DETONAR: Detection of Routing Attacks in RPL-based IoT

DETONAR: Detection of Routing Attacks in RPL-based IoT

Andrea Agiollo*, Mauro Conti†, Pallavi Kaliyar†, TsungNan Lin‡, and Luca Pajola†
*Department of Information Engineering, University of Padua
†Department of Mathematics, University of Padua
‡Department of Electrical Engineering, National Taiwan University

***Abstract*—The Internet of Things (IoT) is a reality that changes several aspects of our daily life, from smart home monitoring to the management of critical infrastructure. The "Routing Protocol for low power and Lossy networks" (RPL) is the only de-facto standardized routing protocol in IoT networks and is thus deployed in environmental monitoring, healthcare, smart building, and many other IoT applications. In literature, we can find several attacks aiming to affect and disrupt RPL-based networks. Therefore, it is fundamental to develop security mechanisms that detect and mitigate any potential attack in RPL-based networks. Current state-of-the-art security solutions deal with very few attacks while introducing heavy mechanisms at the expense of IoT devices and the overall network performance.**

In this work, we aim to develop an Intrusion Detection System (IDS) capable of dealing with multiple attacks while avoiding any RPL overhead. The proposed system is called DETONAR - DETector of rOutiNg Attacks in Rpl - and it relies on a packet sniffing approach. DETONAR uses a combination of signature and anomaly-based rules to identify any malicious behavior in the traffic (e.g., application and DIO packets). To the best of our knowledge, there are no exhaustive datasets containing RPL traffic for a vast range of attacks. To overcome this issue and evaluate our IDS, we propose RADAR - Routing Attacks DAtaset for Rpl: the dataset contains five simulations for each of the 14 considered attacks in 16 static-nodes networks. DETONAR's attack detection exceeds 80% for 10 attacks out of 14, while maintaining false positives close to zero.

***Index Terms*—Internet of Things, Low Power and Lossy Networks, Routing Protocol, Networking attacks, Intrusion Detection System.**

I. Introduction

THE revolution of the internet world is happening in recent years, bringing the name of the Internet of Things (IoT). Initially studied as the evolution of Wireless Sensors Networks (WSNs) [1], recently IoT has gained much popularity and scientific community attention. The new paradigm introduced by IoT has shown the broadest range of applications, from industrial scenarios [2], [3], to smart homes [4], [5], intelligent healthcare [6], and smart cities [7].

The importance of Internet of Things (IoT) applications introduces the need to secure IoT networks [8], [9]. Indeed, many examples of attacks and vulnerabilities can be found for these networks. For instance, in October 2016, the largest Distributed-Denial-of-Service (DDoS) attack was launched using an IoT botnet. This specific attack leveraged the Mirai malware [10]. In 2017, a study developed by the US Food and Drug Administration confirmed that some cardiac devices present serious vulnerabilities that, if exploited, could allow unauthorized access to the devices [11].

There exists a broad variety of communication protocols commonly used in IoT networks, e.g., WiFi [12], IEEE 802.15.4 [13], RFID [14], Bluetooth [15]. Depending on the considered devices, the surrounding environment, and the required communication range, different protocols are leveraged. In this paper, we consider the standardized Routing Protocol for Low Power Lossy Networks (RPL). RPL is considered the *de facto* routing protocol for IoT and can be efficiently used in different applications, including but not limited to healthcare, smart environments, transport, industry, and military applications [16]–[18]. RPL is a proactive protocol developed to allow communication in wireless networks with low power consumption and generally susceptible to packet loss. In particular, we inspect the identified vulnerabilities of RPL. In recent years, research efforts shed light upon routing attacks against RPL (e.g. rank attack, version attack, etc.) [19]–[22].

Given the wide variety of attacks available against this protocol, its secure deployment is difficult. Moreover, RPL's popularity in IoT applications renders the security problem of this protocol of paramount importance. In the recent past, some works have been proposed aiming at securing RPL with alternate success [23]–[29]. Although showing acceptable detection performances, state-of-the-art security systems focus only on few attacks while introducing communication overhead and computation performance issues. Therefore, there is a lack of a reliable and comprehensive IDS that can identify more attacks with low overhead. To this end, we propose DETONAR[1], an Intrusion Detection System used to DETect rOutiNg Attacks in RPL. DETONAR leverages traffic analysis technique. Only a few works explore this concept in RPL security, such as [30], [31]. However, these works focus only on the identification of a few attacks. The proposed IDS aims to identify the maximum number of attacks while introducing zero RPL overhead and zero computations at IoT devices. DETONAR leverages traffic sniffer devices and traffic analysis techniques in order to build a centralized IDS. The DETONAR detection scheme's centrality allows it to overcome state-of-the-art drawbacks like RPL overhead and computational power requirements of IoT devices. Moreover, DETONAR introduces a hybrid approach relying on the combination of anomaly-

[1]DETONAR is a Venetian dialectal word that stands for detonating a bomb, as the proposed IDS aims at detonating any possible attacks against RPL.

based detection and signature-based classification techniques. Combining these techniques allows DETONAR to reach reliable attack detection while maintaining low false positives.

Given the unique nature of DETONAR and its use of network traffic analysis techniques, we need to introduce a novel dataset of attacks against RPL, called RADAR, to evaluate its performances. We consider 14 well-known routing attacks against RPL and use the NetSim tool [32] to implement these attacks and extract the corresponding dataset (see Section III). The dataset consists of the packet trace files for each simulation. These files contain the packets that each device has sent during the communication period. The simulation implementation considers only static scenarios, given their popularity among RPL's applications and medium-sized networks (i.e. 16 nodes). To the best of our knowledge RADAR represents the biggest and most comprehensive dataset of routing attacks in RPL. Its introduction represents a step forward in security research as it allows to study a wide variety of attacks simultaneously.

a) Contribution: To summarize, the contributions that our work brings to IoT security are the following:

- We present *RADAR*, a novel Routing Attacks Dataset for RPL. To the best of our knowledge, RADAR represents the first dataset containing RPL traffic for a vast range of attacks. The dataset (see Section III for more details) contains network traffic of simulations for 14 well known attacks. Thanks to NetSim, we make the dataset publicly available[2]. RADAR contains 80 different simulations, each of length 1500 seconds, obtaining on average more than a million packets for each simulation.
- We present *DETONAR*, a novel IDS developed to detect routing attacks in RPL and identify intruders. Due to its novel sniffing approach and the centralized computation paradigm, DETONAR (see Section IV for more details) maintains zero communication overhead at RPL level and requires no device computations or firmware update.
- We show the effectiveness of DETONAR on the proposed RADAR dataset and shows its applicability to small-scale networks. DETONAR's true positive detection exceeds 80% for 10 attacks out of 14 (see Section V-B for more details), while requiring relatively small computation time, due to its hybrid approach.

b) Organization: The next sections of this paper are organized as follows. Section II discusses the basic concepts of RPL, also presenting the known attacks against it in IoT networks. Section III presents RADAR in details, explaining its most relevant features. Section IV describes the proposed IDS, its workflow, and the main advantages that it brings. Section V presents the attack detection parameters optimization process and the performance analysis of our proposed IDS over different IoT scenarios. Finally, Section VI provides conclusions and insight of possible extensions of our work.

II. Background and Related Work

In this section, we briefly introduce the routing protocol for Low Power Lossy Networks (LLNs), called Routing Protocol for Low Power Lossy Networks (RPL) (in Section II-A). Section II-B summarizes well-known networking attacks against RPL and their workflow. Finally, in Section II-C we present the available IDSs and their limitations, motivating our work.

A. Routing Protocol for LLNs (RPL)

RPL is the routing protocol standard for LLNs, developed by the IETF ROLL task force and defined in the Request For Comment (RFC) 6550 [33]. IEEE commissioned the development of RPL to bridge the existing gap in routing for IoT scenarios, designed to meet the requirements of resource, power, and bandwidth constrained devices.

