODAG tree is responsible for increasing the version number of DODAG tree and advertising version number throughout the network for global repair in RPL normally. However, attacker nodes illegitimately increase version number and propagate it, which may cause unnecessarily rebuilding of the networks.
- *Blackhole (BH)*: In this denial-of-service attack, instead of forwarding incoming packets to neighboring nodes, attacking nodes drop all packets. If it is combined with a sinkhole attack, attackers can damage the network in a way that a large portion of the network traffic is lost.
- *Selective Forwarding (SF)*: While blackhole attack drops all packets that are supposed to be forward, selective forwarding randomly selects some specific packets and drops them. Hence, packets belonging to a particular protocol can be filtered, and routing paths can be disrupted.
- *Worst Parent (WP)*: A node chooses a parent node when new incoming packets to be forwarded come according to the OF. In this attack, the attacker nodes exploit the objective function and force the victim node to choose the worst possible parent to forward packets. As a result, unoptimized paths are constructed, causing poor network performance.
- *DAG Inconsistency (DI)*: Downward route are built by DAO messages in RPL. If a child node cannot send packets to the destination due to unavailable downward routes which are built as a result of fake DAO messages, an inconsistency is occurred. When a node receives a packet with the Forwarding-Error 'F' flag set, this indicates that the packet cannot be sent by a child node and sent back to the parent node to choose a new neighbor node. As a result, node or nodes can be separated from network and unoptimized topology is emerged.
- *Hello Flood (HF)*: A node broadcasts an initial message when trying to join the network. It is called a "HELLO" message and is sent to the node's neighbors within its communication range. Attacker exploits this mechanism and regularly sends a good amount of unnecessary "HELLO" messages to its neighbors. The victim node replies with DIO messages to these "HELLO" messages and resets its trickle timer. This attack increases control packet overhead and node energy consumption.

3 Related Work

Early studies generally adopted centralized placement of IDSs where a node, usually a sink/root or LLN Border Router (LBR) node, is chosen as an ID node. In [4–6], trust-based systems in which the monitoring nodes passively collect and send information to the sink node, which is responsible for analyzing the incoming information. In addition, machine learning-based intelligent systems are also used [7–13]. K-means clustering and decision tree-based supervised learning approaches are employed in [7]. Five different machine learning algorithms that use features related to power consumption and network metrics are evaluated on RPL-based networks that use MRHOF and OF0 objective functions in [8]; whereas, 6LoWPAN compression header data (e.g., destination port, context identifier, etc.) are used by six different machine learning algorithms in [9]. In addition, four ensemble learning approaches against seven types of routing attacks are evaluated in [10]. A neural network-based secure system that instruments the source code of the application at compile time is proposed to detect illegitimate accesses to outbound memory in [11]. As a graph-based supervised learning approach, the optimum path forest algorithm is used in [12] to analyze traffic between LLN and the Internet. In [13], the genetic programming-driven transfer learning approach is proposed to develop an IDS that is resistant to change in attack and node types. In addition to machine learning-based approaches, statistical intrusion detection systems are used in [14, 15] to detect rank attacks by evaluating the energy states of the nodes. Knowledge-driven IDS is also proposed in [16] to select the optimal set of detection techniques by dynamically collecting knowledge from the network. Here, a wide variety of RPL attacks is considered and a subset of the detection techniques of these attacks is chosen.

Researchers have also proposed IDSs that rely on a distributed placement strategy in which dedicated nodes monitor the network to eliminate the concerns posed by centralized placement. The most satisfactory approach here is to place the IDS on all nodes, which is not applicable for LLNs however, as it brings about a dramatic burden on the network and the nodes. Therefore, the IDS system must consume less energy and storage in this strategy. Additionally, determining

the monitoring nodes in this architecture is another matter that should be investigated. Specification-based IDS systems relying on a finite-state machine that is implemented at each monitor node are proposed in [17, 18]. A trust-based RPL routing protocol is introduced in [19] for black hole attacks by calculating the trust value for each of the neighboring nodes of the monitoring nodes. An adaptive threshold-based solution is proposed in [20] to ignore erroneous header options, and hence to abolish the inconsistency in the DODAG tree. A mitigation scheme, named Secure-RPL, that is effective for both static and dynamic network is developed in [21]. In [22] they proposed that the system monitors the network traffic of 6LoWPAN through one or more IDS's operating in promiscuous mode. A similar approach to our study is proposed in [23] where the position of the monitoring nodes is investigated and an RPL-based distributed monitoring strategy is proposed. However, only the version number attack is investigated in [23], while in this current study seven different attacks are targeted. In addition, they assume that monitoring nodes have higher capacity which leads to a reduction in the load on the other nodes with constrained resources. This makes another major distinction on the communication strategy of monitoring nodes, where they communicate with the root node through a separate RPL instance in [23]. They choose monitoring nodes in which the communication ranges of all the monitoring nodes cover all the other nodes. While they only explore the use of the root node as the central IDS [23], the location of the central IDS is also explored in this current study. To sum up, different architectures are covered comprehensively with different attacks and attacker locations in the current study.

Hybrid approaches have also been studied to take advantage of both centralized and distributed placement strategies. SVELTE [24] is the first IDS proposed for LLNs that uses a centralized IDS at the root node and a distributed firewall at every node. An anomaly and specification agent-based IDSs located in, respectively, the root and router or leaf nodes are proposed in [25]. Here, the router and leaf nodes in LLN collect information, determine illegitimate nodes, and transmit the analysis results to the root node, which finally evaluates the anomalies in the network.

Only a few studies analyze the effects of attacks in RPL. Although comprehensive, they generally focus on a particular type of attack, such as version [26] [27], rank attacks [28]. Furthermore, they only analyze the effect of the location of the attackers [26–28] or the percentage of the attackers [29] on the network, not their detectability by considering their locations relative to the placed IDSs in the network. To the best of the authors' knowledge, this is the first study to analyze the placement of IDS in a DODAG topology, which is the main contribution of this study.

4 IDS Architectures

Intrusion detection systems developed for the IoT-based network are mainly categorized from different perspectives (e.g., placement strategy, detection method, security threats, and so forth) in the literature [30, 31]. Among them, placement is one of the most important factors that directly determines the efficacy of IDS. In this study, we take into account three IDS architectures that rely on centralized and distributed placement strategies, and we evaluate their advantages and disadvantages when they are subject to different routing attacks. These IDS architectures are briefly explained in the following sections, and evaluations regarding their performance are given in detail later.

