
Article:
Ioulianou, Philokypros, Vasilakis, Vasileios orcid.org/0000-0003-4902-8226 and Shahandashti, Siamak F. orcid.org/0000-0002-5284-6847 (2022) A Trust-Based Intrusion Detection System for RPL Networks: Detecting a Combination of Rank and Blackhole Attacks. Journal of Cybersecurity and Privacy. pp. 124-153. ISSN 2624-800X

https://doi.org/10.3390/jcp2010009

Journal of Cybersecurity and Privacy

MDPI

Article

A Trust-Based Intrusion Detection System for RPL Networks: Detecting a Combination of Rank and Blackhole Attacks

Philokypros P. Ioulianou, Vassilios G. Vassilakis * and Siamak F. Shahandashti

Department of Computer Science, University of York, York YO10 5GH, UK; pi533@york.ac.uk (P.P.I.); siamak.shahandashti@york.ac.uk (S.F.S.)
* Correspondence: vasileios.vasilakis@york.ac.uk

Abstract: Routing attacks are a major security issue for Internet of Things (IoT) networks utilising routing protocols, as malicious actors can overwhelm resource-constrained devices with denial-of-service (DoS) attacks, notably rank and blackhole attacks. In this work, we study the impact of the combination of rank and blackhole attacks in the IPv6 routing protocol for low-power and lossy (RPL) networks, and we propose a new security framework for RPL-based IoT networks (SRF-IoT). The framework includes a trust-based mechanism that detects and isolates malicious attackers with the help of an external intrusion detection system (IDS). Both SRF-IoT and IDS are implemented in the Contiki-NG operating system. Evaluation of the proposed framework is based on simulations using the Whitefield framework that combines both the Contiki-NG and the NS-3 simulator. Analysis of the simulations of the scenarios under active attacks showed the effectiveness of deploying SRF-IoT with 92.8% packet delivery ratio (PDR), a five-fold reduction in the number of packets dropped, and a three-fold decrease in the number of parent switches in comparison with the scenario without SRF-IoT. Moreover, the packet overhead introduced by SRF-IoT in attack scenarios is minimal at less than 2%. Obtained results suggest that the SRF-IoT framework is an efficient and promising solution that combines trust-based and IDS-based approaches to protect IoT networks against routing attacks. In addition, our solution works by deploying a watchdog mechanism on detector nodes only, leaving unaffected the operation of existing smart devices.

Keywords: RPL-lite security; RPL; SRF-IoT; SRF-OF; trust; intrusion detection system; blackhole attack; rank attack

Citation: Ioulianou, P.P.; Vassilakis, V.G.; Shahandashti, S.F. A Trust-Based Intrusion Detection System for RPL Networks: Detecting a Combination of Rank and Blackhole Attacks. *J. Cybersecur. Priv.* **2022**, *2*, 124–153. https://doi.org/10.3390/jcp2010009

Academic Editor: Danda B. Rawat

Received: 27 December 2021
Accepted: 5 March 2022
Published: 9 March 2022

1. Introduction

More smart devices are connecting to the Internet every day to improve our daily lives. Internet of Things (IoT) forecasts suggest that there will be more than 25.4 billion connected devices by 2030 [1]. The growth of IoT will provide multiple benefits for companies and individuals. However, as IoT networks become more popular, they become attractive to malicious actors, and therefore, security issues start to appear.

Many malicious actors try to attack smart devices due to weak or no security measures implemented by manufacturers [2,3]. A set of 33 vulnerabilities in TCP/IP stacks were found recently, affecting millions of IoT devices of over 150 device manufacturers [4]. Four open source TCP/IP stacks were affected, which are found in multiple operating systems (OSes) such as Contiki and Nut/OS [5]. This set of new flaws is called "Amnesia:33" from the fact that the number of vulnerabilities is 33, and most of them might be exploited to carry out denial-of-service (DoS) attacks and remote code execution (RCE). Protecting IoT devices and networks using approaches such as cryptographic protocols and traditional intrusion detection systems (IDS) is not always feasible. Limited computational power and limited energy are characteristics of a smart device. Ensuring confidentiality, availability, and integrity of IoT networks requires new solutions that take into account their limitations.

The routing protocol for low-power and lossy networks (RPL) [6] is a protocol designed for resource-constrained devices. It was proposed by the Internet Engineering Task Force (IETF) ROLL (routing over low power and lossy networks) working group [7]. Existing works show that RPL is vulnerable to many attacks, and countermeasures are proposed by several researchers [8,9]. The RPL-based attacks focus on exploiting the limitations of power and the lossy environment in which IoT devices operate. Some examples are selective forwarding, blackhole, rank, and version attacks [10,11]. Usually, an IDS is deployed as a detection mechanism in the network.

In this paper, the impact of combined rank and blackhole attacks is studied by implementing them in Contiki-NG and simulating them in the Whitefield framework. Malicious devices attack the RPL-based network by advertising false rank and dropping packets of child nodes. As a result, the network has poor performance and devices exhaust their energy.

A novel security framework for RPL-based IoT networks (SRF-IoT) is also designed and proposed to mitigate RPL-based routing attacks. IoT devices that utilise RPL protocol use an objective function (OF) to help them choose the best parent that provides a route to sink/border router (BR). SRF-IoT provides a trust-based OF, called SRF-OF, so that nodes choose the most trusted parent and avoid malicious actors in a network. Moreover, an external IDS, called SRF-IDS, aims to provide smart devices with trust metrics. It consists of an SRF-IDS root that plays the role of BR and SRF-IDS detectors that have promiscuous mode enabled for capturing network packets. Trust metrics are essential for the operation of the trust-based mechanism during parent selection. We combine the trust-based OF and IDS methods to detect attackers with the help of nodes. Specifically, each node computes the trust value of its direct neighbours based on the information collected from an SRF-IDS detector. In this way, do not waste power for network monitoring. The best parent of each node is selected based on the calculated trust value. Devices with high trust value are chosen as parents, whereas those with lower trust values are avoided. The Whitefield simulator is a new framework that is used for SRF-IoT evaluation. Additionally, it provides more realistic results than the Cooja simulator, Contiki-NG's embedded simulator, and allows deploying large scale scenarios with minimum effort.

One of the novelties of our approach is that a successful combination of trust-based and IDS-based methods is achieved. Most studies propose mitigation schemes for RPL routing attacks using either the former or the latter method. Combining the two approaches enhances the detection performance and allows easier detection of attackers. Furthermore, it is straightforward to add mitigation methods for additional RPL attacks. Another benefit is that nodes' energy consumption is minimised. This is because nodes operate normally without any watchdog mechanism as in other trust-based solutions. SRF-IDS, made of an external set of detectors, monitors the network and provides the nodes with trust metrics.

Another novelty of our approach is the deployment of SRF-IDS along with the normal network. In order to monitor the neighbouring network without interruptions, SRF-IDS is deployed in a different RPL instance than the normal network. SRF-IDS is an improved version of the prototype proposed in our earlier work [12]. It consists of a centralised router that hosts detection module and acts as a firewall, whereas decentralised detectors are responsible for traffic monitoring, and local detection. Alerting monitored nodes with trust metrics is also the task of SRF-IDS detectors. RPL protocol was altered to allow communication between SRF-IDS detectors and monitored devices. The specification of the RPL protocol should be modified to allow the operation of SRF-IDS with other systems.

The key contributions of this paper are as follows:

- In our previous work [12], an initial IDS prototype was presented. SRF-IDS, an enhanced threshold-based IDS, is developed and deployed in this paper. It uses an overlay approach to provide a separate mechanism to monitor devices and help neighbouring nodes isolate malicious actors.
- A novel security framework for RPL-based IoT networks (SRF-IoT) has been designed and proposed to avoid routing attacks, including a combination of rank and blackhole attacks. The framework consists of the threshold-based IDS, called SRF-IDS, and a

trust-based OF, called SRF-OF. It has been implemented in the RPL protocol and evaluated in various scenarios under routing attacks.

- Experimental results demonstrate that the SRF-IoT framework can effectively detect and help nodes to avoid malicious nodes. Our proposed framework showed 92.8% PDR, an almost five times reduction in the number of packets dropped, and a threefold decrease in the number of parent switches in scenarios with active blackhole and rank attacks. A new simulator, called Whitefield framework, which combines the NS-3 simulator with the Contiki-NG operating system, is used for evaluating the SRF-IoT framework. To our knowledge, this is the first work in the literature in which the Whitefield framework is used as the main simulation tool.

The rest of the paper is organized as follows. In Section 2, we provide the required background information, including an overview of the RPL protocol, attacks against it, and IDS-based, trust-based, and protocol-based mitigation methods suggested for various RPL attacks. In Section 3, we describe our proposed SRF-IoT framework, including the general operation and the network topology of the system. In Section 4, the various processes and the design of the SRF-IoT framework is described in detail, including the improved external IDS, the procedure of gathering, calculating, monitoring, and updating trust metrics from the IDS, and the detection mechanism. In Section 5, we describe the algorithms and implementation details of the SRF-IoT framework, including the communication module and the development of SRF-OF and trust in RPL-lite, as well as the studied attacks in Contiki-NG, namely the rank and blackhole attacks. In Section 6, we present the evaluation of our SRF-IoT framework, including the scenarios and considered simulation settings, the configuration of evaluation metrics, and the evaluation results. A comparison of the observed results with related works is also provided. Finally, Section 7 concludes the paper and discusses relevant future work.

2. Background and Literature Review

IoT networks are vulnerable to attacks in multiple layers of the IoT stack. This section aims to summarise state-of-the-art research of securing RPL protocol. Solutions for a secure RPL protocol are divided into two categories: IDS-based and trust-based.

2.1. IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)

Routing attacks such as blackhole, sinkhole, rank, version number, selective forwarding, DIS flooding, and DAO flooding attacks are extensively studied in [3,9,13,14]. Below, we briefly explain the RPL protocol and the existing security attacks.

2.1.1. RPL Overview

IPv6 routing protocol for low power and lossy networks (RPL) is a routing protocol used in IoT networks based on IPv6 [6]. It is designed for IoT, as it requires few resources and has self-maintenance. RPL networks create a topology similar to a tree, which is called a directed acyclic graph (DAG). This DAG is built by various nodes. If a root node exists, it is translated to destination-oriented DAG (DODAG), where the default destination is usually the root node (i.e., sink). Devices always try to send data to the root, which can be used as a border router (BR) to communicate with the Internet or with other internal networks. Multiple RPL instances can coexist inside a DAG. This is possible because each RPL instance can have one or more DODAGs. However, all DODAGs in an RPL instance must have the same RPLInstanceID and use the same Objective function (OF). Thus, multiple applications can operate in a network at the same time and separately.

OF is utilised in RPL protocol to select and optimise routes within an RPL instance. Usually, OF is based on some metrics or constraints (i.e., energy, latency, and throughput). The result of the function is the rank of the node, which is an indication of the node's distance from a DODAG root (hops from the root node). Two OFs are supported by RPL: zero objective function (OF0) and minimum rank with hysteresis objective function (MRHOF). The OF0 is based on hop count to calculate rank, whereas MRHOF uses the

expected transmission count (ETX). Rank increases with every hop from the root, thus root has the minimum rank. This metric is also needed in the preferred parent selection process of a node. Nodes with lower rank are preferred. Parents are nodes that forward packets from a child node. Parents are, by definition, closer to the root, which is translated to a lower rank.

RPL has two different modes: non-storing and storing mode. In non-storing mode, nodes just store the routes to arrive at their parents. Hence, if a node wants to communicate with another node, it needs to propagate this request until the desired node is reached. This mode can produce latency when communicating two specific nodes if they are far enough. However, nodes do not need memory resources to store a lot of information. In storing mode, nodes keep information about the routes of all the nodes in the network. Thus, communication between two nodes is simpler but more memory space is needed.

A number of ICMPv6 control messages are used in RPL. DODAG information option (DIO) messages are sent by the root. These messages are needed to keep the DODAG and contain information about the OF, the rank of the broadcasting node, and the DODAG ID. If a DIO message is received by a node, the node will determine its rank (based on received rank) and the cost of getting to the node from itself. DODAG information solicitation (DIS) messages are used to solicit a DIO from an RPL node. In other words, it is used as a neighbour discovery. When a new node joins the DODAG, it multicasts a DIS message and waits to hear for a DIO. A destination advertisement object (DAO) message is sent to propagate information upwards. Nodes send this message to the root, so the message is propagated or forwarded by parent nodes until root node is reached. Then, a destination advertisement acknowledgement (DAO-ACK) is sent as an answer from the root node when receiving a DAO.

In our work, Contiki-NG uses RPL-lite protocol, which is a lighter version of the standard RPL. This version of RPL removes support for the storing mode in favour of the non-storing mode. Usually, RPL-lite shows better performance and has a considerably smaller ROM footprint than ContikiRPL. Regarding the OF, we use the MRHOF, which is the default in Contiki-NG.