The fundamental concept standing at the base of RPL is the topological notion of *Destination Oriented Directed Acyclic Graphs* (DODAGs). The DODAG is a directed graph-oriented towards a root node without loops. The nodes providing access to the internet (gateways) are considered root nodes. All other nodes link to them directly or through a series of parent nodes. Each node selects a preferred parent who forwards application packets. It is selected depending on the *rank* value that a device can obtain. This value represents the position of a node in the DODAG. *Rank* depends on both the distance of the node from the root and the *Objective Function* (OF). It describes how distance and signal-to-noise ratio are used to compute rank.

RPL introduces new control packets and leverages them to build and maintain the DODAG and the communication routes. RPL control packets are defined as a type of Internet Control Messages Protocol version 6 (ICMPv6) control packets. In particular, (i) *DODAG Information Solicitation* (DIS) packet is used by a node to ask for neighborhood information, (ii) *DODAG Information Object* (DIO) packet is used to discover RPL instances, learn DODAG configurations, select preferred parent, and maintain DODAG structure, (iiii) *Destination Advertisement Object* (DAO) packet is used to advertise reverse routes information, creating upward and downward paths between the nodes, and (iv) *Destination Advertisement Object Acknowledgement* (DAO-ACK) used to reply to a DAO packet.

RPL supports different communication paradigms that include Point-to-multipoint (P2MP), point-to-point (P2P), and multipoint-to-point (MP2P). It also provides two *modes of operation*. In *storing* mode, the preferred parent will store routing information in a routing table. In this mode, application packets reach the closest common parent before being redirected to the destination. In *non-storing* mode, the root node is the only device that maintains a routing table. In this mode, application packets reach the root node before being redirected to the destination. A full explanation of RPL implementation details is out of the scope of this paper. Interested readers may find more comprehensive literature on RPL in [33], [34].

B. Attacks on RPL

There exist many networking attacks against RPL. Even if traditional security protocols are implemented (e.g., IPSec [35], SSL [36], etc.), RPL does not guarantee communication and routing security. A malicious user can take possession of a device, modify the DODAG structure, or block application packets. Mayzaud et al. [37] gave a taxonomy in

[2]https://spritz.math.unipd.it/projects/detonar/

order to classify attack against RPL protocol. We suggest [19]–[21] for an in-depth overview of RPL networking attacks. To the best of our knowledge, there exist 16 well-known attacks, each of them presenting specific behaviors.

- *Blackhole and Selective Forward attacks.* The attacker may drop all (*blackhole*) or some (*selective forward*) application packets received from its children. *Goal*: Denial-of-Service.
- *Sinkhole, Rank, and Continuous Sinkhole attacks.* A malicious user can fake its rank value, modifying or disrupting network structure. *Goal*: DODAG modification.
- *HELLO flooding, and DIS attacks.* A malicious device forges and sends a high amount of control packets (DIO or DIS). Forged messages keep the neighbors busy trying to process them. *Goal*: Network flooding.
- *Clone ID and Sybil attacks.* An attacker can advise himself as one or multiple different devices, stealing the identity of a legitimate node. *Goal*: Eavesdropping.
- *Wormhole attack.* Multiple attackers can collaborate to create a tunnel. The created tunnel allows the two malicious nodes to intercept and divert many applications packets. *Goal*: Routes modification, network disruption.
- *Version, Local repair, DODAG inconsistency and Storing mode attacks.* An attacker can forge modified control packets containing anomalous parameters. These packets provide to disrupt communication. *Goal*: DODAG disruption.
- *Replay attack.* A malicious user can replay previously received control packets. Settings inconsistencies make neighbours unable to communicate. *Goal*: DODAG disruption.
- *Worst Parent attack.* An attacker can select a new parent without changing its rank. The new parent is chosen to be the worst possible, creating sub-optimal paths. *Goal*: DODAG modification.
- *DODAG Inconsistency attack.* An attacker can misuse RPL's DODAG repair mechanism to attack the network. Manipulation of few packets' flags can trigger a DODAG repair mechanism, making it impossible for devices to communicate properly. *Goal*: DODAG disruption.
- *Storing Mode attack.* This attack requires RPL to run in storing mode. An attacker can advise many non existing routes to a legitimate device. The advised routes saturate the routing table of the compromised device, preventing it from building correct routes. *Goal*: Routes disruption.

Table I shows the characteristics of each well-known routing attack against RPL. In particular, we consider if a strategy is influencing or disrupting the DODAG structure. We also study if an attack behavior increases the end-to-end delay due to long queues created at each device or by producing sub-optimal routes. Packet reception may also be influenced due to high packet drop rate or communication overhead. Finally, we also consider if collaboration between attackers or packet forgery are required for a specific attack.

C. State-of-the-art Intrusion Detection Systems

Knowing the importance of security issues in RPL, many systems have been proposed to patch this protocol. Intrusion Detection Systems (IDSs) are the most popular mechanisms to detect security threats (e.g., intruders) in a network [38]–[40]. In particular, *anomaly-based* IDS aims at identifying non-legitimate behavior, knowing how the network works when no attacker is present. These systems can work adequately knowing legitimate network traffic only. On the other hand, any fluctuation from legitimate behavior is considered an anomaly. Therefore, anomaly-based systems can be characterized by high false positives. *Signature-based* IDS instead utilizes signatures of attack behavior to identify intruders. These systems are capable of obtaining low false positives but are not flexible. Signatures are found for specific attack patterns, requiring their full knowledge.

Raza et al. [41] propose a system based on report packets containing network information sent by IoT devices to the root node upon request, to secure RPL. The root node reports are then used to reconstruct the DODAG and find anomalies in its structure. DODAG and network information are used to detect sinkhole, blackhole, and selective forward attacks. In [23], the authors present a system based on device location knowledge to identify wormhole attacks. Cervantes et al. [42] propose a modification of the RPL protocol to detect sinkhole attacks by using an IDS mechanism that is based on network clusterization. An extension of [42] is introduced by Surendar et al. [24]. The authors aim to decrease overhead and increase the packet delivery ratio. Gara et al. [25] focus on mobile Wireless Sensor Networks (WSNs), in which they try to identify possible selective forward attacks. The authors in [26] propose an IDS that is based on game-theory strategies, aiming to decrease false positives, energy consumption, and overhead. A trust-based security mechanism is presented by Airehrour et al. [27] and tested against rank and sybil attacks. The authors in [30] propose a deep learning framework based on traffic analysis to detect rank, hello flooding, and version attacks. Finally, Mayzaud et al. [43] present a distributed security mechanism based on monitoring devices capable of sniffing network traffic to detect version attacks.

The state-of-the-art systems show different possibilities to secure RPL based networks. Although showing some advantages, these mechanisms present the following drawbacks:

1) *Scarn variety of attacks*. Existing IDSs can detect a few classes of attacks compared to the well-known ones (see Table II).
2) *High RPL overhead*. Most of the existing systems introduce high communication overhead at RPL level, resulting inapplicable on real RPL-based IoT scenarios.
3) *Computations required at IoT devices*. Security protocols are executed inside IoT devices to secure RPL. Which is not desirable when dealing with constrained devices.

With our work, we aim to propose a comprehensive system (i.e., DETONAR) capable of protecting RPL against 14 different attacks. It represents a significant increment over state-of-the-art mechanisms. Moreover, previous works increase RPL communication overhead, which limits their real-world deployment since it leads to the disruption of communication between devices. Finally, most existing solutions require installing new protocols inside IoT devices, which adds computational requirements unbearable for most networks, as IoT devices are usually power constrained. The

TABLE I
MAIN FEATURES OF ATTACKS AGAINST RPL. THESE FEATURES HELP TO UNDERSTAND ATTACKS BEHAVIOUR AND TO CLASSIFY THEM.

Attack	DODAG	Queueing Delay	Routing Delay	Packet Loss Directly	Packet Loss via Overhead	Collaboration	Forgery
Blackhole				✓			
Selective forward				✓			
Sinkhole	✓		✓				
Continuous Sinkhole	✓		✓				
HELLO flooding		✓			✓		✓
Clone ID			✓	✓			
Sybil			✓	✓			
Wormhole			✓			✓	
Version	✓		✓		✓		✓
Replay	✓	✓	✓		✓		✓
Rank	✓		✓				
Worst parent	✓		✓				
DIS		✓			✓		✓
Local repair	✓		✓		✓		✓
DODAG inconsistency			✓		✓		✓
Storing mode		✓	✓		✓		✓

TABLE II
STATE-OF-THE-ART IDSS AND CORRESPONDING DETECTED ATTACKS. WE CAN SEE THAT SEVERAL ATTACKS ARE NOT COVERED.

