Intrusion Detection Systems for RPL Security: A Comparative Analysis

George Simoglou, George Violettas, Sophia Petridou[1], Lefteris Mamatas

Email: (it1492, georgevio, spetrido, emamatas)@uom.edu.gr

University of Macedonia

Egnatia 156, Thessaloniki, Greece

Abstract

Internet of Things (IoT) is an emerging technology that has seen remarkable blossom over the last years. The growing interest for IPv6 constrained networks has made the Routing Protocol for Low Power and Lossy Networks (RPL) the standard routing solution, which has gained significant attention and maturity in the literature. However, due to the networks' open and possibly unattended environment of operation, as well as to the nodes' constraints, the security of the protocol is a challenging issue, currently under thorough investigation. New and innovative Intrusion Detection Systems (IDSs) have been proposed in the literature over the last years to address the protocol's security issues. In that regard, our survey paper: i) begins with extracting a set of design requirements for RPL-related IDSs based on discussing the diversity of attacks on the protocol and investigating their impact; ii) continues with identifying best practices and gaps in an IDS design which are derived by studying the evolution of the related bibliography ($2013 - 2020$); and iii) concludes with a number of guidelines extracted once we map the 22 IDSs under study to the attacks they encounter and compare them in line with the design requirements we introduce. Our analysis considers feedback from the corresponding authors for a deeper investigation.

Keywords: IoT, RPL Routing Protocol, Security, Attacks, Intrusion Detection Systems, Comparative Analysis

2020 MSC: 00-01, 99-00

1. Introduction

The Internet of Things (IoT) is a broad field of technology and research, part of which is comprised of Low-power and Lossy Networks (LLNs). The nodes of such networks are susceptible to various restrictions and challenges, rendering the existing routing protocols inappropriate. The gap was filled by the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), which has become the de facto standard for IoT routing, beyond initial expectations [1, 2]. RPL has been proven significantly mature to connect IPv6 devices, with moderate control overhead and under challenging conditions, e.g., lossy links, heterogeneous and constraint devices, newfangled threats [3, 4].

Despite its advantages, RPL still has open issues, the most important of which are related to attacks that disrupt the IoT networks' operation [5]. In fact, RPL is unavoidably exposed to a large number of attacks since it is based on the IPv6 open stack and uses mostly wireless media for the nodes' communication. In addition, by exploiting RPL's mechanisms, an intruder can gain access to the network and unleash attacks that originate from within the LLN. In such cases, encryption itself does not suffice to provide security [6]. On this front, the RPL standard specifies three modes of operation, i.e., unsecured mode, preinstalled mode, and authenticated mode [1], while it also defines mechanisms for data confidentially, data authenticity, and replay protection [7].

Although some recent research efforts focus on a partial implementation of RPL's security features [7, 8], up to this time, the majority of RPL implementations assume the unsecured mode of operation. Actually, the RPL security features are characterized as optional [1] and, according to [9, 10], future versions of RPL will address issues such as authenticated security.

Until then, the most realistic approach to deal with attacks is the Mitigation Methods and the Intrusion Detection Systems (IDSs). The former regard lightweight supplementary mechanisms to the standard RPL and deal

with a limited number of attacks. The latter employ a combination of methods, allowing for a broader spectrum of attacks' treatment. Currently, a small number of surveys focus on the RPL aforementioned security issues and the IDSs confronting them. Mayzaud et al. [5] present a definite categorization of RPL attacks, where the IDSs are solely discussed in line with them, while a detailed taxonomy and evaluation of the attacks are missing. Furthermore, [5] includes only three of the new IDSs, available at the time of publication. Raoof et al. [11] discuss RPL attacks and their mitigation methods in general, leaving limited space for description and analysis of specific IDSs; only a list of those considered most influential by the authors are shortly described. In the recent work of Verma et al. [6], the authors also utilize the taxonomy of attacks from Mayzaud et al. [5], and they propose a comparison chart of the contemporary IDSs based on an extensive set of 26 categorization criteria. Despite being a detailed mapping with some potential of providing future insights, at this time, their comparison table is empty up to 92 percent, and, thus, it remains incomprehensible.

The above fact indicates that selecting criteria for analysis is a challenging issue since they should be primarily meant for the context they are proposed, and, secondly, they should facilitate the direct comparison of the subjects (the IDSs in our case) under investigation. To our mind, this can be achieved by a core of narrow and well-thought criteria.

In this context, this survey implements a coherent investigation of RPL-related IDSs according to a novel conceptual framework that defines a three-step methodology. It starts by investigating the diversity and impact of well-known attacks to define essential design requirements for IDSs, based on both a literature review and illustrative simulations. The next step identifies best practices & gaps by studying the evolution of related IDS proposals. The last step involves mapping 22 selected IDSs to the attacks they encounter, while contrasting them in respect to the introduced requirements as comparison criteria. Our analysis concludes with essential design guidelines for future up-to-date IDSs.

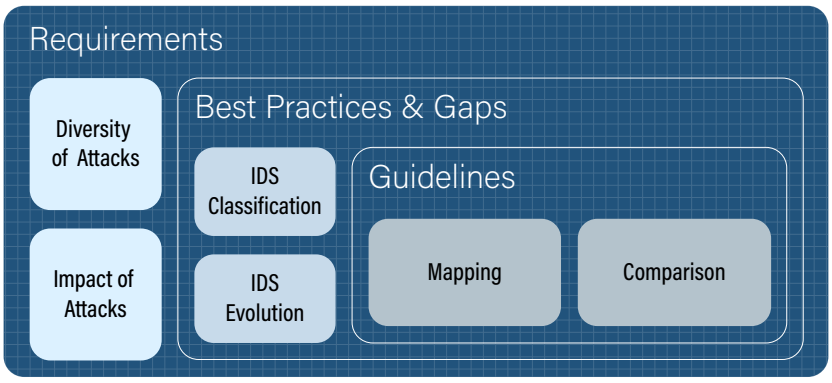

Figure 1: Conceptual framework of the analysis: an abstract representation.

The remainder of this survey is organized as follows: Section 2 presents our conceptual framework that highlights our methodological approach. Section 3 gives a brief overview of the RPL protocol, while Section 4 discusses the RPL-related attacks and their impact to conclude to a set of IDSs' design requirements. Section 5 elaborates on the RPL-related IDSs, providing a classification of them, discussing the evolution of the most recently proposed systems, and highlighting best practices and gaps in the literature. Section 6 summarizes our comparative analysis and compacts our investigation into four guidelines for future systems. Finally, Section 7 concludes this survey.

2. Conceptual Framework & Methodology

This survey adheres to a novel conceptual framework, shown in Fig. 1, that provides the methodological basis of our investigation. It consists of three methodological steps, defined below.

The first one concerns the *requirements' definition* that a successful IDS should address. Our starting point is a better understanding of the problem IDSs tackle, i.e., the mitigation of attacks. For example, Wallgren *et al.* [12] identify the diversity of attacks as the main cause for attack detection accuracy issues in existing IDSs. Other papers, including surveys [5, 11] and IDS proposals [12, 13, 14, 15], do typically base their analysis on identifying the considered

attacks' impact, e.g., increased control overhead or decreased packet delivery ratio (PDR). For completeness, we conduct a literature-based investigation of well-known RPL attacks from a new perspective: a combined study on attacks' diversity and impact.

More precisely, we elaborate on the RPL-related attacks, spanning from *resource depletion* attacks, that shorten the network's lifespan, to *network topology* attacks, that degrade the paths created by RPL or isolate a subset of network's nodes, and *network traffic* attacks, that allow the analysis of packets in order to gain knowledge about the network. Several of them may not be harmful as standalone events. Still, they can be critically detrimental to the network (e.g., control overhead) or the applications (e.g., PDR) in conjunction with others. In this first step, we also provide illustrative simulation results, highlighting the primary outcomes of our combined investigation of attacks' diversity and impact. As an outcome, we define a set of seven design requirements for an RPL-related IDS that are directly connected with the protocol's standard.

Our next step identifies the *best practices & gaps* out of an extensive literature review in respect to the defined design requirements. Our goal is to realize the best approaches of existing works addressing the requirements, understand their evolution, as well as identify associated open issues. We investigate the 22 most recently introduced RPL-related IDSs in the literature ($2013 - 2020$). We firstly discuss their classification in respect to their detection method and their placement strategy. Then, we build up a timeline of their evolution stages along with their principle qualitative (i.e., detection method, placement strategy) and quantitative features (i.e., number of attacks). The adherence level to the requirements and classification criteria is discussed in the textual descriptions of each IDS.

Our last step involves a synthetic process producing our investigation's outcome, which is to *introduce design guidelines* for up-to-date IDSs. We consolidate the outputs of the steps mentioned above by first, including mapping the IDSs to the type of attacks they tackle. Secondly, we provide a

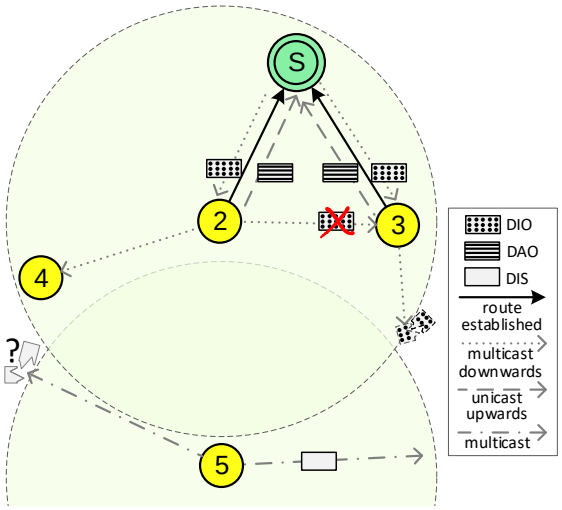

Figure 2: DODAG construction.

summarized comparison viewed under the design requirements we introduce. For the attacks' mapping, we consider both attack detection supported by simulations and those discussed conceptually only. For compliance with the requirements, we are based on the respective authors' claims in the IDS' relevant articles. Since the devised requirements are aligned to the RPL standard objectives, the vast majority of IDSs consider them, and hence, we ended up with a comprehensive comparison that produces and elaborates on four crucial design guidelines for future up-to-date IDSs.

The next section gives a brief overview of RPL, as an essential background for the analysis that follows next.

3. RPL Overview

RPL operates on the IP networking layer, via the 6LoWPAN protocol stack, exploiting Destination Oriented Directed Acyclic Graphs (DODAGs) rooted at a single destination called sink [1]. In practice, the protocol builds a graph of logical paths upon physical network connections, which are directed towards the sink. Parents' selection on paths towards the DODAG root can be

treated as a multi-objective optimization problem since a variety of metrics (e.g., link reliability, latency throughput) and constraints (e.g., nodes' energy, link color) can be exploited to evaluate the nodes' rank [2]. The specified Objective Function (OF) defines how the RPL nodes translate metrics and/or constraints into ranks, and select and optimize routing paths in a DODAG.

As depicted in Fig. 2, the sink-node launches the DODAG's (re)construction based on the exchange of routing control messages, i.e., DODAG Information Object (DIO), Destination Advertisement Object (DAO), DAO-ACK, and DODAG Information Solicitation (DIS). Once the first DIO message is multicasted by the sink, plenty of them are multicasted by nodes getting attached to the graph. DAO messages are used by all nodes, except to the sink, to propagate reverse route information; DIS messages are sent by the not connected (due to their isolated position) or disconnected (due to mobility) nodes in order to solicit DIO messages from other possible connected neighbors and join the graph. DIO messages are critical regarding the graph's construction since they contain the routing metrics and/or constraints, as well as the OF used for the routing paths' establishment.

