Secure Routing and Forwarding in RPL-based Internet of Things: Challenges and Solutions

by

Ahmed Raoof, **M.Sc.**

A dissertation submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
May 2021

Abstract

As the Internet of Things (IoT) becomes an integral part of our everyday life, securing the IoT devices against malicious activities became critical for their deployment, especially with such devices entering homes and controlling essential services. Most IoT devices still have limited resources (i.e., energy, processing power, and memory), complicating the use of traditional security measures. A modified version of the traditional TPC/IP protocol stack was developed for IoT devices, commonly known as the uIP protocol stack. This protocol stack includes either lightweight versions of the traditional protocols, operating at each layer or IoT-suitable replacement protocols. Among these protocols, the Routing Protocol for Low Power and Lossy Networks (RPL) was designed to perform network-layer routing in IoT, while the IPv6 over Low-powered Wireless Personal Area Network (6LoWPAN) protocol was introduced to a new network-sub-layer called the 6LoWPAN adaptation layer.

While exploring the challenges that face the routing and forwarding processes at the Network layer in RPL-based networks (or 6LoWPAN networks), from the work in this dissertation, it was found that both processes suffer from a significant vulnerability: the inability to authenticate the message's immediate sender. This problem is explored in detail, and its effects on the performance and security of IoT devices are thoroughly investigated. A solution is proposed to the authentication problem, in the form of a framework based on Network Coding (NC), which is introduced as a third security mode for RPL: the *Chained Secure mode* (CSM). A prototype for the proposed solution is evaluated, through simulations, for the RPL against several replay attacks, which proved to be effective against the investigated attacks.

An integration of the 6LoWPAN protocol and the CSM framework is proposed to reduce the effect of buffer-reservation attacks. The preliminary evaluation results show that this integration between RPL and 6LoWPAN has potential mitigating and minimizing the effect of the external adversaries of the buffer-reservation attack with minimal resource consumption.

I dedicate this work to my beloved family; my father, Mohammed; my late mother, Manahil; my wife, Noor; my daughter, AYA; and my sisters, Ebaa and Hamsa. Without you and your unlimited love and support, I would not be here today...

Acknowledgments

A sincere gratitude goes to my supervisors, Prof. Ashraf Matrawy and Prof. Chung-Horng Lung, for their unlimited support, motivation, knowledge, and limitless patience. Their guidance during my Ph.D. study was inspiring, to say the least. I do not imagine having better advisors and mentors than them.

I also would like to extend big thanks to my dissertation committee: Prof. Natalija Vlajic (York University), Prof. David Knox (University of Ottawa), Prof. Michel Barbeau (Carleton University), and Prof. Mostafa Taha (Carleton University), for taking some of their time to review this dissertation and providing me with valuable comments and, hopefully, a few challenging questions!

Another thank you is due to my dearest friend, Dr. Danish Sattar, who helped me through the work of this dissertation several times, and our discussion always led me to think about new ideas for my own work.

Of course, the biggest credit goes to my family: my father and sisters who were supporting me over the long distances in every way possible they could, and my wife whom without her beside me, I would have been lost a long time ago. Thank you my dear for standing with and pushing me to reach the sky.

I am extending another thank you to all my friends, whether I met them as part of the Next Generation Networks (NGN) research group or I have known them from my personal life. All your encouraging words and support for my family and me are imprinted in my memory forever.

Finally, a thank you is due to all who gave me a hand with this research and I forgot to mention them in this long acknowledgment.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

6LoWPAN	IPv6 over Low-powered Wireless Personal Area Network.
AES	Advanced Encryption Standard.
AMI	Advanced Metering Infrastructure.
AODV	Ad-hoc On-demand Distance Vector.
ASM	Authenticated Secure Mode.
BH	Blackhole.
CA	CloneID attack.
CC	Consistency Check.
CCM	Counter wih CBC-MAC "Cipher Block Chaining - Message Authentication Code".
CH	Cluster Head.
CNC	Child Node Count.
CoAP	Constrained Application Protocol.
CORPL	Cognitive and Opportunistic RPL.
CR	Cognitive Radio.
CSM	Chained Secure Mode.
CSMA	Carrier-Sense Multiple Access.
DAG	Directed Acyclic Graph.
DAO	Destination Advertisement Object.
DAO-ACK	DAO Acknowledgement.
DDoS	Distributed Denial-of-Service.
DHT	Distributed Hash Table.
DIO	DODAG Information Object.
DIS	DODAG Information Solicitation.

DODAG	Destination Oriented Directed Acyclic Graph.
DODAGID	DODAG Identification.
DoS	Denial of Service.
DTLS	Datagram Transport Layer Security protocol.
DTM	Dynamic Threshold Mechanism.
E2E	End-to-End.
ER	Emergency.
ERNT	Extended RPL Node Trustworthiness.
ESP	Encapsulated Security Header.
EXT	Expected Transmission Count.
EXTOF	Expected Transmission Count Objective Function.
GPS	Global Positioning System.
IANA	Internet Assigned Numbers Authority.
ICMPv6	Internet Control Message Protocol.
IDS	Intrusion Detection System.
IETF	Internet Engineering Task Force.
IoT	Internet of Things.
IPSec	Internet Protocol Security.
IPv6	Internet Protocol version 6.
LBOF	Load Balancing Objective Function.
LLC	Logical Link Control.
LLN	Low-power Lossy Networks.
LOADng	Low power and Lossy Networks On-demand Ad-hoc Distance-vector routing protocol - Next Generation.
LOADng-CTP	LOADng with Collection Tree Protocol.
LoWPAN	Low-powered Wireless Personal Area Network.
M2M	Machine-to-Machine.

MAC	Message Authentication Code.
MAC	Medium Access Control.
MC	Multicast.
MoP	mode of operation.
MoP	Mode of Operation.
MP2P	Multi-Point to Point communication.
MP2P	Multi-Point-to-Point.
MRHOF	Minimum Rank with Hysteresis Objective Function.
MTU	Maximum Transmission Unit.
NA	Neighbor attack.
NC	Network Coding.
OF	Objective Function.
OF0	Objective Function Zero.
OS	Operating System.
P2MP	Point to Multi-Point communication.
P2MP	Point-to-Multi-Point.
P2P	Point to Point communication.
P2P	Point-to-Point.
PDR	Packet Delivery Rate.
PDR	packet delivery rate.
PDU	Protocol Data Unit.
PHY	Physical layer.
PSM	Preinstalled Secure Mode.
PTR	Packet Transmission Rate.
QoS	Quality of Service.
RA	Rank attack.
RDC	Radio Duty-Cycle.
RFID	Radio Frequency Identification.

ROLL	Routing Over Low-power and Lossy Networks working group.
RPL	Routing Protocol for Low Power and Lossy Networks.
RSA	Rivest-Shamir-Adleman encryption.
RSS	Received Signal Stregnth.
RSSI	Received Signal Stregnth Indicator.
RTT	Round-Trip Time.
SC	Secret Chaining.
SF	Selective-Forward.
SH	Sinkhole.
SHA	Secure Hash Algorithm.
SPRT	Sequential Probability Ratio Test.
SRPL	Secure RPL.
SRR	SC Recovery Request/Response.
SRReq	SC Recovery Request.
SRRes	SC Recovery Response.
TAOF	Traffic Aware Objective Function.
TCP	Transmission Control Protocol.
TOF	Trust Objective Function.
TPM	Trusted Platform Module.
TRAIL	Trust Anchor Interconnection Loop.
UC	Unicast.
UDP	User Datagram Protocol.
UM	Unsecured Mode.
VA	Version attack.
VeRA	Version attack and Rank Authentication.
WH	Wormhole.
WSN	Wireless Sensor Network.

Chapter 1

Introduction

1.1 Preface

The Internet of Things (IoT) is currently on the rise, with some organizations expecting that there will be more than 14 billion IoT devices around the world by 2023 [1]. To accommodate the constrained nature of IoT, a modified version of the traditional TCP/IP protocol stack was developed, commonly known as the uIP protocol stack. This protocol stack includes either lightweight versions of the traditional protocols operating at each layer or IoT-suitable replacement protocols. Among these protocols, the Routing Protocol for Low Power and Lossy Networks (RPL) was explicitly designed to perform Network Layer routing in IoT, while the IPv6 over Low-powered Wireless Personal Area Network (6LoWPAN) protocol was introduced to a newly-created sub-layer, below the Network Layer, called the 6LoWPAN Adaptation Layer.

This dissertation is focused on the security aspects for the deployment of IoT networks that use both protocols, RPL and 6LoWPAN, commonly known as the 6LoWPAN networks. More specifically, the dissertation thoroughly investigates the security threats against RPL and 6LoWPAN protocols and the challenges that face the mitigation of such threats. As will be discussed in Chap. 3, the inability to authenticate the immediate sender of a packet presents a significant threat to the existing RPL-based IoT network, which is presented in the form of authentication-based attacks, such as the replay attacks in RPL (see Chap. 2) and the fragmentation attacks in 6LoWPAN (see Chap. 4). The dissertation, then, proposes a solution to the discussed vulnerability, based on a relatively new concept, Network Coding (NC). The proposed solution, presented as a framework, is evaluated (via simulations) against common authentication-based attacks, where it showed a significant reduction of the

investigated attacks' effects, and in many cases resulted in complete mitigation of the attacks in the conducted experiments, without extensively exhausting the nodes' resources.

This chapter includes the following: the motivations and the scope of the work, the questions being answered, the description of the research problem and the methodology being used, and summarizes the main research contributions. Finally, a list of the related publications is provided, along with the dissertation organization.

1.2 Motivations

Despite the large amount of research on securing IoT, many security threats continue to pose a serious risk for the IoT devices without a proper detection or a mitigation method. One of these threats is the inability of a receiver to authenticate the sender of the received messages [2]. This dissertation proves this for RPL and 6LoWPAN protocols in chapters 3 and 4. Such vulnerability allows an adversary to launch several authentication-attacks, e.g., replay, forging, or identity-cloning attacks. This is where the first motivation of this work comes in; proposing a proper solution to this serious vulnerability.

As can be seen later in this dissertation, most of the currently proposed solutions to mitigate Network Layer's common attacks simply ignore the already-existing security features of some Network Layer protocols, more specifically, RPL's optional security features (see Chap. 2). This negligence may complicate the development and the operation of such proposed solutions. In addition, the use of these security features, e.g., control message encryption, could negate the basis of some of these proposed defences. Hence, incorporating the existing security features of the investigated protocols into any new defence become the second motivation behind this work.

NC, as a proposed method to enhance network throughput or reliability by combining several streams, has gained a great deal of research momentum since its introduction in 2000 [3]. Besides the many improvements NC could bring to multicast networks, it also showed some security advantages to both multicast and unicast streams. One such advantage is the creation of a "chained" series of messages between the neighbors (in the case of intra-flow NC - see Chap. 5), which may be used as a method of source authentication. As the third motivation, a framework based on using NC to authenticate senders of messages for the RPL and 6LoWPAN protocols is proposed.

1.3 Dissertation Scope

This dissertation aims to enhance the Network Layer's (of the 6LoWPAN's protocol stack) security by increasing the routing and forwarding processes' resilience against authentication-based attacks, such as replay attacks and buffer-reservation attacks. Mainly, the emphasis is on securing two protocols: RPL and 6LoWPAN, as they became the popular choice for many IoT networks. However, the proposed framework is designed to work along with, rather than replace, any other security measures at the Network Layer or the other layers. In other words, other security measures can be integrated with the proposed framework, which in turns provide a much-needed extensibility feature to the framework.