4.1 Central ID with Local Information (CIDwL)

Here, the IDS is placed in a centralized manner, where a border router or a dedicated node is responsible for monitoring traffic and raising alarm. The main advantage of this architecture is that it does not bring about additional communication costs. Moreover, the lifetime of the network is not affected much since only a single node, in most cases a powerful device, is responsible for monitoring the traffic and extracting features. The downside, however, is the 'single point of failure' phenomena that occurs when the IDS node is down by an intruder, leading to a network that is open to harmful attacks. In addition, the IDS node reaches a decision based on its local data naturally collected.

4.2 Central ID with Global Information (CIDwG)

In this architecture, the IDS node is also placed in a central node. However, other nodes might participate in intrusion detection by forwarding some features extracted from their local traffic. Hence, the central IDS node, who makes the decision, has richer information about the network. However, the single point of failure is still a major handicap of this approach. In addition, this approach *i)* results in an increase in communication cost as the extracted local features are periodically sent to a central ID node, *ii)* reduces the average lifetime of the network, and finally *iii)* requires nodes to have larger memory size as the features are periodically extracted at the participating nodes.

4.3 Distributed and Collaborative ID (DICD)

Unlike previous architectures, more than a single ID node is dedicated such that they function independently. Therefore, each ID node individually performs *i*) traffic monitoring, *ii*) feature extraction, and *iii*) local alarm raising operations. The local alarms are then received by an analyzer module and a global alarm is issued depending on the local alarm rates. The main advantage of this architecture is that the network is still under protection unless all distributed IDs are down.

5 Comparative Analysis of Architectures

There are a number of factors that directly affect the performance of the secure systems developed towards IoT networks. These include the detection model, the adopted architecture, attack types, attacker location, and the like. Among them, no doubt, the IDS architecture plays a key role as it enables the ID model to function efficiently. Therefore, the evaluation of the optimal IDS architecture is one of the important research directions today. Here, we explore the role of three architectures from the point of view of intrusion detection capability, communication cost, and security.

5.1 Simulation Setting and Environment

The grid topology, shown in Figure 1, is used in the experiments. As seen here, we have deployed 30 nodes, including the root node in the simulations because at least 25 nodes are suggested in RPL-based networks to see the multihop characteristics of RPL [32].

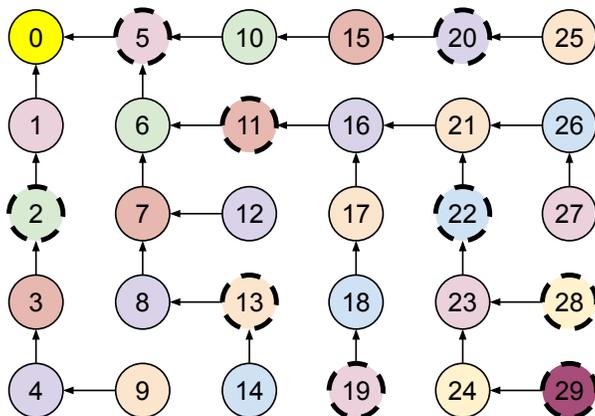


Figure 1: The network topology.

The nodes in the DODAG shown in the figure are leveled according to their ranks, and each level is represented with different face colors in the figure. As seen, there are 10 levels to involve 30 nodes in the DODAG. The nodes are 20 m away from each other, and the transmission range in the network is set to 25 m so that each node can communicate only its one-hop neighborhood in the DODAG. The arrows in Figure 1 represent the preferred parent of child nodes in DODAG under benign environment. The dashed borders in the figure, however, represents the attacker nodes. It is worth noting here that attacker nodes are arbitrarily chosen from every level in DODAG. Taking this DODAG into account, several network scenarios are generated separately for the three types of architecture. An exemplar scenario for the CIDwG and DCID architectures is separately given in Figure 2.

In order to set up the architectures and implement routing attacks, the Cooja simulator [33], a Java-based simulator that emulates sensor nodes running the Contiki operating system [34] (version 2.7), is used in the experiments. Cooja supports simulating wireless sensor networks involving different mote types, and here we adopted the Zolertia Z1 platform due to its larger memory capacity than the other platforms. This enables us to implement attacks on Contiki OS. The parameter values that we adopt in the simulations are listed in Table 1.

In the experiments, we employ a machine learning-based detection method in order to classify network traffic as ‘malicious’ or ‘benign’. Here, the Random Forest (RF) algorithm [35], one of the most popular algorithms applied in different machine learning tasks in the literature, is used for each architecture and scenario. We have used ‘scikit-learn’ [36] library for the implementation of the RF algorithm. The default parameters of the RF algorithm adopted in the library are used in the experiments, and the 10-fold cross-validation is applied. As for the input to train the RF