2.1.2. RPL Security

Below, we briefly present recent work in the field of RPL security in IoT networks.

A detailed study about RPL security was presented by Verma et al. [15]. Authors provided a comprehensive overview of RPL attacks and categorised them based on their targets, including resources, topology, and traffic. Moreover, they evaluated existing RPL security solutions for several attacks and compared their performance based on various metrics. The authors concluded that specific IDS and RPL protocol mitigation techniques are still in the early stages of RPL protocol. Thus, more research is needed to completely secure IoT networks.

Raoof et al. [16] presented a comprehensive study of RPL attacks. They classified RPL attacks into those inherited from wireless sensor networks and those attacks specific to the RPL. The latest mitigation methods were also discussed and classified for RPL-based networks. The authors reported that, although there are some IoT-based IDSes, RPL-specific attacks, such as DIS attacks, have no appropriate mitigation method to date. Moreover, the majority of studies do not present complete implementations for the mitigation of the various RPL-specific attacks.

Another extensive study of RPL security attacks and their impact on network performance was conducted by Wallgren et al. [17]. The authors implemented well-known RPL routing attacks, including rank attack, in ContikiOS, and demonstrated that the protocol is vulnerable to these attacks. A heartbeat protocol was also developed as a security mechanism to protect the IoT network against selective-forwarding attack. However, their solutions worked well only if IPSec was used in the network because an attacker may choose to not filter ICMPv6 packets and thus, avoid being detected.

Ribera et al. [18] studied blackhole and greyhole attacks in an RPL network. The authors analysed the impact of the attacks in Contiki-NG using metrics such as CPU, memory usage, and TX/RX rates. A novel UDP-based heartbeat detection technique was then implemented and evaluated using a Cooja simulator. Results showed high accuracy with low overhead in terms of CPU usage and battery consumption. However, traffic overhead caused by the proposed technique was not studied.

Boudouaia et al. [14] discussed the latest work about RPL-based rank attack. The rank property of RPL protocol can be exploited and may cause poor network performance and a waste of energy. Mitigation methods as well as the damage caused on network parameters were explained in their work. In addition, the authors compared different attacks including rank attack using the Friedman test, and emphasised on the importance of rank attack.

2.2. IDS-Based Mitigation Methods

Over time IDSes have been considered by researchers as security measures for keeping IoT networks secured [19]. However, traditional network detection algorithms have different requirements than those based on IoT. Below, we present the latest IDS solutions for IoT.

Strainer-based intrusion detection of blackhole in 6LoWPAN for the Internet of Things (SIEWE) was proposed by Patel and Jinwala [20] to detect blackhole attacks in RPL networks. Blackhole attackers attract nodes by advertising a greater routing metric to neighbouring nodes so it is selected as the preferred parent. SIEWE uses this fact and adds the nodes IDs in a suspicious list. Amart devices that have suspicious nodes in their vicinity then analyze the behaviour of these nodes and inform BR about their findings. Thus, SIEWE includes only those nodes that have suspected nodes in their vicinity rather than requiring each node in the network to process and check suspicious nodes. Evaluation results on a Cooja simulator indicated that SIEWE increased PDR of the network. A drawback of this approach is that only RSSI is utilised as parent selection metric. Attackers could send with the same signal power in order to bypass the detection mechanism.

A sink-based intrusion detection system (SBIDS) was presented in [21] for detecting rank attacks in RPL networks. The authors used a rule-based approach to compare node's current rank with the node's parent rank as well as the minimum rank of their siblings. If a node advertised a greater rank than its parent, it was considered malicious. Evaluation of the scheme showed that SBIDS achieved good detection performance of rank attacks.

Belavagi et al. [22] studied the different RPL attacks using simulations and tried to identify multiple intrusions for varied network size by using an IDS. Rank attack, selective forwarding, wormhole, and denial of service (DoS) attack were identified by the algorithms discussed in their work. A Cooja simulator with ContikiOS and ETX as an objective function was used for simulating scenarios of multiple attacks. Evaluation results showed that as the number of attackers increases, network performance was reduced. Therefore, a machine learning approach would be more suitable to identify various RPL routing attacks.

Another remarkable work in the field is the SVELTE IDS [23]. This is a signature- and anomaly-based IDS, aiming to protect smart devices from routing attacks based on RPL. Some of the considered attacks included altering information, sinkhole, and selective forwarding. The authors developed and evaluated SVELTE in ContikiOS. Results indicated that SVELTE had a high true positive rate but also some false alarms during simulations. However, SVELTE showed high traffic overhead due to the reconstruction of the network initiated by the router.

All in all, existing IDS-based mitigation methods might detect attackers successfully with some limitations. First, suggested solutions generate too much traffic overhead in the network. Second, using a single metric to detect an attacker may lead to the wrong results. Last but not least, most of the studies detected one or more simple routing attacks. For those reasons, a novel solution that can detect a combination of attacks using multiple metrics and keeping low traffic overhead is needed.

2.3. Trust and Protocol-Based Mitigation Methods

Several studies exist in the literature implementing trust-based IDS or RPL protocol-based mitigation techniques. The most important of these studies are reviewed below.

A metric-based RPL trustworthiness scheme (MRTS) for addressing RPL attacks was introduced in [24]. In this scheme, every node evaluated the behaviour of its neighbouring nodes based on indirect suggestions and direct observations. Nodes then needed to calculate the extended RPL node trustworthiness (ERNT) for their neighbours. The node with higher trust value, more energy, and better link quality was selected as the preferred parent. MRTS used the ERNT as a routing metric to form the network. Results showed that it helped nodes to avoid malicious nodes. In addition, it had low energy consumption and high packet delivery ratio. However, nodes need to be in promiscuous mode to observe neighbour's behaviour.

A secure-RPL (SRPL) protocol was presented in [25]. The aim of the proposed solution was to prevent malicious nodes from changing their rank multiple times, creating fake topologies. SRPL introduced a threshold to limit the number of rank changes. A hash chain authentication technique was also utilised to authenticate nodes when moving in the DODAG and modifying rank values. The authors suggested that SRPL can be used to detect other RPL attacks such as sinkhole, blackhole, selective forwarding attacks, etc. For these attacks, they recommended deploying anomaly-based algorithms to improve detection rate. Simulation results showed that SRPL successfully protected the network from rank attacks. However, there was an increase in RPL control messages.

A secure RPL routing protocol (SRPL-RP) for identifying and isolating rank and version attacks was proposed in [26]. The authors extended the work in [21,27] in which mitigation methods identified both rank and version attacks. For rank detection, if a node's rank was greater than node's parent rank, it was considered malicious. It was then removed from the monitoring table as a legitimate node and added in the blacklist table. For version attack, a similar threshold-based approach was followed. SRPL-RP was implemented and simulated in Cooja with ContikiOS using various topologies. SRPL-RP showed better detection and mitigation accuracy in comparison with other solutions.

Airehrour et al. [28] implemented the SecTrust-RPL protocol to overcome RPL routing attacks such as rank and sybil attacks. The suggested secure framework enhanced RPL protocol by using a trust-based system to detect attackers. The concept was to have each node in the RPL network to be in promiscuous mode and sniff neighbour packets. They then computed direct and recommended trust values for each of its neighbours. Direct trust was calculated based on the packet forwarding behaviour of the neighbour of the node. Recommended trust was given by another node, and it was an estimation of how reliable and trustful a node is that was located at 2-hops or more. Evaluation results showed that their solution protected against rank and sybil attacks, and it was more reliable and consumed less energy than standard RPL protocol.

Iuchi et al. proposed a secure parent node selection scheme in [29]. It aimed to allow nodes to choose legitimate nodes only as parents and avoid attackers. In this scheme, every node could decide if a node's rank is legitimate or not based on the average obtained ranks from neighbouring nodes. Therefore, nodes select parents that do not have too low rank and avoid malicious nodes. Simulation results showed that the proposed scheme avoided attackers, and the network operated better than standard RPL protocol.

Generally, many studies proposed enhancements to secure RPL protocol and detect attackers. Yet, most trust-based methods require devices to be in promiscuous mode and monitor the network. This results in a waste of energy, as devices are always active and sniff network traffic. Apart from that, large storage capacity is required by devices so that essential monitoring information is stored. Another disadvantage is that the majority of studies do not present complete implementations of their proposed mitigation methods.

In conclusion, many studies focus on how RPL routing attacks affect network operations and solutions are developed to stop these attacks. Table 1 depicts scientific efforts that study and propose a mitigation method for RPL attacks. Most of the suggested solutions are either IDS-based or protocol-based utilising the features of RPL protocol. A new method should consider combining these two fields to achieve higher and better detection performance.

Table 1. Relevant works on RPL attacks.

Study	Implementation Method	Detected Attacks
SRPL-RP [26]	RPL integrated with threshold-based detection	Rank and version attacks
Secure-RPL (SRPL) [25]	RPL integrated with threshold-based detection	Rank attack
SecTrust-RPL [28]	RPL integrated with trust scheme	Rank and sybil attacks
Secure Parent Node Selection Scheme [29]	RPL with integrated threshold-based detection	Rank attack
MRTS [24]	RPL integrated with trust scheme	Rank and blackhole attacks
SVELTE [23]	Signature/Anomaly-based IDS	Sinkhole, and selective forwarding
Sink-Based Intrusion Detection Systems (SBIDS) [21]	Rule-based IDS	Rank attack
SIEWE [20]	Anomaly-based IDS	Blackhole
Belavagi [22]	Threshold-based IDS	Rank, selective forwarding, wormhole, and DoS attacks
Our SRF-IoT	RPL with trust scheme and external IDS collaboration	Rank, blackhole, and DIS attacks

In our previous work [12,30,31] we implemented two types of DoS attacks, namely DIS flooding and version number modification. Simulations showed that these attacks affect devices' power consumption. Moreover, we provided a high-level design of a threshold-based IDS for protecting IoT networks from these attacks. In our latest work [12], we implemented a first prototype of the proposed IDS in ContikiOS. IDS consisted of two special devices: IDS root and IDS detectors. The developed system had the detection module embedded into the BR, and sensor-like devices, called IDS detectors, executed lightweight algorithms for detecting and reporting malicious behaviour to the BR. Results showed that a high detection rate can be achieved if there are three or more IDS detectors.

In this work, a collaboration between RPL-based devices and IDS is achieved. Our aim is to identify and avoid routing attackers by embedding a trust-based objective function (OF) in RPL protocol. This OF will allow nodes to securely select a parent. The trust-based method works along with an external IDS that provides the monitored devices with useful metrics. Our solution is evaluated against rank and blackhole attacks in a simulation environment, but it can be extended to detect other attacks. Compared to rule-based approaches such as SBIDS [21], our proposed solution supports the detection of multiple attacks and the isolation of attackers by embedding trust concept in the RPL protocol and using an external IDS. The combination of IDS and trust-based methods will help enhance detection performance.

3. Proposed Security Framework for RPL-Based IoT Networks (SRF-IoT)

SRF-IoT is a security framework designed for IoT RPL-based networks. It aims to support the security of an IoT network by identifying and avoiding malicious devices. This is achieved by embedding trust in RPL protocol for choosing the best parent. Additionally, an external IDS, called SRF-IDS, is used to shield the network from internal attackers. A first prototype of SRF-IDS is proposed in our previous work [12,30,31]. It consists of an SRF-IDS root that plays the role of BR and SRF-IDS detectors that have a watchdog mechanism

for capturing network packets. Moreover, DIS flooding can be detected by SRF-IDS. In this work, an SRF-IoT framework is proposed as an extra security measure for RPL-based networks. We focus on protecting the network from rank and blackhole attacks, as those could severely disrupt network operation [32].

3.1. General Concept

In IoT networks, smart devices provide services to their peer connected devices. Evaluating the reliability of a device would enhance networks' security and performance. Bearing this in mind, trust is used in our proposed solution to form a secure network by avoiding malicious actors. According to [33], trust in wireless networks may be defined as a degree of belief to forecast a node's forthcoming actions, which depend on its previous experience and information gained from a device's behaviour. SRF-IoT uses the trust concept to evaluate the reliability of deployed nodes.

Trust is utilised in this work to enhance the RPL routing protocol. Although some security mechanisms exist in RPL, attacks such as blackhole and rank attacks may occur in a real-world IoT network. Embedding trust in RPL will enable devices to learn their neighbours and choose the best parent based on this value. This knowledge is gained through IDS, which processes sniffed packets and sends various metrics to monitored smart devices. In the monitored network, IoT devices get the data from SRF-IDS, and based on some algorithms they calculate the trust value. More weight is applied on the current behaviour of a node than its history. This is to avoid cases where a malicious node has a high trust value initially, and then starts to attack others. Ranking and selecting neighbouring nodes based on their latest trust value helps to avoid malicious actors. Therefore, trust and IDS concepts are utilised as a method of defence to detect and avoid these attacks. A detailed description of the framework is discussed in Section 4.