Attack	[41]	[23]	[42]	[24]	[25]	[26]	[27]	[30]	[43]
Blackhole	✓					✓			
Selective forward	✓				✓				
Sinkhole	✓		✓	✓		✓			
Continuous Sinkhole									
HELLO flooding						✓		✓	
Clone ID									
Sybil						✓	✓		
Wormhole		✓				✓			
Version								✓	✓
Replay									
Rank							✓	✓	
Worst parent									
DIS									
Local repair									
DODAG inconsistency									
Storing mode									

introduction of security systems at the device level also requires the installation of software and updates. Companies that produce IoT devices would be required to introduce security firmware on their products, while final users would need to update their devices to maintain network safeness periodically. Device users usually lack security awareness, reducing the utility of state-of-the-art security systems [44]. DETONAR, on the other hand, thanks to its sniffing approach, introduces zero RPL overhead while requiring no computations or new protocols at IoT devices level. The obtained system applies to small-scale IoT networks, avoiding further consideration regarding communication reliability, power availability, and device maintenance.

III. RADAR: ROUTING ATTACKS DATASET FOR RPL

In this section, we present RADAR, a novel Routing Attacks DAtaset for RPL. To the best of our knowledge, there exist no exhaustive datasets containing RPL traffic for a vast range of routing attacks. We aim at filling this lack by proposing RADAR. This dataset represents a novelty both in terms of the variety of considered routing attacks and extracted traffic. We use Netsim [32] to implement 14 of the attacks (five simulations for each attack) presented in Section II-B. Only *DODAG inconsistency* and *storing mode* attacks are not implemented due to NetSim's software limitations. NetSim does not implement specific RPL flags that are required for these two attacks to be simulated. These flags have been introduced in later developments of RPL and are not required for its proper functioning. To the best of our knowledge, given the broad variety of attacks, RADAR represents the most significant dataset for routing attacks in RPL.

RADAR contains five simulations for each attack considered. Implemented networks deploy 16 IoT devices and a single border router belonging to a single DODAG structure. RADAR also contains ten legitimate simulations. For each simulation, NetSim stores a packet trace file containing the list of packets exchanged during the simulation. Table III shows the average amount of packets recorded during simulations of different attacks. It is possible to notice that the amount of packets recorded on average depends on the attack considered. Attacks aiming to disrupt DODAG structure or influence control flow introduce high amount of packets. On the other hand, attacks aiming at diverting traffic or steal information do not introduce heavy traffic, resulting in smaller traces.

Netsim stores the following features for each recorded packet: packet type, application name, source, destination, transmitter and receiver identities, arrival and start time for application, network, data link, and physical layers and payload size for the same layers. Source, destination, gateway, and next-hop IP addresses are also recorded along with the rank and version values for RPL control packets. All the features considered by NetSim can be extracted from un-encrypted network traffic. The un-encrypted mode is usually deployed in RPL-based networks due to its lightweight requirements and the heavy constraints of IoT devices. Instead, if encrypted version is considered, RADAR's features can be extracted by knowing security keys. It is reasonable to assume that a trusted system has the full knowledge of these keys, as it serves for the network's security, and its role can be compared to certificate authorities' role. The packet trace file's knowledge corresponds exactly to the deployment of sniffing devices which redirect network traffic to the centralized IDS server.

TABLE III
RADAR'S SIMULATED SCENARIOS AND THE CORRESPONDING AVERAGE AMOUNT OF PACKETS COLLECTED.

Scenario	Average number of packets	Average number of control packets	Average number of application packets
Legitimate	657K	624K	33K
Blackhole	1.8M	1.8M	27K
Selective forward	1.8M	1.8M	29K
Sinkhole	1.6M	1.6M	33K
Continuous Sinkhole	2.3M	2.3M	29K
HELLO flooding	1.0M	1.0M	33K
Clone ID	185K	151K	33K
Sybil	185K	151K	33K
Wormhole	257K	218K	38K
Version	2.3M	2.2M	34K
Replay	3.3M	3.2M	32K
Rank	2.3M	2.3M	34K
Worst parent	185K	152K	33K
DIS	220K	187K	32K
Local repair	2.3M	2.2M	35K

RADAR contains simulations that last 1500 seconds each. Attacks are set to start at a random time between 500 and 700 seconds, since most security systems require some attack-free calibration time. Indeed, RADAR is primarily meant and designed for IDS performance testing. We considered an interval of 500 seconds over a simulation of 1500 seconds to be long enough to satisfy most IDS's calibration time requirements. Except for the wormhole attack, one attacker was selected for each simulation. The attacker node is explicitly selected to show the effects of the attack on the network, avoiding negligible impacts. For example, a blackhole attack on a leaf node would not drop any application packet. Therefore, its significance level would be null.

To represent as precisely as possible real scenarios, IoT devices, in RADAR's simulations, send application packets periodically with a period of 1 second [2], [45]. Moreover, the path loss in RADAR's simulations follows a Friis free space path loss model with an exponent equal to 2. The free-space path model represents realistically little to moderately urbanized environments. Aforementioned realistic settings and the use of a realistic simulator like Netsim allow RADAR to represent real-world networks faithfully. The extraction of RADAR was completed using a Windows 10 machine with 64 GB of RAM and an Intel(R) Xeon(R) CPU E5-2620 v3 @2.40GHz processor. RADAR required around 400 hours of run-time to complete all simulations.

To summarize, RADAR's characteristics are the following:

- It contains packet trace files of 80 different simulations, with more than a million packets stored on average for each simulation.
- 14 well-known attacks and legitimate scenarios are simulated. Five simulations for each attack (see Section II-B) and 10 simulations for legitimate scenarios.
- Each simulation contains 16 IoT devices and a single border router that builds a single DODAG structure. The considered devices are static to recall the most common RPL real-world applications.
- Each IoT device forwards application packets with a period of one second. This setup recalls RPL real-world applications in which IoT devices periodically report information to final users.
- Each simulation lasts for 1500 seconds. In attack simulations, the malicious behavior starts randomly between the second 500 and 700.

IV. PROPOSED RPL ATTACKS DETECTOR: DETONAR

In this section, we propose DETONAR, a novel security mechanism to DETect rOutiNg Attacks in RPL. We first present an overview of the proposed mechanism in Section IV-A, followed by a detailed explanation of DETONAR's pipeline: *traffic collection* in Section IV-B, *features extraction* in Section IV-C, *anomalies detection* in Section IV-D, *attack classification* in Section IV-E and *attacker identification* in Section IV-F.

A. Overview

State-of-the-art detection mechanism introduces several challenges in RPL detector systems (see Section II-C). We now summarize a list of properties that an RPL detector system should guarantee:

- P1 *No RPL-communication overhead*. RPL networks suffer from communications' overhead (see Section II-B). The desired detection system should be an RPL-network independent entity, and it should not use RPL-communication channels.
- P2 *No RPL-nodes overhead*. The addition of operation in RPL-nodes increases their energy consumption. The desired detection system should not impact nodes' computational processes.
- P3 *Attacks Resistant* The desired detection system should face several RPL network attacks.
- P4 *Network Independent*. The desired detection system should work with different RPL network' topologies (i.e., nodes' connection, nodes' numerosity).
- P5 *Implementation flexibility*. Already existing RPL networks should integrate the detection system easily.

DETONAR's design aims to face properties P1-P5. However, in this work we do not focus on P4, while we test DETONAR only on small-size networks. We now briefly introduce DETONAR's pipeline, consisting of 5 steps, as shown in Figure 1.

1) *Traffic Sniffer* (Section IV-B). An ensemble of packet sniffers sense networks' traffic and forward it to a server. The sniffers are RPL-networks independent.
2) *Feature extraction* (Section IV-C). Extraction of a set of features describing the collected network traffic.
3) *Anomaly detection* (Section IV-D). A mechanism that analyzes nodes' traffic patterns to find potential anomalies.
4) *Attack classification* (Section IV-E). A signature-based mechanism that analyzes anomalies to identify potential attacks.
5) *Attacker Identification* (Section IV-F). A signature-based mechanism that identifies compromised nodes.