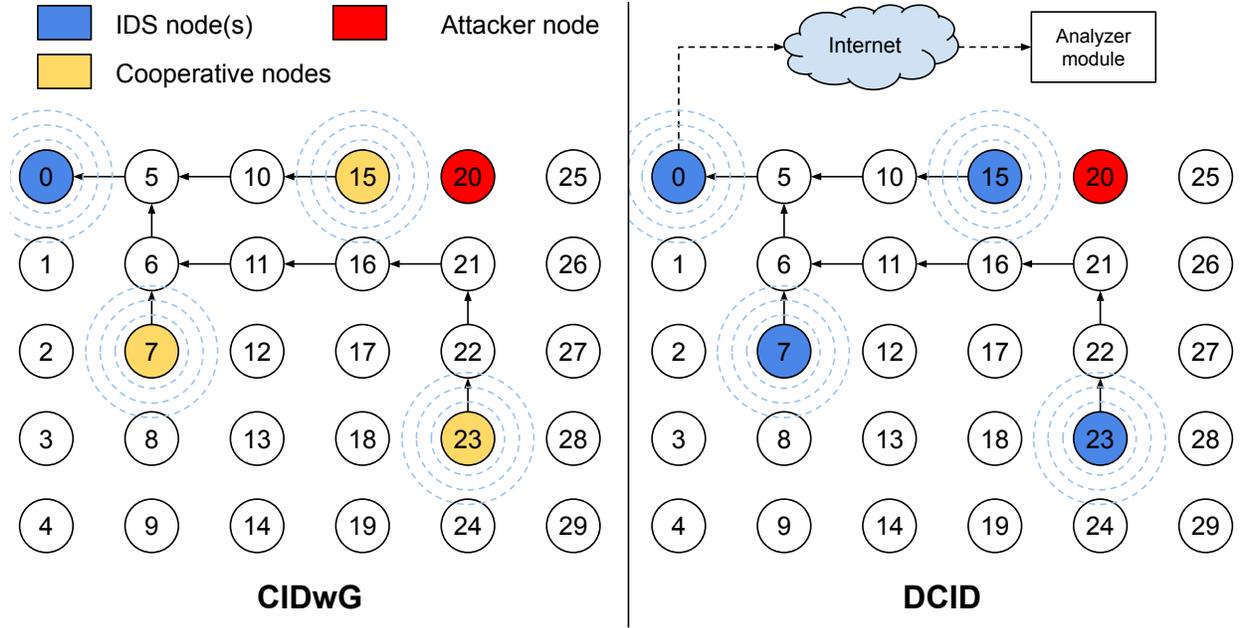


Figure 2: A scenario comparatively shown for CIDwG and DCID architectures.

Table 1: Simulation Parameters

Parameters	Values
Radio Environment	UDGM: Distance Loss
Objective Function	MRHOF-ETX
TX Range	25 m
Simulation Time	5 hours
Area of Deployment	100×80
Number of Sink Node	1 node
Number of Sensor Node	29 nodes
Traffic Pattern	UDP packets sent every 15 s

algorithm; the entire traffic flow is first split into 60 s time windows by the nodes in charge, and the features are then extracted from each windowed interval. Note that we have adopted 35 features that are proposed in [13]. The produced models are evaluated by the following metrics: accuracy, true positive rate (TPR), and false positive rate (FPR).

5.2 Evaluations of Architectures

We aim to evaluate three IDS architectures against seven routing attacks by seeking to answer the following three research questions (RQs). The *i*) main motivation, *ii*) the method adopted in the experiment, and finally *iii*) the findings are successively explained in detail for each RQ below. Note that the experiments are conducted on the basis of these RQs. In order for the experiments to be replicable by the researchers, the data used throughout the experiment are shared¹.

RQ1: Is one central IDS enough for effectively detecting all types of attacks that are performed at different locations?

Motivation: It is no doubt that RPL attacks have a varying effect on the traffic and on the nodes of the network. Depending on their characteristics as well as the position of the attacker node with respect to the root node and the ID node, these attacks can harm a very limited scale (e.g., DIS flooding) or, if not the entire, a very large portion of the network (e.g., increased version). Therefore, prior to the development of a detection system in a central IDS architecture, it is undeniably important to investigate the best, or at least close to the optimal, position of the ID node with respect to the position of the attacker node in the network and the attack type. Here, we evaluate the position of

¹<https://wise.cs.hacettepe.edu.tr/projects/IDArch/results.zip>

the ID node as a function of the distance from the root node and the attacker node for each routing attack. We believe that this attempt will give an important insight to network security practitioners on the optimal placement of IDS when developing a security solution in the central architecture.

Method: In order to reach this goal, we have conducted a massive experiment in which a great number of network scenarios are generated based on the CIDwL architecture. Here, we adopt a strategy to enroll a single node as the ID node and another node as the attacker node. As stated earlier, we used the grid topology shown in Figure 1 in the experiments. From this topology, we selected a node from each DODAG level and set them to the attacker node. Therefore, given in order of DODAG level, the nodes 5, 2, 11, 20, 13, 22, 19, 28, and 29 are taken as an alternative to attacker node (they are represented by the dashed line in the figure). As for the ID nodes, there are 30 nodes (from node 0 to node 29) that can be an alternative to the ID node. For each attack type, we take the combination of all the alternatives of the node selection cases to generate the scenarios in the experiment; therefore, 1,827 network scenarios are generated in total. It is worth stating here that the occurrences in the combination where the IDS and attacker nodes point to the same node are excluded.

Results: The detection accuracy is taken into account to evaluate the performance of IDS under the CIGwL architecture. These detection performances are individually shown for each attack with a heat map given in Figure 3. In the figure, the x- and y-axes represent the positions of the attacker and the ID nodes in the topology, respectively. Therefore, the upper-left and lower-right parts of the maps simply represent the regions in which both the attacker and the ID nodes are closest and farthest to the root node, respectively; the major diagonal part, however, represents that the ID node is only a few hops away from the attacker node (or vice versa). Beware that the detection performances of the scenarios where the position of ID node and attacker represents the same node (e.g., node 5) are evaluated as ‘0.0 accuracy’ in the heat maps because they are not studied. So, they can be ignored in the figure.

As it is clearly seen in Figure 3 that the performance of IDS on CIDwL architecture is highly dependent on the attack type, the distance to the root node, and also the distance to the attacker node. As for the evaluation on an attack basis; no matter wherever the ID or attacker nodes are positioned, the detection accuracy of IDS for the SF attack is very poor (max 74% accuracy observed); while it is remarkably high for the IV attack (min 99% accuracy observed). This proves that the ID model fails to discriminate malicious traffic from benign traffic by examining the feature set for SF attack. It is by no means surprising because the data packets may already be dropped in a typical wireless medium, hence features from lower layers might be needed for effective detection of SF [37]. On the contrary, the ID node can easily detect IV attack as the intruder globally alters the version number in DIO packet which makes it an easy-to-detect attack in this study, particularly with the version-related features (e.g., MAX_VERSION). This also emphasizes the contribution of our study compared to other studies analyzing on only one attack [23].