1.4 Research Questions

This dissertation tries to answer the following research questions:

Question 1, ***What are the current security threats and challenges that RPL and 6LoWPAN face?*** There has been a significant degree of research investigating different attacks on the routing and forwarding processes in the RPL-based IoT networks, represented by two protocols: RPL for routing and 6LoWPAN for packet adaptation and fragments forwarding. This dissertation surveys the current literature (see Chap. 2) to provide an up-to-date view on these threats, and it shows that there is a significant vulnerability in the design of the two protocols; neither one provides authentication of the immediate sender of the control messages (for RPL) or the fragments (for the 6LoWPAN). This vulnerability allows the adversaries to launch authentication-based attacks, e.g., replay, forging, or identity-cloning attacks, which leads to a Denial of Service (DoS) attack inside the network. This dissertation shows that such attacks are hard to detect and mitigate (see Chap. 3.)

Question 2, ***How effective are the current proposals to mitigate authentication-based attacks on RPL and 6LoWPAN?*** A few attempts have been made to mitigate those attacks - see Chap. 2. However, this dissertation shows that most of these proposals require extensive modifications to the way each protocol works, e.g., modified control messages, changes to the way routing maps are created and maintained, or a complete overhaul of the investigated protocol. Another direction was to have additional security measures that are independent of the investigated protocol to mitigate the investigated attacks, e.g., Intrusion Detection Systems (IDSs). Both ways are proven to be limited to mitigating a few attacks or

are extensively taxing the IoT devices' constrained resources.

Question 3, ***Can NC be used as an authentication mechanism to mitigate or reduce the effect of authentication-based attacks on RPL and 6LoWPAN?*** In this dissertation, NC is explored as a solution to the vulnerability mentioned above. Furthermore, a framework based on NC is proposed in the form of a third secure mode for RPL, the Chained Secure Mode (CSM), in addition to the existing secure modes of RPL, i.e., the Unsecured Mode (UM), Preinstalled Secure Mode (PSM), and Authenticated Secure Mode (ASM). A prototype of the proposed framework has been evaluated, and it showed promising results in mitigating and reducing the effects of several types of replay attacks, namely the Neighbor attack (NA), the out-of-band Wormhole (WH) attack, and the CloneID attack (CA).

Question 4, ***Is it beneficial to allow integration between the proposed framework and external security measures? Or even between the CSM framework and 6LoWPAN protocol?*** As a security integration case, this dissertation proposes an integration between 6LoWPAN and CSM's framework to mitigate or reduce the effect of *buffer-reservation* attacks (a fragmentation attack, see Chap. 4). This integration is evaluated (via simulation) against the *buffer-reservation* attacks with an external adversary. The results show that such integration can effectively eliminate the effect of the *buffer-reservation* attack. Hence, this work shows a possibility of mitigating or further reducing the effect of many other attacks, whether on RPL or 6LoWPAN, by integrating an external security measure, such as an IDS. This last possibility was left for future extension.

1.5 Research Methodology

The research process started with a thorough survey on the current security threats in RPL and 6LoWPAN protocols, followed by performance and security evaluations of the two protocols under common attacks. These evaluations demonstrated (via simulation) the current vulnerability of the missed immediate-sender-authentication. Hence, the dissertation continued to propose and evaluate a defence framework based on NC for RPL, which showed a significant reduction of the investigated replay attacks' effects. Finally, a security integration case of the framework's *CSM-Trust* interface was demonstrated by allowing the fragments reception process at the 6LoWPAN sub-layer to use the framework and identify the trustworthiness of the immediate sender; hence, accepting or rejecting the fragments individually. This proposal

was evaluated in a simple setup and was found to provide mitigation of the buffer-reservation attack in the case of an external adversary.

This dissertation takes an untapped approach of using NC to secure the routing and forwarding processes in 6LoWPAN networks against authentication-based attacks. Unlike the proposed NC-based framework in this work, the few proposed solutions to this problem (see Chap. 2) require extensive modifications to the investigated protocols or introduce external security measures, with both ignoring the security features provided with the investigated protocols (RPL and 6LoWPAN) standards. More importantly, such proposals require either the use of dedicated hardware [4, 5] or a central system [6], both of which are not suitable to the constrained nature of IoT devices.

For all the evaluations carried out in this work, simulations were used with 95% confidence intervals. These simulations were performed mostly using Cooja [7], the simulator for the popular IoT Operating System (OS), Contiki OS [8]. The exception is the evaluations in Chap. 4, which were performed using NS3 simulation software [9].

In addition, and to mimic the real-world adversaries, all the simulated adversaries in this dissertation were designed to launch their attacks after the simulated network reach its steady-state phase. They will try to join the network and operate as legitimate nodes before launching the designated attack, with the exception of the WH attack where the adversaries are always in promiscuous mode and do not join the routing topology.

1.6 Research Contributions

- **A Detailed Survey on The Routing Attacks and Mitigation Methods for RPL-Based Internet of Things (Chapter 2):** The chapter starts with an extensive review of RPL, then moves to a thorough exploration of the current security threats on RPL. In addition, a novel classification of the mitigation methods used to counter these attacks is also constructed and provided. A review on many proposed IDSs is also presented.

- **An Analysis of RPL's Performance under Common Routing Attacks (Chapter 3):** To the best of my knowledge, this is the first-of-its-kind, in-depth performance and security analysis of RPL, which provides an insight into what RPL's current security features can do against common routing attacks, namely the Blackhole (BH) and Selective-Forward (SF) attacks, NA, and out-of-band

WH attack. In addition, this analysis is the first to highlight that RPL does not authenticate the immediate sender of the control messages; as it is possible for an adversary to replay encrypted and digitally-signed control messages from other nodes of the other part of the network, creating the illusion of having a nearby node while it is actually out of the range. Such attacks would severely affect the network on several levels, and they are very hard to detect and mitigate. The chapter also proposes two techniques (i.e., having more alternative paths toward the root and reducing RPL's preferred-parent-reachable-detection timeout) to reduce the effect of some of the investigated attacks.

- **Investigating the Common Attacks on Forwarding at 6LoWPAN Layer (Chapter 4):** Similar to the previous contribution, this chapter looks into 6LoWPAN's performance under the buffer-reservation attacks. In addition, it evaluates one of the proposed solutions for this attack, the buffer-split mechanism and its scoring system, and proposes a modification to the system in order to enhance 6LoWPAN's response. The chapter also concludes to the same fact of the previous contribution: the lack of immediate-sender authentication for data fragments allows the adversaries to manipulate the legitimate nodes' fragment buffers and can cause a DoS attack without being detected.

- **Introducing the NC-based framework for RPL, the CSM (Chapter 5):** Presenting the main contribution of this research, the chapter provides a detailed view of the proposed framework: how CSM operates, the new type of control messages used for recovery, and extensive evaluation of the whole framework against several replay attacks, namely the NA, CA, and the out-of-band WH attacks.

- **Integrating 6LoWPAN's Security with CSM Framework (Chapter 6):** Presented as a security integration case for the framework, this chapter proposes and evaluates an integration between 6LoWPAN and CSM framework, where 6LoWPAN uses the framework to find the trustworthiness of the immediate sender of any received fragment before processing it. The evaluation shows that, in the case of an external adversary, this integration is able to eliminate the attack without penalties on any of the used metrics.

1.7 List of Publications

As a part of the continuous contribution to this dissertation, the following papers were either published or submitted in respected journals and conferences:-

- Ahmed Raoof and Ashraf Matrawy, "*The Effect of Buffer Management Strategies on 6LoWPAN's Response to Buffer Reservation Attacks*," IEEE International Conference on Communications (ICC 2017), 2017.
- Ahmed Raoof, Ashraf Matrawy, and Chung-Horng Lung, "*Routing Attacks and Mitigation Methods for RPL-Based Internet of Things*," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1582-1606, Second Quarter 2019.
- Ahmed Raoof, Ashraf Matrawy, and Chung-Horng Lung, "POSTER: *Evaluation of RPL Preinstalled Secure Mode Under Common Routing Attacks*," The 7th IEEE Conference on Communications and Network Security (CNS 2019), June 2019.
- Ahmed Raoof, Ashraf Matrawy, and Chung-Horng Lung, "*Secure Routing in IoT: Evaluation of RPL's Secure Mode under Attacks*," IEEE Global Communications Conference (GLOBECOM 2019), 2019.
- Ahmed Raoof, Ashraf Matrawy, and Chung-Horng Lung, "*Enhancing Routing Security in IoT: Performance Evaluation of RPL Secure Mode under Attacks*," in IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11536-11546, Dec. 2020
- Ahmed Raoof, Chung-Horng Lung, and Ashraf Matrawy, "*Introducing Network Coding to RPL: The Chained Secure Mode (CSM)*," IEEE 19th International Symposium on Network Computing and Applications (NCA 2020), 2020.
- Ahmed Raoof, Chung-Horng Lung, and Ashraf Matrawy, "*Securing RPL using Network Coding: The Chained Secure Mode (CSM)*," submitted for review at The IEEE Internet of Things Journal. Available at arXiv.org as arXiv: arXiv:2102.06254 [cs.NI], 2020. In Press.
- Ahmed Raoof, Chung-Horng Lung, and Ashraf Matrawy, "*Integrating 6LoWPAN Security with RPL Using The Chained Secure Mode Framework*," submitted for review at the 2021 IEEE Global Comm. (Globecom) Conference. Available at arXiv.org as arXiv:2104.14422 [cs.CR], 2021. In Press.

1.8 Dissertation Organization

Chapter 2 provides an in-depth look at RPL, its working principles, and common routing attacks and their proposed mitigation methods. To better understand the effects of common routing attacks on the security and performance of RPL and its security mechanisms, chapter 3 experimentally investigates this matter and provides insights and suggestions to reduce the effects of the investigated attacks. Chapter 4 examines the fragmentation attacks at the 6LoWPAN Adaptation Layer, more specifically the buffer-reservation attacks, and proposes a modified solution to reduce their effects. The main contribution of this dissertation, the proposal of NC-based framework to RPL, is discussed and evaluated in Chap. 5 as a prototype proof-of-concept new secure mode for RPL. in Chap. 6, a security integration case of using the framework integration capability with 6LoWPAN Layer is explored and evaluated. Finally, the dissertation is concluded in Chap. 7.

Chapter 2

Background Review

2.1 Introduction

IoT is a network of *things* that are uniquely identified and connected to the Internet. These things can range from Radio Frequency Identification (RFID) tags and small sensors and actuators in Wireless Sensor Network (WSN), smart grids, and Machine-to-Machine (M2M) networks, all the way up to smartphones and connected vehicles. Hence, IoT networks are heterogeneous and use various standards [11, 12], one of which is the 6LoWPAN.

6LoWPAN networks have the following main characteristics [13, 14]: resource-constrained nodes (energy, memory, and processing power), lossy links, and low data rates ($\sim$ 250kbps). In addition, most of the traffic in these networks is either Point-to-Multi-Point (P2MP) (root to leaf nodes) or Multi-Point-to-Point (MP2P) (leaf nodes to root node). Besides, these networks use the 6LoWPAN protocol stack [15], developed by Internet Engineering Task Force (IETF). This protocol stack is based on Internet Protocol version 6 (IPv6) protocol [16], with the addition of an adaptation layer (6LoWPAN)) [17, 18] that handles header compression, and fragmentation and reassembly of IPv6 packets (among other duties). Fig.2.1 shows a comparison between 6LoWPAN's and TCP/IP's protocol stacks. Additional details about other protocols and standards used in this protocol stack can be found in [13, 19, 20].