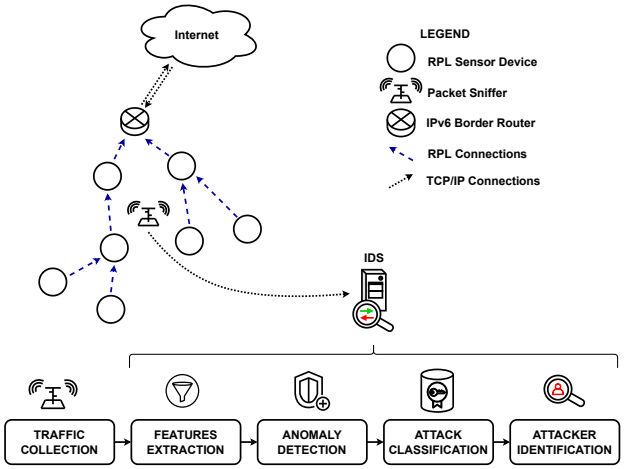


Fig. 1. DETONAR deploys sniffing devices to sense the network traffic and forward it to the centralized IDS server. IDS server is in charge of detecting possible anomalies, attacks, and compromised devices.

B. Traffic Collection

DETONAR employs sniffing devices to capture RPL networks' traffic. The optimal placement of these devices can follow [46], [47]. The collected traffic is then forwarded through secure channels (e.g., SSL) to an external server on-site or in the cloud. The transmission is conducted periodically, with time windows of a predefined size $\omega[s]$. ω is a DETONAR's hyperparameter. Information retrieved by sniffing devices contains general knowledge of packets exchanged by RPL devices. This information contains the type of packet sent (e.g., DIO, DIS, DAO, DAO-ACK, application), the address of the sender, receiver, source, and destination devices. If control packets are sniffed, additional information is considered as rank and version values.

Sniffing devices do not rely on the underlying RPL network to communicate the retrieved information with the server. Otherwise, the quality of the RPL communication would be affected by DETONAR's workflow. Passive sniffing devices can communicate securely with the external server, leveraging various communication protocols (e.g., 4G, satellite). These protocols allow the safe deployment of DETONAR, introducing reasonable costs. The design and cost of specific communications between sniffing devices and external servers are not in the scope of this paper.

The introduction of RPL-agnostic sniffing devices allows us to achieve P1 and P2. Moreover, DETONAR achieves P5 since RPL-nodes do not require any additional computational effort (e.g., software update).

C. Features Extraction

DETONAR monitors each RPL node's activities through the received traffic collected in the last time window. Formally, be $\mathcal{N}$ an RPL network with $|\mathcal{N}|$ nodes and W_i^t the traffic collected at time window t for the $i-th$ node n_i. Starting from W_i^t, DETONAR defines a set $\mathcal{F}$ of 11 features as described in Table IV. $\mathcal{F}$ describes quantitatively and qualitatively the sensed traffic. Features $f_1 - f_8$ are quantitative, as they express the amount of received or forwarded packets by each RPL device (e.g., number of forwarded/received DIO). Quantitative features give a measure of the traffic density that each node sustains. Features $f_9 - f_{11}$ are qualitative, as they express the considered node's information (e.g., rank).

TABLE IV
SELECTED FEATURES OF OUR IDS. IT IS POSSIBLE TO NOTICE THAT EACH FEATURE HELPS DETECT ONE OR MORE ATTACKS.

#	Feature	Attacks Detected
f_1	Number of DIO received	HELLO flood, Local repair, Sinkhole, Continuous Sinkhole, Rank, Replay, DIS
f_2	Number of DIO transmitted	HELLO flood, Local repair, Sinkhole, Continuous Sinkhole, Rank, Replay, DIS
f_3	Number of DAO received	Worst parent, Sinkhole, Rank, Replay, Version
f_4	Number of DAO transmitted	Worst parent, Sinkhole, Rank, Replay, Version
f_5	Number of DIS transmitted	DIS
f_6	Number of application packets received	Blackhole, Selective forward, Wormhole, Clone ID, Sybil
f_7	Number of application packets transmitted	Blackhole, Selective forward, Wormhole, Clone ID, Sybil
f_8	Transmitted vs Received applications rate	HELLO flood, DIS
f_9	Rank	Sinkhole, Continuous Sinkhole, Rank, Replay, Local Repair
f_{10}	Version	Version
f_{11}	Next hop IP	Wormhole, Worst parent

DETONAR represents W_i^t as a feature vector F_i^t

$$F_i^t = [f_1(W_i^t), ..., f_{11}(W_i^t)]. \quad (1)$$

The extracted feature vector F_i^t is finally appended in the node n_i behavioral history B_i:

$$B_i = [F_i^0, ..., F_i^t]. \quad (2)$$

Figure 2 shows the network representation schema. Each node n has its own history, with a different pattern across the various features.

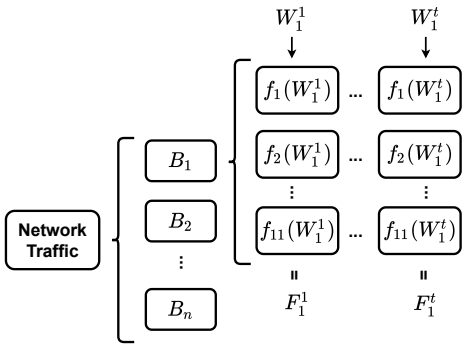


Fig. 2. DETONAR feature extraction overview. Feature vectors F_i^t are extracted for each node i at each time window W_i^t as the composition of the 11 representative features selected. Node behaviour B_i is built as the composition of feature vectors F_i^t.

Figure 3 shows an example of different patterns between features f_6 and f_2 (i.e., the number of received applications and the number of forwarded DIO) among three nodes (i.e., the root node, sensor 5, and sensor 12) at different depths of the RPL's structure in a RADAR simulation.

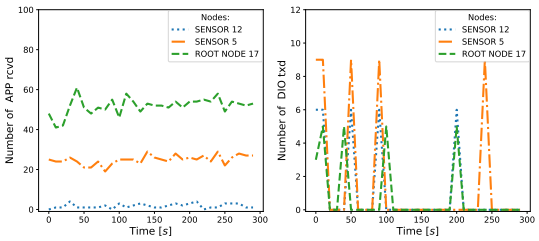


Fig. 3. DETONAR extracts features from traffic received at a node to describe device's behavior. Feature series may differ significantly depending on considered feature and device.

D. Anomalies Detection

As previously introduced, DETONAR relies on a hybrid approach: i) detection of anomalous traffic behaviors and ii) identification of the corresponding attack and compromised device(s). This section describes the anomaly detection stage, aiming to inspect each node's activities to find inconsistencies between its past and current status. This stage allows DETONAR to be resilient to different unknown attacks, as shown in [48]. With this component, DETONAR achieves P3.

Based on the extracted features in the previous step at time t (Section IV-C), DETONAR's anomaly detector inspects the minimum set of features $\mathcal{F}_\mathcal{A} \subset \mathcal{F}$ that allows identifying the presence of the 14 attacks presented in RADAR. The set $\mathcal{F}_\mathcal{A}$ corresponds to: *number of DIO received*, *number of DAO transmitted*, and *number of applications received*. The usage of a minimum set of features allows DETONAR to reduce the number of false alarms and the computational cost.

One of the challenges in RPL anomaly detection is the different traffic nature that each node has. To overcome this issue, DETONAR employs an anomaly detection algorithm $\mathbb{A}_{i,j}$ for each pair n_i, f_j in $\mathcal{N}$, $\mathcal{F}_\mathcal{A}$. Formally, given the node n_i and its node behavioral history B_i, $\mathbb{A}_{i,j}$ analyses node's f_j history $h_{i,j}$:

$$h_{i,j} = [f_j(W_i^{t-\lambda}), ..., f_j(W_i^{t-1})], \tag{3}$$

where λ indicates the history size that we consider. λ is a DETONAR's hyperparameters.