Trust value is calculated as the number of successfully forwarded packets between the node and its neighbours for a specific time period. As discussed in other trust systems [24,28,34], there are two types of trusts: direct and recommended trust. Direct trust is calculated by the node after monitoring its direct neighbour's packet forwarding behaviour [28]. In contrast, recommended trust can be seen as a recommendation from a third party node. Basically, recommended trust is the trust value given by a third node that is 2-hops away, and it recommends its direct neighbour to other nodes. However, recommended values cannot always be trusted, and a third-party node might provide wrong information. Therefore, only selected nodes can be used to provide nodes with trust recommendations.

In our work, we use direct trust for securing RPL protocol. Each node computes the trust value of its direct neighbours based on the information received from an SRF-IDS detector. Thus, nodes do not waste energy on neighbour monitoring. The SRF-IDS is used as an information collector entity. The resulting trust value is used by each node to select the best parent. Devices with high trust value are selected as parents, and those with lower trust values are avoided. Nodes that fall below a trust threshold are blacklisted. The main goal is to secure the network by (i) routing packets through nodes with high trust scores, and (ii) avoiding malicious nodes or nodes with low trust scores. The proposed framework consists of four procedures: gathering information from SRF-IDS, calculating trust, monitoring trust, and identifying malicious nodes.

Figure 1 presents a high-level architecture of the SRF-IoT scheme. On the left side, the external IDS, called SRF-IDS, is shown along with the internal components. SRF-IDS is responsible for packet sniffing, monitoring nodes' behaviour, and updating monitored nodes with trust metrics. The transfer of trust metrics from SRF-IDS to monitored nodes is done by transmitting special control packets. These three procedures belong to the Trust Monitoring (TM) module that is embedded into SRF-IDS detectors. Moreover, the SRF-IDS root has an embedded detection module for detecting attacks such as DIS flooding. A traditional firewall is also used to block external attackers. On the right side, the basic functionality of SRF-OF, which is embedded into monitored nodes, is represented. Basically,

a monitored node receives the trust metrics from SRF-IDS. It then calculates the new trust values and updates the blacklist with suspicious nodes. The parent selection algorithm is based on the calculated trust value of the specified node.

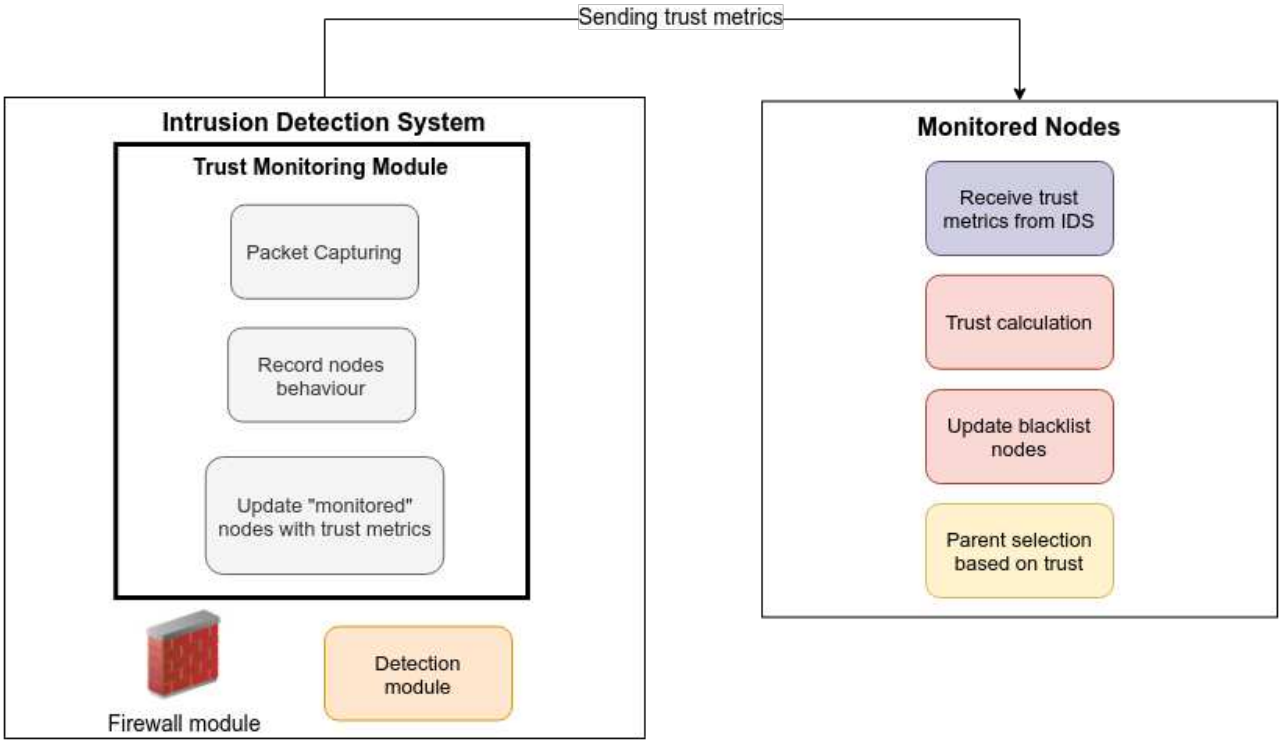

Figure 1. SRF-IoT framework high-level architecture.

3.2. Topology

IoT networks usually are deployed in mesh topology. Devices in mesh topology route packets with each other directly. In our case, the topology is shown in Figure 2. Our approach takes advantage of RPL specification that allows operation of one DODAG with two different RPL instances [6]. This is already supported in operating systems (OS) such as Contiki-NG that implement RPL protocol, so no extra modifications are needed. It is a different architecture from our initial topology shown in the first version of SRF-IDS. This is because having two RPL instances allows the operation of two networks at the same time. The RPL instance with ID equal to zero is the one that is monitored for suspicious activity and we call it a monitored network. It has one sink/router that acts as BR, and several devices that can be benign or malicious.

SRF-IDS forms a second network inside the DODAG with RPL instance ID equal to one, which contains the SRF-IDS root and SRF-IDS detectors. These two networks belong to the same DODAG but are two different RPL instances. This helps SRF-IDS to distinguish packets coming from neighbour monitored network easily. We assume that the RPL instance of SRF-IDS is secured by allowing only authenticated SRF-IDS devices to participate and encrypting packet contents using symmetric encryption. Therefore, it is protected from malicious attackers.

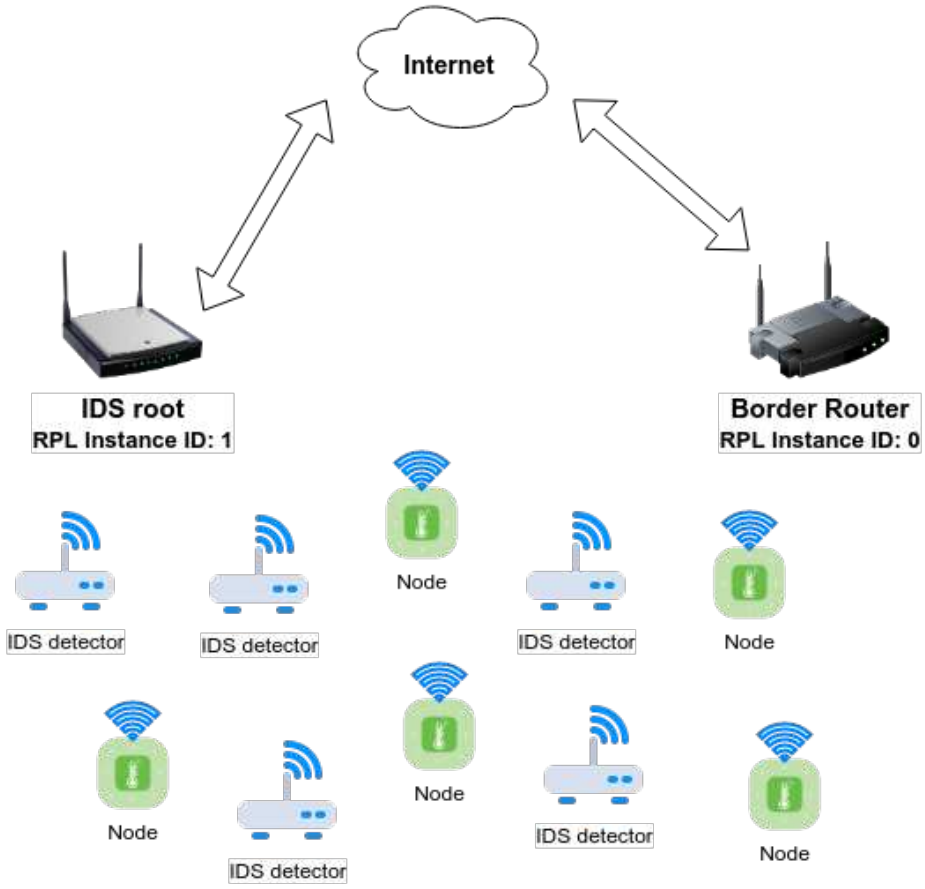

Figure 2. Network topology.

4. SRF-IoT Design

The proposed framework consists of an anomaly-based intrusion detection system (SRF-IDS) and an improved trust-based version of RPL protocol. The IDS from our previous work is enhanced to allow the operation with the monitored network. Moreover, decisions are taken in a distributed way. Apart from IDS, the RPL protocol is modified to consider trust values as method of evaluation for parent selection. We define the following terms:

Monitored network: The network that is being monitored by the SRF-IDS.
Monitored nodes: The nodes that belong to the network that is being monitored by SRF-IDS.
Neighbour: The node N_b is a neighbour of N_a only if N_b is in transmission (TX) range of N_a. That means N_b could provide a route to sink/BR.

Below, the various processes and the design of SRF-IoT scheme are described in detail.

4.1. External SRF-IDS

A main component of SRF-IoT scheme is the external SRF-IDS. It consists of two types of devices: an SRF-IDS root and SRF-IDS detectors. The SRF-IDS root is actually a router responsible for taking final decisions for malicious nodes, whereas SRF-IDS detectors are sensor devices operating in promiscuous mode to gather information from the monitored network.

One of the novelties of this work is the ability of SRF-IDS to operate independently in a different RPL instance without interfering with the monitored network. This is achieved by using a unique RPL InstanceID in packets, only for SRF-IDS nodes. We revised our hybrid-based IDS from previous work [12,30,31] by adding more detection mechanisms for RPL-based attacks, allowing local decision-making by SRF-IDS detectors, and enabling communication between SRF-IDS detectors and neighbour RPL networks. Specifically,

the new SRF-IDS can monitor a network for blackhole and rank attackers. However, the actual detection is done by the monitored nodes themselves and not by SRF-IDS. A malicious actor usually combines blackhole and rank attacks so that nodes select the attacker as parent, and then all packets going through the attacker are dropped. To detect blackhole attackers, SRF-IDS detectors are deployed near the devices of the monitored network so that network packets are captured and analysed. After processing network traffic, SRF-IDS detectors can decide locally if a suspicious node exists or not, and in the former case an alert with useful metrics is sent to nearby monitored devices. Hence, SRF-IDS operates in a decentralised way to combat malicious actors more efficiently.

Another feature is that SRF-IDS detectors are able to send RPL control messages to IoT devices in the monitored network to inform about the possible attacker. Implementing this required a minor modification in the monitored devices so they can correctly parse the SRF-IDS RPL packet. Also, a different RPL InstanceID was needed as discussed in Section 3.2. In this way, a communication between the SRF-IDS network and the monitored network is achieved and malicious nodes are isolated. As an extra security feature, the SRF-IDS root has an embedded firewall to block malicious IPs. In case a malicious node is detected, its IP is forwarded to the SRF-IDS root for blacklisting. For other attacks, such as DIS flooding, SRF-IDS detectors report suspicious nodes to the SRF-IDS root for further decision making [12].

4.2. Gathering Information from SRF-IDS

The first and most important process of the system is to collect information about the network. SRF-IDS is responsible for this, using the deployed SRF-IDS detectors. Specifically, SRF-IDS detectors capture network traffic from monitored networks and save packet metadata in a local database. An algorithm then runs to confirm if packets are forwarded or not to the next hop. The outcome of the algorithm is communicated with the devices belonging to the monitored network for further processing. Only monitored devices that are in TX range of SRF-IDS detectors will receive the metrics. Operating SRF-IDS in promiscuous mode in a different RPL instance is a novelty that allows easy deployment like a plug-n-play system. Additionally, no extra energy is consumed from smart devices in the monitored network to sniff any traffic. All the processing is done in SRF-IDS detectors, and only useful metrics are transferred to monitored network. It is important to note here that nodes in the monitored network can be benign or malicious. SRF-IDS detectors will try to communicate with any type of neighbour in its TX range because it is impossible to know which node is an attacker or not. We assume that SRF-IDS packets will be encrypted to avoid being exploited by attackers.