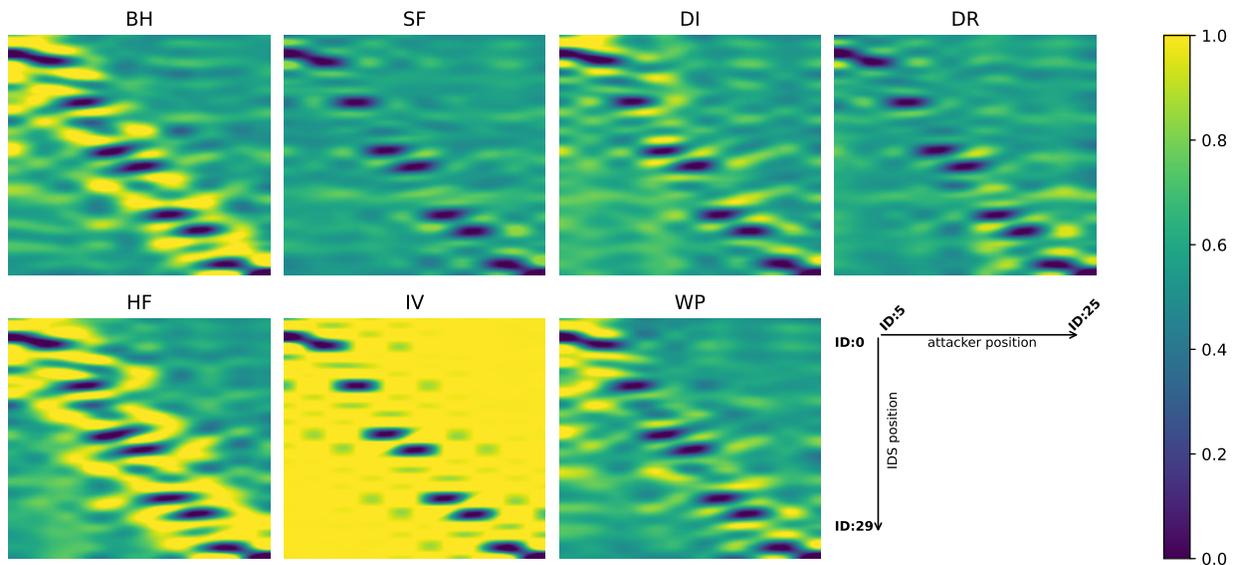


Figure 3: The detection accuracy performance of IDSs with respect to the attacker location in CIDwL architecture. The x- and y-axis represent, respectively, the location of attacker and IDSs nodes (given in order of distance from root to leaf).

The locations of nodes are important when evaluating for other types of attacks (i.e.; BH, HF, DI, WP, and DR). Actually, the detection performance increases dramatically when the ID node is positioned a few hops away from the attacker node for BH and HF attacks. Because the BH attack leads to a continuous drop of data packets sent by the child node, and the HF attack emerges overwhelming DIS packets towards the nodes that are neighbors of the attacker, they mainly harm the attacker’s locality in the network. That’s why, as the findings suggest, the ID nodes are better to be positioned near the attacker nodes for these two types of attacks. As for DI and WP attacks, IDS also shows a satisfactory detection performance when it is close not only to the attacker node but also to the root node. This is due to the nature of these attacks. Speaking concretely, the WP attacker intentionally selects its worst parent node, leading to data packets being forwarded through a suboptimal route. Because data traffic is concentrated at the root node, the WP attacker near the root node yields even a much higher deviation in terms of misrouted data traffic compared to the benign environment. Therefore, the ID node in the attacker’s locality can easily capture malignant traffic by analyzing the change in the number of data packets it forwards in malicious and benign environments. This is very alike to what happens with the DI attack where, due to the illegitimate setting of the flags, almost all data packets are dropped, especially when the attackers are near the root node. In contrast, the DR attack can be only detected effectively when the ID and attacker nodes are far away from the root node but close to each other. This is because of the objective of the attack, that is, misleading the benign nodes by illegitimately advertising them a lower rank value. The nodes near the root node have already lower rank values in DODAG, and the attackers are unable to generate a deviation compared to the benign environment when they are also near the root node. In such a circumstance, the ID node fails to discriminate whether the change in rank values is legitimate or not.

In addition to a general discussion of the CIDwL architecture, we also assess to which degree the attacker position can affect the detection accuracy of IDS under the central architecture. To do that, we consider the scenarios where the IDS is placed to the root node (node 0) because the plethora of the studies in the literature adopts the placement of IDS to the root node so that root node can analyze all the traffic exchanged between LLN and the Internet [30]. The accuracy performances reported separately for each attacker location are shown in Table 2. Note that, the best-performing accuracy values are highlighted with the gray color. The last column indicates the major difference in the accuracy performance. The results reveal that the degradation on the accuracy performance of IDS reaches from 13.17% (DR attack) to 49.50% (DI attack) for attacks where the optimal positioning of IDS is important (i.e., BH, DI, DR, HF, and WP). Therefore, it can be concluded that satisfactory detection performance with a single central ID can only be obtained when the position of the attacker is close to the IDS, maximum 2-levels away as shown in Table 2. In addition to the accuracy performance, Table 3 gives the best and worst TRP and FPR performances corresponding to the best and worst accuracy values in Table 2, respectively, to also reveal the performance difference (Δ) in terms of detection rate and false alarm rate. The results are highly correlated with those of Table 2, and the difference between TPR and FPR can reach 51% and 48.7%, respectively.

Table 2: Detection accuracy of the root node (ID:0) when attackers are at different levels (L).

Attack	Attacker IDs									Major Difference
	5 (L1)	2 (L2)	11 (L3)	20 (L4)	13 (L5)	22 (L6)	19 (L7)	28 (L8)	29 (L9)	
BH	0.988	0.832	0.533	0.575	0.530	0.535	0.540	0.515	0.562	0.473
SF	0.580	0.507	0.528	0.577	0.540	0.505	0.508	0.540	0.512	0.075
DI	0.967	0.992	0.557	0.687	0.520	0.553	0.570	0.497	0.503	0.495
DR	0.500	0.515	0.568	0.568	0.535	0.632	0.528	0.593	0.538	0.132
HF	1.000	1.000	0.802	0.667	0.582	0.560	0.592	0.517	0.553	0.483
IV	0.998	1.000	1.000	0.998	0.998	0.998	0.997	0.997	0.993	0.007
WP	0.927	0.805	0.517	0.503	0.510	0.575	0.522	0.537	0.493	0.433

Table 3: Best and worst TPR and FPR performances of the root node.

Attack	TPR			FPR		
	best	worst	Δ_{TPR}	best	worst	Δ_{FPR}
BH	0.983	0.523	0.460	0.007	0.493	0.487
SF	0.553	0.493	0.060	0.393	0.483	0.090
DI	0.997	0.487	0.510	0.013	0.493	0.480
DR	0.653	0.460	0.193	0.390	0.460	0.070
HF	1.000	0.500	0.500	0.000	0.467	0.467
IV	1.000	0.990	0.010	0.000	0.003	0.003
WP	0.943	0.467	0.477	0.090	0.480	0.390

RQ2: What is the minimum number of IDS required for effective detection?