In DETONAR implementation, $\mathbb{A}_{i,j}$ is an AutoRegressive Integrated Moving Average (ARIMA) model [49]. Being an autoregressive technique, ARIMA fits on previous values of the feature series to predict its future behavior. In ARIMA, an autoregressive mechanism is applied to the series rendered stationary via differentiation procedure. Autoregression (AR) and error moving average (MA) are leveraged to predict future values. Once the forecast is computed on the stationary process, the integration (I) operation is applied to compute the final prediction value. For an in-depth overview, we suggest [49]. ARIMA is a parametric function over: p, the number of autoregressive terms, d, the number of differentiation steps needed to make the series stationary, and q, the number of lagged forecast errors in the prediction equation. The selection of p, q, and d hyperparameters is fundamental for a correct fit. There exist automatic search algorithms that have been proposed to optimize such parameters. In our work, we applied a variation of the Hyndman-Khandakar algorithm [50], following ARIMA's implementation proposed in *Pmdarima*[3].

ARIMA estimates $f_j^t(W_i^t)'$ using past node history $h_{i,j}$:

$$f_j^t(W_i^t)' \pm \mu = \mathbb{A}_{i,j}([f_j(W_i^{t-\lambda}), ..., f_j(W_i^{t-1})], \alpha), \tag{4}$$

where $f_j^t(W_i^t)'$ is the forecast value, and α is the confidence value (a DETONAR's hyperparameter); μ represent the prediction's boundary. DETONAR raises an anomaly for node n_i on feature f_j at time t if the following condition does not hold:

$$f_j^t(W_i^t)' - \mu \leq f_j^t(W_i^t) \leq f_j^t(W_i^t)' + \mu. \tag{5}$$

Figure 4 shows the anomaly detection mechanism applied to the same device and the same feature f_1 in a legitimate and an attack scenario. It is possible to notice that in legitimate traffic, the ARIMA raises no alarm. While, in the attack scenario, sinkhole produces an anomalous increment in the number of DIO packets received.

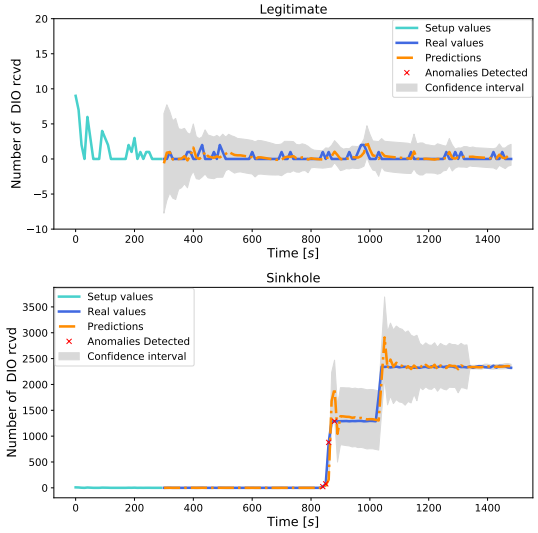


Fig. 4. ARIMA applied to the number of received DIO packets of a device in a legitimate and a sinkhole attack scenario.

We remark that the use of ARIMA introduces the need for a setup period free of attacks. This setup phase's size depends on history size λ, a DETONARS's hyperparameter, and the time window size ω chosen for feature extraction. Another aspect to consider is the choice of one ARIMA per node and traffic behavior, which makes DETONAR limited in the scalability, and thus it does not achieve P4.

E. Attack Classification

The anomaly detection stage allows DETONAR to detect any potential attack leveraging the minimum amount of features. However, it presents a couple of drawbacks:

[3]https://alkaline-ml.com/pmdarima/index.html

- *High false positives*. Any anomalous behavior causes ARIMA to raise an alarm. There exists no guarantee that an attack causes the alarm. Therefore, we need to filter out those anomalies caused by slight variations of legitimate behaviors.
- *No attack/attacker identification*. Alarms raised by the first stage of DETONAR simply signal that there exist some inconsistencies of behavior. These alarms do not allow compromised devices to be identified. Therefore, we need to introduce a classification technique that allows DETONAR to locate the attacker correctly.

To identify attack class, attacker identity and reduce the false positives, solving anomaly detection's drawbacks, we use several tests to profile the anomaly. In detail, once the anomaly detection mechanism raises the alarm, a list of suspected devices is produced. This list contains the devices which caused the alarm and their neighbors. Some attacks influence only features representing the neighborhood behavior while leaving attacker behavior features untouched. The list is passed to the attack classification mechanism, which is in-charge of classifying anomalies into different classes of attacks. This mechanism is implemented to make it possible to return false alarms whenever no attack is detected, allowing to identify and discard potential false positive alarms. To classify the attack correctly, DETONAR considers the 8 unused features of Section IV-C. These features consider the device's behavior and network structure. In particular, the DODAG is reconstructed at the centralized server due to the ability to record DAO packets exchanged between devices.

To classify anomalies into attacks and identify attackers, DETONAR implements a classification flowchart shown in Figure 5. For each node of the tree, a different rule (anomaly or signature-based) is applied to choose the path to follow. We now describe the different rules used by the classification mechanism:

- *Clone Identity - signature*. This rule compares active identities at W_i^t with the legitimate identities collected during the setup phase. In particular, we recall that DETONAR considers RPL devices sending application packets periodically (e.g. one packet per second), where each device can have different period. We define m as the maximum devices' period. Given the lossyness of RPL networks, DETONAR compares the identities between the setup phase and each possible sub-window of W_i^t of size $m \cdot c$, where the overlap is set to one second. Finally, DETONAR identifies a clone identity or a sybil attack if there is at least one mismatch among the comparisons. m represents an RPL network-related parameter, while c is a DETONAR's hyperparameter.
- *Changing DODAG - signature*. This rule checks if the attack impacts the DODAG. Since we are considering static scenarios, a modification of the DODAG can only be due to an attack. Change in DODAG structure are detected comparing DODAG at time window W_i^t with its predecessor at time window W_i^{t-1} This rule helps to identify two macro-categories of attacks: attacks on the DODAG, and attacks on the traffic.
- *Changing Rank - signature*. This rule allows to detect attacks that leverage the modification of the rank value. DETONAR filters the DIO packets sent by each device to check this rule, extracting the corresponding advised rank. A device advising a different rank in a static scenario is considered an attacker.
- *Changing Version - singature*. If no rank value has changed while the DODAG has been modified, advised versions are checked in this rule. Version values can be extracted from control packets in the same way as ranks. If a node has changed the version, then a version attack is detected. The corresponding attacker is identified as the first device which advised a new DODAG version. If no version and no rank values have been changed, but the DODAG was modified, then a worst parent attack is detected. The attacker is then detected as the device which changed the preferred parent.
- *Changing Transmitted Applications Rate - anomaly*. DETONAR checks if the considered node saw any change in the application packets that it transmitted. This rule is considered since attacks aiming at traffic can either manipulate application traffic or control traffic. Like the *changing DODAG* rule, this one helps to subdivide attacks against the traffic into attacks against application traffic and attacks against control traffic. This rule consists of an anomaly-based detection scheme. Indeed, DETONAR applies ARIMA on the series of transmitted application packets. This approach is identical to anomaly detection, but it changes only the considered feature.
- *Children Changing Destination - signature*. If an attack against application traffic is detected, the proposed rule checks if any node is changing its next-hop. If this happens, then the node changing next-hop is considered to be part of a wormhole attack. With respect to the *state-of-the-art* the proposed approach is the simplest enabling the detection of wormhole attacks. No considerations regarding devices' positions or power of transmission is done, and the resulting performances are surprising.
- *Incoming vs Outgoing Traffic - anomaly*. When no change in next-hop is detected, our IDS checks the ratio between received and transmitted application packets. In legitimate scenarios, this ratio's trend should remain almost constant. Instead, in blackhole and selective forward attacks, it decreases significantly. DETONAR applies anomaly-based detection scheme (i.e., ARIMA) to identify possible anomalies in this ratio sequence. If an anomaly is found, then the attacker is the device analyzed by the ARIMA. Otherwise, a false alarm is sent.
- *Produce New Control Packets - signature*. When no attack against application traffic is detected, the proposed rule checks if any node produces unnecessary control packets. The forged control packets may be either DIO or DIS. The same check is done for DIO and DIS control packets. DETONAR considers the number of control packets sent by a suspected device in the last time window W_i^t. If this value is bigger than the previous maximum value of control packets transmitted in a time window W_i^j with $j \in [1, t-1]$, then an attack is detected. Otherwise, a