4.3. Calculating Trust

The calculation of trust value of a node until time T occurs in this process. The formula to calculate the direct trust that device D_a holds for device D_b until time period T is given by $DT(D_a,D_b)_T$. The number of packets successfully transmitted between devices D_a and D_b until time period T is given by $PT_{ab}(T)$. The total number of packets forwarded by device D_b on behalf of device D_a until time period T is given by $PF_{ba}(T)$. Intuitively, the higher the number of packets D_b drops (i.e., $PT_{ab}(T) - PF_{ba}(T)$) the lower the trust D_a has in D_b should be. In order to be able to compare between devices, we normalised this by dividing it to the total number of packets that D_b forwards. Therefore, now the higher the proportion of the number of packets D_b drops to the total number of packets forwarded by D_b (i.e., $[PT_{ab}(T) - PF_{ba}(T)]/PF_{ba}(T)$) , the lower the trust D_a has in D_b should be. The outcome of this formula, i.e., $[PT_{ab}(T) - PF_{ba}(T)]/PF_{ba}(T)$, ranges between 0 (if all packets are forwarded) and infinity (if D_a sends many packets, but none are forwarded). Normalisation was the next step so that we get a value between 1 and 0, with 1 corresponding to the former

case (maximum trust) and 0 corresponding to the latter case (minimum trust). In order to achieve it, the following function was used:

$$f(x) = \frac{1}{(1+x)} \tag{1}$$

The above function takes $x = 0$ to $f(x) = 1$ and $x = \infty$ to $f(x) = 0$. The value achieved from $[PT_{ab}(T) - PF_{ba}(T)]/PF_{ba}(T)$ needs to be scaled to reflect varying degrees of trust in different situations. For example, if a device is initially forwarding all packets, it has high trust value. In a later moment, if it behaves maliciously and drops the packets, its trust value should be reduced. For this purpose, a weight factor w is added before applying the f function. A weight factor is added to the equation to punish or reward nodes that may change packet forwarding behaviour accordingly.

Based on the previous explanations, direct trust calculations are computed as follows:

$$DT(D_a, D_b)_T = \frac{PF_{ba}(T)}{PF_{ba}(T) + w \cdot [PT_{ab}(T) - PF_{ba}(T)]} \tag{2}$$

which is the result of multiplying the numerator and denominator of initial formula f by $PF_{ba}(T)$. Weight factor w can take the following values:

$$w = \begin{cases} 0.6, & \text{if } node\_verified_T \equiv 0 \text{ and } PFI(t) \equiv 0 \\ 0.8, & \text{if } node\_verified_T \equiv 0 \text{ and } PFI(t) > 0 \\ 0.85, & \text{if } node\_verified_T \equiv 1 \text{ and } PFI(t) \equiv 0 \\ 0.5, & \text{if } node\_verified_T \equiv 1 \quad \text{and } PFI(t) > 0 \text{ and} \\ & \qquad\qquad PF_{ba}(T) > minimum\_fw \\ 0.0, & \text{otherwise} \end{cases}$$

where $PFI(t)$ represents the number of packets forwarded, sniffed by SRF-IDS, for the specified node at time t. To calculate the total number of packets forwarded until time T we use:

$$PF_{\text{ba}}(T) = \sum_{n=1}^{T} PFI(n) \tag{3}$$

The parameter $node\_verified_T$ means that a node is verified by SRF-IDS that is behaving normally until moment T. We use this indicator to increase the weight factor so that a node is trusted by benign nodes, whereas an unverified node has a smaller weight. The fourth case has a condition that w is smaller if the node is verified, forwards packets, and the total number of packets forwarded are greater than the minimum number of forwarded packets ($minimum\_fw$). The value of $minimum\_fw$ is 5, and it is used as an indicator to check if a node keeps forwarding packets after the initial verification. If a node is verified but the total number of forwarded packets are less than this parameter, weight becomes zero and trust is 100%. This ensures that a verified benign node will be fully trusted until it reaches the threshold $minimum\_fw$. Weight is then applied in the formula.

Weight factor plays an important role because trust value depends on the obtained behaviour of the node. The general idea of weight factor is to use it in the OF formula so that a high trust score is calculated for an unverified node that behaves normally, keep the trust score at same levels if a node keeps forwarding packets to avoid unnecessary parent switches, and assign low trust score once a node does not or selectively forwards data packets. The values were chosen after various experiments so that a fair and balanced value is calculated for each node by the OF. The methodology followed was to initially assign zero weight was for fully trusted nodes and 1 to malicious nodes. However, simulation results showed that SRF-IDS detectors were not accurate in detecting malicious attackers. This happened especially in cases where a device is trusted initially and later attacks the network. Therefore, we decided to create different weight values with specific conditions to represent different cases and to improve the detection performance. As the network

starts forming, we wanted to give all devices the chance to have high trust value. For this reason, initially the weight factor is 0.6 if nodes do not forward packets and are not verified. Moreover, in the case where nodes are still unverified but some packets are forwarded, the weight value increases by 0.2. This was done to stop nodes that behave suspiciously at the beginning but later behave normally. In the other case where a node is treated as benign and then becomes an attacker, we wanted to lower trust immediately. That is why the value of weight factor was 0.85. In the last case, we assigned 0.5 to weight factor to balance the trust value and assign a fair value to the node that is verified as benign. The zero weight factor, as explained, is assigned to a verified benign node which has not reached the minimum packets forwarded threshold and it is fully trusted.

The verification of a node is done with the help of the SRF-IDS component. The SRF-IDS detector keeps the following fields in a structure for a monitored neighbour node:

PF_{ba} = the total number of packets forwarded from device D_b on behalf of device D_a.

$dstIP_a$ = an array of IP addresses that device D_a is usually sending the packets. The value is taken from the destination IP field in the packet.

$verifiedIP_a$ = a field indicating if the IP address of a node is verified or not. Initially, all nodes are unverified until SRF-IDS verifies them. A node is verified if it forwards a packet to next hop.

4.4. Monitoring/Updating Trust

Keeping the trust value up to date is significant to avoid interruption of network operation from malicious actors. Hence, SRF-IDS constantly monitors the network and sends updated metrics to the monitored nodes. SRF-IDS sends the updated metrics in two modes: interval-based and trickle-based. Interval-based transmission occurs every 3 min from SRF-IDS detectors to devices in monitored network so that malicious nodes are identified in a short time. Before packet transmission, SRF-IDS detectors verify that new metrics are actually available to send to monitored nodes; otherwise they skip the procedure.

Trickle-based transmission is based on RPL trickle timer implementation for transmitting DIO packets [35]. Trickle timer is a dynamic mechanism embedded into RPL that tries to minimise the transmission of RPL control packets. SRF-IDS detectors send packets with updated metrics each time the trickle timer resets. Sending packets in two different modes ensures that SRF-IDS packets arrive successfully and at the proper time to monitored nodes to choose their best parent. Without any metrics, benign nodes use MRHOF. Once a benign monitored node receives a packet from SRF-IDS, it calculates candidate parent's confidence value based on the new measurements. Metrics for monitored nodes are stored in SRF-IDS detectors and they are reset every 15 min to avoid storage capacity problems.

Trust values have different scales as shown in Table 2. We defined the interval for the first three trust levels to be 25 because the weight factor may affect the calculations and cause rapid changes to the trust value. As a result, a big interval was needed to avoid unnecessary changes when the best parent algorithm is executed. In the higher levels of "High Trust" and "Full Trust", the interval is defined to be 11 and 14, respectively. There is no specific reason for the interval. However, it is important to note that once a node reaches a trust value of 76 or more, it is considered trusted and the weight factor becomes zero. Therefore, the node will be in the "Full Trust" scale. If trust falls below the *min_threshold* (less than 26), then a device is considered malicious and it is blacklisted. In addition, RPL local repair is triggered to allow nodes find new parents. In the opposite case, if a node is blacklisted and the trust value is above 50, the node is removed from the list of malicious nodes. Initially, trust value 63 is assigned by default to all nodes joining the network. This is to allow nodes to choose the best parent using other metrics, such as rank, until trust metrics become available. Lists with malicious nodes are stored locally in each benign node so that parent selection algorithm avoids blacklisted nodes. In the case where a node stops attacking, the SRF-IDS will recalculate it's trust value. However, weights are adjusted and the node will gradually become fully trusted.

Table 2. Trust scale.

Values	Explanation	Actions
≥0 and ≤25	No Trust	Avoid node
≥26 and ≤50	Low Trust	Select only if no other option exist
≥51 and ≤75	Medium Trust	Select only if other nodes are below this rank
≥76 and ≤86	High Trust	Best candidate parent
≥87 and ≤100	Full Trust	Ideal parent, select without comparisons

4.5. Identifying Malicious Nodes

In case a node starts attacking the network using blackhole and rank attacks, SRF-IDS will immediately notice this behaviour in the packet forwarding metric. A packet is then sent to the monitored network so that nodes will assign a low trust for these neighbours. The algorithm used by SRF-IDS detectors to detect the various routing attacks is shown in Figure 3. The procedure starts by enabling a promiscuous mode in SRF-IDS detectors to start sniffing network traffic. Once a packet is captured, its packet type is checked and proper thresholds are used to determine if the network is under DIS flooding attack. Specifically, if the packet interval is above the predefined $threshold_{DIS}$, then the SRF-IDS detector alerts the network administrator for a possible DIS flooding attack, the node is reported to the SRF-IDS root for blacklisting, and the SRF-IDS detector continues packet sniffing. If the packet interval is at normal levels, the packet is checked if it needs to be forwarded by the received node. This is done by checking the destination IP (*dstIP*) field to be different with the next hop IP (*next_hopIP*). In the case where *dstIP* is equal to the received node IP, the packet is discarded and the procedure restarts. Otherwise, the *dstIP* is stored in a table for further checks. SRF-IDS detectors continue to sniff packets in order to decide if the neighbouring node actually forwards the packets. This is translated into the following condition: if the captured packet source IP (*srcIP*) is equal to *next_hopIP* then it means the neighbour (next hop) node transmitted the packet. The next check is to validate the destination node. If the stored packet IP (*stored_pkt_dstIP*) is equal to the new destination IP (*dstIP*) then the packet is forwarded correctly, and the node's packet counter is increased by one point. In case *srcIP* equals *next_hopIP* but the *dstIP* is not expected, it means the packet is not forwarded, and the SRF-IDS flags the device as a possible malicious node. The last step is to discard the packet and continue capturing network packets for other nodes.

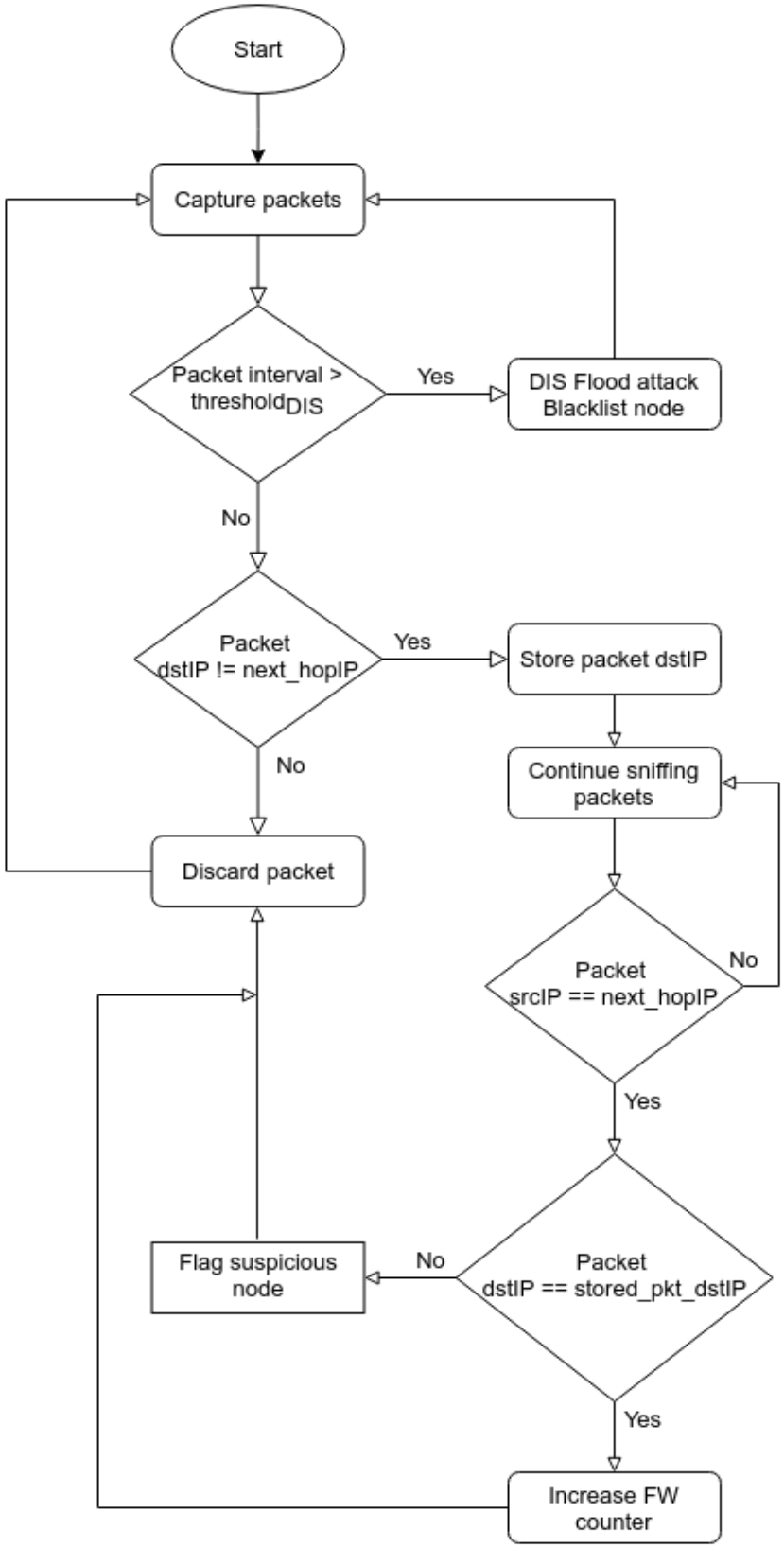

Figure 3. SRF-IDS detectors operation flow chart.