Motivation: In the CIDwL architecture, only one node is running as a second defence of line. While this brings about some advantages such as lower communication cost, increased network lifetime, the performance of IDS depends on the type of attack and the attacker position, which is clearly shown by the findings in RQ1. Hence, this architecture is not scalable, and a distributed architecture seems more suitable for RPL-based IoT in order to obtain higher detection accuracy. For this purpose, here we consider CIDwG architecture where more than one node take part in intrusion detection. Unlike the CIDwL architecture, running this architecture increases *i)* average power consumption and *ii)* communication overhead, since participating ID nodes are responsible for monitoring and extracting the features and sending to a central ID node. Here, we mainly attempt to reveal how much the detection performance improves by including other nodes in intrusion detection. We also try to answer how many nodes are enough for effective detection and explore trade-offs between detection accuracy and cost (i.e., power consumption and communication cost).

Method: The main task is to generate scenarios that rely on CIDwG architecture using the network topology shown in Figure 1. Because it is not applicable to make all the nodes as participating nodes for a realistic scenario, we have chosen one node from each DODAG level as ID (except attackers as in RQ1). Therefore, the nodes 0, 1, 10, 15, 8, 25, 14, 23, and 24 (given in order of DODAG level) are selected randomly as the participating nodes in CIDwG. The combination of all these nodes (with lengths of two to nine) is generated to obtain participating ID node sets (i.e., {0, 1}, {0, 10}, ..., {0, 1, 10}, ..., {0, 1, 10, 15, 8, 25, 14, 23, 24}). By adopting this strategy, we have generated 31,626 scenarios in total for CIDwG architecture.

Results: Firstly, the overall accuracy performance of CIDwL and CIDwG as a function of the number of ID nodes are compared. To do that, we first obtained the accuracy performances of CIDwG architecture separately for different attacker locations, and then grouped them with respect to the number of ID nodes in the network. Then, we take the average of each of these groups to reveal the overall accuracy performance of the CIDwG architecture based on the number of participating nodes. The same procedure is also applied for CIDwL architecture except for the grouping as only one ID node is enrolled there. Please note that, in order to ensure a fair comparison, for the CIDwL architecture, only the accuracy performances of the ID nodes used in the CIDwG architecture (i.e., 0, 1, 10, 15, 8, 25, 14, 23, 24) are taken into account.

The overall performance of the CIDwL and CIDwG architectures is shown comparatively in Figure 4. Note that the line and bar plot shows the overall accuracy performances of CIDwL and CIDwG architectures, respectively. Therefore, it gives not only the comparative overall performances of the architectures, but also the change in the accuracy according to the number of ID nodes in CIDwG architecture. As seen here, the performance improves as the number of nodes increases for BH, SF, DI, DR, and HF attacks in CIDwG architecture, while a satisfactory performance is already obtained for IV attack with two ID nodes only. As for the comparison of the architectures, CIDwG architecture yields better performance than the CIDwL architecture for all types of attacks, where the accuracy difference of the best performances ranges from less than 1% (for IV attack) to 29.33% (for BH attack). The TPR and FPR performances of the CIDwG architecture with two and nine participating nodes that give the worst and the best accuracy performance, respectively are also analyzed in detail. These performances are given in Table 4. It is also seen here that attacks have a strong impact on the performance of TPR and FPR, and the difference in the performance of TPR and FPR reaches 21.0% (for BH attack) and 21.8% (for BH attack), respectively when the IDS is developed in CIDwG architecture with nine participating nodes. On the other hand, the participating nodes that give such an improvement (e.g., nine nodes are necessary to effectively detect the BH attack) impose an additional communication cost to the network. Therefore, the trade-off between accuracy, TPR, FPR, and network performance has to be balanced, which is discussed further in the next RQ.

Table 4: Comparative TPR and FPR values obtained with two and nine participating nodes in the CIDwG architecture.

Attack	with two nodes		with nine nodes	
	TPR	FPR	TPR	FPR
BH	0.783	0.222	0.994	0.003
SF	0.605	0.375	0.618	0.358
DI	0.694	0.292	0.744	0.249
DR	0.630	0.347	0.640	0.320
HF	0.831	0.165	1.000	0.003
IV	0.997	0.002	0.998	0.003
WP	0.706	0.292	0.704	0.304

Here, the best performance of each architecture is also given. To do that, separately for each attack type, we first report the maximum accuracy and its corresponding number of participating nodes obtained with respect to the attacker

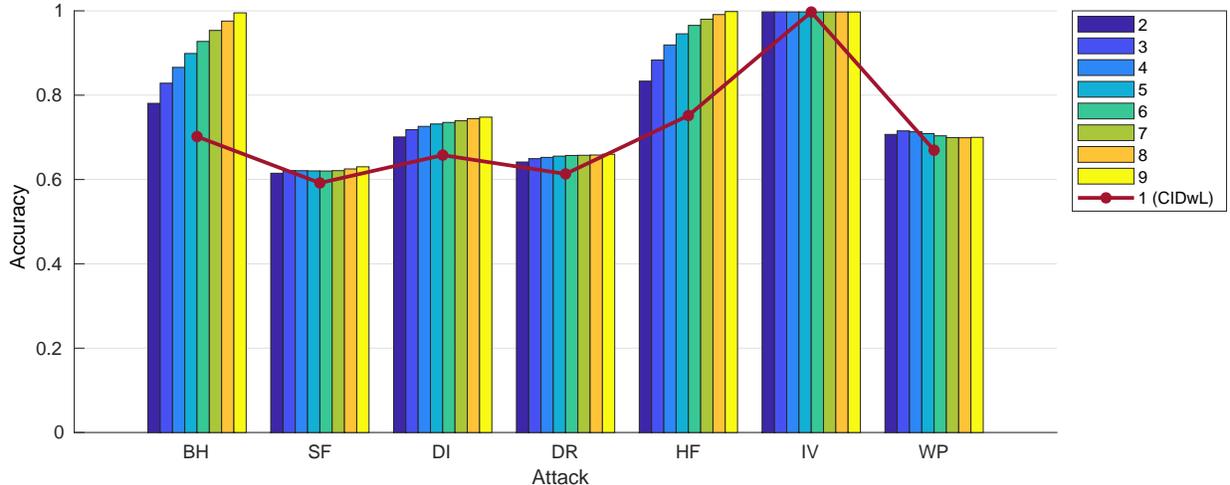


Figure 4: The comparison of the overall accuracy obtained by CIDwL and CIDwG architectures.

location. Then, the average of the best detection performances for all different attack positions in each attack type is taken for comparison of CIDwL and CIDwG. Again, for the CIDwL architecture, only the accuracy performances of the ID nodes used in the CIDwG architecture (i.e., 0, 1, 10, 15, 8, 25, 14, 23, 24) are taken into account.