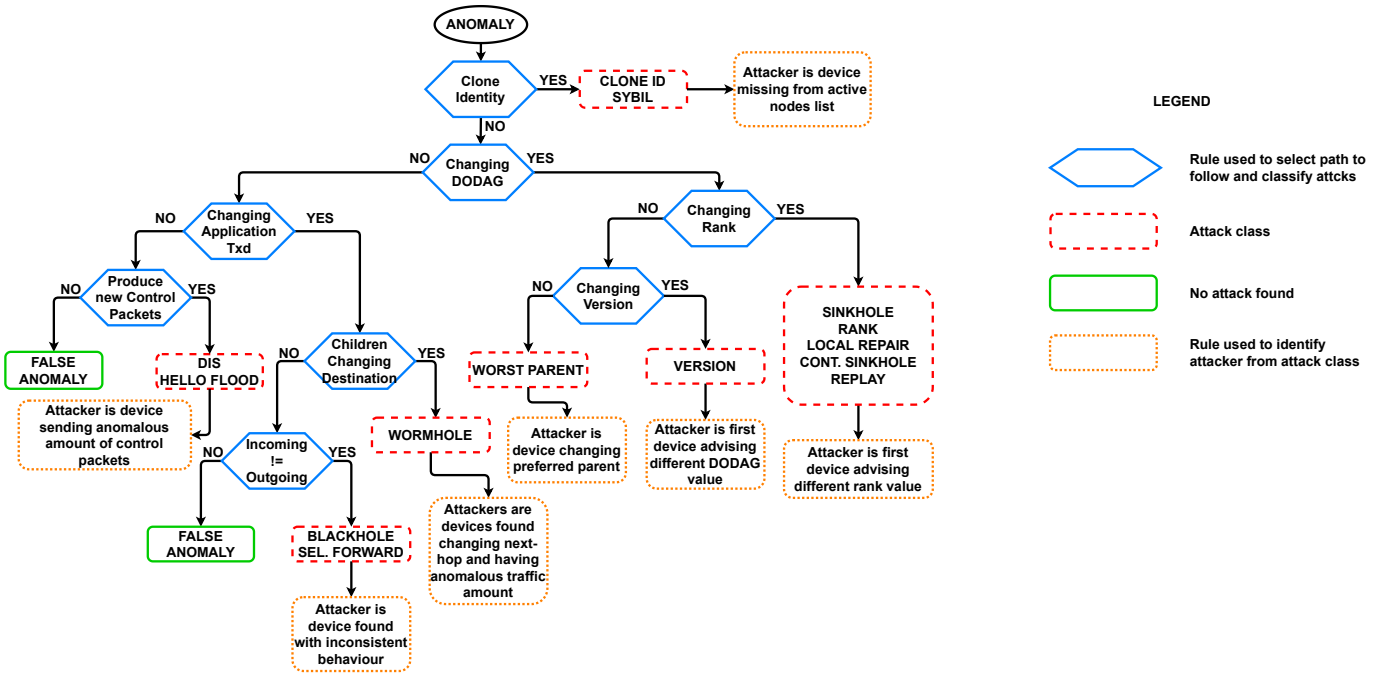


Fig. 5. DETONAR's decision flowchart. Each rule is based on considerations regarding RPL networks that help classify the final attack. Attacker identity depends on the attack classified.

false alarm message is raised, as there exists no significant proof of an attack on control traffic. Mathematically, an HELLO flooding attack is detected, if $DIO(W_i^t) > max\{DIO(W_i^{t-1}), DIO(W_i^{t-2}), ...\}$. Where the DIO function counts the number of DIO packets transmitted by a device i in the time window W_i^t.

The combination of the proposed rules allows DETONAR to detect the most known attacks against RPL. Moreover, the proposed mechanism is flexible against new attacks. New rules may be added to classify novel attacks upon their discovery.

F. Attacker Identification

A security system should automatically identify attacker's identity to remove the malicious device from the network. DETONAR introduces specific rules to identify the attacker location from the attack class identified following Section IV-E. The knowledge of attack class is sufficient to identify the attacker correctly since the attacker behavior of different attacks is specific. We now present the attacker identification mechanism that is used for each class of attacks:

- *Clone ID and Sybil attacks*. As already mentioned, DETONAR's check on active devices immediately identifies the attacker's identity. The device missing from communication corresponds to the attacker's original identity.
- *Attacks on rank*. This class of attacks comprehends sinkhole, local repair, rank, continuous sinkhole, and replay. To find the attacker's identity DETONAR checks which is the first device that advised a changed rank. One effect of these attacks is to change multiple nodes' ranks. Therefore, to find the attacker, it is necessary to find the oldest change in rank values. This approach is possible since the proposed IDS knows each device's rank from the sniffed DIO packets.
- *Version attack*. In this case, the approach to identify the attacker is very similar to the attacks on rank. The proposed IDS identifies the attacker as the first node that advised a different version in a DIO packet.
- *Worst parent attack*. To identify the attacker in the worst parent attack, DETONAR checks what device changed next-hop IP in the time window W_i^t. This straightforward principle effectively detects the attacker in a complex scenario like the worst parent attack. DETONAR can use such a simple principle due to the attack classification mechanism that relies on more complex decisions.
- *Wormhole attack*. DETONAR's attack classification mechanism detects those devices that changed next-hop IP and transmitted an anomalous amount of application packets in wormhole scenarios. This approach by itself allows the proposed IDS to find attackers' identities. Attacker devices are the only nodes that satisfy the two conditions presented above.
- *Blackhole and Selective forward attacks*. For these two attacks, DETONAR detects an attack only for those devices that drop an anomalous amount of application packets. Therefore, DETONAR's attack classification mechanism is already identifying attacker identities.
- *HELLO flooding and DIS attacks*. In HELLO flooding and DIS scenarios, the malicious device sends an anomalous amount of control packets. DETONAR's attack classification mechanism already identifies the attackers as those devices transmitting an anomalous amount of control packets.

Figure 5 shows the rules for attackers' identification.

V. Implementation and Performance evaluation

This section first presents the implementation details of the DETONAR mechanism (see Section V-A). Finally, in Section V-B we present the results obtained using DETONAR on RADAR dataset. In particular, we present DETONAR's true detection percentage, false positives, and the computation time needed by the anomalies detection step.

A. Implementation & Hyperparameters tuning

We make our implementation of DETONAR publicly available[4]. DETONAR's scripts take as input a packet trace corresponding to a simulation and return as output DETONAR's runtime information. This information contains the anomaly detection mechanism's decision at each time window W_i^t. If the anomaly detection algorithm raises the alarm, then the attack classification results are also present in the runtime information. Running DETONAR's scripts from the packet trace files belonging to RADAR corresponds to the real-world application of packet sniffers that communicates with the centralized server.

We now describe the DETONAR's hyperparameter tuning among the time window size ω, the test significance α, and the history size λ (see Section IV). Tuning is conducted on a training set containing five legitimate simulations. The goal of the tuning is to minimize the false positives (FP) of DETONAR. Figure 6 shows the false positives for different ω, α, and λ values. To select ω, we fix $\lambda = 30$ and $\alpha = 10^{-4}$. ω affects with different trends different features, not giving statistical relevant results. We set $\omega = 10$ due to computational performance reasons only. To select λ, we fix $\alpha = 10^{-4}$ and $\omega = 10$. We can notice that when increasing ARIMA history size, we reduce the false positives. To reduce the setup time as much as possible while allowing DETONAR to detect eventual attacks accurately, we set $\lambda = 30$. We fix $\lambda = 30$ and $\omega = 10$ to select α's value. A small α lead to small FP. We thus set $\alpha = 10^{-4}$. Finally, concerning c, we recall that in RADAR devices are deployed sending one application packet every second, i.e., $m = 1$. Therefore, we set $c = 3$.