5. Implementation Steps

The implementation details of this work are described in this section. First, the development of the specific RPL routing attacks is explained. Next, the implementation of the SRF-IDS communication components is described. The last subsection presents the algorithms developed for the SRF-OF component.

5.1. Rank and Blackhole Attacks

In our previous work [12], we extensively studied the impact of IoT-specific DoS attacks, namely DIS flooding. This attack is based on the RPL routing protocol and affects the availability of the network. In this work, a combination of rank and blackhole attacks is implemented in Contiki-NG. Contiki provides an implementation of RPL, called ContikiRPL [36], whereas Contiki-NG implements a lighter version called RPL-lite.

According to [14], in rank attacks usually a malicious node may intentionally advertise a lower rank in order to attract neighbouring devices to select it as the preferred parent. A parent node is needed in order to form the DODAG network and allow creation of routes reaching the BR. In cases where networks are small, the best parent is the BR itself. In other cases, metrics such as rank and ETX are used to select the best parent. If a malicious node manages to be chosen as the best parent of several nodes, it can attack the network, affecting its availability and integrity. As a result, the malicious node is the single point of failure of the network. In this work, rank attack is implemented in RPL-lite protocol of Contiki-NG. The implementation is shown in Algorithm 1. Rank attackers advertise a fake rank with value 129, whereas the root node has a value of 128.

Another interesting attack is the blackhole. A blackhole attacker can degrade the performance of the IoT network by dropping all incoming packets. As a result, no packets are forwarded from this node. Algorithm 2 shows our implementation in Contiki-NG. All incoming packets are dropped only if they need to be forwarded. The attack begins after 2 min of simulation and lasts until the end of the simulation. Dropped packets by malicious attackers can be easily distinguished from the packets dropped due to interference because blackhole attackers first receive a packet and, then, the algorithm decides if it will be dropped or not. In the case of interference, nodes are unable to receive packets, and thus packets are not considered in the calculations.

A combination of blackhole and rank attacks is implemented and tested in our experiments. They are combined to achieve the highest impact in the network by attracting neighbouring nodes and dropping all packets. A special node, called *udp-client-malicious*, is running these attacks in our scenarios.

Algorithm 1 Rank attack implementation

1: **Function** $rpl\_icmp6\_dio\_output(ip\_address)$:
2: $dio\_packet \leftarrow create\_dio\_packet(ip\_address)$;
3: $dio\_packet.dag\_rank \leftarrow 129$;
4: **Send packet**

Algorithm 2 Blackhole attack implementation

1: **Input:** *Packet* Pkt_{ij}
2: **Function** $uip\_process(Pkt_{ij})$:
3: $next\_hop \leftarrow Pkt_{ij}.next\_hop$;
4: $time\_attack \leftarrow 2$;
5: **if** $next\_hop \neq NULL$ and $current\_time \geq time\_attack$ **then**
6: **drop packet**
7: **end if**

5.2. Communication between SRF-IDS and Monitored Devices

Communication and routing among devices is handled by the RPL-lite protocol. The SRF-IDS and monitored networks operate as usual in two different RPL instances. Nodes accept packets destined only for their RPL instance. Therefore, a solution was needed to allow SRF-IDS to alert monitored devices about attackers. As discussed in Section 4.4, the SRF-IoT scheme works with the help of an external SRF-IDS. A first prototype of SRF-IDS was presented and evaluated for detecting DIS attacks in our previous work [12]. However, as already discussed in Section 4.1, more features are added. The new implementation includes the possibility of monitoring devices operating in a different RPL instance, detecting packet forwarding behaviour of a monitored node, and communicating with neighbouring devices. Apart from that, the detection mechanism for DIS attacks is not affected.

Allowing SRF-IDS to communicate with monitored nodes is essential to enable SRF-OF to calculate trust correctly. Trust is calculated by each node using the metrics received from SRF-IDS detectors. To achieve this, we implemented a new ICMPv6 control message in Contiki-NG. The implementation was done for SRF-IDS detectors and the nodes that could be monitored. SRF-IDS detectors are able to send special packets to nodes in RPL InstanceID zero, and include metrics noted in Section 4.3.

Algorithm 3 presents the actual implementation that is used by SRF-IDS detectors to send packets to neighbouring devices. As it is shown, SRF-IDS detectors iterate over their neighbours and create a buffer that contains the metrics. For each neighbour, SRF-IDS detectors send the IP address of the node, the *verifiedIP* flag, and the number of forwarded packets. After adding the metrics into the buffer, the SRF-IDS detector sends the ICMPv6 message using a custom RPL code to the neighbouring device. Moreover, the detector sends the same information to SRF-IDS root for detecting potential blackhole attackers. If no information is available for a device, the SRF-IDS detector does not send any packets to neighbours. Custom ICMPv6 packets are parsed by benign/malicious nodes to extract metrics and calculate trust using the appropriate formulas. We assume that these ICMPv6 packets are encrypted and only nodes participating in the monitored network can read its contents. Although malicious nodes can receive and read packet contents, the metrics will be related to their parents. Therefore, it will be useful even for them to select a benign parent based on metrics.

In cases where a monitored node receives metrics from multiple SRF-IDS detectors, an appropriate mechanism is in place to handle these cases. For example, let M_a be the monitored node, IDS_a and IDS_b the SRF-IDS detectors, and D_a a candidate parent. If IDS_a sends a packet with metrics to M_a for device D_a that contains $verified_a = 1$ and $packets\_forwarded_a = 5$, the M_a will store it normally. In a later moment, if IDS_b sends another packet that contains $verified_a = 0$ and $packets\_forwarded_a = 0$, node M_a will aggregate the knowledge and calculate trust with the proper weight factor. Node M_a will consider its actual transmitted packets to check if $packets\_forwarded_a = 0$ is correct or not. A candidate parent is verified after consecutive notifications arrive by multiple SRF-IDS detectors.

Algorithm 3 SRF-IDS communication function

```
1: Function ids_output_to_benign(ip_addr) :
2: for each N in nbr_table do
3:   linkaddr * lladr ← nbr_table_get_lladr(nbr_table, N);
4:   uip_ipaddr_t * ipaddr2 ← NULL;
5:   ipaddr2 → u8[0] ← 254; {Add address prefix}
6:   ipaddr2 → u8[1] ← 128;
7:   unsigned char *buffer ← UIP_ICMP_PAYLOAD; {Create new buffer}
8:   uint16_t pos ← 0; {Set RPL instance ID}
9:   buffer[pos++] ← 0;
10:  buffer[pos++] ← N → destParents; {Number of neighbours}
11:  for int j = 0; j < N → destParents; j++ do
12:    buffer[pos++] ← N → destIP[j];
13:    buffer[pos++] ← N → verifiedIP[j];
14:    buffer[pos + 2] ← N → count_fw_packets[j];
15:    N → count_fw_packets[j] ← 0;
16:    N → verifiedIP[j] = 0;
17:    N → destIP[j] = 0;
18:  end for
19:  if N → destParents > 0 then
20:    uip_icmp6_send(ipaddr, ICMP6_RPL, RPL_CODE_IDS_NORM,
     sizeof(buffer));
21:    uip_ipaddr_t addr2;
22:    if get_root_ipaddr(addr2)! = NULL then
23:      uip_icmp6_send(addr2, ICMP6_RPL, RPL_CODE_IDS2, sizeof(buffer));
24:    end if
25:  else
26:    printf("No information available");
27:  end if
28: end for
```

5.3. Security Framework Objective Function (SRF-OF)

The trust concept is implemented as a new OF in RPL-lite protocol, called security framework objective function (SRF-OF). The SRF-OF algorithm for choosing the best parent is presented in Algorithm 4. First, nodes check if neighbours are acceptable as parents. An acceptable node has low link metrics or path cost. Next, checks are done to avoid blacklisted and malicious nodes that were detected in previous attempts. A neighbour node with high trust value and smaller rank than the current node's rank is selected as the parent (lines 17–21). In case Algorithm 4 reaches the last condition (line 27), it returns the parent with the lowest ETX value. The last condition as well as the whole SRF-OF implementation includes appropriate mechanisms to achieve stability in parent selection and to avoid unnecessary parent switches. It is important to have a stable network and minimise parent switches to reduce energy consumption overhead.

The trust value for each neighbour is computed in Algorithm 5. Specifically, monitored nodes receive the packet at time T from SRF-IDS, extract metrics from the packet for a specific neighbour, and store the measurements to the corresponding variables. For the trust calculation, a node uses the formula shown in Section 4.3, which takes into account the actual number of packets sent to the neighbour until time T to the number of packets forwarded by neighbour and captured by SRF-IDS detectors until time T.

SRF-OF utilises the following metrics during best parent calculations: trust value, rank, and ETX. A combination of these metrics would allow nodes to choose the most trusted and reliable parent. SRF-OF is implemented in Contiki-NG for both benign and malicious nodes. SRF-IDS uses the default MRHOF as the objective function, and the rest nodes use SRF-OF.

Algorithm 4 SRF-OF algorithm

Input: Neighbour nodes nbr1 and nbr2 from nbr table
Output: Best neighbour/parent to route packets
Function $best\_parent(nbr1, nbr2)$:
$int\ nbr1\_is\_acceptable \leftarrow (nbr1! = NULL$ and $nbr\_is\_acceptable\_parent(nbr1))$;
$int\ nbr2\_is\_acceptable \leftarrow (nbr2! = NULL$ and $nbr\_is\_acceptable\_parent(nbr2))$;
if nbr1!=NULL **and** nbr1_is_acceptable **and** is_blacklisted(nbr1) **then**
  if nbr2_is_acceptable **then**
    return $nbr2$;
  end if
  return $NULL$;
else if nbr2!=NULL **and** nbr2_is_acceptable **and** is_blacklisted(nbr2) **then**
  if nbr1_is_acceptable **then**
    return $nbr1$;
  end if
  return $NULL$;
end if
if $(((nbr1 \rightarrow trust\_value > nbr2 \rightarrow trust\_value)$ **or** $(nbr2 \rightarrow trust\_value < 38))$ **and** $(nbr1 \rightarrow rank < current\_rank))$ **then**
  return $nbr1$;
else if $(((nbr2 \rightarrow trust\_value > nbr1 \rightarrow trust\_value)$ **or** $(nbr1 \rightarrow trust\_value < 38))$ **and** $(nbr2 \rightarrow rank < current\_rank))$ **then**
  return $nbr2$;
end if
if $nbr1 \equiv current\_preferred\_parent$ **and** $within\_hysteresis(nbr1)$ **then**
  return $nbr1$;
else if $nbr2 \equiv current\_preferred\_parent$ **and** $within\_hysteresis(nbr2)$ **then**
  return $nbr2$;
end if
if $nbr\_link\_metric(nbr1)$ <$nbr\_link\_metric(nbr2)$ **then**
  return $nbr1$;
else
  return $nbr2$;
end if

Algorithm 5 Trust calculation at time T

Input: Read buffer received from SRF-IDS
for each neighbour **do**
  $nbr \leftarrow$ Get neighbour node details from SRF-IDS buffer
  $node\_verified_T \leftarrow nbr \rightarrow verified_T$
  $node\_pkts\_forwarded_t \leftarrow nbr \rightarrow pf\_from\_ids$
  if $node\_verified_T == 0$ **then**
    if $node\_pkts\_forwarded_t == 0$ **then**
      $direct\_trust \leftarrow (nbr{\rightarrow}total\_packets\_fw/(nbr{\rightarrow}total\_packets\_fw + 0.6 * (nbr{\rightarrow}total\_packets\_tx - nbr{\rightarrow}total\_packets\_fw)))$
    else if $node\_pkts\_forwarded_t > 0$ **then**
      $direct\_trust \leftarrow (nbr{\rightarrow}total\_packets\_fw/(nbr{\rightarrow}total\_packets\_fw + 0.8 * (nbr{\rightarrow}total\_packets\_tx - nbr{\rightarrow}total\_packets\_fw)))$
    else
      $direct\_trust = 0$
    end if
  else if $node\_verified == 1$ **then**
    if $node\_pkts\_forwarded_t == 0$ **then**
      $direct\_trust \leftarrow (nbr{\rightarrow}total\_packets\_fw/(nbr{\rightarrow}total\_packets\_fw + 0.85 * (nbr{\rightarrow}total\_packets\_tx - nbr{\rightarrow}total\_packets\_fw)))$
    else
      $direct\_trust \leftarrow (nbr{\rightarrow}total\_packets\_fw/(nbr{\rightarrow}total\_packets\_fw + 0.5 * (nbr{\rightarrow}total\_packets\_tx - nbr{\rightarrow}total\_packets\_fw)))$
    end if
  end if
end for

6. Experimental Evaluation of SRF-IoT Framework

The evaluation of the SRF-IoT framework is presented in this section. The scenarios explored are initially described along with the simulation tools. Next, metrics and relevant configurations used are presented. Evaluation results are described in detail in the last subsection.