As WSN routing methods do not work effectively in IoT [14, 21] (e.g., depends on

This chapter is a revised version of the paper "*Routing Attacks and Mitigation Methods for RPL-Based Internet of Things*," published in IEEE Commun. Surveys and Tutorials [10], 2018

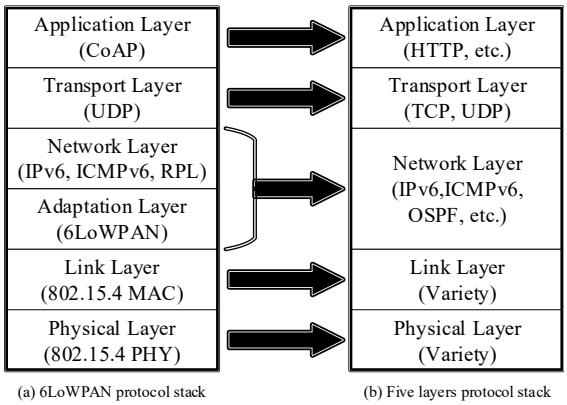


Figure 2.1: (a) 6LoWPAN protocol stack, compared to (b) Internet's five layers protocol stack. The main difference is the new sub-layer below the Network Layer; the 6LoWPAN Adaptation Layer. In addition, new protocols were introduced to suit the resource-constrained nature of IoT devices.

Link Layer addresses), and the traditional Internet routing protocols are not suitable for the resource-constrained IoT devices, several solutions have been proposed for routing in 6LoWPAN networks (and IoT in general) [22–25]. Among these standards, the RPL was introduced by IETF [13]. RPL is designed from scratch to meet the routing needs of IoT networks and minimize resource consumption along the routing path. The protocol is also flexible to adapt to different environments it works within; this is done by using a suitable *Objective Function* (OF) – see Sec. 2.3. It also has several energy-efficient repair mechanisms.

IoT has gained tremendous attention, and security is one of the major concerns in IoT. Moreover, RPL is designed explicitly for IoT networks (including smart grids and M2M networks) to meet routing and efficient resource consumption requirements. Hence, there is a critical need to investigate the security aspects of RPL for a better understanding of the routing attacks on RPL and their mitigation techniques.

To get a better idea on the current security-related issues with RPL and how it became the main motive for the work in this dissertation, this chapter focuses on the routing attacks on RPL and their mitigation methods, whether these mitigation methods are part of the current protocol implementation/standard or proposed in the literature. The chapter contributions can be summarized in the following:-

- A thorough review of RPL's specifications is presented, including a recent "security-minded" proposal to update and modify the standard [26].

- Recent literature (up to the beginning of 2018) on routing attacks for RPL was

investigated. Further, a classification based on the origin of the attacks was also devised and presented to facilitate better understanding and differentiation between these attacks.

- A novel classification for RPL attacks' mitigation methods was introduced. This is the first time such a classification is conducted, which provides an easier way to understand and track current and upcoming mitigation proposals.
- IDSs for RPL were discussed with great details on their working principles, including their latest proposals or extensions.
- A list of the current issues, challenges, and suggested research directions (regarding RPL's security) is provided in light of the above-mentioned study.

The rest of this chapter is organized as follows: Section 2.2 shows a brief comparison between routing in WSNs and RPL-based networks. Section 2.3 presents a comprehensive review of RPL, its working principle, Destination Oriented Directed Acyclic Graph (DODAG) creation and maintenance, self-healing mechanisms of RPL, and its built-in security features. Classification of RPL's routing attacks and their mitigation methods are held in section 2.4. Section 2.5 thoroughly discusses IDS proposals for RPL-based IoT networks. A comprehensive study on routing attacks against RPL (according to their classification) is conducted in sections 2.6 and 2.7. The current issues, challenges, and future work (in regard to mitigating RPL's routing attacks) are discussed in section 2.8. The chapter is concluded in section 2.9.

2.2 WSNs and IoT: Similarities and Differences

Due to the fact that most of IoT concepts are coming from WSN, there is a lot of confusion between the two. In general, WSNs and IoT networks share various similarities in their characteristics [21, 23, 27]:-

- Nodes are deployed depending on the network application: deployment can be deterministic or random, mobile, or static. This situation creates a significant challenge when routing the data for all traffic types – see Sec. 2.3.3.
- The resources of nodes are constrained: usually, many of the nodes depend on batteries or energy-harvesting technologies to operate, and most of them have limited storage/memory capacities and low processing powers. This requires lightweight protocols to be implemented.

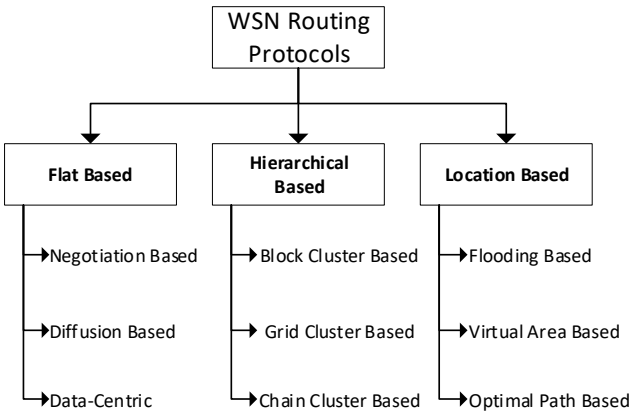


Figure 2.2: Taxonomy of WSN routing protocols (According to [21].)

- Networks may work in harsh conditions: lossy links, nodes that go on and off (due to energy conservation schemes or simply are exhausted), and node mobility (in some applications). In addition, IoT nodes may be scattered in large areas. Such conditions require routing protocols to be scalable in a fast, energy- and resource-efficient ways.

- Due to their constraints, nodes are highly susceptible to attacks, internal or external.

However, WSNs and IoT networks also have their differences: firstly, WSNs are designed originally [11] to coordinate data collection and activate actuation, without any intelligence at the nodes (all processing and decision making is done outside the sensor network). On the contrary, IoT is designed to introduce *smartness* to the *things*, e.g., having more powerful nodes (within the IoT network) that can handle some of the decision making tasks. IoT may also presents a direct communication between each node and the Internet, so that intensive processing can be carried out in the edge or the cloud and response is sent back directly to that node.

The second difference lies in the fact that IoT networks use the IP protocol stack. This feature also means that routing in IoT networks is based on IP addresses, which are a global addressing scheme. However, this is not the case for WSNs, which use different routing techniques, as explained in the following paragraph.

There has been a significant amount of research on routing protocols for WSNs. Recent surveys conducted by Kumari and Prachi [27], and Singh and Singh [21] presented a comprehensive overview of these protocols. Fig. 2.2 shows a general

taxonomy of WSN routing protocols, as classified by [21]. The classification was performed based on the structure of the network and the routing concept: In *Flat-based* routing protocols, all nodes have the same functionality, and sensing is performed by a collaboration between a large number of nodes. The routing here is conducted by queries from different regions of the network based on data types. *Hierarchical-based* routing, on the other hand, divides the network into clusters; each has at least one Cluster Head (CH), which is responsible for processing and aggregating data collected from other cluster member nodes and transmitting the aggregated data toward the sink node. Lastly, *Location-based* routing uses the location of the sender and the sink node to route the data. The details of WSN routing protocols and their operation are beyond the scope of this dissertation as they have been thoroughly investigated in the literature.

2.3 RPL Overview

RPL was developed by the IETF Routing Over Low-power and Lossy Networks working group (ROLL) workgroup to provide routing service in Low-power Lossy Networks (LLN), where devices are highly constraint in resources. It is a distance-vector routing protocol that organizes network devices into *Directed Acyclic Graphs (DAGs)* [28]. A DAG represents a network where all nodes are connected in a way such that there are no round-trip paths and traffic is routed to reach one or more *root* nodes. In the DAG reside one or more *Destination Oriented Directed Acyclic Graph (DODAG)*, in which the topology has one root node only (usually is the gateway or border router "6BR") and all the data are *sinked* [13, 22]. To enable several applications to work simultaneously, but independently, inside the network, several *RPL instances* can coexist inside a DAG; each can have one or more DODAGs. All DODAGs inside an RPL instance share the same *RPLInstanceID* and use the same *Objective Function* (more on that below) [13]. Fig. 2.3 shows an illustration of a DAG with two RPL instances and three DODAGs.

Before going further into RPL operations, it is essential to clarify a few critical terms: an **Objective Function (OF)** defines essential configurations such as routing metrics used, optimization objectives, how to calculate the rank, and how to select parents in the DODAG [29]. Since there are multiple and different applications for IoT deployment, link attributes that are important for each application are different

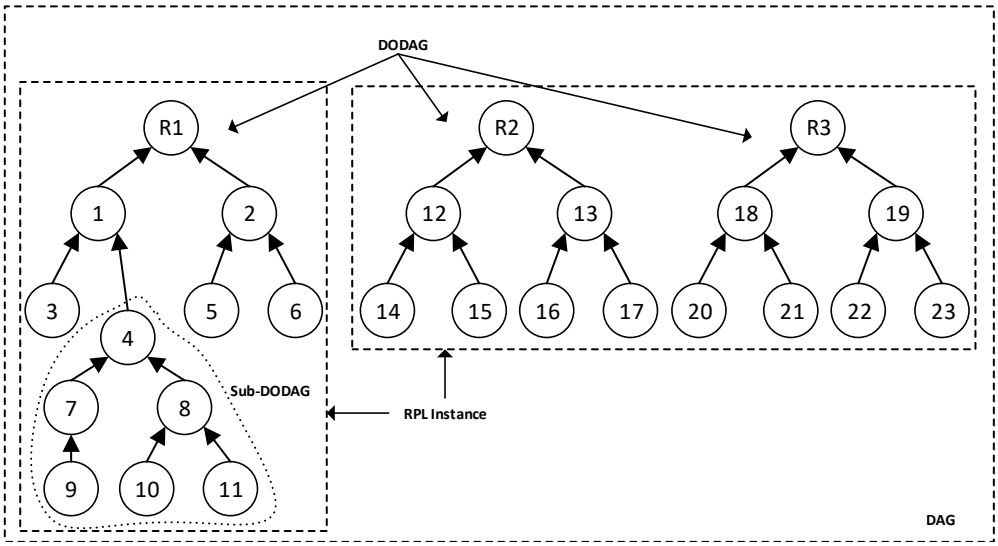


Figure 2.3: An example of a typical RPL-based IoT network, showing all the building blocks: RPL Instances, DAG, DODAG, and Sub-DODAG.

from the others; hence, the OF will be different. The details of the typical IoT applications' requirements can be found in [30–33], but they are out of the scope of this dissertation. There are currently two fully-defined and three draft OFs, set by IETF:-

- *Objective Function Zero (OF0):* OF0 simply adds a predetermined value to the previous rank value [34].
- *Minimum Rank with Hysteresis Objective Function (MRHOF):* It selects routes that minimize a metric while using hysteresis to reduce churn in response to small metric changes [35].
- *Expected Transmission Count Objective Function (EXTOF)* - *Draft:* This OF selects paths that minimize the number of packet transmissions for packet delivery [36]. However, EXTOF is an expired Internet draft, even though it is still widely used in many IoT networks.
- *Load Balancing Objective Function (LBOF)* - *Draft:* LBOF adds *Child Node Count (CNC)* as a metric, and uses it to select paths in a way that maintain a balanced number of children per preferred parent in the DODAG [37]. This will balance the traffic between the nodes, resulting in lower power consumption (hence longer network lifetime), a lower possibility of bottlenecks, and a better

delivery rate. Evaluation for this OF was carried in [38] with a comparison to OF0 and MRHOF, and it shows that LBOF provides a longer network lifetime (by 16-40%) and a better delivery rate (by 10-15%). However, with larger networks, the LBOF seems to consume more energy due to parents churn. The current draft of LBOF suggests using a combination of routing metrics to reduce this churn.

- *Traffic Aware Objective Function (TAOF)* - *Draft:* TAOF uses a combination of Expected Transmission Count (EXT) and Packet Transmission Rate (PTR) as routing metrics, and uses it to select paths with less traffic toward the root [39].

In addition to these OFs, each RPL implementation can have its own OF.