The average of the best accuracy performances is shown in Table 5. The detailed best performances on the attacker location basis are also reported separately for each attack type in¹. Note that the best-performing architecture is represented with gray shades in the table. It can easily be seen from the table that very similar detection performance is observed for BH, HF, and IV attacks, while superior detection performance is observed when IDS is developed with the CIDwG architecture for SF, DI, DR, and WP attacks. The overall improvement on the detection accuracy with the CIDwG architecture ranges from less than 1% to around of 2.7% on overall. While the improvement is not dramatic, please note that the best performances are obtained with different placements of ID nodes each time. Hence, it is not applicable in real life. The best results also show that, depending on the attack type and attacker location, two to five ID nodes are enough to reach the best performances with CIDwG architecture.

Table 5: Average of the best detection accuracy performances for CIDwL and CIDwG.

Attack	Accuracy		Number of Nodes		
	CIDwL	CIDwG	max	min	median
BH	0.996	0.996	3	2	2
SF	0.697	0.715	4	2	3
DI	0.847	0.856	5	2	2
DR	0.718	0.744	5	2	4
HF	0.999	0.999	2	2	2
IV	0.999	1.000	3	2	2
WP	0.845	0.852	5	2	2

RQ3: How do IDSs make decision together efficiently from the communication cost perspective?

Motivation: We show that enrolling multiple nodes as part of IDS dramatically increases the detection accuracy on overall. However, as stated earlier, involving additional nodes get down the network’s performance from the point of communication cost. This is because LLNs are characterized by low data rates, limited frame size, and high packet loss [2]. When considering a scenario in which a number of participating nodes periodically transmit the local features through a sequence of fragmented frames, the effectiveness of the CIDwG architecture is highly questionable. Moreover, because these nodes have to transmit their local information within a much shorter interval to ensure ‘early’ detection capability of IDS, the applicability of this architecture on such a network should even be discussed. Therefore, in DCID architecture, the enrolled ID nodes are operating individually and, rather than the massive feature data, only alarms are transmitted periodically. The main goal here is to analyze if DCID architecture yields better than or at least similar detection performance to the CIDwG architecture does. Hence, the aforementioned communication cost, network lifetime problems accelerated by CIDwG architecture could be overcome. The major question to be addressed in DCID architecture is to find optimal voting ratio to raise a global alarm in the network.

Method: As in RQ2, we here have created a large number of scenarios to implement DCID architecture on the network topology given in Figure 1. The same ID nodes as in RQ2 are considered for a fair comparison; and therefore, the combination of these nodes (with a length of two to nine) is generated to obtain distributed node sets. Contrary to the CIDwG architecture, a voting scheme has to be tuned in DCID architecture. It is simply evaluated that of all local alarms arising from the distributed ID nodes, how many are necessary to raise a global alarm. Here, DCID architecture is evaluated with two voting schemes for each attack type: *i*) minority voting and *ii*) majority voting. In minority voting, a global alarm is issued, provided that at least a local alarm is sent by one of the ID nodes. In the majority voting, however, the ratio of local alarms ($R | R \in \{50, 60, 70, 80\}$) matters, and a global alarm is issued as long as $R\%$ of the ID nodes arises local alarms. It is worth stating here that because the lower or greater R values (i.e., $R < 50$ or $R > 80$) lead to a performance degradation, we here disregard studying them further. As in RQ2, we have generated 31,626 scenarios for each voting scheme in the DCID architecture.

Results: Because we mainly aim to investigate if the DCID architecture can be a good alternative to the CIDwG architecture, here we compare them with each other in terms of overall accuracy performance. In order to accomplish that, we first find out the best voting scheme that is to be adopted for DCID architecture. To do that, we collect the accuracy results obtained by different voting schemes for different attacks and then calculate the average of these accuracy values. The overall comparative performance of the voting schemes in the DCID architecture is outlined in Table 6, and again the best performances are highlighted in gray. From the comparative results, it is clear that minority and 80% voting schemes have shown the poorest accuracy, whereas 50% voting scheme (followed by 60% voting scheme) has shown the highest accuracy. Therefore, we adopted the 50% voting scheme for the DCID architecture to compare its overall accuracy performance with that of the CIDwG architecture.

Table 6: Comparison of overall accuracy performances of voting schemes in DCID architectures.

Attack	DCID				
	minority	50%	60%	70%	80%
BH	0.614	0.805	0.798	0.734	0.681
SF	0.548	0.648	0.642	0.604	0.577
DI	0.592	0.737	0.727	0.677	0.638
DR	0.560	0.677	0.671	0.630	0.599
HF	0.653	0.866	0.857	0.787	0.728
IV	0.996	0.998	0.998	0.998	0.997
WP	0.597	0.754	0.748	0.695	0.652

In order to compare the CIDwG and DCID architectures (with 50% voting scheme) as a function of the number of distributed nodes, we first collect the accuracy performances obtained with two to nine distributed nodes and then calculate the average of these performances. The overall accuracy performance of the DCID architecture is shown in Figure 5. Note that the results for other schemes can also be found in¹. It is worth stating that the corresponding accuracy values obtained by the CIDwG architecture are reported in Figure 4, and are also indicated in this figure by horizontal red markers for the sake of a clear performance comparison of the two architectures.

From the results, it can be seen that a better detection accuracy can be reached with DCID architecture, as more distributed nodes are participating for all attack types except IV attack, where a satisfactory performance can already be obtained with even two IDs. It is important to note that the participation of more nodes also increases the reliability of the DCID architecture. This is because even if the intruder targets and prohibits some of these nodes from operating, the IDS can still function with a detection performance that may degrade to some extent. On the contrary, the major downside of this setting is the reduced average lifetime of the network, as resource-constrained devices periodically extract features. Therefore, a trade-off exists for the DCID architecture that should be properly balanced taking into account the detection performance, reliability of the IDS, and constrained nature of the devices. When it comes to the comparison of the performances of the architectures, the findings here suggest that, no matter how many nodes are participating, CIDwG outperforms DCID for BH and HF attacks, and a slight performance improvement is also observed when CIDwG architecture with two participating nodes; whereas DCID shows better detection performance for SF, DI, DR, and WP attacks with three or more participating nodes.