6.1. Scenarios

We examined three main scenarios: normal, malicious using MRHOF as OF, and malicious using SRF-OF as OF. Specifically, a normal scenario is an environment without any attackers. Nodes are operating normally and are using the default MRHOF to choose a parent.

In the malicious scenario, one or more attackers exist, and all nodes are using the default MRHOF. This scenario is studied to understand the impact that rank and blackhole attacks have in the network. The last malicious scenario is simulating an environment where again benign and one or more compromised nodes exist but nodes are using the new implemented objective function called SRF-OF. SRF-IDS operates in a different RPL instance, and, thus, nodes are using the default MRHOF. SRF-IDS is deployed in all scenarios only for comparison reasons as we increase SRF-IDS detectors in the network to find the optimal case. SRF-IDS detectors capture metrics only and do not help in parent selection in normal and BHR scenarios.

The different types of nodes used in our simulations are shown in Table 3. As can be seen, SRF-IDS nodes have different RPL InstanceID than other nodes. In addition, Sink/BR and SRF-IDS Root play the role of sink for both networks. Benign and malicious nodes are configured to join the network and start sending UDP packets to BR once the DODAG network is formed. However, malicious nodes start attacking the network after 2 min.

The number of nodes deployed in each scenario is presented in Table 4. There are 30 benign nodes in all scenarios and six malicious nodes in the malicious scenarios. A varying number of SRF-IDS detectors are deployed in the network to study and find the optimised

number of detectors to avoid the attackers. In normal and malicious scenarios with MRHOF, SRF-IDS detectors are deployed with the Trust Monitoring module disabled. SRF-IDS is deployed in these two scenarios for comparisons and to help us generate results. The Trust Monitoring module, as noted in previous sections, is enabled only when the SRF-IoT scheme is evaluated and, thus, monitored nodes use the SRF-OF.

Table 3. Node types and configuration.

RPL InstanceID	Node Type	Description
0	BR	Acts as a sink node. Receives messages and sends only UDP replies.
0	Benign node	Uses RPL-lite to create a mesh network and sends data periodically to BR.
0	Malicious node	Uses RPL-lite to join the network and advertises low rank (rank attack) and drops all incoming packets (blackhole attack). Also, it sends UDP packets like benign nodes.
1	SRF-IDS Root	Plays the role of sink in IDS and collects all the information from SRF-IDS detectors.
1	SRF-IDS Detector	Sniffs traffic of monitored network to detect malicious nodes. Stores information about messages exchanged and packets forwarded.

Table 4. Number of node types in each scenario.

	BR	IDS Root	Benign Nodes	Malicious Nodes	IDS Detectors	Total
Normal scenario with MRHOF	1	1	30	-	5 to 15, Trust Module Disabled	37 to 47
Malicious scenario with MRHOF	1	1	30	6 Blackhole and Rank	5 to 15, Trust Module Disabled	43 to 53
SRF-IoT scenario with SRF-OF	1	1	30		5 to 15, Trust Module Enabled	

6.2. Simulation Tools

Several simulators are being used in the literature. In many studies, Contiki-NG is simulated using a Cooja simulator [37], which is included in the Contiki OS. In our previous work [12,31], Cooja was used for emulating hardware. However, as the network becomes larger, the need for more CPU and memory resources increases. These issues are resolved with a new simulator, called the Whitefield framework [38]. According to its author, Whitefield provides a simulation environment for wireless sensor networks by combining realistic PHY/MAC layer simulation with the native mode use of popular IoT operating systems. In our case, we use Contiki-NG as the OS, which provides the network layer and above, whereas the NS-3 simulator provides the PHY/MAC/RDC layer. Moreover, Whitefield generates logs and pcap files for each simulation. This is really useful for monitoring and auditing simulation results. The only drawback of using Whitefield is that deployed nodes are native processes running in NS-3 and not emulated hardware. Therefore, monitoring energy consumption or other hardware-specific metrics is not possible.

6.3. Configuration and Metrics

The settings and metrics utilised for evaluation purposes of the SRF-IoT framework are discussed in this subsection. We considered the median value in our calculations as it is robust against outliers. In many experiments, due to the random behaviour of the attackers,

observed results differ significantly. Therefore, to get a more realistic picture of the results and avoid outliers, we used median value. The metrics used are the following:

- Median Packet Delivery Ratio (PDR): Indicates the median value of the ratio of the total number of unicast packets received by BR up to the total number of unicast packets generated by all benign and malicious nodes. It does not include UDP retransmitted packets.
- Median Parent Switch: Presents the median value of the number of parent switches that benign and malicious nodes execute during the simulation. A parent switch happens to select a better route to the BR. Changing parent results to changing the rank of a node.
- Median Packets Dropped: Shows the median percentage of packets dropped by the attackers during the simulation. Malicious nodes drop all types of packets that normally should be forwarded to next hop.
- Median IDS Packet Overhead: Indicates the median percentage of the SRF-IDS detector's packets sent to SRF-IDS root and the monitored network during simulation. The value is calculated from the number of packets exchanged in N repetitions.

Below, the mathematical definitions for the metrics are provided. Let

$$S_r = \frac{Rcvd}{\sum_{k=1}^{n} P_k} \tag{4}$$

be the packet delivery ratio for repetition r where *Rcvd* is the total number of packets received at BR, k is the number of node sending the packets, n is the total number of nodes, and P_k is the total number of packets sent from node k. The Median PDR, *E[DRpkt]*, is given by:

$$E[DRpkt] = Median(S) \tag{5}$$

where r is the repetition number, m is the total number of repetitions for each simulation, and *Median(S)* is the median value of the set S. Packet delivery ratio from all repetitions are included in the set S to calculate the *Median(S)* value.

The Median Parent Switch, *E[PS]*, is given by:

$$PS_r = \sum_{k=1}^{n} P_k \tag{6}$$

$$E[PS] = Median(PS) \tag{7}$$

where P_k the parent switch of node k, PS_r the sum of parent switches for repetition r, r the number of repetitions for each simulation, and *Median(PS)* is the median value of the set PS. The set PS contains all parent switches for all repetitions so that the *Median(PS)* value is calculated.

The Median Packets Dropped, *E[Dpkt]*, is given by:

$$D_r = \frac{Drop_r}{\sum_{k=1}^{n} TA_k} \tag{8}$$

$$E[Dpkt] = Median(D) \tag{9}$$

where $Drop_r$ the total packets dropped in repetition r, TA_k the total number of packets transmitted from node k including multicast and unicast packets, D_r the percentage of packets dropped in repetition r, m the number of repetitions for each simulation, and *Median(D)* is the median value of set D. The total packets dropped from all repetitions are included in set D so that the *Median(D)* is calculated.

The Median IDS Packet Overhead, *E[IDS]*, is given by:

$$Sent_r = \sum_{k=1}^{n} TA_k \tag{10}$$

$$I_r = \frac{\sum_{a=1}^{b} IDS_a}{\sum_{r=1}^{m} Sent_r} \tag{11}$$

$$E[IDS] = Median(I) \tag{12}$$

where k is the number of node sending the packets, n is the total number of nodes, TA_k in (10) is the total transmitted packets from node k including multicast and unicast packets, $Sent_r$ is the sum of packets sent in repetition r, a is the number of SRF-IDS detector, b is the total number of SRF-IDS detectors IDS_a is the total number of SRF-IDS packets sent from SRF-IDS detector a to SRF-IDS root, m is the total number of repetitions for each simulation, I_r the SRF-IDS packet overhead percentage in repetition r, and *Median(I)* is the median value of set I. The calculated values from all repetitions are added in set I to calculate the *Median(I)* value.

Regarding simulation configuration, the simulator called Whitefield framework provides a configuration file in which proper settings were defined. In that file, various simulation options can be defined by the user. The options used are shown in Table 5. Simulation execution time is defined as 60 min. Normal and SRF-IoT scenarios using SRF-OF are repeated 10 times, whereas malicious scenarios using MRHOF are repeated only four times. The reason for repeating the malicious scenarios only four times is that SRF-IDS had the TM module disabled, and no significant difference was observed when the scenarios were repeated four or more times. Regarding normal and SRF-IoT scenarios, the number of repetitions is higher as the observed results varied, and we wanted a large sample for analysis. SRF-IDS detectors in normal and SRF-IoT scenarios are used only to capture traffic and calculate metrics based on the observed measurements. Consequently, using more SRF-IDS detectors allow us to collect a large sample for analysis. In contrast, SRF-IoT scenarios have SRF-IDS deployed with TM enabled. This is to investigate how well the proposed security framework isolates attackers and find the recommended number SRF-IDS detectors that have to be deployed in a network for best protection. The seed number plays a significant role in simulations. It affects the behaviour of the nodes regarding processing and packet transmission times. Therefore, we use random seed in each repetition so that the Whitefield simulator produces random results in each run. The simulator's default configuration of MAC packets re-transmissions is three times, and the maximum number of packets waiting in the buffer in the MAC layer is 20.

Table 5. Simulation configurations.

Parameter	Value
Grid size	70 × 70
Topology	Random
Simulation time	60 min
Seed number	Random in each execution
Max MAC packet retries	3
Max buffered packets in MAC layer	20
Operating System	Contiki-NG 4.4
Simulator	Whitefield simulator

6.4. Evaluation Results

Evaluation results from simulating the SRF-IoT scheme are presented in this subsection. From this point, we reference to malicious scenario using MRHOF as BHR scenario, normal scenario using MRHOF as normal scenario, and malicious scenario using SRF-OF as SRF-IoT scenario.

Starting with Figure 4, the Median (PDR) of SRF-IoT framework is presented for scenarios where 5 to 15 SRF-IDS detectors are deployed. Overall, PDR in monitored network is kept at high levels with the help of the SRF-IoT framework. This can be clearly identified from the figure, as the median PDR starts from 90.8% when 5 detectors are deployed, then reaches the maximum of 92.8% in scenario with 7 detectors, and then the PDR declines as the number of detectors increases, reaching the minimum of 82.4% with 15 SRF-IDS detectors. This is expected because multiple SRF-IDS detectors generate extra overhead in the network and monitored nodes may receive other nodes' metrics, not related to their direct neighbours. Therefore, monitored nodes drop more unrelated packets, reducing the PDR metric.

A comparison of the Median PDR of different scenarios is shown in Figure 5. Normal scenario has a median value of almost 95% in all simulated cases, whereas in the BHR scenario this percentage drops more than 15%. This big difference shows how attackers can degrade the performance of the network. Comparing the median number of PDR in SRF-IoT scenarios versus normal scenarios, we can see that a small difference of a minimum 5% and maximum 13% exist. This means that as more SRF-IDS detectors are deployed, the SRF-IoT framework's performance declines. However, those results indicate that SRF-IoT framework can assist nodes to avoid extra processing and energy overhead and operate in the same levels as in normal scenarios.

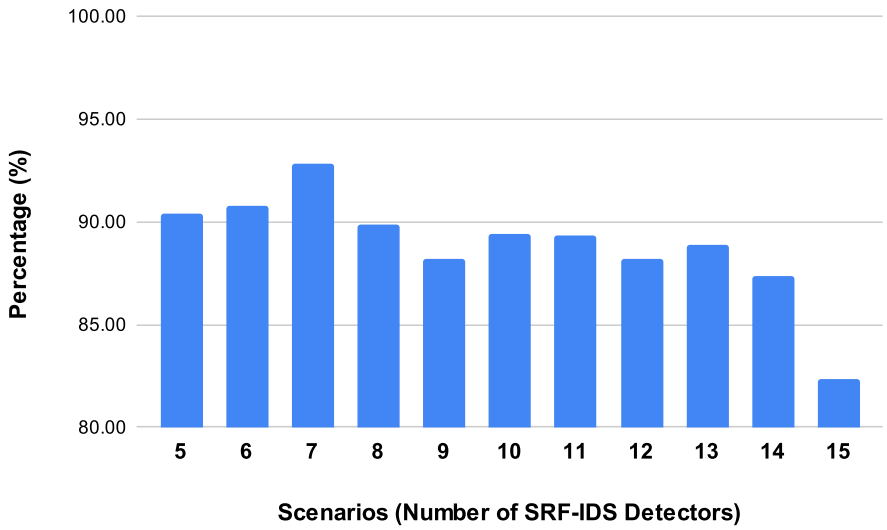

Figure 4. Median packet delivery ratio (PDR) results in SRF-IoT scenario after deploying 5 to 15 SRF-IDS detectors.