The second term is the *Node's Rank*, an unsigned integer value that defines the node's relative position to DODAG's root node. It is calculated using routing metrics defined in the OF – as simple as actual distance or hop count, or as complex as including other properties, such as link cost, link/node status, and nodes constraints. The rank strictly increases going down the DODAG (root node always has the lowest rank) [13]. Rank is used by nodes to decide what set of parents to choose and the preferred parent, as discussed in Sec. 2.3.2, in addition to eliminating routing loops.

Other terms are the *DODAGID*, which is the IPv6 address of DODAG's root node, used as a unique identifier for the DODAG inside an RPL instance, and *DODAG Version*, which defines a specific iteration "version" of the DODAG with a DODAGID. As will be discussed later, *DODAG Version Number* is a counter used to keep up with DODAG's version, and it is incremented only by DODAG's root node.

The following sections cover important RPL features: control message types used in RPL, the formation of the DODAG and downward routing tables, self-healing mechanisms of RPL, and RPL's security features.

2.3.1 RPL Control Messages

There are five types of control messages used in RPL; each has two versions: base and secure, except for the Consistency Check messages, which have only a secure version [13]. All RPL messages are sent as Internet Control Message Protocol (ICMPv6) messages, with the "*Type*" field in its header equal to 155 – as set by Internet Assigned Numbers Authority (IANA) – and the "*Code*" field identifying the type of the RPL control message [13]. In the following, the base versions of these messages are listed:-

DODAG Information Object (DIO) Message

This message is advertised by each node when it wants to join a DODAG, create a DODAG (root node only), or maintain the DODAG. It contains information used to identify RPL's instance (RPLInstanceID), DODAG's ID, DODAG's version number, RPL's mode of operation, the rank of sending node, DODAG configuration (including the OF adopted), and other information and options [13].

Usually, The root node broadcasts the first DIO message, while all other nodes broadcast their DIO messages once they receive DIO messages from their neighbors. All the information inside any DIO message is a copy of the ones set by the root node in the first broadcasted DIO message. The only exception is the rank, which is computed by each node before it sends its own DIO message [14, 15]. Another case where a DIO message must be sent is when a node receives a DODAG Information Solicitation (DIS) message from another node [13].

DODAG Information Solicitation (DIS) Message

DIS message is used by nodes when they want to join a DODAG, and when they did not receive any DIO messages for some time. Nodes basically probe their neighbors looking for available DODAGs. Responding to a DIS message depends on how it was sent: if *unicasted* then each receiving node will unicast a DIO message to the sender with DODAG configuration and will not reset its trickle timer; if *multicasted* then each receiving node will reset its trickle timer and broadcast DIO message to the neighboring nodes [13].

The use of DIS messages, however, does introduce a vulnerability which could be used to start an attack – see Sec. 2.7.4. IETF intended to solve this problem by introducing some flags in the DIS message [26]. However, this proposal was never followed up and now is expired.

Destination Advertisement Object (DAO) and Acknowledgment (DAO-ACK) Messages

While RPL uses DIO and DIS messages to create and maintain the upward routes (toward the root node) in the DODAG, it uses DAO duo to find the downward routes (from the root or parent node toward children or leaf nodes) [13, 22]. DAO contains path information for reachable nodes by its sender; this information is used to create

routing tables at receiving nodes for Point-to-Point (P2P) and P2MP (downward) communications.

Sending and responding to DAO messages depends on RPL's mode of operation (MoP) (see Sec. 2.3.3): in the Storing mode, DAO messages are unicasted to node's parent(s); while in the Non-Storing mode it will be unicasted toward DODAG's root node [13]. Depending on the flags field of the DAO message, the receiver may respond by sending a DAO Acknowledgement (DAO-ACK).

Consistency Check (CC) Messages

CC messages are used by RPL to synchronize security counters/timestamps between each pair of nodes and provide a challenge-response mechanism as a basis for control messages' reply attacks' protection [13,22] – see Sec. 2.3.5. These messages are always sent as secured messages; hence, only available in RPL's secure modes. For more details and an example on when to use these messages, see Sec. 2.3.5.

2.3.2 DODAG Formation and Maintenance

As mentioned in Sec. 2.3.1, DODAG formation is done by exchanging DIO messages, and it always starts from the root node. The root node should set most of the DIO base fields: RPL instance ID, DODAG ID, DODAG version, and RPL mode of operation (see Sec. 2.3.3). Also, it sets its rank to ROOT_RANK [13], which is equal to the minimum rank increase per-hop *"MinHopRankIncrease."*

Upon receiving a DIO message [13, 15, 24], each node should:

1. Calculate its own rank (using the specified OF and based on the information received from the DIO message).
2. Decide on which DODAG to join (if there are multiple).
3. Select at least one preferred parent from a set of possible parents, and
4. Multicast its replica of the DIO message. In the new DIO message, the node should only change the rank (in the DIO base fields) to reflect its own, while maintaining all the rest as originally set by the root node [13].

According to the current RPL specifications [13], each node – except the root node – should maintain a set of possible parents, which it selects from the list of its

neighbors based on their rank and the OF: possible parents' ranks must always be lower than the node's rank. From this set, the lowest-ranked parent is chosen as the preferred parent. If more than one parent has the same lowest rank, and based on the implementation and OF, the node may select all or some of them as preferred parents. The use of rank to select the parents prevents the creation of routing loops.

In order to maintain the routing topology, DIO messages should be exchanged on a regular basis, and nodes may decide to discard it if the new DIO message does not result in any changes to the current configurations of the DODAG at receiving node (e.g., DODAG version number change) or a change in the preferred parent of the node [13].

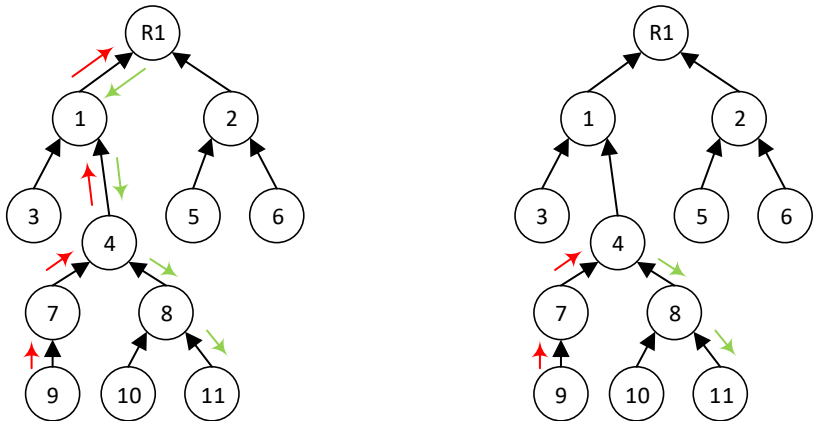
To reduce the effect on nodes' limited resources, RPL employs the *Trickle* algorithm [40] to control when to send DIO messages [13, 24]: each node will maintain a ***DIO counter*** and a ***trickle timer***, with a pre-defined *threshold* for the former and *trickle time* for the latter. DIO message will be sent when the trickle timer reaches its time or after receiving a DIO message that causes changes in RPL configuration. Now, whenever a DIO message is received and discarded, the DIO message counter will be increased. If the counter reaches the threshold value, both the counter and trickle timer will be reset and trickle time will be doubled. In addition, DIO counter and trickle time will return back to their original settings when a change is made due to a received DIO message. This procedure allows for fewer broadcasts of DIO messages when the network is stabilized, and for quick topology update when there is a change [24].

2.3.3 Downward Routes and RPL Mode of Operation (MoP)

RPL supports the three types of communications in 6LoWPAN networks (see Sec. 2.1): MP2P, which resembles the upward traffic; P2MP, which resembles the downward traffic; and P2P, resembling the traffic between two nodes who none are root nodes of the DODAG.

The first type, MP2P, is guaranteed by the DODAG topology created using the method explained in the previous section, while P2MP is done by the root node (as it keeps the full downward routing table). For P2P, it has to be done using one of three ways [13, 28]:

1. ***Non-Storing Mode***: Traffic goes all the way up to the root node, which in turn sends it down toward its destination using source routing [41] (see Fig. 2.4(a)).



(a) RPL non-storing mode (b) RPL storing mode

Figure 2.4: RPL modes of operation for P2MP and P2P communication [13].

2. ***Storing Mode***: each node will keep a downward routing table for its sub-DODAG and use it to forward P2P traffic. In other words, traffic will go upward until it reaches a common ancestor (of the sender and destination), which will route it downward to the destination node – see Fig. 2.4(b).

3. In addition, RPL provides the ability for one-hop P2P traffic [13].

As mentioned in Sec. 2.3.1, DAO messages are used to create the downward routing table. Depending on RPL's mode of operation [13]:-

- In the **storing mode**, each node will send DAO messages to its *DAO parent* (which must be one of its possible parents set but not necessarily the same as the upward parent). Receiving nodes should use the information contained within the DAO message to create and maintain downward routing tables. This process should be done by all non-leaf, routing-capable nodes only.

- In the **non-storing mode**, nodes will send DAO messages toward their DODAG root node, which is responsible for creating and maintaining source routing tables using the received DAO information.

According to RPL's specifications, both modes cannot operate simultaneously and the active MoP must be selected by the DODAG root node only. Also, there are no mechanisms to ensure that routing-capable nodes are storing/maintaining entries for all of their sub-DODAG nodes [13].

2.3.4 Self-Healing Mechanisms of RPL

Current RPL specification includes several self-healing mechanisms: one global repair mechanism and several local ones. The ***Global Repair*** mechanism is typically called when the root node finds significant inconsistencies in the DODAG [15, 28]. At that time, the root node broadcast a DIO message with an incremented DODAG version number. This step will start the regeneration of the DODAG and force all nodes to discard the current DODAG formation, reset their trickle timers, and start joining the new DODAG. The only node allowed to start a global repair is the root node [13]; however, the number of inconsistencies or their degree of severity that triggers the mechanism is left for the implementation.

The main ***Local Repair*** procedure for the downward paths consists of resetting DIO trickle timers and exchanging updated DIO messages. It also includes poisoning the sub-DODAG (by sending DIO message with *Rank* field set to INFINITE_RANK) [13], which forces child nodes to detach the sending node from being a preferred parent and selecting another one. It is called local because it only affects the sub-DODAG of the nodes.

Another local repair procedure occurs when a node finds out its preferred parent is no longer available (either through lost link-layer acknowledgments, not receiving DIO messages for an extended period of time, or after receiving a poisoning DIO message). The node then will have to select another preferred parent from its possible parents set, and if none were available or the set was empty, it will start the main local repair procedure [13, 42].

Besides these procedures, two recovery mechanisms are also available, which are summarized in the following [13, 43]:

- **DAG Inconsistency Loop Recovery:** DODAG inconsistency is detected whenever the direction of a packet contradicts the rank relationship between the sender and the receiving node. For example, a packet with Down "O" flag is set, but the sender is of a higher rank (the packet is directed downward but received as upward); or a packet with Down "O" flag clear, but the sender is of a lower rank (the packet is directed upward but received as downward) [13, 44]. This may occur during a global repair procedure (rebuild of DODAG with the increase of DODAG version number) [13] when a node still on the previous DODAG receives a packet from a sender who already joined the new DODAG

with a changed rank value. Another possibility for having a loop is when a node wants to detach from the DODAG, and some of DIO messages get lost or not delivered, so this particular node joins the same sub-DODAG with a different rank [13].

RPL has a repair mechanism for this situation, called *DAG Inconsistency Loop Recovery*, in which the receiving node forwards the packet with Rank-Error "R" flag set. Whenever a receiver detects DODAG inconsistency, it will check the "R" flag of the inconsistent packet: if the flag is clear, the receiver will set it and forward the packet. If the flag is set, the receiver will drop the packet, reset DIO trickle timer and start the main local repair procedure (sending updated DIO messages to its neighbors) [13].