B. Results

We now present DETONAR's performance computed on our proposed dataset, RADAR. We remark that we do not compare DETONAR with state-of-the-art IDS since they are not implemented using NetSim and their re-implementation is not trivial. We evaluate three aspects: false positives, attack and attacker(s) detection accuracy, and finally, time performance. We need to make some considerations for the attack detection accuracy. We know the attack's starting time τ_{Att} and the compromised device(s) for each simulation. We consider an attack to be detected correctly if DETONAR raises the alarm and classifies the attack correctly after τ_{Att}. Instead, the attack is considered misclassified if DETONAR does not raise any alarm or raises an anomaly correctly, but it does not classify the correct class of attack. We are bound to consider these metrics for classification's performance for the following reasons:

- Some attacks start at time $\tau_{Att} + \epsilon$ since they need the reception of a specific packet to be triggered. For example, in sinkhole attack, the attacker waits for the reception of DIO packets to trigger the publication of forged rank value. No assumption can be made on the duration of ϵ. Depending on the attack considered, the size of the network and the simulation time ϵ may vary significantly. Therefore, it is impossible to identify the actual attack starting time.
- No assumption can be made on the label of packets in the network traffic, since the attacks can indirectly affect the performance of non-victim devices. For example, sinkhole attack induces the attacker's neighbors to change their rank, and the attacker's neighbors advise new rank values in forged DIO packets. The aftermath is in a complex labeling process, which we avoid.

1) False Positives Performances: To analyze FP, we test DETONAR over five legitimate simulations (separate from the five used for tuning) and measure the number of identified attacks (∼2000 predictions). In detail, we measure the FP in both anomaly detection and attacker identification stages (see Section IV). Table V shows the FP rate results. In particular, the anomaly detection stage based entirely on ARIMA has high FP, while the second stage with both anomaly and rule-based signature pushes the score close to zero.

TABLE V
False positives for Anomaly Detection (AD) and AD + Attack Classification (AC) of DETONAR in five legitimate simulations.

Simulation ID	AD	AD + AC
6	2.13%	0%
7	1.83%	0.10%
8	3.06%	0%
9	2.08%	0.05%
10	5.19%	0.05%
Overall	2.86% ± 1.24	0.04% ± 0.04

2) Detection and Identification Performances: We test the ability of DETONAR to identify the attack and attacker(s) over five simulations for each of the 14 attacks presented in RADAR. Table VI summarizes the detection performance. DETONAR successfully detects with 100% of accuracy 8 out of 14 attacks; in these attacks, the attacker is always successfully identified. DETONAR seems to suffer only blackhole, continuous sinkhole, and local repair attacks. Concerning blackhole attack, we notice that in some simulations, the attack affected nodes with few application packets, resulting in challenging detection. In continuous sinkhole and local repair, instead, DETONAR misses the detection only for those simulations in which the attack does not produce any change in the DODAG structure; in these scenarios, DETONAR identifies reasonably a hello flooding attack.

3) Time Performances: Finally, we analyze DETONAR's time performance. We find the computational bottleneck in ARIMA models (see Section IV, anomaly detection stage), which are applied for each window W_i^t. We measure ARIMA

[4]https://github.com/AndAgio/DETONAR

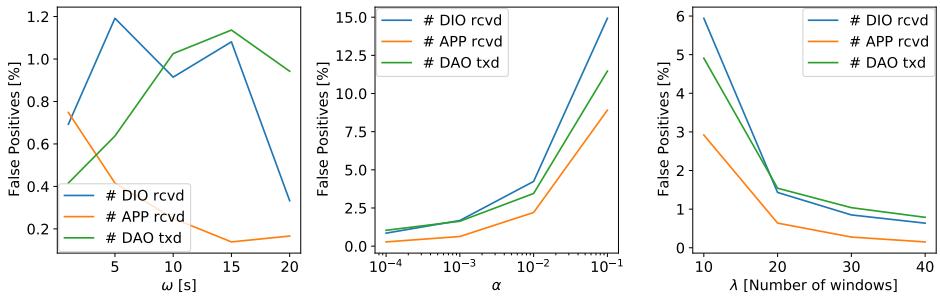


Fig. 6. ARIMA's false positives are influenced by DETONAR's time window size (i.e., ω), history size (i.e., λ) and test significance (i.e., α).

TABLE VI
DETONAR CAN DETECT MOST ATTACKS WITH SATISFACTORY ACCURACY BOTH IN TERMS OF ATTACK DETECTION (DET) AND ATTACKER IDENTIFICATION (ID). IN WORMHOLE, ID CONTAINS THE NUMBER OF ATTACKERS IDENTIFIED CORRECTLY OUT OF THE TWO EXISTING.

Attack	1		2		3		4		5		Overall	
	DET	ID	DET	ID	DET	ID	DET	ID	DET	ID	DET	ID
Blackhole	✓	✓	✓	✓	✓	✓					60%	60%
Selective Forward	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Sinkhole	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Continuous Sinkhole	✓	✓			✓	✓	✓	✓			60%	60%
HELLO Flooding	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Clone ID	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Sybil	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Wormhole	✓	2/2		0/2	✓	2/2	✓	2/2	✓	1/2	80%	70%
Version	✓	✓	✓	✓			✓	✓	✓	✓	80%	80%
Rank	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Replay	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Worst Parent	✓	✓	✓	✓	✓	✓			✓		80%	60%
DIS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100%	100%
Local Repair		✓		✓	✓	✓	✓	✓		✓	40%	100%

performance on 170 thousand predictions using a standard laptop (i.e., Intel Core i5-3230M CPU, 8 GBs DDR3 RAM). The average prediction time is 1.1 seconds. Given this performance, and the possibility to distribute the computation among several cores, we can state that DETONAR can be deployed in real-world small-scale networks.

VI. Conclusion and Future Work

In this work, we implement and detect 14 well-known routing attacks against RPL in IoT networks. Using the various network logs obtained while simulating these attacks, we build our RADAR dataset. RADAR represents the largest and most significant dataset of routing attacks against RPL. We believe that the availability of such a comprehensive dataset is a step forward in the research field of IoT security. Based on the RADAR dataset, we propose a novel and complete security mechanism called DETONAR, capable of detecting 14 well-known attacks. The simulation results show that DETONAR provides excellent attacker identification results (i.e., low false positives) with no RPL communication overhead, thanks to its sniffing approach. DETONAR does not require any heavy computation or firmware modification at IoT devices, which makes it a practical solution. It also introduces future flexibility as, upon discovering novel attacks, one can modify the attack classification mechanism by adding new rules to previously unknown attack rules. Finally, DETONAR's flexibility allows its quick deployment on the underlying network, as it does not require any IoT devices update. In the future, we plan to do a more in-depth analysis of DETONAR concerning the following aspects: (i) test its performance on a real-world testbed, (ii) investigate its performance in dynamic networks, (iii) extend its attack detection algorithm to generalize features behaviours among different devices, (iv) test its performance on large-scale networks, and (v) compare its performance with state-of-the-art IDS implementation using NetSim.

Acknowledgment

This research is partially sponsored by the Ministry of Science and Technology, Taiwan, under Grant no. MOST 110-2634-F-002-037. Moreover, the authors would like to thank Tetcos organization and their NetSim support service. Thanks to their kind help we were able to develop our work quickly and successfully.

References

[1] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*, 2011, pp. 1–6.

[2] H. P. Breivold and K. Sandström, "Internet of things for industrial automation–challenges and technical solutions," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, 2015, pp. 532–539.

[3] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[4] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on smart homes system using internet-of-things," in *2015 International Conference on Computation of Power, Energy, Information and Communication*, 2015, pp. 0330–0335.

[5] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of internet of things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.

[6] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[7] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[8] N. Tariq, M. Asim, Z. Maamar, M. Z. Farooqi, N. Faci, and T. Baker, "A mobile code-driven trust mechanism for detecting internal attacks in sensor node-powered iot," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 198 – 206, 2019.

[9] U. Khalid, M. Asim, T. Baker, P. C. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for iot systems," *Cluster Computing*, pp. 1–21, 2020.

[10] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[11] S. L. CNN, "FDA confirms that St. Jude's cardiac devices can be hacked," https://money.cnn.com/2017/01/09/technology/fda-st-jude-cardiac-hack/.