Looking at the Median Parent Switch in Figure 6, results depict that our proposed framework requires fewer than 200 parent switches when deploying fewer than 15 SRF-IDS detectors. Specifically, the bar chart shows that SRF-IoT scheme has a median of 140 parent changes with 5 deployed SRF-IDS detectors, declining to 97 changes with 7 detectors, and then going up to 258 when SRF-IDS detectors are 15. It is clear that when having more than 12 SRF-IDS detectors, parent changes are almost doubled. An explanation is that as we increase SRF-IDS nodes, multiple detectors monitor similar nodes, and they send multiple metrics for the same monitored nodes. This could lead to high or low trust values which in turn leads to parent changes. The normal scenario indicates that parent switch should occur fewer than three times per node on average.

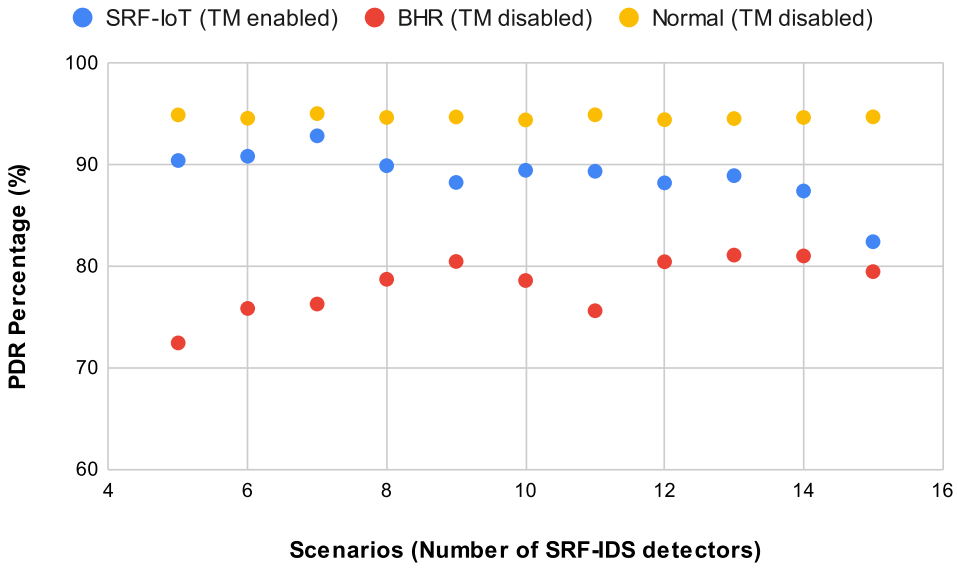

Figure 5. Comparing median packet delivery ratio (PDR) per scenario after deploying 5 to 15 SRF-IDS detectors.

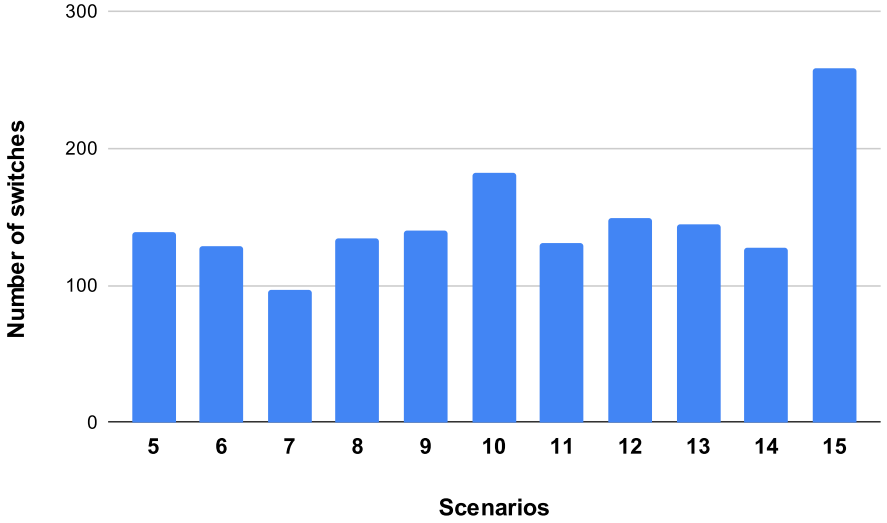

Figure 6. Median parent switch results in SRF-IoT scenario after deploying 5 to 15 SRF-IDS detectors.

Figure 7 compares the values of the Median Parent Switch metric for three different scenarios: SRF-IoT, BHR, and normal. Generally speaking, the scatter chart indicates that SRF-IoT framework achieves low parent changes, with a small increase from the levels of normal scenarios. Looking more closely, using the SRF-OF, nodes change parents almost three times less than in BHR. Attacking the network causes approximately 600 parent changes on average, whereas in SRF-IoT we have fewer than 190 switches, and in normal scenarios we have a median of 30 switches. The only exception is the case with 15 SRF-IDS detectors in which SRF-IoT sees a surge in parent changes. The high parent switch number explains the low PDR that we previously saw in that scenario. Assuming the selected parent is an attacker, we expect to have more dropped packets, affecting the PDR metric. This means malicious actors successfully affect the network performance using rank and blackhole attacks. Another conclusion is that the SRF-IoT scheme drastically reduces parent switches in almost all cases, and even in the worst case it keeps the network more stable than in BHR scenario in which both PDR and parent switches metrics are high.

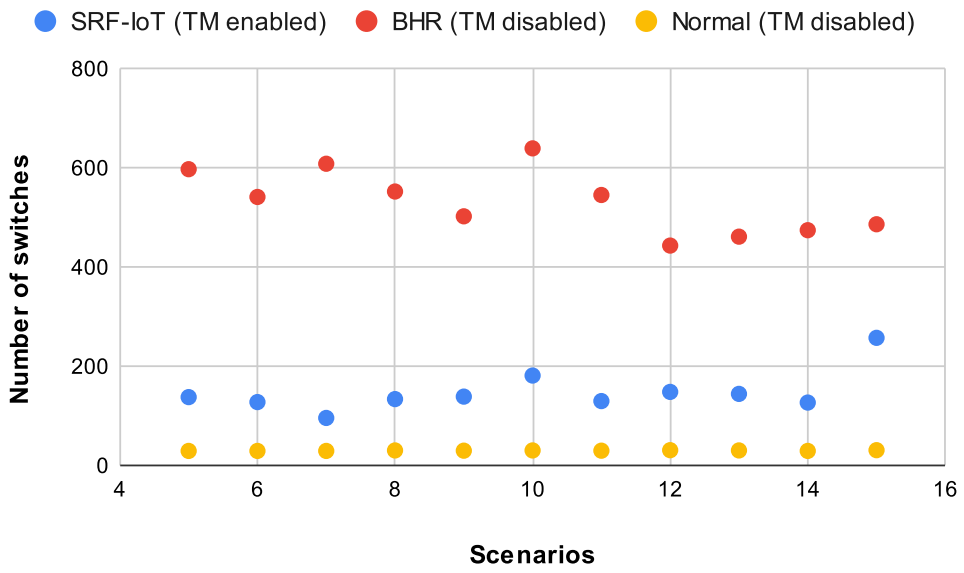

Figure 7. Comparing median parent switches per scenario after deploying 5 to 15 SRF-IDS detectors.

The Median Packets Dropped obtained after deploying 5 to 15 detectors in SRF-IoT and BHR scenarios are displayed on Figure 8. Looking at the BHR scenario, the median percentage of packets dropped is initially 38.6% when 5 SRF-IDS detectors are deployed. The percentage fluctuates around 40% until the scenario with 12 SRF-IDS detectors, in which the percentage declines to 35%. This is related to the previous parent switch metric because in scenarios with higher packets dropped, more parent changes occur. Regarding the SRF-IoT scenario, it has an increasing trend of dropping packets as the number of SRF-IDS detectors becomes larger. A median of 14.2% of the packets are dropped when five SRF-IDS detectors are deployed, falling to 8.2% with seven detectors and then rising to 16.5% with eight detectors. The median value then remains at the same level apart from the scenarios with 12 and 15 SRF-IDS detectors in which the median dropped packets climb up to 20.2% and 25%, respectively. Therefore, our framework may assist the network to avoid blackhole attackers and reduce dropped packets.

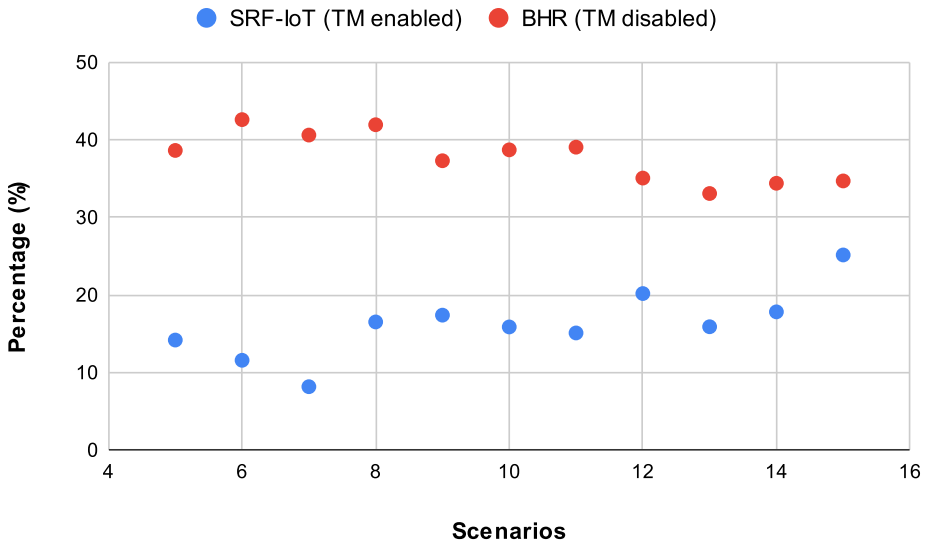

Figure 8. Median packets dropped per scenario after deploying 5 to 15 SRF-IDS detectors.

As a last metric, the Median SRF-IDS Packet Overhead is illustrated in Figure 9. In all scenarios, SRF-IDS generates less than 2.5% traffic overhead in the network. The only case where the SRF-IDS packet overhead is relatively high is in the scenario with 15 SRF-IDS detectors. The median value reaches 2.2% because the monitored nodes are trying to avoid attackers. We have the highest median parent switches in this scenario—the number of dropped packets is also increasing, and thus, SRF-IDS detectors attempt to help monitored nodes by sending them trust metrics. All the previous metrics indicate that the monitored network is greatly affected by attackers in that specific scenario. Generally, the number of packets sent from SRF-IDS detectors depends on the number of monitored nodes. For example, SRF-IDS detectors that are deployed near multiple nodes of the monitored network send more packets in order to update nodes with trust metrics. In our case, SRF-IDS nodes are randomly deployed in the simulated scenarios. As depicted in the column chart, SRF-IDS helps monitored nodes avoid attackers with very low packet overhead. A relatively high packet overhead percentage is depicted in the scenario with nine SRF-IDS detectors. The reason for the high median percentage in the scenario with nine detectors in comparison with other scenarios such as 13 or 14 SRF-IDS detectors is that in the specific case, the SRF-IDS detectors detect malicious attackers in the network and try to help monitored nodes by sending them many packets to alert neighbouring nodes. Some of these packets are lost and others are received by benign nodes. As a result, nodes receiving those alerts try to change parents, and, thus, the median parent switch metric shown in Figure 6 is also high in this scenario.

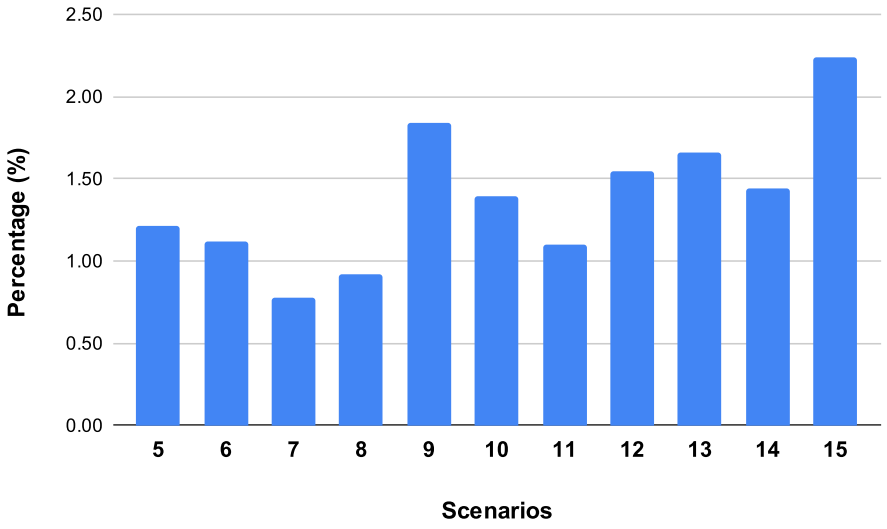

Figure 9. Median SRF-IDS packet overhead in SRF-IoT scenario after deploying 5 to 15 SRF-IDS detectors.