The DODAG's maintenance is a functionality placed at the very core of the RPL. Hence, a dedicated algorithm, namely the *Trickle timer*, synchronizes the propagation of DIO messages upon which the network's convergence time is based. The critical aspect in DIO multicasting process is the attainment of a short network setup time and, thus, the reinforcement of the network's metrics, e.g., PDR, while restricting the control overhead towards lowering the node's power consumption. To achieve the aforementioned trade-off, the DIO messages are sent periodically; their interval ranges from I_{min} (Minimum Interval) up to I_{max} (Maximum Interval), where $I_{max} = I_{min} * 2^{I_{doubling}}$. For example, the default RPL configuration specifies $I_{min} = 2^{12} = 4.096\ ms$ and $I_{doubling} = 8$ which entails $I_{max} = 2^{12+8} = 17.5\ min$. Actually, the timer's duration is doubled each time it fires. Moreover, any change in the DODAG, e.g., an unreachable parent or a new parent selection, resets the *Trickle timer* to I_{min} [3]. According to the algorithm, DIO messages will be sent at a higher rate when

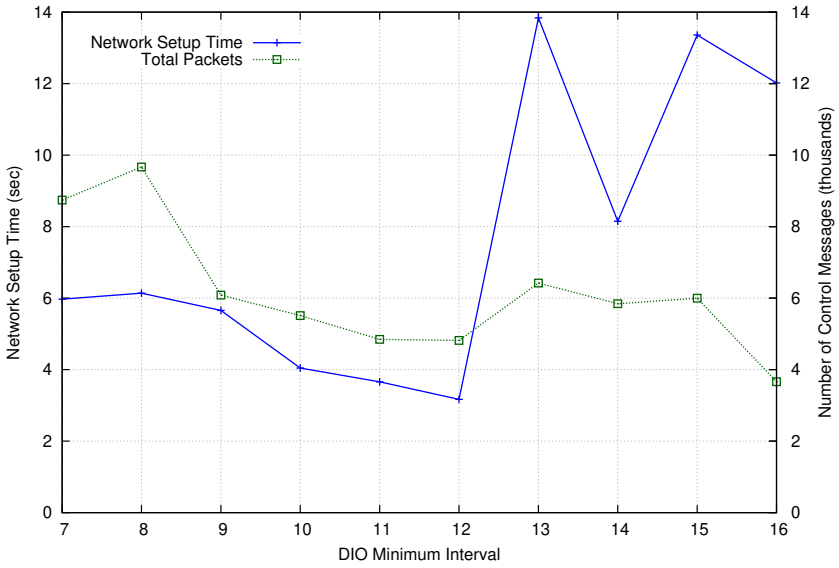

Figure 3: The network setup time and control overhead in respect to the DIO I_{min}.

the network is unstable and slower otherwise, i.e., to reduce protocol overhead and save energy.

The impact of DIO sending frequency in RPL is depicted in Fig. 3. We derive the graph by simulating a WSN in Cooja, which is embedded with Contiki OS [16]. Our explanatory simulation considers a network of one sink and 10 nodes that perform measurements' collection and forwarding them over multi-hop communication. Fig. 3 shows the impact of DIO I_{min} values on the network setup time (left axis - blue squared-dot curve) and on the network control overhead measured in line with the total number of DIO, DAO, and DIS messages (right axis - green x-marked curve). According to the results, high values of I_{min}, i.e., infrequent DIO transmissions, cause delays in network setup time due to the nodes that have not yet received DIO messages and thus remain unconnected. On the opposite, frequent DIO messages entail lower setup time. I_{min} equal to 12, which is the default value in Contiki RPL implementation, provides the best performance concerning the setup time. Regarding control overhead, Fig. 3 validates that higher interval values

produce less network traffic since the frequency of DIO messages is low. Fig. 3 is in compliance with our findings in [3].

Since the *Trickle timer* is the most responsible algorithm for the protocol's performance and along with the DODAG and the sink-node are fundamental parts of the RPL protocol, it is undoubtedly a profound target for a series of attacks.

In the following section, we give a taxonomy and describe such attacks, including those exploiting RPL mechanisms and/or weaknesses. We pay special attention to their impact, since in fact, several attacks may not cause severe damage by themselves. Still, they can have bothersome effects on the network (e.g., control overhead) or on the applications (e.g., PDR) when combined with others.

4. Attacks on RPL-based IoTs

Routing in the RPL-based networks is an incredibly challenging task basically due to the power, storage, memory and processing constraints of the connected devices. The RPL protocol offers several configuration parameters to satisfy diverse requirements regarding deployments of different scale, heterogeneity, and mobility [1, 17] as well as mechanisms to adapt to changes. However, such network contexts, including *resource-constraint nodes*, supporting *dynamic topologies*, and based on the *passive nature of the wireless medium*, do inevitably attract malicious actions, including but not limited to denial of service attacks (DoS), physical damages, and/or extraction of sensitive information, e.g., DODAG version, nodes' rank values, and IDs. In fact, some nodes can get compromised by exploiting the RPL mechanisms themselves; if the node happens to have a significant role in the network, e.g., the sink or parent nodes, then a combination of attacks can be applied with serious effects, spanning from resource-depletion of nodes, due to a sharp increase in the control overhead, to severe degradation of the protocol's performance in terms of data delivery.

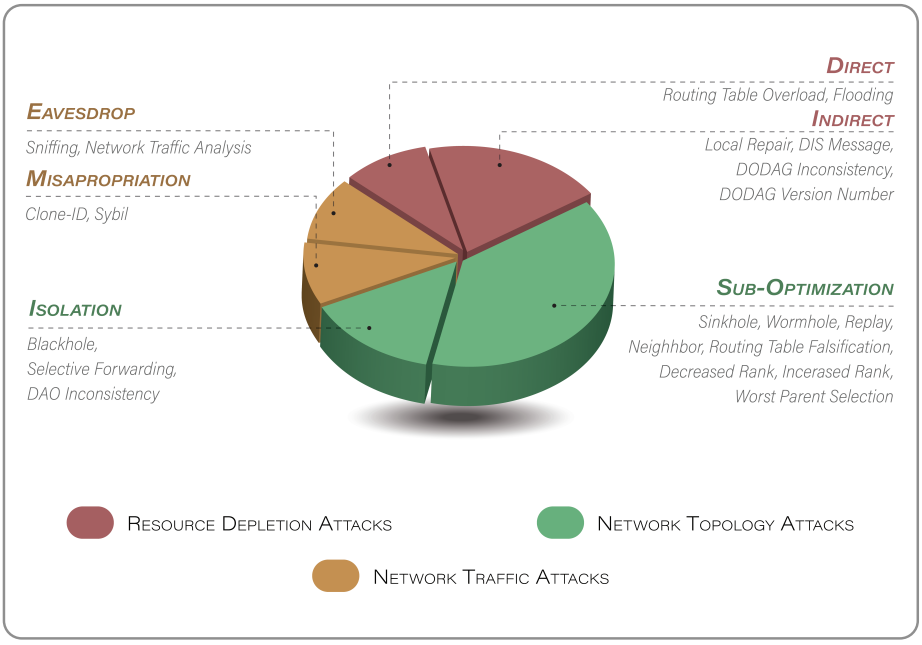

Figure 4: Classification of RPL attacks.

Right afterward, a comprehensive list of the most common and disrupting attacks on the RPL protocol is presented. The network attacks that do not mainly target RPL are not included since they are not part of the paper's scope, e.g., (Distributed) Denial of Service, (D)DoS attacks.

4.1. Diversity of Attacks

Reflected to the aforementioned characteristics of the RPL-based IoTs, i.e., *resource-constraint nodes*, *dynamic typologies* and *passive nature of the wireless medium*, the RPL-related attacks are rather divergent and classified into: *Resource depletion* attacks, *Network topology* attacks and *Network traffic* attacks [5]. Fig. 4 provides a panorama of them along with their classes and sub-classes.

More specifically, the *Resource depletion* attacks include malicious actions that intend to deplete nodes' computing, memory, or energy resources by

creating a false impression of continuous operation. Given that the node's operation is inextricably linked to processing, memory, and energy assets' utilization, any overhead is equitable to excessive consumption of their resources. Consequences may be local or, even worse, affect the overall network availability and performance, leading to routing loops, unnecessary network traffic, and congestion [18, 19, 20].

Attacks against resources are distinguished into *Direct* and *Indirect*, according to the fashion of their execution. In direct attacks a malicious node overloads a subset of nodes-victims and affects their status or operation. Common examples are *Routing Table Overload* [18], and *Flooding Attacks* [11, 18]. On the other hand, indirect attacks manipulate intermediate nodes as a means of broadly affecting the network by, for example, causing unnecessary control traffic. *Local Repair* [19, 21], *DIS Message* [18, 13], *DODAG Inconsistency* [20], and *DODAG Version Number* [22, 23] attacks are typical examples of this sub-category.

The *Network topology* attacks are divided into *Sub-Optimization* and *Isolation* attacks that disrupt the nodes' communication and DODAG's structure, respectively. In practice, the sub-optimization attacks impact the network's optimal convergence ability, i.e., they prevent the establishment of the optimal routes, and thus, affect the network traffic and degrade the network services. Some of the most common consequences include topology inconsistencies, significant packet losses, increased end-to-end delays, network congestion and nodes resources' depletion. The aforementioned effects can be particularly detrimental to dynamic networks due to the nodes' mobility. *Sinkhole* [24], *Wormhole* [25, 26], *Replay* [27, 28], *Neighbor* [18], *Routing Table Falsification* [15], *Decreased Rank* [21], *Increased Rank* [15, 29], and *Worst Parent Selection* [29] attacks are well-known sub-optimization attacks.

Isolation Attacks exploit the tree topology of the RPL network; they aim at cutting off part(s) of the network by interrupting the nodes' communication with either their parent- or sink-node. Amongst their effects are loss of network traffic, end-to-end delay increase, significant service quality

deterioration (e.g., PDR), and isolation of sub-graph parts along with starvation of their participating nodes. The most common isolation attacks are *Blackhole* [19, 30, 31], *Selective Forwarding* or *Greyhole* [19, 24, 30, 31], and *DAO Inconsistency attacks* [5, 11]. These attacks can be severe when combined with others, e.g., decreased rank and blackhole attacks.

The *Network traffic* attacks intercept and monitor the network traffic to acquire or deduce information, e.g., DODAG version or rank value, which can be exploited by attacks launched later on. Depending on how the traffic is affected, they are classified into *Eavesdropping* and *Misappropriation* attacks. In the first case, the intruder monitors the network's transmissions and analyzes the packets either through a breached node or by directly "listening" to the wirelessly transmitted packets. This way, he/she gains access to the topology and routing-related information or even to the actual content of the transmitted packets. The most known eavesdropping attacks include *Sniffing* [5] and *Network Traffic Analysis* [5].

In the latter case, the attacker impersonates other network nodes to extract information about the network topology or gain knowledge of other parameters. The node with the greatest interest in such attacks is the sink due to its crucial role. Appropriating a network node's identity negatively affects the routing service. It also confuses the rest nodes leading to potential incorrect messages' forwarding since, for example, instead of reaching their legitimate destination are delivered to the attacker. *Clone-ID* [5, 11, 24] falls in this category and can be the first stage of further hostile actions causing serious troubles in the network; *Sybil attacks* [24, 32, 33] are an escalated type of Clone-ID attacks which eventually can cause increased network control traffic, high energy consumption and degradation in PDR.

Diversity and/or combination of attacks may affect different aspects of an RPL-based IoT network. The next section provides some indicative examples through simulation.

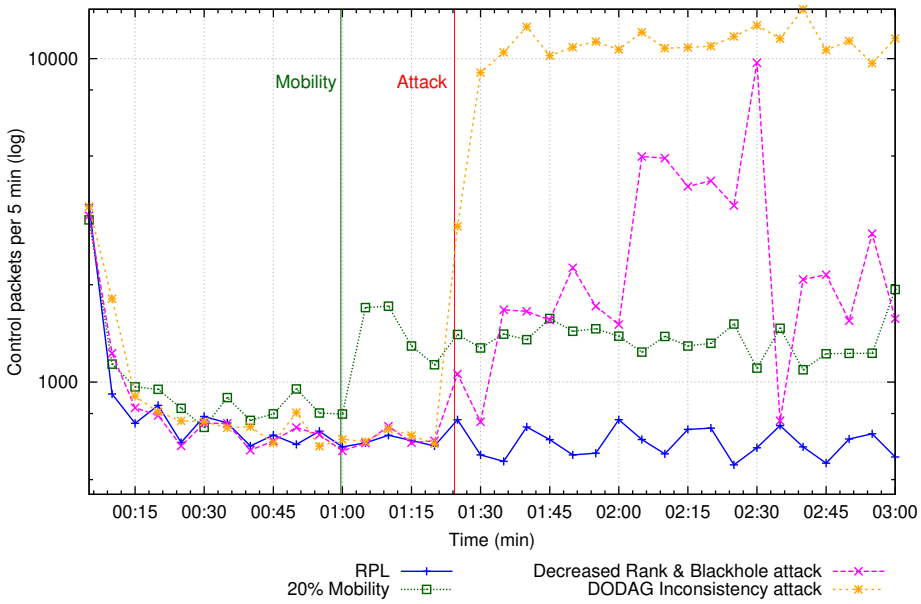

Figure 5: Control overhead under attack and mobility over time.