- **DAO Inconsistency Loop Recovery:** When a node receives a packet from a parent node that is targeted toward one of its child nodes (or their children), it checks the routing table to find the path. If one of these routes (that had been learned from a previous DAO message) is no longer available, it will create inconsistency in the network as other nodes are still not aware of the change [44].

  RPL has an optional repair mechanism for this situation called *DAO Inconsistency Loop Recovery* [13]: the packet will be returned to the parent of the node with Forwarding-Error "F" flag set and Down "O" flag left unchanged (set). When the parent receives a packet with F flag set, it will remove routes to the destination through that neighbor, clear the "F" flag from the packet and send it to the destination again through a different neighbor. The process will recurse if that neighbor also has inconsistency or will end by delivering that packet - See Fig. 2.5. It is worth noting that this process is not applicable for non-storing RPL mode, as only the root node has a routing table and uses the source routing. If a route is not available, an "Error in Source Routing Header" ICMPv6 message is sent back to the root.

2.3.5 RPL Security Features

According to RPL specifications [13], the protocol has several security features, which are listed below. However, the specification states that implementing these security features (partially or fully) is "optional":

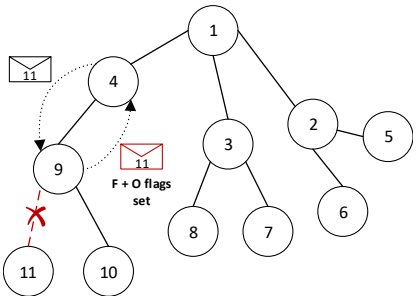


Figure 2.5: DAO inconsistency loop recovery procedure.

1. RPL is designed to use link-layer security mechanisms when they are available to secure message transmission.

2. Beside link-layer security, RPL has its own security mechanisms, provided mainly through three security modes:-

   - **UM:** The default mode for RPL when it depends on link-layer security. In this mode, no security measures are applied to RPL control messages and are exchanged in clear text.
   - **PSM:** Preinstalled encryption key (symmetrical key) is manually pre-configured on the nodes, which use this key to encrypt and decrypt the secure versions of RPL control messages when joining the DODAG and for its maintenance. This mode is recommended when secure routing is desired with a lot of constrained devices.
   - **ASM:** Here, two different encryption keys are used; a preinstalled one (similar to PSM) to allow any node to join the DODAG as a *leaf* node (a node without children and does not perform routing), and an authenticated key used only by routing-capable nodes to create and maintain the DODAG. In this case, routing nodes must have both keys to allow communication with the leaf nodes (using the preinstalled key) and other routing nodes (using the authenticated key). These authenticated keys must be obtained from an authentication authority. However, how this authority authenticates the nodes or distributes the keys to them are left for the implementation.

3. RPL also has an optional "*Replay Protection*" mechanism, called the Consistency Check. These checks compare a non-repetitive value (Counter wih CBC-MAC "Cipher Block Chaining - Message Authentication Code" (CCM) Nonce), which is sent within the CC secure messages – see Sec. 2.3.1, and the stored status information (the originating node's address and CCM nonce value last received from that node) to check if the received CCM nonce value has been used before from the originating node; hence, the mechanism detects a replay attack.

In the PSM and ASM, secure versions of RPL control messages are exchanged. To support these messages' confidentiality, integrity, and authenticity, RPL uses AES/CCM (the Advanced Encryption Standard (AES) in CCM mode) with 128-bits key to generate 32-bit and 64-bit Message Authentication Code (MAC). These MACs are used to assure the integrity of the messages. Also, RPL uses Rivest-Shamir-Adleman encryption (RSA) with Secure Hash Algorithm (SHA)-256 for digital signatures of the messages, to provide the confidentiality and authenticity, with optional 2048 and 3072-bit signatures.

It is worth noting that, up to this moment, the current implementations of RPL provide it in UM only. Even the most commonly used operating systems (Contiki OS [8] and TinyOS [45]) do not have these security features implemented in their RPL implementations. Perazzo *et al.* in [46, 47] provided a partial implementation of RPL's security features by adding PSM and the replay protection to ContikiRPL (Contiki OS implementation of RPL). The authors evaluated the performance of their implementation via simulations in Contiki. They showed that implementing RPL in PSM by itself (no replay protection) will not add significant increments to network formation time, RPL's control overhead, or power consumption. On the other hand, implementing the replay protection mechanism would induce significant delays to network formation time and control overhead, leading to higher power consumption, especially in large networks (with more than 25 nodes).

2.4 Classification of RPL Attacks and Mitigation Methods

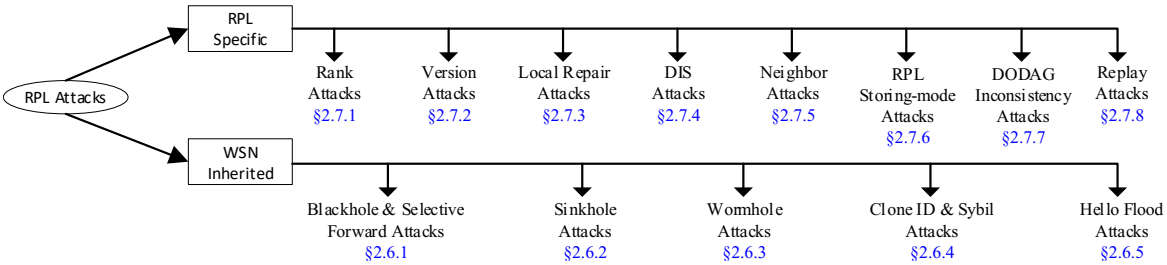


Figure 2.6: Classification of RPL attacks.

2.4.1 Classification of RPL's Attacks

As mentioned in Sec. 2.1, IoT has some similarities to WSNs, but it extends the communication with the Internet all the way down to the *things*. This means that IoT networks inherit attacks from both WSNs and the traditional Internet-based networks [22]. In this chapter, the focus is on routing attacks that RPL is susceptible to, which can be classified (based on the origin of the attack) to attacks inherited from WSNs and attacks based on RPL own vulnerabilities (RPL-specific attacks). These two classes of attacks, along with specific common attacks that belong to each class, are illustrated in Fig. 2.6. A comprehensive discussion of these attacks, along with their mitigation methods, is presented in Sec. 2.6 and Sec. 2.7.

2.4.2 Classification Scheme for RPL Attacks Mitigation Methods

Through this chapter, mitigation methods can be divided into *mitigation mechanisms* and *IDSs*. Mitigation mechanisms usually are either added procedures to RPL standard or modifications to the current RPL procedures. In general, the mitigation mechanisms are used to deal with specific types or limited numbers of the attacks (typically one or two types). However, such mechanisms have lower consumption to nodes' resources. On the other hand, IDSs are designed more toward detecting and mitigating several types of attacks at once, and can be extended to cover attacks that they were not originally designed to detect. In addition, IDSs require some degree of collaboration between the nodes.

To better understand the mitigation methods for RPL's attacks (both mechanisms and IDSs), a novel classification scheme was created for these mitigation methods based on the various techniques adopted [10]. Such taxonomy aids the researchers

and provides them with a better understanding of the current proposed solutions, both current and future ones. This classification scheme is presented as follows:-

- **Acknowledgment-based Methods:** The concept of these methods is to send a message and receive an acknowledgment. This message should be signed in a way to prevent alteration. An example is the Heartbeat protocol [14], where the root node sends an ICMPv6 echo message to each node and waits for an acknowledgment. If nothing is received from that node, it will be considered as a malicious node. In general, these methods are not sufficient by themselves and are often used in combination with other methods.

- **Trust-based Methods:** Nodes have to monitor their neighbors, rate them using some algorithm, and create trust relations based on these ratings. This type is usually effective combating WSN-inherited attacks and some of the RPL-specific attacks, such as the decreased rank attack – see Sec. 2.7.1.

- **Location-based Methods:** These methods relate nodes or packets to their locations within the network, whether geographically; e.g., using Global Positioning System (GPS) systems, or logically; e.g., via Round-Trip Time (RTT) calculations or using information from the Link/PHY layers. A good example would be Packet Leashes [48]. Currently, these are the only reliable methods to detect wormhole and Sybil attacks – see Sec. 2.6.3 and Sec. 2.6.4.

- **Statistical/Mathematical-based Methods:** Here, detection/mitigation is based on statistical or mathematical models. For example, Surendar and Umamakeswari proposed an IDS [49] that uses Evidence Theory to detect sinkhole attacks – see Sec. 2.6.2.

- **Specification-based Methods:** These methods use RPL's own specifications (such as rank and DODAG version) to detect the attacks. An example would be the work of Glissa *et al.* in [50] that finds a rank threshold value based on monitored DIO messages, and uses that to detect rank attacks – see Sec. 2.7.1.

The specific methods that belong to each class are described later in this chapter. This includes the following section, which presents more detailed information about IDSs and their current proposals for RPL-based IoT networks.

2.5 Current Trends of IDSs for RPL-Based IoT

IDSs are generally classified based on either their detection method or their placement strategy [51]. Detection methods divide IDSs into:

- **Signature-based IDSs:** These IDSs use a database of signature patterns of the attacks to detect them [52]. However, even though it consumes less resources than other types of IDSs, it is not efficient in detecting unregistered attacks.
- **Anomaly-based IDSs:** These IDSs monitor the network traffic to create a normal behavior profile, then compare network activities to this profile and consider any anomaly as a possible attack [51]. Normally these IDSs are efficient in detecting known attacks and possibly new ones that cause anomalous behaviors. However, they usually have false-positive/false-negative detections, which limits their potentials. Also, they are considered more resources-consumers than the previous type.
- **Specification-based IDSs:** Similar to anomaly-based detection methods, these IDSs also create a normal behavior profile of the network. However, they differ in that this type of IDSs creates the network profile based on network (or protocol's) specifications, normally defined manually by the operator [51, 52]. This results in much lower false-positives and false-negative rates than anomaly-based methods and almost no training period. On the other hand, using a manual definition of the specifications makes it harder to adapt to the environment's changes. Le *et al.* in [53] introduced a semi-auto profiling technique for a specification-based IDS that uses trace files to create the rules for their IDS.
- **Hybrid IDSs:** Combining two of the methods mentioned above in one IDS creates a hybrid system that usually takes their advantages and minimizes the drawbacks [51].

On the other hand, using IDS placement for classification, IDSs can be divided into:

- **Centralized IDS Placement:** Here the IDS resides either on the root node (e.g., border router) or a dedicated host (e.g., CH node) [51], and uses the traffic passing through to detect attacks. In many cases, it is required that the central node of the IDS send periodic requests for updates from the monitored

network. The advantages of centralized IDS are that most heavy work occurs inside a powerful node, which provides the ability to perform extensive security checks. Also, it is usually capable of protecting the network from the Internet-side attacks and botnets, due to performing as a firewall. On the down side, it would be challenging to monitor the network during an actual attack.

- **Distributed IDS Placement:** In this type of IDSs [51,54], each node will have a full IDS implementation, making it responsible for detecting attacks around it. Usually, there is a collaboration among the nodes to increase the efficiency of attacks' detection and mitigation. However, this approach results in a high consumption of resources on all IoT devices. Hence, another approach is used: distributing monitoring nodes (*watchdogs*) within the network responsible for the monitoring task, then they communicate with IDS units inside other nodes to mitigate the attacks. The main advantage of these IDSs is better monitoring and detection of insider attacks. Since the major parts of the IDS, namely the decision making parts, are implemented within each node, this generally results in higher resources consumption at the nodes. It is usually required to optimize the IDS periodically to minimize this effect.