In conclusion, the experimental evaluation of the SRF-IoT framework against rank and blackhole attackers showed higher PDR, lower packets dropped, as well as lower parent switches in comparison to the malicious scenarios where SRF-IoT had TM disabled. Results indicate that the proposed framework can aid nodes to choose the proper nodes as parents and avoid the compromised ones. According to the evaluation results, deploying seven SRF-IDS detectors in a network with at least 36 nodes generates the best results, assuming that one-sixth of them might be compromised. Moreover, results depicted that deploying 5 to 10 SRF-IDS detectors still helps the IoT network to isolate and avoid attackers.

6.5. Comparison with Related Works

This section focuses on the discussion and the comparison of the results obtained from the proposed SRF-IoT framework with other similar studies.

In Table 6, a comparison of our work with four similar studies is presented. Specifically, the evaluation results from the studies of MRTS [24], SRPL-RP [26], and SecTrust-RPL [26,28] are compared with our proposed SRF-IoT framework. The metrics used in the comparison are the PDR, parent switches, and packets dropped. In addition, the total number of nodes deployed in each experiment is compared.

The related works that are used for comparison purposes explore rank or blackhole attackers in medium scale networks. As we wanted to have a similar basis for comparing all the related studies, the results from Section 6 are referenced and used. That specific section evaluates the SRF-IoT framework in a scenario where 30 benign nodes are deployed, and 6 nodes launch a combination of blackhole and rank attacks. Thus, simulations have similar configurations with the other work.

As it can be seen from Table 6, most of the studies provide the PDR metric, whereas only one of them provides the parent switch and another one the packets dropped metrics. In our work, we provide all of the aforementioned metrics. Looking at the PDR metric, the highest PDR value is achieved by SRPL-RP with 98.48% in a small network of 16 nodes plus the 4 rank attackers. The second highest PDR value is from our SRF-IoT framework, which achieves 92.8% in a network of 30 benign nodes plus 6 attackers that launch combined blackhole and rank attacks. The MRTS study follows with a PDR up to 90% in a network with 27 nodes plus the 3 blackhole attackers. SecTrust-RPL exhibits the lowest PDR with a value of 80%, with the same number of nodes are deployed in the network as in MRTS. Comparing the rest of the metrics, parent switches of the SRF-IoT framework are slightly more than the 80 switches observed in the MRTS work. Regarding packets dropped, our work keeps the percentage near 8%, which is very low in comparison with the 22–23% recorded in the SecTrust-RPL study.

Table 6. Comparison of results with similar IoT-related studies. BH: Blackhole attack, Rank: Rank attack.

Study (Attack)	PDR	Parent Switches	Packets Dropped	Number of Nodes and Attackers
MRTS (BH)	up to 90%	>80	-	27 nodes, plus 3 attackers
SRPL-RP (Rank)	98.48%	-	-	16 nodes, plus 4 attackers
SecTrust-RPL (Rank)	80%	-	22–23%	27 nodes, plus 3 attackers
Our proposed SRF-IoT (Rank and BH)	92.8%	97	8.2%	30 nodes, plus 6 BH + Rank attackers

All in all, it has been shown that the proposed SRF-IoT is an effective solution that achieves the best results among the current related studies that deploy fewer nodes and study only single attacks. SRF-IoT was evaluated in a larger network than other studies using a combination of blackhole and rank attacks and demonstrates superior performance in most cases. Nevertheless, given the variation in the results with the variation in the number of detector nodes, the study of the application of machine learning techniques to dynamically optimise the number of detector nodes could be considered as a future enhancement.

7. Conclusions and Future Work

In this work, we studied and implemented rank attack along with blackhole attack in Contiki-NG. We then designed and developed a novel security framework called SRF-IoT for detecting RPL attacks. The proposed method is a trust-based system that utilises an external SRF-IDS to get intelligence and choose the best route for network packets. SRF-IoT is evaluated in Whitefield simulator. Obtained results indicate that the proposed scheme helps nodes to avoid malicious attackers successfully, reduces parent switches, and

improves network performance. As a future work, we plan to extend our work to detect more attack such as sinkhole attack and enable detection of unknown attacks with the help of a machine learning model in the detection module of SRF-IDS root.

Author Contributions: Conceptualization, P.P.I. and V.G.V.; Methodology, P.P.I. and S.F.S.; Software: P.P.I.; Validation: P.P.I., V.G.V. and S.F.S.; Writing, P.P.I., V.G.V. and S.F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available in Zenodo at 10.5281/zenodo.5828193.

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

References

1. Holst, A. IoT Connected Devices Worldwide 2019–2030 | Statista. 2021. Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 26 December 2021)
2. Gemalto. The State of IoT Security. Available online: http://www2.gemalto.com/iot/index.html (accessed on 1 February 2021).
3. Nawir, M.; Amir, A.; Yaakob, N.; Lynn, O.B. Internet of Things (IoT): Taxonomy of security attacks. In Proceedings of the 3rd International Conference on Electronic Design (ICED), Phuket, Thailand, 11–12 August 2016; pp. 321–326. [CrossRef]
4. Lindsey, O. 'Amnesia: 33' TCP/IP Flaws Affect Millions of IoT Devices. 2020. Available online: https://threatpost.com/amnesia33-tcp-ip-flaws-iot-devices/161928/ (accessed on 26 December 2021).
5. egnite GmbH. Nut/OS. 2009. Available online: http://www.ethernut.de/en/firmware/index.html (accessed on 26 December 2021).
6. Winter, T.; Thubert, P.; Brandt, A.; Hui, J.W.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, J.P.; Alexander, R.K.; et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *rfc* **2012**, *6550*, 1–157.
7. Ko, J.; Terzis, A.; Dawson-Haggerty, S.; Culler, D.E.; Hui, J.W.; Levis, P. Connecting low-power and lossy networks to the internet. *IEEE Commun. Mag.* **2011**, *49*, 96–101.
8. Tsao, T.; Alexander, R.; Dohler, M.; Daza, V.; Lozano, A.; Richardson, M. A security threat analysis for the routing protocol for low-power and lossy networks (RPLs). *RFC* **2015**, *7416*, 131.
9. Medjek, F.; Tandjaoui, D.; Romdhani, I.; Djedjig, N. Security Threats in the Internet of Things: RPL's Attacks and Countermeasures. In *Security and Privacy in Smart Sensor Networks*; IGI Global: Hershey, PA, USA; 2018; pp. 147–178.
10. Pongle, P.; Chavan, G. A survey: Attacks on RPL and 6LoWPAN in IoT. In Proceedings of the 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015; pp. 1–6.
11. Kumar, A.; Matam, R.; Shukla, S. Impact of packet dropping attacks on RPL. In Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 22–24 December 2016; pp. 694–698.
12. Ioulianou, P.P.; Vassilakis, V.G. Denial-of-Service Attacks and Countermeasures in the RPL-Based Internet of Things. In Proceedings of the 2nd International Workshop on Attacks and Defenses for Internet-of-Things (ADIoT) in Conjunction with ESORICS, Luxembourg, 26 September 2019.
13. Kamble, A.; Malemath, V.S.; Patil, D. Security attacks and secure routing protocols in RPL-based Internet of Things: Survey. In Proceedings of the 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI), Pune, India, 3–5 February 2017; pp. 33–39.
14. Boudouaia, M.A.; Ali-Pacha, A.; Abouaissa, A.; Lorenz, P. Security Against Rank Attack in RPL Protocol. *IEEE Netw.* **2020**, *34*, 133–139. [CrossRef]
15. Verma, A.; Ranga, V. Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review. *IEEE Sens. J.* **2020**, *20*, 5666–5690. [CrossRef]
16. Raoof, A.; Matrawy, A.; Lung, C.H. Routing Attacks and Mitigation Methods for RPL-Based Internet of Things. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1582–1606. [CrossRef]
17. Wallgren, L.; Raza, S.; Voigt, T. Routing attacks and countermeasures in the RPL-based internet of things. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 794326. [CrossRef]
18. Ribera, E.G.; Alvarez, B.M.; Samuel, C.; Ioulianou, P.P.; Vassilakis, V.G. Heartbeat-based detection of blackhole and greyhole attacks in RPL networks. In Proceedings of the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2020; pp. 1–6.
19. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
20. Patel, H.B.; Jinwala, D.C. Blackhole detection in 6LoWPAN based internet of things: An anomaly based approach. In Proceedings of the TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Kochi, India, 17–20 October 2019; pp. 947–954.
21. Shafique, U.; Khan, A.; Rehman, A.; Bashir, F.; Alam, M. Detection of rank attack in routing protocol for Low Power and Lossy Networks. *Ann. Telecommun.* **2018**, *73*, 429–438. [CrossRef]
22. Belavagi, M.C.; Muniyal, B. Multiple intrusion detection in RPL based networks. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 467. [CrossRef]

23. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
24. Djedjig, N.; Tandjaoui, D.; Medjek, F.; Romdhani, I. Trust-aware and cooperative routing protocol for IoT security. *J. Inf. Secur. Appl.* **2020**, *52*, 102467. [CrossRef]
25. Glissa, G.; Rachedi, A.; Meddeb, A. A secure routing protocol based on RPL for Internet of Things. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–7.
26. A Almusaylim, Z.; Jhanjhi, N.; Alhumam, A. Detection and Mitigation of RPL Rank and Version Number Attacks in the Internet of Things: SRPL-RP. *Sensors* **2020**, *20*, 5997. [CrossRef] [PubMed]
27. Arış, A.; Yalçın, S.B.Ö.; Oktuğ, S.F. New lightweight mitigation techniques for RPL version number attacks. *Ad Hoc Netw.* **2019**, *85*, 81–91. [CrossRef]
28. Airehrour, D.; Gutierrez, J.A.; Ray, S.K. SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things. *Future Gener. Comput. Syst.* **2019**, *93*, 860–876. [CrossRef]
29. Iuchi, K.; Matsunaga, T.; Toyoda, K.; Sasase, I. Secure parent node selection scheme in route construction to exclude attacking nodes from RPL network. In Proceedings of the 2015 21st Asia-Pacific Conference on Communications (APCC), Kyoto, Japan, 14–16 October 2015; pp. 299–303.
30. Ioulianou, P.P.; Vassilakis, V.G.; Moscholios, I.D.; Logothetis, M.D. A signature-based intrusion detection system for the Internet of things. In Proceedings of the IEICE Information and Communication Technology Forum (ICTF), Graz, Austria, 11–13 July 2018; pp. 1–6.
31. Ioulianou, P.P.; Vassilakis, V.G.; Logothetis, M.D. Battery Drain Denial-of-Service Attacks and Defenses in the Internet of Things. *J. Telecommun. Inf. Technol.* **2019**, *2*, 37–45. [CrossRef]
32. Airehrour, D.; Gutierrez, J.; Ray, S.K. Secure routing for internet of things: A survey. *J. Netw. Comput. Appl.* **2016**, *66*, 198–213. [CrossRef]
33. Yuan, W.; Guan, D.; Lee, Y.K.; Lee, S.; Hur, S.J. Improved trust-aware recommender system using small-worldness of trust networks. *Knowl.-Based Syst.* **2010**, *23*, 232–238. [CrossRef]
34. Perrey, H.; Landsmann, M.; Ugus, O.; Schmidt, T.C.; Wählisch, M. TRAIL: Topology authentication in RPL. *arXiv* **2013**, arXiv:1312.0984.
35. Levis, P.; Clausen, T.; Hui, J.; Gnawali, O.; Ko, J. The trickle algorithm. *Internet Eng. Task Force RFC* **2011**, *6206*, 1–13 .
36. Tsiftes, N.; Eriksson, J.; Dunkels, A. Low-power wireless IPv6 routing with ContikiRPL. In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, Stockholm, Sweden, 12–16 April 2010; pp. 406–407.
37. Osterlind, F.; Dunkels, A.; Eriksson, J.; Finne, N.; Voigt, T. Cross-level sensor network simulation with Cooja. In Proceedings of the 31st IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 14–16 November 2006; pp. 641–648. [CrossRef]
38. Jadhav, R. Whitefield Framework. 2020. Available online: https://github.com/whitefield-framework/whitefield (accessed on 26 December 2021).