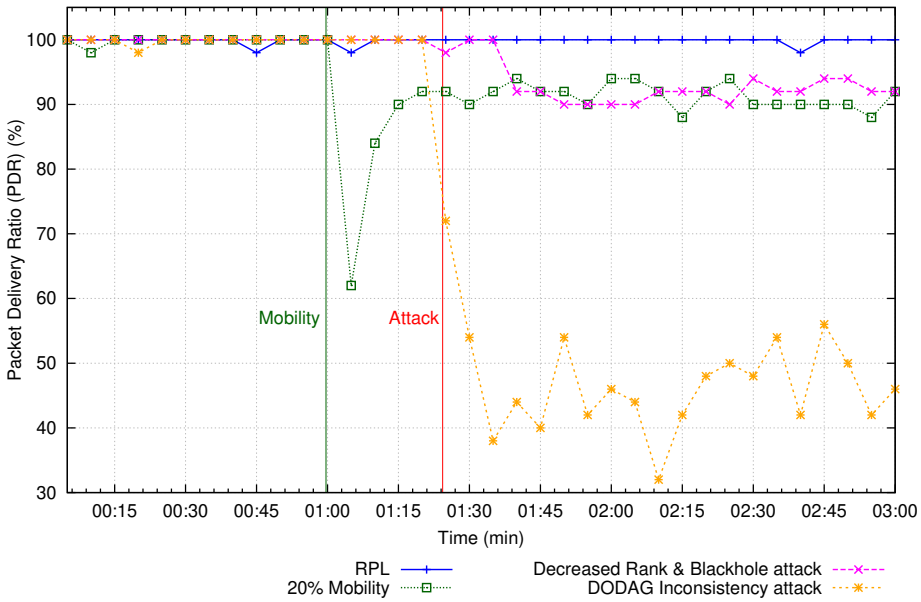

Figure 6: PDR under attack and mobility over time.

4.2. Impact of Attacks

To indicatively illustrate the impact of attacks on an RPL network, we simulate (in Contiki Cooja [16]) a multi-hop network with one sink and 50 nodes randomly placed around it; the outcome is shown in Fig. 5 and Fig. 6. In practice, we run the simulation for three hours (x-axis) and consider that 20 percent of nodes become mobile at 01:00 hour (vertical green line). Regarding attacks, we select one from the resource depletion class, i.e., DODAG inconsistency (yellow curve), and a combination of attacks from the network topology class, i.e., decreased rank and blackhole attack (purple curve). Attacks start at 01:20 hour (vertical red line), for visualization clarity reasons.

Fig. 5 shows the impact of attacks on the network concerning the *control overhead* which is calculated in line with the total number of ICMP packets. The RPL standard operation (blue curve) expresses the ground-truth performance which is contrasted with the performance under attacks' scenario. In our simulation, we notice a heavy impact on control overhead in case of DODAG inconsistency attack, i.e., 750 percent (on average), since a big part of the network is isolated and many nodes are forced to constantly update and recalculate ranks and paths to find routes to the sink. Significant deterioration, i.e., 153 percent (on average), is also caused by the decreased rank and blackhole attacks, launched in combination. This deterioration happens because the attacker advertises a lower rank value compared to all other legitimate nodes in a network's neighborhood, causing the affected nodes to send an excessive number of ICMP packets in their try to find paths to the sink.

Our previous experience with nodes' mobility [3, 4] urges us to investigate further the attacks' impact in comparison to the effects of mobility. The graph confirms our intuition, i.e., trying to get attached to the graph after being disconnected, mobile nodes can create control overhead easily misinterpreted as the effect of an attack, depending on the observation's time-window, e.g., the green and purple curves on the period 01:30 - 02:00.

Apart from the network, attacks also affect the application, e.g., by

Table 1: Design requirements.

i.	RPL Specification Compliance
ii.	Low Overhead
iii.	Scalability
iv.	Robustness
v.	Extendability
vi.	Low False Positives/Negatives
vii.	Mobility Support

aggravating the rate of data packets' delivery. Fig. 6 shows the impact on the PDR which is defined as the received UDP packets (rUDP) over the total number of packets being send (sUDP), i.e., $PDR = rUDP/sUDP$ [3]. While RPL rarely fails to deliver a UDP packet, e.g., 100 percent PDR in the graph, its performance drops to 49 percent on average and to 38 percent on the worst case under DODAG inconsistency attack, since there are no paths to deliver the packets of nodes that are being detached from the DODAG due to the attack. A mild impact, but again very similar to the mobility case, is caused by the rank and blackhole attacks, where the intruder attracts as a parent many neighboring nodes only to drop their data packets once received.

All the above make clear that RPL-based networks must integrate adequate security mechanisms, which will be able to detect and mitigate the attacks of along-coming intruders. According to the literature [5, 11, 34], IDSs are a suitable approach to encounter malicious activities since they aim at detecting several attacks at once, and ideally can be extended to deal with attacks that are not initially included in their design goals. However, the design of an RPL-related IDS has further requirements derived from the protocol itself as well as from the impact of its related attacks. Next section elaborates on such design requirements.

4.3. Design Requirements of an RPL-related IDS

The design of an IDS that aims to shield an RPL network is a challenging task since it should consider the issues of LLNs, the objectives described in RFC 7416 [35], and the heterogeneity of IoT devices, combining them with its principal mission. In that regard, Table 1 presents a set of seven design requirements of an RPL-related IDS whose selection is justified right afterwards.

i. *RPL specification compliance*: In fact, an RPL-related IDS should be primarily compliant with the RPL standard [1], i.e., the fundamental way in which the protocol operates. This includes, among others, the DODAG's construction, the rational of control messages' exchange, the *Trickle timer* algorithm. The advantages of compliance are twofold: firstly, the IDS exploits data that are meaningful in the context of the protocol itself, i.e., the rank value, the number of nodes attached to a single parent-node, which may prevent false positives due to misinterpretations, e.g., attack instead of mobility, as we saw on the previous section. Secondly, it preserves the protocol's efficiency, for example, in terms of time needed for the graph's convergence, packet delay, as well as resource consumption, which is essential in constrained environments.

ii. *Low overhead*: Any security solution should take into consideration resources' availability, let alone when the solution is intended for LLNs. Fig. 5 indicates that a "low budget" approach should take care of control messages exchanged and aim at exploiting the standardized ones to train the system and detect any abnormal event. Keeping the control overhead at regular levels entails energy preservation in transceivers, which are the significant consumers of constraint devices. In addition, components that serve to monitor the network, collect and/or analyze data or perform more sophisticated tasks should be hosted by the nodes with the corresponding processing, memory, storage, and power capabilities.

iii. *Scalability*: In [4] we argue that RPL can cover a wide range of IoT

deployments. Once the LLNs and their routing approaches inherit IoT characteristics, such as large-scale deployment, it is reasonable to evaluate an IDS in terms of its ability to shield the protocol even when the network's size, in terms of connected devices, is significantly increased. Obviously, satisfying scalability should not jeopardize the low overhead requirement.

iv. *Robustness*: The diversity of attacks previously described entails the necessity of an IDS that is able to detect a range of attacks. If an IDS does not protect the network against different types of attacks, the adversary can compromise a node, in the worst case a central one, and affect both the network and applications, as we saw in Fig. 5 and Fig. 6.

v. *Extendability*: Apart from their primary performance with respect to the attacks they cope with, many IDSs can be extended to encounter additional cases. Some systems exhibit a "static", binary rationale that recognizes a known threat pattern or not and proceeds accordingly with the decision. However, new attacks and security issues emerge following the progress of research and development on the IoT. Systems should exploit all current technology assets to remain up-to-date and able to deal with threats that might be currently unknown. To our mind, an IDS can be extendable once its detection method becomes intelligent and its placement is sophisticated.

vi. *Low false (positive or negative) detections*: The effectiveness and detection accuracy of a system is associated with the number of false positives and/or negatives. Thus, beyond being robust and extendable, an IDS should exhibit a high accuracy rate; this means that the system sends alarms for precise attacks while minimizing the cases that attacks are overtaken. To satisfy this requirement, it is necessary to monitor different aspects of the network's operation, e.g., the control overhead combined with the number of times a node changes its parent, or the PDR in line with the local repairs triggered by the RPL itself. This enables more accurate decisions, including differentiating regular but unexpected operations from attacks.

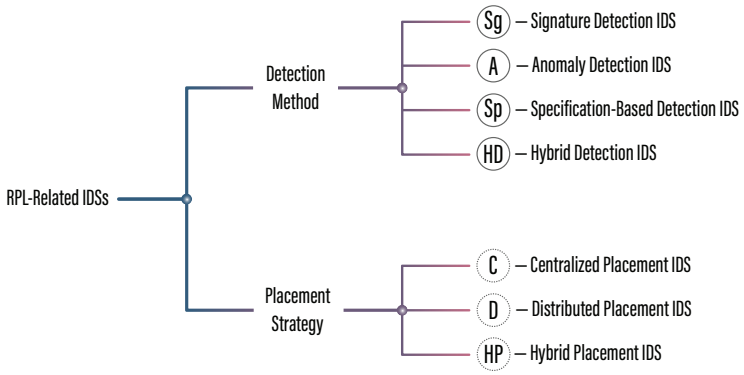

Figure 7: Classification of IDSs in respect to their detection method and their placement strategy.

vii. *Mobility support*: Many applications with mobile IoT devices have emerged over the last decade, and RPL operation under mobility is the leading research challenge since it entails connectivity hand-overs and additional control overhead to maintain the topology [3, 4, 36]. Thus, we should not underestimate or surpass the mobility issue when it comes to the IDS design. Security mechanisms, similar to the basic ones, should consider both fixed and mobile nodes, and the literature has shown, so far, that there are no straightforward solutions.

Apparently, to design an IDS able to satisfy all the above requirements is a great challenge. In the next section, we provide classification and present the evolution of most recent IDSs in the literature as a means of identifying best practices and possible gaps in the so-far related research.

5. RPL-related IDSs

5.1. Classification of RPL-related IDSs

The RPL-related IDSs in the literature are classified according to two main criteria [37]: (i) the detection method they employ, and (ii) their network

placement, as depicted in Fig. 7. Based on the detection method, the IDSs fall into one of the four distinctive categories that follow [11, 13, 37]:

1. The *Signature Detection (S_g)* IDSs identify specific patterns in the network traffic that signify a particular attack [38]. They usually rely on databases [19], which contain known malicious signatures. While these systems consume limited resources, they are not effective against unknown threats [11], since their effectiveness depends on threat awareness.
2. The *Anomaly Detection (A)* IDSs rely on network traffic monitoring and machine-learning or statistical analysis. They develop a healthy network behavior profile, and then compare it to any future network state, intending to recognize possible discrepancies that signal malicious activity. They can detect events that correspond to known or unknown threats at the expense of having high false detection rates [11, 13, 39].
3. The *RPL Specification-based (S_p)* IDSs are similar to the previous ones in the sense that they detect attacks based on divergent network behaviors' observation. However, they build healthy network models by monitoring RPL-related data specified under the security goals [11, 19, 13, 37]. This category's IDSs present high efficiency and low false detection rates while requiring less training time than the *Anomaly Detection* IDSs. Though, in the case of regularly changing environments, their manual configuration reduces their effectiveness.
4. The *Hybrid Detection (HD)* IDSs are a combination of at least two of the categories mentioned above. They tend to inherit the advantages of the combined categories while minimizing their drawbacks [37]. The prevailing hybrid scheme, at this time, is signature along with anomaly detection; to the best of our knowledge, currently, there are five HD systems [40, 41, 42, 43, 44], spanned across the time evolution of IDSs, and three of them, i.e., [40, 41, 42], employ signature and anomaly detection. Signature-based techniques are simple [43] and can be executed very quickly and

efficiently [45], because they rely on pattern matching. Hence, they are a favored choice of combination to detect the known attacks effectively. In contrast, the unknown ones are left to be caught by the mechanism which is combined with, e.g., anomaly detection [40, 41, 42] or specification-based detection [44].

Regarding their placement strategy, the RPL-related IDSs are classified into three categories [37]:

1. *Centralized (C) IDSs* are installed and operate at the root-node of DODAG or at a subset of network nodes [11, 37] assuming that resource-intensive processes are being handled by nodes that are sufficiently equipped [11]. Due to the centralized strategy, these systems are not effective in detecting simultaneous malicious activities in different network locations, e.g., in broad networks. Additionally, such IDSs could render the network exposed in failures at the single point of defense, e.g., the sink-node [46, 47].
2. *Distributed (D) IDSs* on the opposite side, are decentralized and fully implemented in every node of the network. They usually require cooperation between the network nodes [11], whose availability may be highly fluctuated [47]. Detection mechanisms are usually implemented in specific nodes-guards distributed across the network and are responsible for monitoring, whereas the attack mitigation functions are implemented at each node. The benefit of these systems is that threat mitigation is performed from within, as all the nodes are involved in protecting the network [11]. In this manner, the network's scalability and adaptability with a high-security level can be achieved [47]. Nonetheless, the resource consumption of these IDSs remains a significant issue.
3. *Hybrid Placement IDSs (HP)* combine the two previous categories as a means of balancing the pros and cons [11, 19, 24, 37]. In practice, they delegate the resource-demanding processes, such as monitoring, analysis, and decision-making, to the central nodes, while assigning the

lightweight tasks to the rest. Nevertheless, the IDSs of this category require continuous optimization; the central nodes' deployment should be done wisely and may variate for each RPL network [11].