The TPR and FPR performances of the DCID architecture with two and nine participating nodes are also evaluated to reveal the change in the performances with these two settings. The results are comparatively given in Table 7 where the best performances are again highlighted in gray. The results here suggest that DCID architecture with two participating nodes yields better performance than that with nine participating nodes in terms of TPR for all types of attack except HF. The difference here ranges from less than 1% (for the IV attack) to 14.7% (for the SF attack). Regarding FPR performance, a dramatic improvement is observed for all types of attack when nine nodes participate. The difference here ranges from less than 1% (for the IV attack) to 37.8% (for the BH attack). These findings imply that a much lower

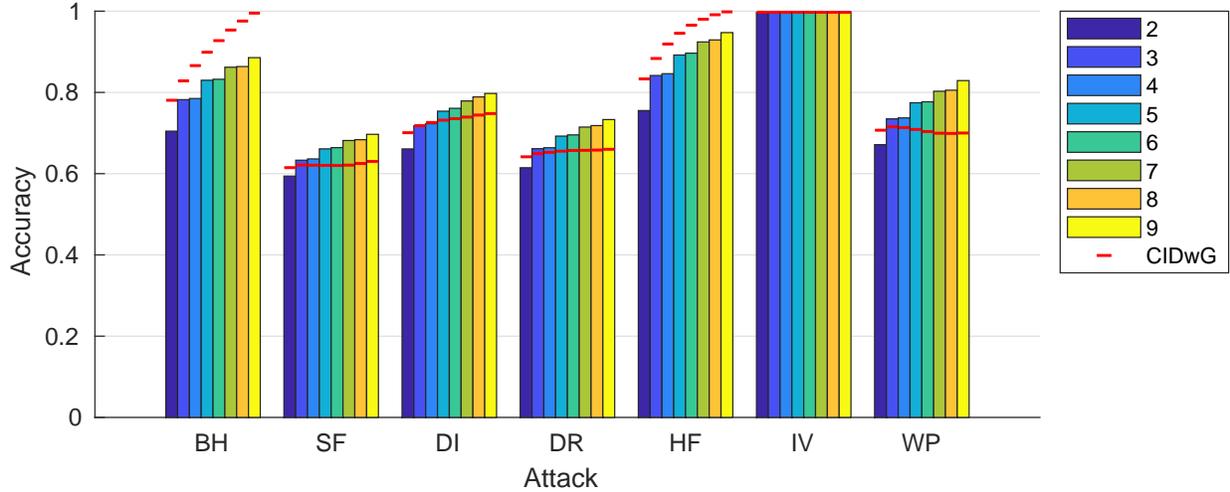


Figure 5: The comparison of the overall accuracy as a function of varying number of nodes in DCID architecture (50% voting scheme).

false alarm rate for benign traffic flows is guaranteed at the cost of a reduced detection rate for malicious flows as more nodes participate in the architecture, resulting in higher accuracy performance overall.

Table 7: Comparative TPR and FPR values obtained with two and nine participating nodes in the DCID architecture.

Attack	with two nodes		with nine nodes	
	TPR	FPR	TPR	FPR
BH	0.908	0.499	0.892	0.121
SF	0.821	0.633	0.674	0.281
DI	0.864	0.542	0.774	0.180
DR	0.831	0.602	0.703	0.237
HF	0.932	0.421	0.940	0.045
IV	0.998	0.004	0.997	0.000
WP	0.876	0.534	0.820	0.161

6 Discussion

In this rigorous analysis, seven types of attacks are performed at different locations and their detectability by ID nodes placed at different locations and work independently or collaboratively are explored.

Effectiveness & Cost: The results clearly show that a central node is not enough to detect attacks at different locations. Hence, a distributed and cooperative architecture is more appropriate for RPL topologies. As the results suggest, at least one ID node could be placed at each level for effective detection. This is also important for some types of attack, such as WP that have a higher impact locally. Although such architectures increase communication cost, it is worth placing ID nodes at each level for reasonable accuracy and scalability. Moreover, a voting scheme could be preferred over sending features periodically for a lower communication cost, which also overcomes fragmentation issues in LLN. This architecture provides high accuracy for some types of attack (BH, HF, IV). In the future, the features collected from the lower layer could be included for a better detection of other types of attack, namely SF and DR as in [37].

Threats to validity: As shown in the results, attackers' locations have a clear impact on their detectability. Hence, if attackers know the placement of IDs, they could evade detection by performing their attacks far from ID nodes. Therefore, placing an ID node at each level becomes more important, as suggested in the findings of the current study.

A central ID node is one of the main drawbacks of all architectures. For such single point of failure points, there should always be another backup node that could work in case of a failure in the main node. Another solution could be to place the central ID in a cloud, outside of the network. Another option is to collect alarms in each ID node in DCID. Even though this increases communication cost, it is fairly acceptable for such short alarm messages.

Other factors: In the future, adversarial attacks against ID nodes could be explored thoroughly. Another area that needs investigation is the effect of mobility on intrusion detection. On the one hand, a mobile attacker could evade intrusion detection. On the other hand, its mobility might limit its impact in the network. In our ongoing study, we are analyzing the effect of mobility from various aspects in RPL security.

7 Conclusion

While many of the advancements in RPL security are in the area of attacks' analysis and developing methods for detection of these attacks, the placement of intrusion detection nodes plays a vital role, since attacks could come from different locations. In this study, this research problem is explored with extensive simulations, and three intrusion detection architectures are discussed with different aspects that are performance, communication cost, and security. We believe and hope that this study could give guidelines for researchers that work on developing suitable (e.g., lightweight with low communication cost) intrusion detection techniques for RPL.

Acknowledgment

This study is funded by the Scientific and Technological Research Council of Turkey (TUBITAK-122E331). We would like to thank TUBITAK for its support.