- **Hybrid IDS Placement:** To get the best of both previous placement strategies, a hybrid solution is desirable [14, 51, 52]: central node(s) that have more resources and are responsible for computationally intensive IDS tasks (such as analyzing data gathered from monitoring nodes, decision making, etc.) and normal nodes that are responsible for performing lightweight IDS duties (such as monitoring neighbor nodes, sending data about traffic passing through them, and responding to mitigation control messages from central nodes). This approach has the advantage of better and faster detection of the attacks than the centralized approach and lower resource consumption than the distributed approach. However, continuous IDS optimization is required, and the placement of central nodes should be chosen wisely, depending on the application of the network and its environment.

In the IoT paradigm, and due to the highly constrained nature of the devices, a hybrid detection approach with hybrid placement IDS is the most preferred option for implementation. As it is clear that this combination reduces resource consumption and increases the detection rate, and that is what most of the current research is

working on [14,24,52]. There are many proposals for IDSs for IoT and WSNs; however, only a few of them stand out because of their suitable features for RPL-based IoT.

In the following, a summary of these most influential IDS for IoT, designed more specifically for RPL-based IoT networks, is provided.

2.5.1 SVELTE

This is a specification-based, hybrid detection, hybrid placement IDS, proposed by Raza *et al.* [55]. The system consists of three modules:-

- **6LoWPAN Mapper (6Mapper):** Residing on the root node, this module is responsible for constructing a full DODAG with each node's neighbors and parent information.
- **Intrusion Detection Module:** This module also resides on the root node. It is responsible for detecting the attacks using both signature-based and anomaly-based methods, mainly by using the rank and DODAG information collected by the 6Mapper module. It is designed with the ability to be extended to cover newer attacks, as claimed by the authors. Also, it includes several algorithms to reduce false-positive and false-negative rates.
- **Distributed Firewall and Response Module:** This module is installed on every node. Its responsibility is to prevent attacks from outside the network, even when the data is encrypted. Also, it is responsible for sending information requested by the 6Mapper module.

One of the requirements of SVELTE is to use all possible encryption at the Application layer (Constrained Application Protocol (CoAP) with Datagram Transport Layer Security protocol (DTLS)), the Network layer (6LoWPAN with Internet Protocol Security (IPSec) enabled), and the Link layer (encryption enabled). This is to assure indistinguishability for IDS control messages.

The original version of this IDS showed promising results (80 to 90% true-positive rate, as demonstrated by the authors) against BH, SF, Sinkhole (SH), and DODAG inconsistency attacks – see Sec. 2.6 and Sec. 2.7. An extended version [56] added geographical hints (as location information) and introduced a new metric (EXT), to lower the false-positive and false-negative rate, and to provide detection ability for Sybil and WH attacks.

As pointed out by many researchers (Le *et al.* in [53], Alzubaidi *et al.* in [57], and SVELTE's authors [55], among many others), the original SVELTE IDS tends to have more false-positives and false-negatives in medium to large IoT networks. This is due to the way SVELTE detects the "*invalid inconsistencies*": it depends on the number of reported ranks of the nodes, then it uses a simple threshold (the difference between the ranks of the nodes) to detect such inconsistencies (normally, the threshold is set to 20%). This shortcoming has been investigated intensively by many researchers [49, 51, 54, 58].

2.5.2 Intrusion Detection of Sinkhole Attacks on 6LoWPAN for the Internet of Things (INTI)

This IDS is a statistical/mathematical-based, anomaly detection, hybrid placement IDS proposed by Cervantes *et al.* [54] to overcome original SVELTE's drawbacks: not supporting node mobility [54], high false-positive rates (when there are many attacks, as per the authors) [59], and high resource consumption (specifically energy and memory [58]). The system divides the network into clusters. Each cluster consists of one *Leader*, at least one *Associated Node* (a node that receives messages from another leader in order to forward it toward the root node), and *Member Nodes* (normal nodes who joined the cluster and did not have a connection to another leader). Using the rank and nodes statistics, leaders calculate reputation and confidence values for each node in their cluster. These values are used to detect and isolate adversaries in the network by applying evidence theory.

The system was evaluated against the SH attack, showing results similar to those obtained from SVELTE, but with a lower false-positive rate [51, 54]. However, unlike SVELTE, INTI supports nodes mobility. Energy consumption was not investigated, and it should have been due to the high number of calculations performed within the system. The authors also mentioned the ability to extend their system to detect other types of attacks. However, they did not disclose any details in this regard.

2.5.3 Intrusion Detection and Response System (InDReS)

Proposed by Surendar and Umamakeswari [49] as an enhancement to INTI – Sec. 2.5.2, this IDS is also a statistical/mathematical-based, anomaly-detection, hybrid placement system. The system divides the network into clusters; within each

cluster, a node is elected as a *Leader* based on probability. All nodes in the cluster will send their rank to the cluster leader, who will use it to detect and isolate the adversary(ies) based on *Dempster-Shaffer* evidence theory. Whenever an adversary is detected, the root node will be notified by the cluster leader. The root node then will reconstruct the DODAG, excluding adversary nodes from it.

The system was designed originally to detect and mitigate SH attacks only. However, the authors claimed that their system is extendable to cover other types of attacks, such as Rank attack (RA) and Version attack (VA) attacks. Unfortunately, no details on such extensibility were provided. The effect of node mobility is also yet to be investigated.

2.5.4 Game Theory IDS

Based on an original concept, Sedjelmaci *et al.* proposed [59] a statistical/mathematical-based, hybrid-detection, distributed-placement IDS. Their concept is to use the signature-detection method to detect common attacks, while anomaly-detection is used only when the traffic is suspected to be malicious. The use of either method is decided by setting a game between IDS entities and the adversaries: anomaly-detection is trained by turning on periodically in the beginning, and it is triggered if traffic pattern hits a threshold set up by applying gaming theory to the learned patterns. The authors also combined their system with a reputation system to minimize the false-positive rate.

The IDS was evaluated against a SH attack. The evaluation showed similar results to that obtained from SVELTE but with lower energy consumption. Detections of other attacks (WH, BH, Sybil, etc.) were also described in the original design. However, they were never evaluated nor for node mobility.

2.5.5 RPL Specification-based IDS

This is a unique IDS, proposed by Le *et al.* [53]. The IDS is considered as a specification-based, specification-detection, hybrid placement IDS. A profile of RPL specification for routing is generated by the IDS, using trace files, and this profile is used to detect and mitigate various attacks. The IDS was originally designed for SH, RA, local repair, NA, and DIS attacks, but due to its design, it can be extended to cover other attacks as well, as claimed by its authors. Using this approach, a network

will be divided into clusters, each with a CH that has the IDS agents, which are responsible for the profiling, detecting, and mitigation of the attacks.

Simulation results showed a high detection rate with a small percentage of false-positive and false-negative detections (between 0-7%), while energy consumption was up by 6.3% only. However, the authors have not considered the effect of mobility in their investigation, which would be a challenge to the profiling stage of their IDS.

2.5.6 Distributed IDS for Version Number Attacks Detection

Introduced in [58] by Mayzaud *et al.*, this specification-based, signature-detection, hybrid-placement IDS is designed mainly for the detection and mitigation of VA. Several "*monitoring nodes*" are distributed strategically inside the network to monitor DIO messages, check DODAG version number within them, and prepare a "*Possible Attackers*" list based on the fact that a version number change should not come from a higher rank node before lower ones. These lists are then sent periodically to the root node where they are aggregated, and a curated list of the adversaries is created. This final list is distributed to all nodes for the exclusion (isolating the adversaries).

The authors conducted an evaluation of this IDS. Their experiment results showed excellent detection rates. However, it was also shown that to minimize the false-positive detections, at least three monitoring nodes should monitor each normal node.

2.5.7 Real-time IDS for Wormhole Attacks Detection

In general, to detect and mitigate WH attacks is to use location-based methods, and this IDS is no exception. Here, Pongle and Chavan proposed a location-based, signature-detection, hybrid placement IDS [5] to detect two types of WH attacks: Packet Encapsulation and Packet Relay – see Sec. 2.6.3.

The system works as follows: the root node will have all nodes' locations and their transmission ranges pre-stored. Each node then will periodically send information about their neighbors and the *Received Signal Stregnth Indicator (RSSI)* to the root node. The root node then uses all the information to calculate the distances between the nodes and determines if a new neighbor has been formed but out of the transmission range, which means a WH attack. It also can ask individual nodes to monitor their surrounding nodes and report RSSI information to pinpoint adversaries.

Simulation evaluation of the IDS showed promising detection rate (94% for attack

only detection and 87% for both attack detection and pinpointing the adversary) with very low resource usage (except for the root node). However, there was a higher number of control messages than normal RPL (that is, RPL in UM), due to the nature of the IDS. In addition, the IDS is not effective with mobile nodes, according to the authors themselves. The authors mentioned the possibility to extend the IDS for detecting other attacks, namely Sybil, BH and SF, VA, and local repair attacks. However, details of these extensions were not discussed nor evaluated.

2.5.8 SPRT-based IDS for Selective-Forwarding Attack Detection in IPv6-based Mobile WSNs

Based on the Sequential Probability Ratio Test (SPRT) statistical method [60] and combined with an adaptive threshold, Gara *et al.* [61] proposed this statistical/mathematical-based, anomaly-based, hybrid-placement IDS to detect malicious nodes performing SF attacks in RPL-based networks with mobile nodes.

The IDS consists of two modules: a central decision-making module at the root node, and distributed monitoring-and-responding modules at routing nodes. The distributed modules hold a neighbors table that records the number of packets sent and received from each neighbor. Then, these statistics are sent periodically toward the central module at the root node in the form of "*Hello*" messages. These *Hello* messages are sent through random paths (other than the preferred parent) to mitigate the possibility of being filtered by a malicious parent.

Using these information, the central module analyzes the packets sent and received by each node and its parent, to determine if one of them is lying (hence detecting a falsification attack against the IDS). After the analysis, SPRT is used to determine if each node is legitimate or malicious. Once all the decisions are made, the elimination of the detected malicious nodes is conducted by initiating a global repair using a modified DIO message. This modified version of DIO contains the malicious nodes' identifiers. Each receiving node then should discard the malicious nodes from its parent list, which results in full isolation to the adversaries.

The authors performed an evaluation for their IDS using simulations (through Cooja [7], the simulator of Contiki OS). It showed that the IDS is capable of eliminating the SF attacks, with the detection rate reaching %100 with some parameters optimization. However, this excellent performance comes at the expense of having

much high overhead traffic (due to the *Hello* message), which was noted by the authors. It is understood that such overhead would increase drastically with larger networks and slow them to a crawl. In addition, such high overhead traffic results in much higher energy consumption; hence, shortening the life of the IoT network significantly. The last observation points out that the IDS does not use encryption, including for the introduced *Hello* messages. This may introduce a new threat as the adversaries can take advantage of the lack of encryption and send fabricated *Hello* messages on behave of their victims; and so, hindering the detection process.

2.5.9 Summary and Insights

From the discussion above, it is clear that there is a lot of work regarding introducing IDSs for RPL. Observations from the previous sections show the following: The dominant theme for IDS placement is the hybrid-placement system, as it minimizes resources consumption. In regard to IDS detection methods, no theme is dominant (in this chapter, an equal number of IDSs were discussed per detection-method). However, both specification-based and hybrid-based detection methods showed higher rates of detections with medium-to-low resources consumption than signature-based IDSs, which suits the constrained nature of IoT devices. From the mitigation method point of view, the most commonly used methods are the statistical/mathematical-based methods, with specification-based methods coming second. It was also noticed that no IDS has used the secure modes of RPL nor required the use of encryption, with the exception being SVELTE IDS, which only suggests using security features for the other layers, ignoring RPL's security measures.

2.6 WSN-Inherited Attacks

Since a large part of IoT is originated from WSNs, many of WSN routing attacks have migrated to IoT networks, with small changes in their methods to cope with IoT paradigm. These attacks are discussed in the following sections.