Remarks: As an outcome, we notice that *Signature Detection* IDSs' major weakness is their ineffectiveness against unknown threats. In contrast, the *Anomaly Detection* ones can detect even unknown threats, but they suffer from high false positives' rates. Exploiting data related to the protocol seems promising, and thus, the relevant systems dominate the detection method. However, it is interesting that only two out of five *Hybrid Detection* systems employ them in combination with either signature [43] or anomaly detection methods [44]. This leaves room for investigating the potentiality of hybrid systems that indeed contains RPL specification-based methods.

Apart from the attack detection approach, the design of modern IDSs demands an energy-aware efficient placement strategy due to the resources' limitations of the IoT devices. The decision to place the IDS at the root-node (i.e., *Centralized*) keeps the computationally intensive tasks away from the constrained devices; however, it bequeaths the disadvantages of the single point of failure solutions, i.e., the root-node can be compromised or cut-off. *Distributed IDSs* do not face this problem, plus they can be scaled easily but require some tasks to be executed by the constrained nodes. *Hybrid Placement* logic attempts to blend the above two approaches by keeping the "heavy" tasks for the root-node and delegating the lightweight ones to the rest.

Nowadays, there is a trend towards this category, since it seems to bring satisfactory results. Our experience advocates that this trend can be further enhanced by the emergence of the softwarization paradigm [3, 4, 36]; we discuss this challenge later in the paper.

We now summarize the most recently proposed IDSs based on the above taxonomy, along with a timeline highlighting their evolution.

5.2. The evolution of RPL-related IDSs

The research field of IDSs is vast, but only a restricted subset is appropriate for LLNs [19], i.e., considering the resource-constraints and lossy nature of the latter. In this survey, we identified 22 relevant works that have been proposed in the literature over the last seven years, i.e., from 2013 to 2020. We summarize these RPL-related IDSs in Fig. 8, which illustrates their time evolution along with their qualitative features, i.e., the incorporated detection method and the placement strategy, as well as their quantitative feature, i.e., the number of attacks they encounter.

5.2.1. Signature detection IDSs

Authors in [25, 48, 49, 50, 14, 51, 52] introduce signature detection systems. Regarding their placement, the majority of them [25, 50, 14, 51, 52] are hybrid schemes, while *DEMO* [48] is a distributed and *ELNIDS* [49] is a centralized approach.

DEMO [48] is an adaptation of "Suricata", an open-source IDS, developed in the context of the "EBBITS" European project and deals with flooding attacks. DEMO includes a frequency agility manager (FAM), and a security information and event management system (SIEM). At the same time, it defines two particular non-RPL node types: the IDS node, which is responsible for the attack detection, and the monitoring nodes that monitor the network traffic and send the relevant data via a wired connection (to prevent jamming) to the IDS node for further analysis. The system is scalable and effective in detecting the attacks. Regarding its extendability, the authors propose hosting the Simple Network Management Protocol (SNMP) along with special modules into the system to detect additional attacks and combine DEMO with SVELTE [43] to create a hybrid solution. Overall, exploiting non-RPL nodes and wired connectivity incurs no overhead to the RPL network but also entails a solution that is not totally RPL-compliant.

Compliant with the RPL specification and hybrid regarding its placement, the *Real time IDS for wormhole attacks* [25, 51] exploits measurements

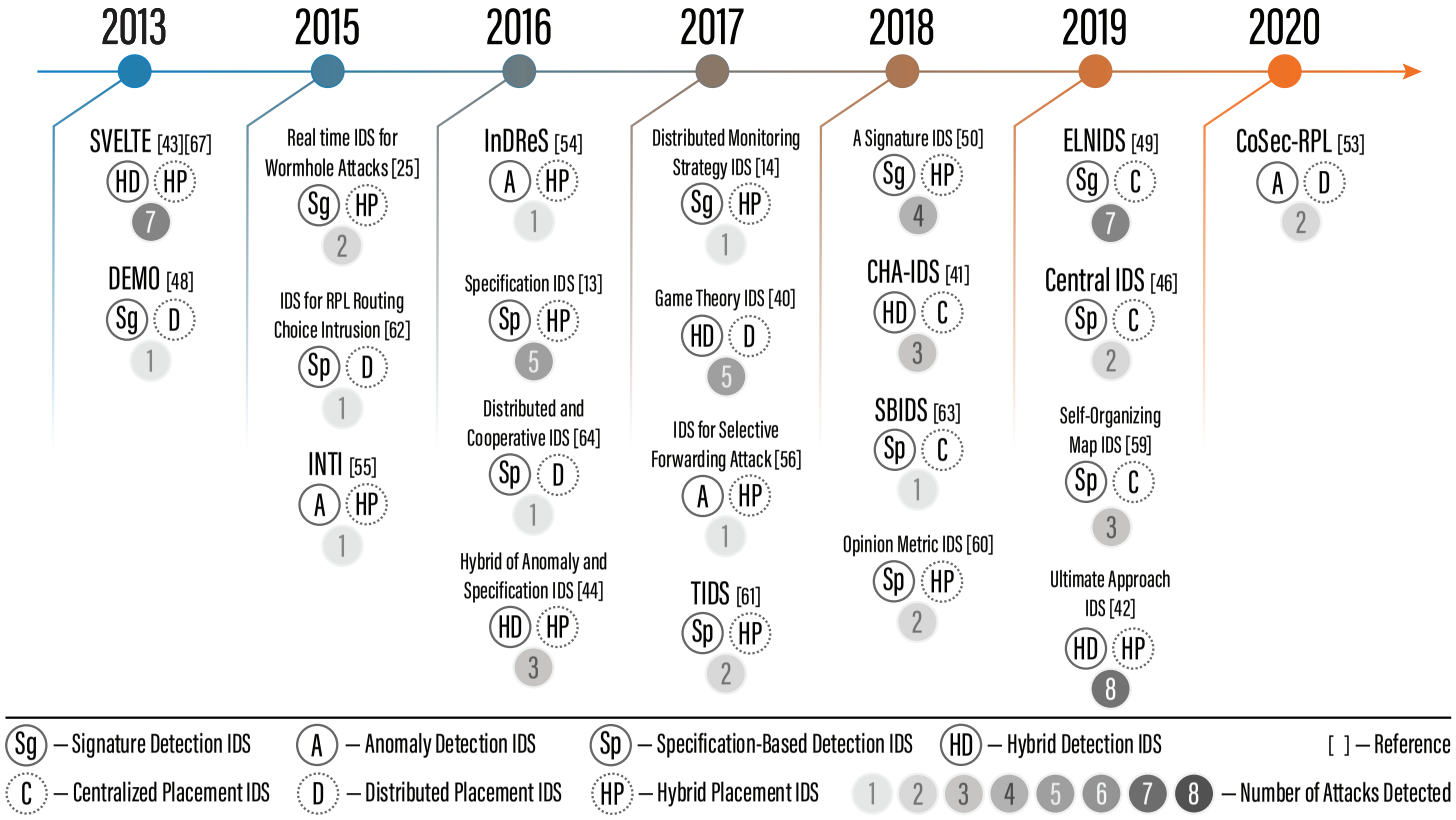

Figure 8: The RPL-related IDSs in a timeline.

regarding the nodes' Received Signal Strength Indicator (RSSI) as a means of cross-checking the network's topology. It deals with two types of wormhole attacks, i.e., by packet encapsulation and by packet relay, as well as with neighbor attacks. More specifically, during the network setup, the root-node records topology-related data and receives by the rest nodes their neighbors' RSSI values. Then, it exploits such information to estimate the distances between the nodes and compare them to the pre-saved topology data to detect discrepancies that indicate an attack. The system demands low resources and has low false detection rates. It can be extended to detect more attacks, such as clone-id, sybil, DODAG version number, and local repair attacks. However, it bases its operation on static topology information ignoring mobility issues that networks usually face.

Distributed monitoring strategy IDS for the detection of version number attacks [14] is also a hybrid placement IDS that focuses on DIO, DODAG version, and nodes' rank monitoring. The IDS defines several monitoring nodes responsible for identifying and sending to the DODAG root a list of malicious nodes detected by tracking the RPL's specification parameters. Once the root receives and merges all the incoming lists, it notifies the network nodes to interrupt further contact with the adversaries. The system behaves effectively in small and medium-scale networks, but its performance deteriorates in high false positives/negatives rates in large networks. An idea to overcome this disadvantage is to cross-monitor each node by at least two other ones.

Another hybrid placement system proposed in 2018 is the *Signature-based IDS for the IoT* [50, 52], which is designed to detect sinkhole, selective forwarding, and clone-ID attacks. It assigns the central role to the IDS router and defines a subset of nodes as IDS detectors. The router serves both as a network traffic monitoring node and a firewall and is capable to access the required resources. The detectors narrow the monitoring operation in their neighborhood and forward any useful information derived by a local, lightweight decision-making algorithm. Among the parameters that the IDS monitors are the RSSI and the packet drop rate. A security scheme is used for

wireless communications' protection; however, the authors suggest the IDS nodes are wire-connected to avoid signal jamming and eavesdropping. The system is extended [50] to also detect the DIS message attacks by monitoring the DIS sending rate and comparing it to a pre-defined threshold. The evaluation shows high accuracy and low false positives even in large networks [50]; concerning the trade-off between performance and overhead, the authors conclude that three to eight detectors should be deployed.

The most recent signature detection system is *ELNIDS* [49] that utilizes artificial intelligence and machine-learning mechanisms on central premises. It is based on ensemble learning to encounter sinkhole, blackhole, selective forwarding, sybil, clone-ID, flooding, and local repair attacks. The IDS relies on the following modules: the sniffer, the sensor events/traffic repository, a feature extraction module, the analysis engine, the signature database, and the alarm/attack notification manager. The sniffer module monitors the network traffic and records the information in the storage unit. The feature extraction module distinguishes the network traffic characteristics that aid in a later classification performed by the analyzer using ensemble models. An event is classified as an attack if any database known signature is detected. According to its evaluation, ELNIDS exhibits high accuracy; however, similarly to the other S_g IDSs discussed, it does not consider nodes' mobility.

Remarks: We can notice that early signature detection systems [14, 25, 48, 51] aim at a special attack by design and operate deterministically. On the contrary, the latest systems of this category [49, 50, 52] expand their impact to a broad range of attacks either by adopting a hybrid placement strategy [50, 52] or by employing centralized machine-learning mechanisms [49], e.g., ensemble learning.

5.2.2. Anomaly detection IDSs

Anomaly detection systems are proposed in [53, 54, 55, 56]; most of them are hybrid regarding their placement [54, 55, 56], while *CoSec-RPL* [53] is the most recent one (published on May 2020) and adopts distributed placement

logic. Both *CoSec-RPL* [53] and *INTI* [55] belong to the IDSs' minority which supports mobility.

Anomaly detection in *INTI* [55] relies on separating the network into clusters (i.e., group of nodes). Each cluster consists of a leader-node, at least one associated-node, and the member nodes. The system bases its functionality on trust estimation, using the nodes' ranks and statistics. The attack detection and the malicious nodes' isolation is performed using the Dempster-Shafer evidence theory [57]. Evaluations [11, 55, 37] showed that the system mitigates sinkhole attacks at the cost, however, of high computational processing requirements. According to the authors [55], INTI is an extendable IDS and takes into account nodes' mobility.

InDReS [54] is an improvement of *INTI* [55] that keeps the main principles of functionality while limiting the computational overhead, thus preserving resources which is critical for LLNs. Once the system identifies malicious nodes, it reconstructs the network's topology, excluding them. However, compared to its predecessor, InDReS' performance was not evaluated in terms of false positives/negatives and mobility support.

The *IDS for selective forwarding attack* [56] was proposed in 2017 and uses the Sequential Probability Ratio Test (SPRT) combined with an adaptive threshold. Its mechanism relies on two modules: the first is responsible for decision making and is implemented at the root-node. The second, used for incoming and outgoing packet monitoring, operates on the rest routing nodes. The monitoring nodes send information to the root via randomly selected paths. The root analyzes the data it receives using the SPRT and assigns every node with a probability of being malicious. The decision making is based on a threshold above which a node is classified as malicious. Then, the root notifies the non-malicious nodes about the adversaries' presence and initiates a DODAG global repair in order to isolate the possible intruders. The system's evaluation indicates its effectiveness, which comes at the cost of being resource-intensive. Due to the high resource requirements, the IDS is not scalable.