References

- [1] Statista. Number of internet of things (iot) connected devices worldwide from 2019 to 2030. [accessed 19-August-2022].
- [2] Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, March 2012.
- [3] Anth ea Mayzaud, Remi Badonnel, and Isabelle Chrisment. A taxonomy of attacks in rpl-based internet of things. *I. J. Network Security*, 18:459–473, 2016.
- [4] Zeeshan Ali Khan and Peter Herrmann. A trust based distributed intrusion detection mechanism for internet of things. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, pages 1169–1176, 2017.
- [5] Faiza Medjek, Djamel Tandjaoui, Imed Romdhani, and Nabil Djedjig. A trust-based intrusion detection system for mobile rpl based networks. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 735–742, 2017.
- [6] Ashwini Nikam and Dayan Ambawade. Opinion metric based intrusion detection mechanism for rpl protocol in iot. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–6, 2018.
- [7] Prachi Shukla. MI-ids: A machine learning approach to detect wormhole attacks in internet of things. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 234–240, 2017.
- [8] John Foley, Naghmeh Moradpoor, and Henry Ochenyi. Employing a machine learning approach to detect combined internet of things attacks against two objective functions using a novel dataset. *Security and Communication Networks*, 2020:2804291, Feb 2020.
- [9] Mohamad Nazrin Napiyah, Mohd Yamani Idna Bin Idris, Roziana Ramli, and Ismail Ahmedy. Compression header analyzer intrusion detection system (cha - ids) for 6lowpan communication protocol. *IEEE Access*, 6:16623–16638, 2018.
- [10] Abhishek Verma and Virender Ranga. Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–6, 2019.
- [11] Ahmed Saeed, Ali Ahmadinia, Abbas Javed, and Hadi Larjani. Intelligent intrusion detection in low-power iots. *ACM Trans. Internet Technol.*, 16(4), dec 2016.
- [12] Mansour Sheikhan and Hamid Bostani. A security mechanism for detecting intrusions in internet of things using selected features based on mi-bgsa. *International Journal of Information & Communication Technology Research*, 9:53–62, 03 2017.

- [13] Selim Yilmaz, Emre Aydogan, and Sevil Sen. A transfer learning approach for securing resource-constrained iot devices. IEEE Transactions on Information Forensics and Security, 16:4405–4418, 2021.
- [14] Usman Shafique, Abid Khan, Abdur Rehman, Faisal Bashir, and Masoom Alam. Detection of rank attack in routing protocol for low power and lossy networks. Annals of Telecommunications, 73(7):429–438, Aug 2018.
- [15] R Stephen and L Arockiam. E2v: Techniques for detecting and mitigating rank inconsistency attack (RInA) in RPL based internet of things. Journal of Physics: Conference Series, 1142:012009, nov 2018.
- [16] Daniele Midi, Antonino Rullo, Anand Mudgerikar, and Elisa Bertino. Kalis - a system fmi-bgsaor knowledge-driven adaptable intrusion detection for the internet of things. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pages 656–666, 2017.
- [17] Anhtuan Le, Jonathan Loo, Yuan Luo, and Aboubaker Lasebae. Specification-based ids for securing rpl from topology attacks. In 2011 IFIP Wireless Days (WD), pages 1–3, 2011.
- [18] Anhtuan Le, Jonathan Loo, Kok Keong Chai, and Mahdi Aiash. A specification-based ids for detecting attacks on rpl-based network topology. Information, 7(2), 2016.
- [19] David Airehrour, Jairo Gutierrez, and Sayan Kumar Ray. Securing rpl routing protocol from blackhole attacks using a trust-based mechanism. In 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), pages 115–120, 2016.
- [20] Anuj Sehgal, Anth a Mayzaud, R mi Badonnel, Isabelle Chrisment, and J rgen Sch nw lder. Addressing dodag inconsistency attacks in rpl networks. In 2014 Global Information Infrastructure and Networking Symposium (GIIS), pages 1–8, 2014.
- [21] Abhishek Verma and Virender Ranga. Mitigation of dis flooding attacks in rpl-based 6lowpan networks. Transactions on Emerging Telecommunications Technologies, 31(2):e3802, 2020. e3802 ett.3802.
- [22] Prabhakaran Kasinathan, Claudio Pastrone, Maurizio A. Spirito, and Mark Vinkovits. Denial-of-service detection in 6lowpan based internet of things. 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 600–607, 2013.
- [23] Anth a Mayzaud, R mi Badonnel, and Isabelle Chrisment. A distributed monitoring strategy for detecting version number attacks in rpl-based networks. IEEE Transactions on Network and Service Management, 14(2):472–486, 2017.
- [24] Shahid Raza, Linus Wallgren, and Thiemo Voigt. Svelte: Real-time intrusion detection in the internet of things. Ad Hoc Networks, 11(8):2661 – 2674, 2013.
- [25] Hamid Bostani and Mansour Sheikhan. Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach. Computer Communications, 98:52–71, 2017.
- [26] Anth a Mayzaud, Anuj Sehgal, R mi Badonnel, Isabelle Chrisment, and J rgen Sch nw lder. A study of rpl dodag version attacks. In IFIP international conference on autonomous infrastructure, management and security, pages 92–104. Springer, 2014.
- [27] Ahmet Aris, Sema F Oktug, and S Berna Ors Yalcin. Rpl version number attacks: In-depth study. In NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, pages 776–779. IEEE, 2016.
- [28] Anhtuan Le, Jonathan Loo, Aboubaker Lasebae, Alexey Vinel, Yue Chen, and Michael Chai. The impact of rank attack on network topology of routing protocol for low-power and lossy networks. IEEE Sensors Journal, 13(10):3685–3692, 2013.
- [29] Cansu Dogan, Selim Yilmaz, and Sevil Sen. Analysis of rpl objective functions with security perspective. In SENSORNETS, pages 71–80, 2022.
- [30] Bruno Bogaz Zarpel o, Rodrigo Sanches Miani, Cl udio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things. Journal of Network and Computer Applications, 84:25–37, 2017.
- [31] Ali Seyfollahi and Ali Ghaffari. A review of intrusion detection systems in rpl routing protocol based on machine learning for internet of things applications. Wireless Communications and Mobile Computing, 2021:8414503, Aug 2021.
- [32] Hyung-Sin Kim, Jeonggil Ko, David E Culler, and Jeongyeup Paek. Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): A survey. IEEE Communications Surveys & Tutorials, 19(4):2502–2525, 2017.
- [33] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In Proceedings. 2006 31st IEEE Conference on Local Computer Networks, pages 641–648, 2006.

- [34] Contiki-Ng. [contiki-ng/contiki-ng](http://contiki-ng.org), 2004. [accessed 13-July-2021].
- [35] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [37] Erdem Canbalaban and Sevil Sen. A cross-layer intrusion detection system for rpl-based internet of things. In *International Conference on Ad-Hoc Networks and Wireless*, pages 214–227. Springer, 2020.