[12] L. Li, H. Xiaoguang, C. Ke, and H. Ketai, "The applications of wifi-based wireless sensor network in internet of things and smart grid," in *2011 6th IEEE Conference on Industrial Electronics and Applications*, 2011, pp. 789–793.

[13] L. Davoli, L. Belli, A. Cilfone, and G. Ferrari, "From micro to macro iot: Challenges and solutions in the integration of ieee 802.15. 4/802.11 and sub-ghz technologies," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 784–793, 2017.

[14] X. Jia, Q. Feng, T. Fan, and Q. Lei, "Rfid technology and its applications in internet of things (iot)," in *2nd International Conference on Consumer Electronics, Communications and Networks*, 2012, pp. 1282–1285.

[15] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the iot," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 125–131, 2018.

[16] H. Kharrufa, H. A. Al-Kashoash, and A. H. Kemp, "Rpl-based routing protocols in iot applications: A review," *IEEE Sensors Journal*, vol. 19, no. 15, pp. 5952–5967, 2019.

[17] F. Gara, L. B. Saad, R. B. Ayed, and B. Tourancheau, "Rpl protocol adapted for healthcare and medical applications," in *2015 International wireless communications and mobile computing conference*, 2015, pp. 690–695.

[18] H.-S. Kim, J. Paek, and S. Bahk, "Qu-rpl: Queue utilization based rpl for load balancing in large scale industrial applications," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking*, 2015, pp. 265–273.

[19] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the rpl-based internet of things," *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, 2013.

[20] P. Pongle and G. Chavan, "A survey: Attacks on rpl and 6lowpan in iot," in *International conference on pervasive computing*, 2015, pp. 1–6.

[21] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing attacks and mitigation methods for rpl-based internet of things," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1582–1606, 2018.

[22] M. Conti, P. Kaliyar, and C. Lal, "A robust multicast communication protocol for low power and lossy networks," *Journal of Network and Computer Applications*, vol. 164, p. 102675, 2020.

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

[24] M. Surendar and 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*, 2016, pp. 1903–1908.

[25] F. Gara, L. B. Saad, and R. B. Ayed, "An intrusion detection system for selective forwarding attack in ipv6-based mobile wsns," in *2017 13th International Wireless Communications and Mobile Computing Conference*, 2017, pp. 276–281.

[26] H. Sedjelmaci, S. M. Senouci, and T. Taleb, "An accurate security game for low-resource iot devices," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9381–9393, 2017.

[27] D. Airehrour, J. A. Gutierrez, and S. K. Ray, "Sectrust-rpl: A secure trust-aware rpl routing protocol for internet of things," *Future Generation Computer Systems*, vol. 93, pp. 860–876, 2019.

[28] P. Kaliyar, W. B. Jaballah, M. Conti, and C. Lal, "Lidl: Localization with early detection of sybil and wormhole attacks in iot networks," *Computers & Security*, vol. 94, p. 101849, 2020.

[29] M. Conti, P. Kaliyar, M. M. Rabbani, and S. Ranise, "Split: A secure and scalable rpl routing protocol for internet of things," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct 2018, pp. 1–8.

[30] F. Y. Yavuz, Ü. Devrim, and G. Ensar, "Deep learning for detection of routing attacks in the internet of things," *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 39–58, 2018.

[31] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach," *Computer Communications*, vol. 98, pp. 52–71, 2017.

[32] "NetSim by Tetcos," https://www.tetcos.com/netsim-std.html.

[33] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, R. K. Alexander *et al.*, "Rpl: Ipv6 routing protocol for low-power and lossy networks." *RFC*, vol. 6550, pp. 1–157, 2012.

[34] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.

[35] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6lowpan with compressed ipsec," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops*, 2011, pp. 1–8.

[36] W. Jung, S. Hong, M. Ha, Y.-J. Kim, and D. Kim, "Ssl-based lightweight security of ip-based wireless sensor networks," in *2009 International Conference on Advanced Information Networking and Applications Workshops*, 2009, pp. 1112–1117.

[37] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016.

[38] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Networks*, vol. 90, p. 101842, 2019.

[39] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[40] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," in *International Symposium on Networks, Computers and Communications*, 2016, pp. 1–6.

[41] S. Raza, L. Wallgren, and T. Voigt, "Svelte: Real-time intrusion detection in the internet of things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[42] C. Cervantes, D. Poplade, M. Nogueira, and 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)*, 2015, pp. 606–611.

[43] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in rpl-based networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 472–486, 2017.

[44] X. Feng, X. Liao, X. Wang, H. Wang, Q. Li, K. Yang, H. Zhu, and L. Sun, "Understanding and securing device vulnerabilities through automated bug report analysis," in *SEC'19: Proceedings of the 28th USENIX Conference on Security Symposium*, 2019.

[45] Y. Tian and R. Hou, "An improved aomdv routing protocol for internet of things," in *2010 International Conference on Computational Intelligence and Software Engineering*, 2010, pp. 1–4.

[46] Y. Ji, S. Biaz, S. Wu, and B. Qi, "Optimal sniffers deployment on wireless indoor localization," in *2007 16th International Conference on Computer Communications and Networks*, 2007, pp. 251–256.

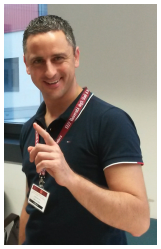
[47] S. Biaz, Y. Ji, and P. Agrawal, "Impact of sniffer deployment on indoor localization," in *2005 International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2005, p. 10.

[48] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks." Association for Computing Machinery, 2003, p. 251–261.

[49] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[50] R. J. Hyndman and Y. Khandakar, "Automatic Time Series Forecasting: The forecast Package for R," *Journal of Statistical Software*, vol. 27, no. 1, pp. 1–22, Jul. 2008.

Andrea Agiollo is currently a Ph.D. student in Computer Science and Engineering at the University of Bologna, Italy, with the collaboration of Electrolux Professional S.P.A. He received his Master Degree in Information and Communication Technologies for Internet and Multimedia at the University of Padova, Italy, in 2020. The same year, he also received his Master Degree in Communication Engineering at the National Taiwan University, Taiwan. He received his Bachelor of Information Engineering at the University of Padova, Italy, in 2018. He is conducting research on fields including Machine Learning, Internet of Things, and Explaible Artificial Intelligence.



Mauro Conti is Full Professor at the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 400 papers in topmost international peer-reviewed journals and conferences. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and has been Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and General Chair for SecureComm 2012, SACMAT 2013, CANS 2021, and ACNS 2022. He is Senior Member of the IEEE and ACM. He is a member of the Blockchain Expert Panel of the Italian Government. He is Fellow of the Young Academy of Europe.

Pallavi Kaliyar is currently a Ph.D. student in school of Brain Mind and Computer Science at the University of Padova, Italy with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). Here, she is part of the SPRITZ Security and Privacy Research Group research group under the supervision of Prof. Mauro Conti. She received the Masters of Technology in Computer Science and Engineering in 2012 and Bachelor of Engineering in Computer Science and Engineering in 2008. She is conducting research on fields including security and communication reliability related to the Internet of Things and Software Defined Networking.

Tsung-Nan Lin Tsung-Nan Lin (SM'03) received his B.S. degree from National Taiwan University, Taipei, Taiwan, in 1989 and his M.S. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1993 and 1996, respectively. He then joined EPSON Research and Development, Inc., San Jose, CA, USA, and EMC Corporation, Hopkinton, MA, USA. Since February 2002, he has been with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University. He had been the Director of the Division of Network Management of Computer and Information Networking Center, National Taiwan University and Vice President and Director General of Cybersecurity Technology Institute, Institute For Information Industry. Dr. Lin is a member of the Phi Tau Phi Scholastic Honor Society

Luca Pajola is currently a Ph.D. student in school of Brain Mind and Computer Science at the University of Padova, Italy. Here, he is part of the SPRITZ Security and Privacy Research Group research group under the supervision of Prof. Mauro Conti. He received my MSc in Computer Science in 2018 at University of Padova, Italy. He is conducting research on fields including security and machine learning.