CoSec-RPL [53] has been lately introduced and deals with a combination of flooding and replay attacks, namely "copycat attacks". To detect anomalies and analyze the statistical data, the system relies on a modified version of the Interquartile Range (IQR) Outlier Detection (OD) method [58], which uses the median instead of the mean value and entails less implementation complexity. The idea behind CoSec-RPL is to identify the nodes with significantly diverse behavior. The authors tune the IDS's thresholds appropriately via multiple experiments. CoSec-RPL is triggered whenever a DIO message is received from any neighbor and monitors the time difference between consecutive DIO messages. When measurements surpass certain thresholds, a node is initially considered suspicious, and its state is characterized accordingly as "suspected". In this state, communication with the node is still allowed; however, when a second threshold is reached, the node is considered malicious, and its state becomes "blocked"; in this case, no further communication with it is permitted. Even though the system's memory requirements are not negligible, since it demands a neighboring table in every node to store relative information, they are not prohibitive for IoT devices, and thus it does fit inside a Z1 mote. *CoSec-RPL* is evaluated under both static and mobile network scenarios and is proved to be very useful. However, it performs better in fixed topologies (since mobility affects the intervals of DIO messages transmissions). It can be extended to detect more attacks, particularly DIS flooding, DAO insider, wormhole, and spoofed copycat attacks.

Remarks: The anomaly detection IDSs are a minority of the systems under analysis (four out of 22), probably because anomaly detection is, by definition, a general method, loosely coupled with the RPL itself. So far, most systems [54, 55, 56] have been exercised with only one attack type, but they can potentially detect unknown attacks. Such a feature relates to the anomaly detection mission, which identifies unusual or even unknown "behavior" and attributes it to an attack. They mainly exploit intelligent mechanisms, e.g., clustering, probability theory, and statistical parametric or non-parametric tests, along with appropriately defined thresholds. Of course, thresholds'

tuning is an important issue since it may result in either high false positives or negatives. As we will see later in this section, combining the advantages of anomaly detection with other detection methods brings very positive results [40, 41, 42, 44]. It is indicative, for example, that they dominate as a component of the *Hybrid Detection (HD)* systems.

5.2.3. Specification-based detection IDSs

IDSs of this category [13, 46, 59, 60, 61, 62, 63, 64] share the feature of taking into account RPL-related information, e.g., control messages, rank value, DODAG information, and try to identify an attack exploiting such knowledge. Regarding their placement, there is a shared trend.

IDS for RPL routing choice intrusion [62] is a distributed placement system that relies on monitoring DIO messages' fields, nodes' parents and rank values, as well as the number of nodes connected to a single parent to detect decreased rank attacks. The idea is that a low-rank value advertised by a node that presents an increased number of nodes attached to it indicates that this node is probably malicious. Energy requirements were taken into account, and the IDS can operate in large networks.

The IDS proposed in [13] is a hybrid placement system that, similarly to the *INTI* [55], divides the network into clusters and uses specification-based detection to mitigate the attacks. It is designed to repel sinkhole, worst parent selection, local repair, neighbor, and DIS message attacks. The system is effective, it presents low false detection rates, and due to its low energy demands, it is scalable. It can be extended to detect a broader range of attacks; however, it does not address mobility issues.

The *Distributed and Cooperative Verification IDS to defend against DODAG version number attack* [64] suggests that when the nodes receive a DIO message containing an increased DODAG version, the message should be accepted once it is confirmed. In case that the sender is the root-node, the receiver will accept the message; otherwise, the receiver requests the DODAG version number from its two-hops-distant neighboring nodes. This functionality demands two additional

message types, the "CVQReq" for the request and "CVQRep" for the reply. Evaluation results show that the IDS is effective against the DODAG version attack; however, the false detection rate increases in proportion to the attacking nodes' number. Furthermore, the control overhead is significantly low.

TIDS: Trust-based IDS [61] is a hybrid placement system that mitigates sinkhole and selective forwarding attacks using the notion of trust. TIDS relies on Subjective Logic [65], incorporating variables both for trust and uncertainty, and considers a node as malicious when its disbelief value is higher than its belief value. Trust values are calculated based on the level of nodes' good cooperation and conformity with the RPL specification. Each node observes its neighbors and forwards the recorded data to the root-node using a new control packet, namely "Trust Information (TRU)". The root-node has the required resources for the purpose and calculates the trust values. The system was evaluated and found to successfully detect sinkhole attacks even in large topologies (at the expense of high energy demands on the root-node), while selective forwarding attack was discussed only in a theoretical context. According to the author, TIDS is useful in topologies comprised solely of static nodes, and it can be extended to mitigate version number attacks additionally.

SBIDS: Sink-based Intrusion Detection System [63] is a centralized system designed to detect decreased rank attacks in non-storing RPL networks. The root-node, which is considered trusted by default, marks a node as malicious by monitoring the rank changes and defining thresholds accordingly, i.e., it records the previous and current ranks of parent-nodes, and establishes a threshold for parent switching. SBIDS considers both static and mobile nodes. Its evaluation revealed high accuracy in large networks in both cases; however, its performance degrades as the number of attacking nodes increases, especially when mobility is considered. Concerning the power consumption, the IDS incurs an overhead of around 20 percent compared to the unprotected network consumption. Finally, SBIDS can be extended to accommodate more routing metrics and, thus, repel additional attacks.

Opinion Metric based Intrusion Detection System for RPL Protocol in

IoT [60] is a hybrid placement IDS, able to mitigate sybil and flooding attacks, utilizing an opinion metric-based mechanism which is based on subjective logic [65]. The nodes monitor their neighbors' transmissions and rate them according to their compliance with the RPL specification. Nodes that behave as per specification principles are rated positively, whereas the diverging ones are rated negatively. The ratings are later aggregated to the root-node, where the subjective logic (the "Θ" consensus operator) is employed for the malicious nodes' detection. A node is considered malicious when the aggregated degree of disbelief exceeds a threshold. The system is solely evaluated in terms of detection performance, and a considerable number of false detections were recorded. Nevertheless, the authors plan to extend their work and consider additional routing attacks using a neural network trust model.

A Central IDS able to mitigate flooding and DODAG version number attacks was proposed in [46]. The system is implemented at the root-node and uses genetic programming to generate the IDS's algorithm automatically. The root continuously analyzes the network traffic and extracts 50 features, which are later used for the constitution of the genetic programming trees. The last generation's best individual (tree) is evaluated for both flooding and DODAG version number attacks, and two corresponding detection algorithms are obtained. In its current version, a central logic is adapted. The root-node executes the resource-demanding tasks; the authors also suggest a decentralized fashion of operation, but this entails further challenges to be addressed. The system is highly effective, probably due to centralized monitoring, which provides a global network view. Aspects such as resource requirements, scalability, extendability, and mobility support, were left out of the system's evaluation.

Self-Organizing Map IDS for RPL Protocol Attacks [59] exploits machine-learning and more precisely Self-Organizing Maps (SOM), built centrally to the RPL network, to detect flooding, sinkhole, and DODAG version number attacks. The authors elaborate on the way that several modules collaborate to generate the maps. Initially, synthetic data from numerous simulations of different real-

life scenarios were produced and used as input to the "aggregator" module. This module utilizes six packet fields (i.e., message type – DIO/DIS/DAO, IP addresses of the sender and destination nodes, current DODAG version, current sender node rank, Unix timestamp), pre-processes the input data and provides as an output six features (i.e., DIS, DIO, DAO, DODAG version changes, rank changes to total messages ratios in the timeframe, average power consumption on the destination node in the timeframe). These features are getting normalized by the "normalizer" module, to be used by the "trainer" module to generate the maps. Simulations run by the authors indicate that the IDS is able to identify the attacks.

Remarks: Not surprisingly, eight out of 22 systems (36.4 percent), according to the Fig. 8, fall in this category. Either intuition or experience leads the researchers to exploit the cardinal RPL data structure, i.e., the graph, and its relevant information, e.g., control messages and *Trickle timer* algorithm, in IDS design. However, judging by the outcome, the specification-based detection, either as a single detection method or in combination with others, performs moderately regarding the number of attacks. In the worst-case, systems detect one attack [63, 62, 64], while it is remarkable that they perform better once hybrid placement strategy is adopted [13, 60, 61], or when RPL-related information is processed by machine-learning mechanisms [59, 60]. Indeed the specification-based systems that exploit clustering, trust schemes, genetic programming, and artificial neural networks to process the RPL-monitoring parameters outperform those that take these parameters into account without any kind of intelligence.

Here, the aftermath is that tight coupling with the protocol itself is not sufficient; it is a step to start with. Mixing techniques can help to develop robust systems that do not jeopardize performance and cost.

5.2.4. Hybrid detection IDSs

SVELTE [43] is one of the oldest RPL-related IDSs. It is a hybrid placement system that consists of three modules: (i) the 6LoWPAN Mapper

(6Mapper), implemented at the root-node, maps and keeps track of the DODAG along with the parent and neighboring information of each node; (ii) the intrusion detection module, which is also executed centrally, relies on the RPL specification, signature and anomaly detection to specify the attacks, and; (iii) the distributed firewall and response module that prevents the out-of-network attacks and is implemented in every node. SVELTE combines all three detection methods and tries to achieve a trade-off between the storage cost of S_g and the computing cost of anomaly detection techniques. The system's evaluation revealed its effectiveness against blackhole, selective forwarding, sinkhole, and DODAG inconsistency attacks. However, since SVELTE uses a rank threshold to detect anomalies, it suffers from high rates of false positives/negatives [13, 43, 54, 37]. In addition, it has significant resource requirements and does not take into account mobility issues. Improvements of SVELTE [66, 67] reduce false detections and add geographical hints of the malicious nodes, increasing the IDS's robustness by allowing it to discover clone-ID, sybil and wormhole attacks additionally.

Hybrid of Anomaly-Based and Specification-Based IDS for IoTs Using Unsupervised OPF Based on MapReduce Approach [44] is a full hybrid approach that encounters selective forwarding, sinkhole, and wormhole attacks. The system combines an Anomaly Agent-Based IDS (AA-IDS) with several Specification Agent-Based IDSs (SA-IDSs) and considers the leaf-nodes traffic solely to the root. The SA-IDSs, implemented at the router-node(s), are used for traffic monitoring and the identification of malicious nodes. Once traffic is analyzed, the output data are embedded into data packets and forwarded to the root-node, where the AA-IDS resides. AA-IDS employs the unsupervised Optimum-Path Forest (OPF) algorithm [68] to cluster the collected data and proceed with the anomaly detection. The decision that classifies a node as malicious or not is based on a voting mechanism that considers both local results of SA-IDS agents and the global analysis of the AA-IDS. The system can also be extended to mitigate blackhole and decreased rank attacks.

The authors developed a dedicated RPL WSN simulator for their

evaluation analysis and provided high accuracy rates regardless of the network size, justifying this way their system's scalability; their evaluation, however, considers only a static topology. Regarding the energy requirements, abundance was taken for granted for all kinds of nodes. Still later in a theoretical context, it was concluded that the IDS could be used in real-world IoT applications by offloading the resource-intensive tasks from the root-node to an external device; obviously, such assumptions leave space for improvements.

Game Theory IDS [40] is a distributed placement IDS that combines signature detection for the known attack patterns and anomaly detection for the unknown ones. In this way, the system is proved to encounter a considerable number of attacks, i.e., flooding, sinkhole, blackhole, sybil, and wormhole attacks. The Nash Equilibrium Game Theory is used to set a game between the IDS entities and the attackers; when the system detects a traffic pattern that reaches a threshold, it considers it an anomaly. To reduce false detections, the authors combine the IDS with a reputation system. The evaluation of the IDS assumes both fixed and mobile nodes and reveals low requirements on resources.

CHA–IDS [41] is a centralized system that elaborates on the IPv6 compressed header's analysis using machine-learning. In fact, the root-node extracts data from the network traffic, which are later used as an input to the "J48" algorithm [69] for the attacks' detection. In this way, it detects flooding, sinkhole, and wormhole attacks, taking place either individually or in combination, with high accuracy. According to the authors, the system exhibits a good performance regarding the trade-off between performance and overhead. However, in its current version, it does not succeed in locating the attacker's position; future extensions and possible combinations with other distributed placement schemes could offer this capability. Furthermore, extensions could improve the system to additionally mitigate sybil, clone-ID, DODAG version number, and local repair attacks.

Lastly, the *Ultimate Approach IDS of Mitigating Attacks in RPL Based*

Hybrid Detection IDS	Detection Method			Placement Strategy		Number of Attacks	Attacker Localization
	Signature Detection	Anomaly Detection	Specification Detection	Centralized Placement	Distributed Placement		
SVELTE [43][67]	✓	✓	✓	✓	✓	7	✓
Hybrid of Anomaly and Specification IDS [44]		✓	✓	✓	✓	3	✓
Game Theory IDS [40]	✓	✓			✓	5	
CHA-IDS [41]	✓	✓		✓		3	
Ultimate Approach IDS [42]	✓	✓		✓	✓	8	✓

Figure 9: Overview of the Hybrid Detection IDSs.