2.6.1 Blackhole (BH) and Selective-Forward (SF) Attacks

In a Blackhole (BH) attack [52], malicious node(s) will drop all packets it receives, instead of forwarding them, resembling a "blackhole" in the network and causing a

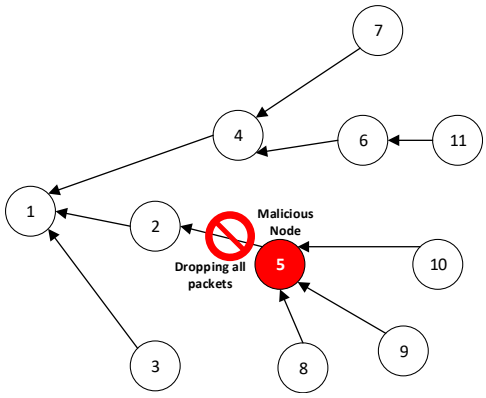


Figure 2.7: RPL network with BH attack. The malicious node (node 5) had participated in the topology, after which it started to drop all received packets.

DoS - see Fig. 2.7. A more sophisticated attack would forward packets from selected protocols and drop the rest, e.g., forwarding RPL control messages and ICMPv6 packets, and dropping everything else. This is known as *Selective-Forward (SF)* [14] or *Greyhole* attack. It has been shown in [62] that BH attacks by themselves are not very effective unless a large number of adversaries were deployed strategically, as RPL self-healing features will eventually remove these BH adversaries from the topology. However, combining the BH attack with other types of attacks, e.g., SH attack, can degrade the Quality of Service (QoS) in the network and even results in DoS to some parts of the network.

The main difference between BH and SF attacks lies in their purpose: BH attack is intended to create a DoS inside the network, while SF's target is to make a significant disruption in the routing topology [14, 52]. Because of their nature, SF attacks cannot be detected or mitigated by the self-healing mechanisms of RPL, as malicious nodes usually pass RPL's control messages and participate in DODAG creation and maintenance as the legitimate nodes [14, 52].

Mitigation techniques: There have been a lot of work to mitigate BH and SF attacks, with mainly IDSs used for BH and encryption for SF attacks. A summary of these mitigation methods goes as follows:-

1. Using source routing, or creating a disjoint path between leaf nodes and the root minimizes the effect of SF attacks [14]. However, it is not easily applicable to large networks.

2. Also, encrypting RPL messages mitigate SF attacks from external adversaries[1], as it eliminates the adversary's participation in the DODAG and its ability to identify the used protocols [52]. However, using encryption will slightly increase the E2E delay.

3. *Acknowledgment-based methods:* Two methods have been proposed:-

   - *Parent Fail-Over mechanism:* Proposed by Weekly and Pister [63] as part of their solution to mitigate a combined SH and BH attack. In this mechanism, root node marks a node as a BH if it did not hear from it for 10 seconds, and sends a list of these *"Unheard Nodes"* in a modified DIO message to all the nodes to blacklist it. However, and according to the authors, parent fail-over is not efficient when the BH attack is combined with a Sybil attack.

   - *Heartbeat mechanism:* Proposed by Wallgren *et al.* [14] to detect/mitigate SF attacks. With this mechanism, an ICMPv6 echo message is sent from the root node to each node, with its reply expected. It is advised by the authors to use IPSec with *Encapsulated Security Header (ESP)* to get the full potential of the protocol and prevent adversary node(s) from identifying ICMPv6 messages.

4. *Trust-based methods:* several solutions were proposed, including:-

   - Airehrour *et al.* in [64] proposed a trust-based mechanism to detect and isolate BH. Each node will calculate "*Trust Values*" for all its neighbors (by monitoring their transmissions), then uses them to choose its preferred parent. However, it is necessary to investigate their mechanism's efficiency when mobile nodes are involved, as authors used static topology for their evaluation.
   - The work of Khan and Herrman in [65] proposes a distributed mechanism that also finds trust values. However, these values are sent to the root node where they are aggregated into "*reputation values*," which in turn are used to identify adversaries. In addition, trust values are affected by the results of several checks, including "*Forwarding check*" for SF and BH attack detection.

[1] More information on external and internal adversaries can be found in Sec. 3.2.3

- Djedjig *et al.* in [66,67] proposed a trust-based version of RPL. Their proposal requires a *Trusted Platform Module (TPM)* co-processor to be implemented within each node to handle the extra cryptography. Nodes within the network will compute *Extended RPL Node Trustworthiness (ERNT)* for their neighbors based on several parameters, such as energy, rank, and neighbor's interaction/collaboration with other nodes. Then, a newly introduced *Trust Objective Function (TOF)* will use ERNT as a routing metric instead of the rank to create the DODAG. However, the authors' evaluation was a simple one, performed by giving only one node a bad reputation and checking if their mechanism will avoid that node (which it did promptly). It would be very interesting to see how the system would react to actual, different routing attacks, and to evaluate its resources consumption and under node mobility.

5. *Statistical/Mathematical-based Methods:* As reported by their authors, InDReS and the Game theory IDSs (see Sec. 2.5.3 and Sec. 2.5.4), can be used to detect and mitigate BH and SF attacks. In addition, SPRT-based IDS (see Sec. 2.5.8) was explicitly designed to eliminate SF attacks, and by extent, BH attacks.

6. *Specification-based methods:* these can be summarized as follows:
   - Many IDSs have been proposed [24] to detect RPL attacks, including BH and SF. As a specification-based IDS, SVELTE (see Sec. 2.5.1) shines as the most comprehensive one, according to [14, 24, 52]. However, other proposed IDSs can be extended to detect and mitigate BH attacks, such as RPL Specification-based IDS (see Sec. 2.5.5).
   - Glissa *et al.* in [50] proposed a security mechanism to be added to RPL, constituting *Secure RPL (SRPL)*. The mechanism uses the rank and the number of descendants to detect mainly SH and RA attacks. However, the authors showed that their mechanisms could also be used to detect other attacks, including BH and SF attacks, provided that a large number of nodes have been disconnected suddenly. The authors proposed to include some anomaly-detection algorithms to enhance the detection of these other attacks.

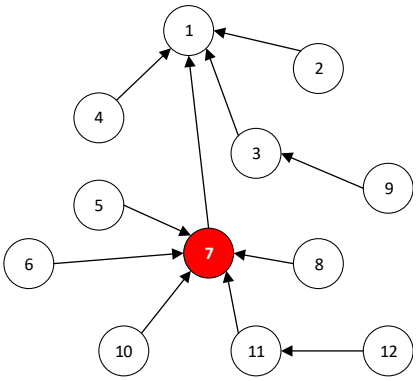


Figure 2.8: RPL network with SH attack. The malicious node (node 7) advertised that it has a better route to the root than its neighbors, resulting in it becoming the parent for many nodes.

2.6.2 Sinkhole (SH) Attack

For this type of attack, malicious nodes try to "*sink*" (hence the name) for as much traffic as possible by advertising a fabricated route with better metrics. This will attract nearby neighbors to send their traffic through the adversaries by selecting them as preferred parents - see Fig. 2.8. By itself, SH attack is not disruptive, but when combined with other types of attacks (e.g., BH or WH attacks); where the passing traffic can be altered, forged, or used for spoofing; the new mixture makes a very powerful attack.

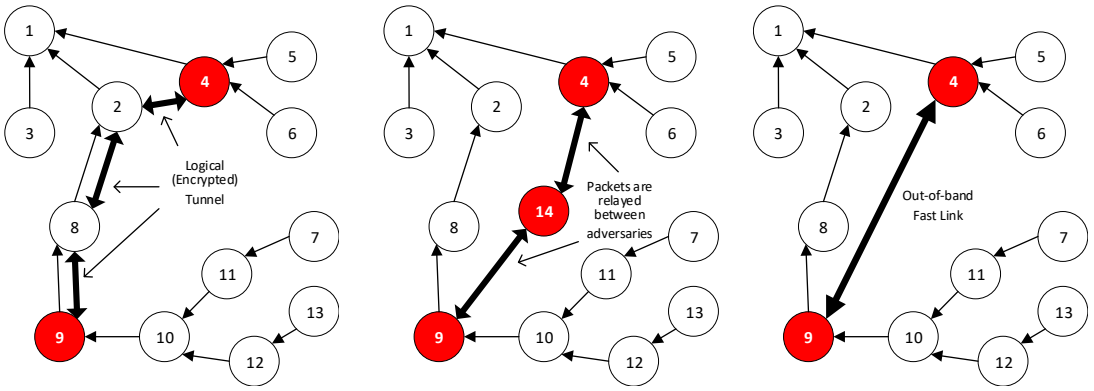
Usually, this attack can be performed in several ways: by advertising a DIO with a better combination of the rank and the OF – see Sec. 2.3 (this version of SH attack is synonymous to the decreased rank attack in RPL, see Sec. 2.7.1). Other ways can be through manipulating preferences or having several adversaries directing all passing traffic toward another adversary (instead of their preferred parent as in the DODAG), creating a sinkhole. However, to the time of this chapter's writing, no literature was found discussing these methods; hence, they should be investigated.

Mitigation techniques:

1. *Trust-based Methods:* The same trust-based methods used to mitigate BH attacks are also valid for SH attacks' mitigation – see Sec. 2.6.1.

2. *Statistical/Mathematical Methods:-*

   - Weekly and Pister [63] proposed two mechanisms to countermeasure SH/BH. Part of that solution is the "Rank Verification" mechanism, which



(a) Packet encapsulation WH (b) Packet relay WH (c) Out-of-band WH

Figure 2.9: The three types of WH attacks. Here, nodes (4, 9, and 14) are malicious, with node 4 forwarding all passing traffic to node 9 instead of the root, causing longer E2E delays and DODAG disruptions.

uses one-way hash-chaining to verify topological ranking.

- IDSs of this type, such as InDReS (see Sec. 2.5.3), Game Theory IDS (Sec. 2.5.4), and INTI (Sec. 2.5.2), can be used to detect and mitigate SH attacks.

3. *Specification-based Methods:* SVELTE (Sec. 2.5.1) and RPL specification-based IDS (Sec. 2.5.5) have been proved effective against SH attacks. Also, the mechanism proposed by Glissa [50] (described in Sec. 2.6.1) can be used for SH attack mitigation.

2.6.3 Wormhole (WH) Attacks

Usually, two (or more) malicious nodes cooperate in this attack, as they create a tunnel [14, 52] between them and transmit traffic (entirely or selectively) through it instead of the regular path dictated by the DODAG – See Fig. 2.9.

In general, there are three ways to create a wormhole [5]:-

- *Packet Encapsulation* [5], where malicious nodes use a legitimate path between them and create a logical tunnel by encapsulating original legitimate packets into encrypted packets – see Fig. 2.9(a). This encryption will hide real hop count from other nodes on the tunnel's path.

- *Packet Relay* [5, 24], in which one (or more) malicious node(s) relay packets between two far legitimate nodes to deceive them into being neighbors, usually this is done by relaying packets without changing the hop count. See Fig. 2.9(b).

- *Out-of-Band Link* [14, 24], here malicious nodes use an out-of-band link (wired or wireless) to communicate with each other – see Fig. 2.9(c). Dangerously, This approach can allow an adversary inside the network to communicate with another adversary outside the networks and bypass the border router or firewalls.

Perazzo *et al.* in [68] conducted an evaluation of WH attack effects using actual testbed. The implemented WH was done via an out-of-band link, and their evaluation was based only on packet loss. The evaluation results showed a significant increase in packet dropping rate due to the lost RPL control messages, which cause longer recovery delays for the affected nodes. In addition, the authors provided critical analysis to some of the countermeasures used to mitigate WH attacks (e.g., the addition of specialized hardware such as GPS, making legitimate nodes monitor their neighbors, timeouts measurements, etc.), and they argued that none of these countermeasures provide a practical solution, due to the extra costs or the high impact on the devices' resources. However, the authors did not support their argument with any substantial evidence, nor did an actual evaluation of such mitigation methods through their testbed network. More work should be done on such evaluation, for example, to include energy consumption and E2E delays effects.