Low Power Lossy Networks [42] follows a holistic approach, is full hybrid regarding its design and encounters the maximum number of attacks, i.e., eight. More specifically, the system encounters sinkhole, DODAG version number, flooding, neighbor, wormhole, decreased rank, clone-ID, and sniffing attacks and can detect events that originate both inside and outside the network. The IDS incorporates many non-mobile sink/sub-DODAG parent-nodes that can detect both known signatures and anomalies. The system uses blockchain and calculates trust values to detect the attacks and isolate the adversaries. The author presents a conceptual framework of their approach, stating its effectiveness along with low resource requirements and its ability to be extended. The system seems to partially support mobile nodes since only the root and the sub-DODAG parents are considered to be fixed-positioned.

Remarks: The time evolution of IDSs (Fig. 8) shows that hybrid detection systems span across the whole investigation period, i.e., $2013 - 2020$, indicating that even in the early systems, such as SVELTE [43], the researchers pinpointed that combining the attacks' detection methods brings advantages to the process. The basic and, probably, the apparent benefit is quantitative and regards the number of attacks that the system can encounter; this ranges from three to eight

as depicted in Fig. 9.

Further benefits include the ability of some systems to localize the adversary [42, 43, 44], as well as the detection accuracy rate in conjunction with low resource overhead, especially when the developed mechanisms are appropriately located both in central and distributed nodes. In particular, appropriately tuning the parameters of *SVELTE* [43] can offer as much as 100 percent of detection accuracy and zero false positives. In comparison, solution [44] shows an average of 93.3 percent accuracy with less than 3.3 false positives for multiple runs. *Game Theory IDS* [40] reports an average of 98.6 percent accuracy and less than 2.5 percent of false positives for a variety of setups, while *CHA–IDS* [41] shows an accuracy within $85.2 - 100$ percent and up to 0.058 percent false positives, in the worst case.

Evaluating these numbers in real-world environments is a challenging issue that certainly deserves a further investigation, e.g., whether they allow a realistic operation of the particular IDSs. This angle of investigation is associated with: (i) the considered use-case in terms of required security level and affordable control overhead or processing cost; and (ii) the type of involved mitigation action and its impact, since this determines the communication or performance issues a false positive causes.

Most of these hybrid systems use machine-learning, i.e., *Game Theory IDS* [40], *CHA–IDS* [41] and [44] employ Nash equilibrium game theory, the "J48" algorithm, and unsupervised data mining, respectively. We omitted a more in-depth discussion and comparative analysis on the involved algorithms in the IDSs at this point of the investigation since we mainly focus on their systemic aspects. Such investigation requires comparisons between different approaches (e.g., machine-learning vs statistics-based) under a given environment or theoretical investigations on their impact on the computational burden, as an example. From our point of view, this exercise diverges from the given scope of the paper. However, this issue is important and complex enough to deserve an independent study. Consequently, it is considered future work.

Next, we provide a brief summary that compacts the individual remarks into

a set of best practices and identified gaps in IDS design.

5.3. Best Practices & Gaps

The so far research, reflected on the IDSs under analysis, reveals best practices in the design of RPL-related IDSs. The most important is that utilizing detection methods in conjunction can bring a high score regarding the number of attacks detected. In particular, anomaly detection contributes as a general method to detect both known and unknown threats and performs excellent with either signature or specification-based methods, which provide some kind of "knowledge" to the process, i.e., patterns or threshold crossings of RPL-related parameters. Another best practice is to exploit both distributed and centralized mechanisms to achieve optimal placement in the detection mechanisms. This includes coarse-grained, lightweight monitoring at every node which conditionally triggers fine-grained, resource-demanding processes executing at central premises, e.g., machine-learning. The third point is that detection by its own narrows the IDSs' mission; some systems [42, 43, 44] go beyond it by identifying the attacker(s) and mitigating the threats using information relevant to the RPL protocol.

This observation combined with the summary of the most robust systems – Fig. 9 – reveals that eventually, a minority of IDSs follows a holistic approach that deals with the threefold mission of detection, identification, and mitigation. Thus, there are several gaps in the literature regarding methods: to identify and then mitigate the intruder, to detect multiple attacks, to deal with false positives decisions, e.g., how and when a blacklisted node comes back to the network and which are the coincidences of its isolation. Our analysis also finds the lack of an architecture beyond a hybrid-wise fashion of combination and builds up a "polymorphic" system able to adapt in dynamic conditions.

Finally, we notice a lack of IDS evaluation in real environments, i.e., test-beds, since the majority of systems in our analysis are evaluated using simulations. More specifically, 16 out of 22 IDSs utilize Contiki Cooja [16], while NS-2, Matlab and TOSSIM simulators are also used for evaluation in

[54], [40] and [49], respectively. Only authors in CHA–IDS [41] document utilizing Cooja in combination with a test-bed facility, however, without providing the details of the latter. Our previous experience with test-beds participating in the FED4FIRE [70] and GENI [71] federations, in the context of 5G network slicing research [72, 73, 74], shows that it would be interesting, but also very challenging, to deploy complete IDSs in test-beds for evaluation reasons and address possible issues that arise. Currently, the Sharing Artifacts in a Cybersecurity Community Hub (SEARCCH) project [75] offers a facility that provides validation, repeatable sharing, and reuse of security-related research results. A relevant initiative for IoT security could establish a common framework where open-source IDS code could be released and comparatively evaluated, e.g., in a common environment with the same methodology and evaluation scenarios.

The section that follows proceeds with a comparative analysis of the IDSs under investigation that includes: (i) a complete mapping of IDSs to the type of attacks they encounter; and (ii) their comparison in the light of the design requirements we introduce. The ultimate goal is a list of four guidelines that, to our mind, a modern IDSs should follow.

6. Comparative Analysis & Insights

6.1. Mapping IDSs to Attacks

We start our comparative analysis by assigning each of the 22 most recently introduced IDSs under discussion to the RPL-related attacks they tackle. This is a challenging and not straight-forward task, since it depends on how an IDS covers the addressed attack(s). To this point, our literature study reveals that different approaches are spanning from simulating all or some of the attacks to conceptually supporting coverage for all or subset of the attacks under study. In the case of simulation approaches, differences also concern the simulation environments as well as the metrics used to evaluate the IDSs' performance.

Table 2: Mapping the IDSs to the type of mitigated attacks.

		Attacks		IDS
RESOURCE DEPLETION ATTACKS	DIRECT	Routing Table Overload		-
		Flooding		**[40]**, **[41]**, [42], **[46]**, **[48]**, **[49]**, [53], **[59]**, **[60]**
	INDIRECT	Local Repair		**[13]**, **[49]**, [25]*, [41]*
		DIS Message		**[13]**, **[50]**, [53]*
		DODAG Inconsistency		[43]
		DODAG Version Number		**[14]**, [42], **[46]**, **[59]**, **[64]**, [25]*, [41]*, [61]*
NETWORK TOPOLOGY ATTACKS	SUB-OPTIMIZATION	Sinkhole		**[13]**, [40], **[41]**, [42], **[43]**, **[44]**, **[49]**, [50], **[54]**, **[55]**, **[59]**, **[61]**
		Wormhole		**[25]**, [40], **[41]**, [42], [43] (D. Shreenivas' version [67]), **[44]**, [53]*
		Replay		**[53]**
		Neighbor		**[13]**, [25], [42]
		Routing Table Falsification		-
		Rank Attacks	Decreased Rank	[42], **[62]**, **[63]**, [44]*
			Increased Rank	-
			Worst Parent Selection	**[13]**
	ISOLATION	Blackhole		[40], [43], **[49]**, [44]*
		Selective Forwarding		**[43]**, **[44]**, **[49]**, [50], **[56]**, **[61]**
		DAO Inconsistency		[53]*
NETWORK TRAFFIC ATTACKS	EAVES-DROP	Sniffing		[42]
		Network Traffic Analysis		-
	MISAPPR-OPRIATION	Clone-ID		[42], [43] (D. Shreenivas' version [67]), **[49]**, [50], [25]*, [41]*
		Sybil		[40], **[49]**, [43] (D. Shreenivas' version [67]), **[60]**, [25]*, [41]*

– IDSs in **[bold]** are evaluated through simulations for the corresponding attack.

– IDSs with the star mark (*) can be extended to encounter the corresponding attack according to the authors' declaration in the relevant publication.

– The rest IDSs are mapped to the corresponding attack according to the authors' declaration in the relevant publication.

To proceed with our mapping, we listed the attacks with respect to the classes they belong to and are illustrated in Fig. 4. Next, to highlight the aforementioned differences, we mark in bold the IDSs in a row when they are evaluated through simulation (e.g., based on Contiki Cooja, NS-2, Matlab, or TOSSIM) for the attack on the same row on Table 2, while regular fonts indicate that no simulation is carried out. Regular fonts with the star mark refer to the IDSs that can be extended to tackle an attack, according to the corresponding authors. The outcome is summarized in Table 2 which synthesizes the knowledge gained from Sections 4 and 5.

To better highlight the mapping process, we give two indicative examples. The authors in [13] utilize Contiki Cooja [16] and evaluate their IDS against sinkhole, worst parent selection, local repair, neighbor, and DIS message attacks; their simulation results include true positives/negatives, false positive/negatives, and energy consumption. For this reason, the reference [13] appears in bold in rows: $3, 4, 7, 10$ and 14 that refer to the aforementioned attacks. On the other hand, *SVELTE* [43] is an example for which the authors declare its effectiveness against selective forwarding, sinkhole, blackhole, and DODAG inconsistency attacks. However, they evaluate it only for the first two attacks using the metrics of true positive rate, energy and memory consumption in Contiki Cooja [16]. Thus, it appears in bold only in rows 7 and 16; the rest entries on the table are with regular fonts. The same applies to SVELTE's improvement [67] where the corresponding authors claim effectiveness against clone-ID, sybil and wormhole attacks due to additions considering the malicious nodes' geographical position. However, relevant to these new attacks results are not provided. The only simulation results refer to the reduction of false detection rates for the initial attacks having already been evaluated, i.e., selective forwarding and sinkhole.

Mapping of Table 2 reveals that the vast majority of the RPL-related IDSs (73 percent) deal with network topology attacks; this is expected since the DODAG and its related mechanisms, i.e., the *Trickle timer* algorithm, and parameters, i.e., DODAG ID and rank values, play a cardinal role on the RPL

networks. An even more interesting fact is that as much as 54.5 percent of the IDSs focus on the *Sinkhole* attacks indicating the sink-node's major role to such networks. On the contrary, network traffic attacks do not attract significant attention, probably due to the passive nature of eavesdropping attacks, which are difficult to be detected. To our mind, energy-awareness, in conjunction with resources' limitations on IoT networks, create an emerging field of research regarding the resource depletion attacks and the corresponding IDSs.

Table 2 also shows that some IDSs [13, 42, 49] are more robust than others since they encounter a greater number of attacks; in fact, they repel different attacks that expand to all three categories, i.e., resource depletion, network topology, and network traffic attacks. Among them, the *Ultimate Approach* [42] introduces a full-hybrid, conceptual framework where the authors discuss but not evaluate their IDS with respect to the attacks encountered. On the contrary, the Specification-Based IDS [13] and ELNIDS [49] tackle five and seven attacks, respectively, for which simulation analysis and results are provided. SVELTE [43] addresses seven different types of attacks, evaluates a subset of them through simulation, and gives an indication towards the potentiality of full-hybrid IDSs to deal with a broad spectrum of attacks. Overall, the majority of works (17) proceed with comprehensive simulation approaches in the sense that they evaluate all the attacks the corresponding authors claim tackling. A small subset of works [25, 40, 43, 50] evaluate through simulation a portion of attacks they investigate, while Kaur [42] introduces a conceptual work that misses simulation results.

In the following section, we elaborate on comparing those RPL-related IDSs in light of the design requirements we introduced.

6.2. IDSs' Comparison

Table 3 presents the comparative overview of the 22 IDSs under analysis (their order is consistent with their time evolution on Fig. 8) in respect to the seven design requirements introduced and discussed in Section 4.3. The

Table 3: Comparative overview of RPL-related IDSs.

IDS	Criteria						
	i	**ii**	**iii**	**iv**	**v**	**vi**	**vii**
SVELTE [43] [67]	✗	✗	–	✓	✓	✗	✗
DEMO [48]	✗	–	✓	✗	✓	–	✗
Real time IDS for Wormhole Attacks [25]	✓	✓	–	✗	✓	✓	✗
IDS for RPL Routing Choice Intrusion [62]	✓	✓*	✓	✗	–	–	✗
INTI [55]	✓	✗	✓	✗	✓	✓	✓
InDReS [54]	✓	✓	–	✗	✓	–	✗
Specification-Based IDS [13]	✓	✓	✓	✓	✓	✓	✗
Distributed and Cooperative Verification IDS [64]	✗	✓	–	✗	–	✓*	✗
Hybrid of Anomaly and Specification Based IDS [44]	✓*	✗	✓	✗	✓	✓	✗
Distributed Monitoring Strategy IDS [14]	✓	–	✓	✗	–	✓*	✗
Game Theory IDS [40]	✓	✓	✓	✓	–	✓	✓
IDS for Selective Forwarding Attack [56]	✓	✗	✗	✗	–	–	✗
TIDS: Trust based IDS [61]	✗	✗	✓	✗	✓	✗	✗
Signature IDS [50]	✓	✗	✓	✗	✓	✓	✗
CHA – IDS [41]	✓	✗	–	✗	✓	✓	✗
SBIDS: Sink-based IDS [63]	✓	✗	✓	✗	✓	✓	✓
Opinion Metric based IDS [60]	✓	–	–	✗	✓	✗	✗
ELNIDS [49]	✓	–	✓	✓	✓	✓	✗
Central IDS [46]	✓	–	–	✗	–	–	✗
Self-Organizing Map IDS [59]	✓	–	–	✗	✓	–	✗
Ultimate Approach IDS [42]	✓	✓*	–	✓	✓	–	✓*
CoSec-RPL [53]	✓	✗	–	✗	✓	✓	✓

Design requirements:

i = RPL specification compliance

ii = Low overhead

iii = Scalability

iv = Robustness

v = Extendability

vi = Low false positives

vii = Mobility support

* = Under certain conditions or estimated but not evaluated

✓ = Satisfied

✗ = Not Satisfied

– = No Information Available

comparison shows if a system satisfies (✓) or not (✗) each of the requirements, while a dash (−) denotes that no information is available. We are essentially based on the respective authors' claims in the relevant articles and, in some cases, we exploit feedback from them for clarifications. This way, we manage to build a table completed as much as 80.5 percent, which indicates that both the design requirements and the comparison itself are meaningful.

Elaborating on RPL-related systems, it is expected that the majority of them are compliant with the protocol. However, even if they are designed for LLNs only one-third of them presents low overhead; the rest are either high-cost solutions or do not clarify their trade-offs in terms of performance and cost. Half of the systems are scalable, and the rest are not evaluated for large-scale deployments.

Regarding the robustness, most of the systems deal with up to four attacks, while almost 37 percent of the IDSs are single-attack solutions (Fig. 8). As a result, 22.7 percent of them appear to be robust, since they claim to cope with five or more attacks; among them, only the Specification-Based IDS [13] and ELNIDS [49] are evaluated for all the attacks they investigate. Despite these relatively low scores, a significant number of IDSs (almost 73 percent) claim that they are extendable and able to detect and mitigate more attacks, once they are modified. Unexpectedly, we notice that robustness is not necessarily associated with a low overhead cost, i.e., three out of five robust systems present low overhead [13, 40, 42], while two of them [13, 40] also combine robustness with low false detection. These findings indicate that research towards balancing the trade-off among security (expressed with robustness and extendability), performance (in terms of low false positives, scalability, and RPL compliance), and cost (associated with low overhead) can bring fruitful results.

Finally, an insightful outcome of Table 3 is that 77 percent of IDSs do not consider the mobility issue, probably due to the difficulties that it entails. We demonstrate, for example, on Fig. 5 and 6 that nodes' mobility causes control overhead comparable to some attacks, e.g., decreased rank and blackhole attack;

this could mislead the decision-making of an IDS with impact on false positives' rate. Indeed, IDSs that deal with sinkhole [13, 40, 41, 43, 44, 49, 50, 54, 55, 59, 61], wormhole [25, 40, 41, 43, 44] and rank attacks [13, 44, 62], mishandle nodes' mobility and interpret it as an attack pattern (since, for example, mobile nodes send control messages from different network places and in irregular intervals compared to the fixed ones). In addition, mobility patterns can be known a priori (e.g., a city-bus, with IoT nodes on it, follows the same route every day) or completely random; in the latter case, even probabilistic or machine-learning models face accuracy issues in predicting nodes' status and, thus, providing appropriate input to an IDS.

These observations make clear that an IDS should monitor and evaluate a number of parameters in conjunction to each other in order to combine high accuracy with low false positives.

6.3. Guidelines

So far, it is clear that there is no one-for-all solution that mitigates a great portion of the RPL-related attacks and, at the same time, meets all the design requirements we introduced. As aftermath, we present here some basic guidelines for an up-to-date IDS.

- *Trade-off between security and performance*: This notice reflects the need for robust and extendable systems while simultaneously presenting high accuracy and ability to operate regardless of the network's scale and be compliant with the RPL to preserve the protocol's native performance. Table 3 shows that only [13, 40] are robust systems and at the same time satisfy the criteria i, ii and vi. Thus, there is room for research and improvements, especially if we consider that out of 21 different RPL-related attacks, a critical portion of the IDSs, 77 percent, deal with up to only four of them. Furthermore, current literature lacks proposals that cope with certain attacks, such as routing table overload and falsification, increased rank, and worst parent selection. Simultaneously, the built-in security mechanisms of RPL have not been thoroughly

investigated and are considered optional features in the RPL specification. Their implementation and further research on their effectiveness against the various attacks may bring positive results for the trade-off between security and performance.

- *Trade-off between security and cost*: Designing security systems for LLNs should take the cost as a primary concern. The fact that 63 percent of IDSs do not satisfy the low overhead and robustness criteria simultaneously, and 27 percent do not provide any cost-related results indicates that current research underestimates this issue. Of course, a high level of security entails cost barriers. However, three systems [13, 40, 42] are robust and entail low overhead simultaneously, while [13] exhibits the best behavior in respect to all the requirements defined. Probably the last seven years are a trial period during which many ideas and approaches are under investigation. Fortunately, the above IDSs provide evidence that we gain knowledge and invest in holistic solutions that combine security, performance, and cost.

- *Mobility support*: Mobility is a trend of modern IoT networks and, among others, contributes to widening the networks' range deployment. Current IDSs' literature is not mature enough to provide solutions that deal with this issue efficiently, i.e., to combine it with robustness and low false positives' rates. In fact, mobility is the least satisfied among our defined requirements. Previously in this section, we justified this weakness, which definitely provides room for research, especially in the light of results and solutions regarding the RPL under mobility [3, 4, 36]. Both from our previous experience [3, 4, 36] and from the systems that support mobility [40, 42, 55], we conclude that hybrid solutions regarding the detection method and/or the placement strategy could efficiently contribute to building efficient IDSs.

- *Alignment to the IoT evolution*: IoT advances towards supporting applications with diverse, challenging requirements, e.g., ultra-low

delays, mobility, or high capacity of nodes, through exploiting Edge Cloud Computing, Software-Defined Networks (SDN) and 5G or Beyond Networks. In this complex ecosystem, new critical IoT installations (e.g., Industry 4.0 or Smart-city) come together with new sophisticated attacks. Consequently, an up-to-date IDS should be extendable, able to tune security/cost and security/performance trade-offs to particular IoT applications, and benefit from such advanced networking, processing, and storage capabilities. For example, Edge Clouds' incorporation brings significant processing and storage resources that can support Artificial Intelligence/Machine-Learning (AI/ML) capabilities, e.g., for data analysis, clustering, or prediction. Such features perfectly match with RPL extensions inspired by the SDN paradigm [3, 4, 36] that enables modularity, adaptation, and dynamicity; e.g., to jointly recognize mobility patterns, detect, and mitigate unknown attacks.

The hybrid approaches are consistent to the above direction since their centralized mechanisms can be driven by intelligent mechanisms deployed at Edge Clouds, their decisions enforced by SDN controllers. Simultaneously, the nodes are assigned with lightweight tasks, such as local monitoring and/or low-complexity algorithms, i.e., for instantaneous reporting or acting upon attacks.

7. Conclusion

The RPL routing protocol is a relatively mature technology that allows IPv6 routing in LLNs. By investigating RPL attacks with special attention on their impact in terms of control overhead and application performance, and evaluating the related IDSs in the literature, we conclude that there is room for research regarding holistic solutions with specific tailored-made characteristics, such as: monitoring and exploiting several features in conjunction, e.g., network conditions and protocols' mechanisms, handling mobility, respecting resource constraints, while at the same time providing a

high level of security reflected in robustness and low false positives. We introduce seven design requirements that a modern RPL-related IDS should satisfy. Moreover, we provide a list of four concrete guidelines that, according to our experience, future approaches should take into consideration. In fact, we are currently working on an SDN-inspired, machine-learning-based polymorphic IDS that exploits our findings and brings promising results.

References

[1] T. Winter, P. Thubert, A. Brandt, et al., RPL: IPv6 routing protocol for low-power and lossy networks, IETF RFC 6550 (2012).
URL https://tools.ietf.org/html/rfc6550

[2] O. Gaddour, A. Koubâa, RPL in a nutshell: A survey, Computer Networks 56 (14) (2012) 3163–3178. doi:10.1016/j.comnet.2012.06.016.

[3] G. Violettas, S. Petridou, L. Mamatas, Evolutionary software defined networking-inspired routing control strategies for the Internet of Things, IEEE Access 7 (2019) 132173–132192. doi:10.1109/ACCESS.2019.2940465.

[4] G. Violettas, S. Petridou, L. Mamatas, Routing under heterogeneity and mobility for the Internet of Things: a centralized control approach, in: Global Communications Conference (GLOBECOM), 2018 IEEE Conf. on, IEEE, 2018, pp. 1–7.

[5] A. Mayzaud, R. Badonnel, I. Chrisment, A Taxonomy of Attacks in RPL-based Internet of Things, International Journal of Network Security (2016). doi:10.6633/IJNS.201605.18(3).07.

[6] A. Verma, V. Ranga, Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review, IEEE Sensors Journal 20 (11) (2020) 5666–5690. doi:10.1109/JSEN.2020.2973677.

[7] A. Arena, P. Perazzo, C. Vallati, G. Dini, G. Anastasi, Evaluating and improving the scalability of RPL security in the Internet of Things, Computer Communications (2020). doi:10.1016/j.comcom.2019.12.062.

[8] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, G. Dini, An implementation and evaluation of the security features of rpl, in: International Conference on Ad-Hoc Networks and Wireless, Springer, 2017, pp. 63–76.

[9] P. O. Kamgueu, E. Nataf, T. D. Ndie, Survey on RPL enhancements: a focus on topology, security and mobility, Computer Communications 120 (2018) 10–21. doi:10.1016/j.comcom.2018.02.011.

[10] J. Granjal, E. Monteiro, J. S. Silva, Security for the internet of things: a survey of existing protocols and open research issues, IEEE Communications Surveys & Tutorials 17 (3) (2015) 1294–1312. doi:10.1109/COMST.2015.2388550.

[11] A. Raoof, A. Matrawy, C.-H. Lung, Routing attacks and mitigation methods for RPL-based internet of things, IEEE Communications Surveys & Tutorials 21 (2) (2018) 1582–1606. doi:10.1109/COMST.2018.2885894.

[12] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the rpl-based internet of things, International Journal of Distributed Sensor Networks 9 (8) (2013) 794326.

[13] A. Le, J. Loo, K. K. Chai, M. Aiash, A specification-based IDS for detecting attacks on RPL-based network topology, Information 7 (2) (2016) 25. doi:10.3390/info7020025.

[14] A. Mayzaud, R. Badonnel, I. Chrisment, A distributed monitoring strategy for detecting version number attacks in RPL-based networks, IEEE Transactions on Network and Service Management 14 (2) (2017) 472–486. doi:10.1109/TNSM.2017.2705290.

[15] A. Kamble, V. S. Malemath, D. Patil, Security attacks and secure routing protocols in rpl-based internet of things: Survey, in: 2017 International Conference on Emerging Trends Innovation in ICT (ICEI), 2017, pp. 33–39. doi:10.1109/ETIICT.2017.7977006.

[16] A. Dunkels, B. Gronvall, T. Voigt, Contiki-a lightweight and flexible operating system for tiny networked sensors, in: 29th annual IEEE international conference on local computer networks, IEEE, 2004, pp. 455–462. doi:10.1109/LCN.2004.38.

[17] T. Tsvetkov, A. Klein, RPL: IPv6 routing protocol for low power and lossy networks, Network 59 (2011) 59–66.

[18] A. Le, J. Loo, Y. Luo, A. Lasebae, The impacts of internal threats towards routing protocol for low power and lossy network performance, in: 2013 IEEE Symposium on Computers and Communications (ISCC), 2013, pp. 000789–000794. doi:10.1109/ISCC.2013.6755045.

[19] P. Pongle, G. Chavan, A survey: Attacks on RPL and 6LoWPAN in IoT, in: 2015 International conference on pervasive computing (ICPC), IEEE, 2015, pp. 1–6. doi:10.1109/PERVASIVE.2015.7087034.

[20] A. Sehgal, A. Mayzaud, R. Badonnel, I. Chrisment, J. Schönwälder, Addressing DODAG inconsistency attacks in RPL networks, in: Proceedings of Global Information Infrastructure and Networking Symposium (GIIS), IEEE, 2014, pp. 1–8. doi:10.1109/GIIS.2014.6934253.

[21] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, M. Chai, The impact of rank attack on network topology of routing protocol for low-power and lossy networks, in: IEEE Sensors Journal, Vol. 13, IEEE, 2013, pp. 3685–3692. doi:10.1109/JSEN.2013.2266399.

[22] A. Aris, S. F. Oktug, S. Berna Ors Yalcin, Rpl version number attacks: In-depth study, in: NOMS 2016 - 2016 IEEE/IFIP Network Operations and

Management Symposium, 2016, pp. 776–779. doi:10.1109/NOMS.2016.7502897.

[23] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, J. Schönwälder, A study of rpl dodag version attacks, in: Monitoring and Securing Virtualized Networks and Services, LNCS, volume 8508, Springer, 2014, pp. 92–104. doi:10.1007/978-3-662-43862-6.

[24] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the rpl-based internet of things, International Journal of Distributed Sensor Networks 9 (8) (2013). doi:10.1155/2013/794326.

[25] P. Pongle, G. Chavan, Real Time Intrusion and Wormhole Attack Detection in Internet of Things, International Journal of Computer Applications 975 (July, 2015) 8887. doi:10.5120/21565-4589.

[26] D. Airehrour, J. Gutierrez, S. K. Ray, Secure routing for Internet of Things: A survey, Journal of Network and Computer Applications 66 (2016) 198–213. doi:10.1016/j.jnca.2016.03.006.

[27] D. Sharma, I. Mishra, S. Jain, A detailed classification of routing attacks against RPL in Internet of Things, International Journal of Advance Research, Ideas and Innovations in Technology 3 (2017) 692–703.

[28] P. Perazzo, C. Vallati, G. Anastasi, G. Dini, DIO suppression attack against routing in the Internet of Things, IEEE Communications Letters 21 (2017) 2524–2527. doi:10.1109/LCOMM.2017.2738629.

[29] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, Routing Loops in DAG-Based Low Power and Lossy Networks, in: 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 888–895. doi:10.1109/AINA.2010.126.

[30] K. Chugh, A. Lasebae, J. Loo, Case Study of a Black Hole Attack on 6LoWPAN-RPL, in: Proc. of the Sixth International

Conference on Emerging Security Information, Systems and Technologies (SECURWARE), Rome, Italy (August 2012), 2012, pp. 157–162.

[31] A. Kumar, R. Matam, S. Shukla, Impact of packet dropping attacks on rpl, in: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2016, pp. 694–698. doi:10.1109/PDGC.2016.7913211.

[32] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the internet of things, IEEE Internet of Things Journal 1 (5) (2014) 372–383. doi:10.1109/JIOT.2014.2344013.

[33] F. Medjek, D. Tandjaoui, M. R. Abdmeziem, N. Djedjig, Analytical evaluation of the impacts of sybil attacks against rpl under mobility, in: 2015 12th International Symposium on Programming and Systems (ISPS), 2015, pp. 1–9. doi:10.1109/ISPS.2015.7244960.

[34] A. Verma, V. Ranga, Security of rpl based 6lowpan networks in the internet of things: A review, IEEE Sensors Journal 20 (11) (2020) 5666–5690. doi:10.1109/JSEN.2020.2973677.

[35] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, M. Richardson, A security threat analysis for the routing protocol for low-power and lossy networks (RPLs), RFC 7416 (2015) 131.
URL https://tools.ietf.org/html/rfc7416

[36] T. Theodorou, G. Violettas, P. Valsamas, S. Petridou, L. Mamatas, A Multi-Protocol Software-Defined Networking Solution for the Internet of Things, IEEE Communications Magazine 57 (10) (2019) 42–48. doi:10.1109/MCOM.001.1900056.

[37] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, S. C. de Alvarenga, A survey of intrusion detection in Internet of Things, Journal of Network and Computer Applications 84 (2017) 25–37. doi:10.1016/j.jnca.2017.02.009.

[38] B. Lokesak, A Comparison Between Signature Based and Anomaly Based Intrusion Detection Systems, PPT). www. iup. edu (2008).
URL http://www.iup.edu/WorkArea/DownloadAsset.aspx?id=81109

[39] R. A. Sadek, M. S. Soliman, H. S. Elsayed, Effective anomaly intrusion detection system based on neural network with indicator variable and rough set reduction, International Journal of Computer Science Issues (IJCSI) 10 (6) (2013) 227.

[40] H. Sedjelmaci, S. M. Senouci, T. Taleb, An accurate security game for low-resource IoT devices, IEEE Transactions on Vehicular Technology 66 (10) (2017) 9381–9393. doi:10.1109/TVT.2017.2701551.

[41] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, I. Ahmedy, Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol, IEEE Access 6 (2018) 16623–16638. doi:10.1109/ACCESS.2018.2798626.

[42] J. Kaur, An Ultimate Approach of Mitigating Attacks in RPL Based Low Power Lossy Networks, Proceedings of 17th International Conference on Security and Management (SAM'19) (2019).

[43] S. Raza, L. Wallgren, T. Voigt, Svelte: Real-time intrusion detection in the internet of things, Ad hoc networks 11 (8) (2013) 2661–2674. doi:10.1016/j.adhoc.2013.04.014.

[44] H. Bostani, M. Sheikhan, Hybrid of Anomaly-Based and Specification-Based IDS for Internet of Things Using Unsupervised OPF Based on MapReduce Approach, Computer Communications (2016) 52–71doi:10.1016/j.comcom.2016.12.001.

[45] S. Dharmapurikar, J. W. Lockwood, Fast and Scalable Pattern Matching for Network Intrusion Detection Systems, IEEE Journal on Selected Areas in Communications 24 (10) (2006) 1781–1792.

[46] E. Aydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsström, M. Gidlund, A Central Intrusion Detection System for RPL-Based Industrial Internet of Things, in: 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), IEEE, 2019, pp. 1–5. doi:10.1109/WFCS.2019.8758024.

[47] S. M. Othman, N. T. Alsohybe, F. M. Ba-Alwi, A. T. Zahary, Survey on Intrusion Detection System Types, International Journal of Cyber-Security and Digital Forensics 7 (4) (2018) 444–463.

[48] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, M. A. Spirito, An IDS framework for internet of things empowered by 6LoWPAN, in: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 1337–1340. doi:10.1145/2508859.2512494.

[49] A. Verma, V. 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), IEEE, 2019, pp. 1–6. doi:10.1109/IoT-SIU.2019.8777504.

[50] P. Ioulianou, V. Vasilakis, Denial-of-Service Attacks and Countermeasures in the RPL-Based Internet of Things, Katsikas S. et al. (eds) Computer Security. CyberICPS 2019, SECPRE 2019, SPOSE 2019, ADIoT 2019 11980 (2020) 374–390. doi:10.1007/978-3-030-42048-2_24.

[51] S. Deshmukh-Bhosale, S. S. Sonavane, A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things, Procedia Manufacturing 32 (2019) 840–847, 12th International Conference Interdisciplinarity in Engineering, INTER-ENG 2018, 4–5 October 2018, Tirgu Mures, Romania. doi:10.1016/j.promfg.2019.02.292.

[52] P. Ioulianou, V. Vasilakis, I. Moscholios, M. Logothetis, A signature-based

intrusion detection system for the Internet of Things, Information and Communication Technology Form (2018).

[53] A. Verma, V. Ranga, CoSec-RPL: detection of copycat attacks in RPL based 6LoWPANs using outlier analysis, Telecommunication Systems (2020). doi:10.1007/s11235-020-00674-w.

[54] M. Surendar, A. Umamakeswari, InDReS: An Intrusion Detection and response system for Internet of Things with 6LoWPAN, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE, 2016, pp. 1903–1908. doi:10.1109/WiSPNET.2016.7566473.

[55] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015, pp. 606–611. doi:10.1109/INM.2015.7140344.

[56] F. Gara, L. B. Saad, R. B. Ayed, An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs, in: 13th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2017, pp. 276–281. doi:10.1109/IWCMC.2017.7986299.

[57] K. Sentz, S. Ferson, K. Sentz, Combination of evidence in dempster-shafer theory, Tech. rep., US Department of Energy (US) (2002). doi:10.2172/800792.

[58] Barnett, V. and Lewis, T., Outliers in statistical data. 3rd edition., Vol. 37, J. Wiley & Sons, 1994. doi:10.1002/bimj.4710370219.

[59] E. Kfoury, J. Saab, P. Younes, R. Achkar, A Self Organizing Map Intrusion Detection System for RPL Protocol Attacks, International Journal of Interdisciplinary Telecommunications and Networking (IJITN) 11 (1) (2019) 30–43. doi:10.4018/IJITN.2019010103.

[60] A. Nikam, D. Ambawade, Opinion Metric Based Intrusion Detection Mechanism for RPL Protocol in IoT, in: 3rd International Conference for Convergence in Technology (I2CT), IEEE, 2018, pp. 1–6. doi:10.1109/I2CT.2018.8529770.

[61] F. Nygaard, Intrusion detection system in IoT, Master's thesis, NTNU (2017).

[62] L. Zhang, G. Feng, S. Qin, Intrusion detection system for RPL from routing choice intrusion, in: 2015 IEEE International Conference on Communication Workshop (ICCW), IEEE, 2015, pp. 2652–2658. doi:10.1109/ICCW.2015.7247579.

[63] U. Shafique, A. Khan, A. Rehman, F. Bashir, M. Alam, Detection of rank attack in routing protocol for Low Power and Lossy Networks, Annals of Telecommunications (2018). doi:73.10.1007/s12243-018-0645-4.

[64] F. Ahmed, Y.-B. Ko, A Distributed and Cooperative Verification Mechanism to Defend against DODAG Version Number Attack in RPL, in: PECCS, 2016, pp. 55–62. doi:10.5220/0005930000550062.

[65] H. Svensson, A. Jøsang, Correlation of intrusion alarms with subjective logic, in: Proceedings of the sixth Nordic Workshop on Secure IT systems (NordSec2001), Copenhagen, Denmark, Citeseer, 2001.

[66] T. Matsunaga, K. Toyoda, I. Sasase, Low false alarm attackers detection in rpl by considering timing inconstancy between the rank measurements, IEICE Communications Express 4 (2) (2015) 44–49. doi:10.1587/comex.4.44.

[67] D. Shreenivas, S. Raza, T. Voigt, Intrusion detection in the RPL-connected 6LoWPAN networks, in: Proceedings of the 3rd ACM international workshop on IoT privacy, trust, and security, 2017, pp. 31–38. doi:10.1145/3055245.3055252.

[68] L. Rocha, F. Cappabianco, A. Falcão, Data clustering as an optimum-path forest problem with applications in image analysis, International Journal of Imaging Systems and Technology 19 (2009) 50 – 68. doi:10.1002/ima.20191.

[69] S. Sahu, B. M. Mehtre, Network intrusion detection system using j48 decision tree, in: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2015, pp. 2023–2026. doi:10.1109/ICACCI.2015.7275914.

[70] T. Wauters, et al., Federation of internet experimentation facilities: architecture and implementation Federation of internet experimentation facilities: architecture and implementation, in: European Conf. on Networks and Communications (EuCNC) 2014, IEEE, pp. 1–5.

[71] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, I. Seskar, GENI: A federated testbed for innovative network experiments, Computer Networks 61 (2014) 5–23.

[72] P. Valsamas, P. Papadimitriou, I. Sakellariou, S. Petridou, L. Mamatas, S. Clayman, F. Tusa, A. Galis, Multi-PoP Network Slice Deployment: A Feasibility Study, in: 2019 IEEE 8th International Conference on Cloud Networking (CloudNet), IEEE, 2019, pp. 1–6.

[73] P. D. Maciel, F. L. Verdi, P. Valsamas, I. Sakellariou, L. Mamatas, S. Petridou, P. Papadimitriou, D. Moura, A. I. Swapna, B. Pinheiro, et al., A marketplace-based approach to cloud network slice composition across multiple domains, in: 2019 IEEE Conference on Network Softwarization (NetSoft), IEEE, 2019, pp. 480–488.

[74] P. Valsamas, I. Sakellariou, S. Petridou, L. Mamatas, A Multi-domain Experimentation Environment for 5G Media Verticals, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 461–466.

[75] Flux Research Group, The University of Utah, https://www.flux.utah.edu/index (2020).