Mitigation techniques: Out-of-Band WH are very difficult to detect and still present a big challenge [14], but for the other two types of the WH, there are few ways to detect them [5, 14]:-

1. *Location-based Methods:*

    - *Geolocation-Based methods:* Tying up geographical information to nodes and neighborhoods of the network makes it much easier to detect and stop a WH attack [14]. For example, embedding GPS hardware inside the nodes – or using indoor location beacons – and integrating the geographical location into path authentication [5]. However, adversaries could easily deceive this method by sending misleading location information. Besides, this method still faces difficulties in identifying mobile adversaries.

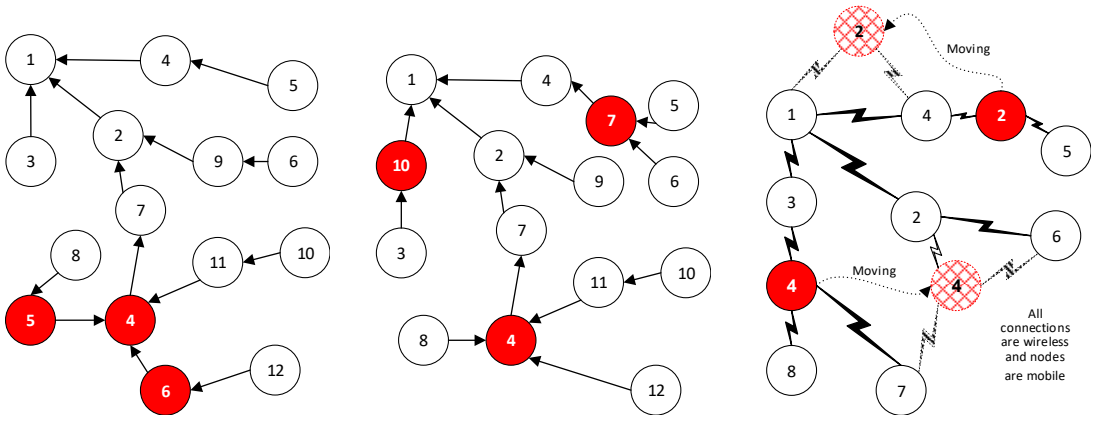
- *Packet-Leashes methods:* First introduced in [48], packet leashes simply constrain the packets' trip to specific limits within the network (geographical or temporal) [5]. This provides a very easy-to-implement solution; however, it needs either GPS hardware integration (geographical leashes) or tight clock synchronization between all the nodes (temporal leashes).
- *RTT-Based methods:* Mainly developed for WSNs, these methods make each node records the RTT for all other nodes in its range and use that information to estimate their distances; hence, a virtual map is constructed and used for WH detection [69,70]. These methods fail if the adversary(ies) are using high-speed connections. So, in many cases, these methods are combined with distance verification methods to enhance true-positive detection and decrease the false-positives.
- Location-based IDSs, such as the one proposed by Pongle and Chavan – see Sec. 2.5.7 – can be used to mitigate WH attacks.

2. *Statistical/Mathematical-based Methods:* Game theory IDS (see Sec. 2.5.4) can detect and mitigate WH attacks via extension, as per its authors. Also, the use of Merkle tree authentication to mitigate WH attacks was proposed by authors in [71].

3. *Specification-based Methods:* SVELTE (see Sec. 2.5.1) and RPL specification-based IDS (see Sec. 2.5.5) can be used to detect and mitigate WH attacks (via extensions). In addition, Lai [72] proposed a detection mechanism that uses the ranks extracted from all DIO messages passing by and uses that to detect the WH adversary. The author evaluated the system using simulation in different environment settings and showed a 100% true-positive detection rate in all settings. However, they did not evaluate the energy and resources consumption of their mechanism.

2.6.4 Clone ID and Sybil Attacks

In the Clone ID attack, a malicious node(s) takes the identity of another legitimate node. A more sophisticated attack, the Sybil attack, can be performed with each malicious node taking several legitimate nodes' identities [14, 52]. Zhang *et al.* [73] classified Sybil attacks according to their social connections and mobility into three types (it is to be noted that each malicious node can take and use several identities):-



(a) SA-1 (b) SA-2 (c) SA-3

Figure 2.10: Sybil attack types, according to [73].

- ***SA-1,*** in which malicious nodes are organized in one area with more bonds between them than the legitimate nodes. The main goal of this type is to manipulate data; e.g., in a sensing system, it can steal the identity of many legitimate nodes and submit forged information to manipulate the system. Also, in SA-1 all the nodes are fixed. See Fig. 2.10(a).

- ***SA-2,*** This is more common in sensing networks, where malicious nodes are distributed among legitimate nodes and not bonded to one area. Also, this type is much harder to detect as malicious nodes have many "socially normal" connections with the legitimate nodes. The main goals of SA-2 attack in RPL-based IoT networks are to disrupt the routing topology and manipulate any reputation-based system. As with SA-1, all the nodes here are fixed. See Fig. 2.10(b).

- ***SA-3,*** Very similar to SA-2, but here the nodes can be mobile, which reflects on their social connections: they do not have long relationships with their neighbors. The goals are the same as in SA-2, but the discovery of these nodes is harder due to mobility. See Fig. 2.10(c).

Besides that, malicious nodes can use *stolen identities* from legitimate nodes, or use *fabricated* ones to communicate. However, detecting and eliminating the fabricated ones is somewhat easier by using whitelists. Also, malicious nodes can use

all of their identities at the same time (*Simultaneous Sybil*) or use a subset (*Non-Simultaneous Sybil*) [74].

Medjek *et al.* [74] made an analytical analysis to estimate the impact of Sybil attacks in RPL, with mobility in mind. It has been shown that a Sybil attack will significantly increase control overhead (due to repeated DODAG repair/construction), decrease packet delivery (either control messages take priority, or packets are delivered to malicious nodes instead of the original legitimate nodes), and increase energy consumption (due to the increased overhead and repeated retransmission of undelivered packets). However, no practical nor simulation-based study has been done to evaluate the effects of Sybil attacks on RPL-based networks.

Mitigation techniques: The best way to mitigate Clone ID/Sybil attacks is by keeping track of instances of each node/identity, better with geographical location info, as two instances of the node cannot be at two locations at the same time [14]. Another way is to use Distributed Hash Table (DHT) to follow the instances [52]. Other methods are summarized in the following:-

1. *Trust-based Methods:* Zhang *et al.* [73] proposed the following trust-based schemes to detect and mitigate Sybil attacks in Social IoT[2] [75]:-

   - *Social Graph-Based Sybil Detection (SGSD):* This scheme depends mainly on creating a "social" graph of the network to identify the malicious Sybil nodes. The detection of the adversaries is based on two assumptions:

     (a) Sybil adversaries have tight connections with each other and limited connections with legitimate nodes.

     (b) Sybil nodes cannot have tight connections with several legitimate nodes in several social communities.

     Each node will implement either one or both of these assumptions to build trust relationships with its neighbors. By looking back at the characteristics of Sybil attack types, the scheme's two assumptions represent an excellent match to SA-1 characteristics. Hence, this scheme is effective against the SA-1 type [73].

   - *Behavior Classification-Based Sybil Detection (BCSD):* In this scheme, the behavior of the nodes within the social graph is the basis for detecting

[2] In Social IoT (SIoT), the network is organized based on the "*social*" relationships between IoT devices in a way similar to how humans make social relationships [75].

Sybil nodes. The assumption here is that adversaries will follow limited, repetitive, and specific behaviors compared to normal users, even though they may have several, tight connections with legitimate nodes in several social communities. Due to the similarities between the characteristics of type SA-2 Sybil attack and the assumption used in this scheme, this detection method is effective against type SA-2 [73].

- *Mobile Sybil Defenses:* Mainly developed to mitigate type SA-3, these schemes swap the need for a global social graph with other methods to overcome the decentralized nature of mobile IoT networks:-
  - *Friend Relationship-Based Sybil Detection (FRSD):* In these schemes, trust relationships are built by maintaining a "friends" list at each node. This list is based on specific social parameters, e.g, nodes belonging to one community or performing similar tasks. However, it was mentioned in [73] that these schemes require knowledge of the communities and ways of identifying nodes' community in advance, which is not efficient.
  - *Cryptography-Based Mobile Sybil Detection (CMSD):* These schemes are based on using crypto-identities or pseudonymous identities and signatures to make trust relationships and hence detect Sybil nodes.
  - *Feature-Based Mobile Sybil Detection (FMSD):* Using network features or characteristics to detect Sybil nodes is the basis for these schemes. For example, receiving packets from several nodes with identical Received Signal Stregnth (RSS) and direction would mean that these are stolen identities of a Sybil node, or detecting that several nodes are transmitting with the same RSS and are moving together.

2. *Specification-based Methods:* By adding extensions that collect geographical information or hints [56], both SVELTE (see Sec. 2.5.1) and RPL specification-based (see Sec. 2.5.5) IDSs can use this information to build a network profile that would be able to detect Sybil/cloneID attacks.

3. *Statistical/Mathematical-based Methods:* As mentioned by their authors, Game theory IDS (see Sec. 2.5.4) and InDReS (see Sec. 2.5.3) also can be used to mitigate the Sybil attacks via extended versions of the IDSs. The first one can have an extended version of its hybrid detection system that observes the

signal strength distribution around each node. Then, it creates a signature pattern for the network to detect Clone ID/Sybil attacks [59]. On the other hand, InDReS can be extended by adding behavior monitoring and introducing numerical analysis to its detection algorithm [49].

2.6.5 HELLO Flood Attacks

When any node wants to join a network, it usually starts by broadcasting a "*HELLO*" message. In RPL, DIO messages are considered as *HELLO* messages. An adversary can send DIO messages with strong routing metrics and/or a strong signal; then, it might just disappear or reduce its transmission power to normal [14, 52]. This is known as Hello Flood Attack. If the malicious node announced itself using higher transmission power then disappeared or reduced its transmission power, packets from legitimate nodes would fail to deliver [24]. Another effect of such an attack is the exhaustion of legitimate nodes and network congestion due to increased control overhead.

Mitigation techniques: RPL's self-healing mechanisms (both local and global) can mitigate this attack. This is done by using the link-layer metrics to calculate the best route: if no acknowledgment is received, the path is considered bad and a different parent is selected [14, 44, 52]. Also, using geographical information to identify the location of the originating node for each packet can help mitigate this attack: if the originating node is out of transmission range, all of its packets will be dropped. However, combining this attack with other types will increase the difficulty of mitigating them [52].

2.6.6 Summary and Insights

It can be seen that most of the WSN-inherited attacks, on their own, do not have a significant effect, with except for the WH and Sybil attacks. The latter two attacks can cause noticeable degradation to the network's QoS, in addition to being much harder to detect and mitigate due to their nature. However, combining two or more attacks could lead to a significant impact on the network and devices' resources. For example, in many cases, the SH attack is combined with other types of attacks (WH, SF, NA, etc.) to withdraw larger traffic toward the adversary, resulting in a bigger impact.

The observation shows that using one of RPL's secure modes can prevent adversaries from joining the network; hence, mitigate all the aforementioned WSN-inherited attacks. However, these secure modes will not be of the same effectiveness if the adversaries have access to the encryption keys used, as they would join the network as any legitimate node.

From the mitigation methods' point of view, location-based methods are the main ways to be used for WH and Sybil attacks, while trust-based methods provide decent mitigation against SH and, at the same time, introduce another layer of protection against some of Sybil attacks variations. In addition, IDSs that use specification-based or statistical/mathematical-based methods proved to be capable of mitigating several types of the WSN-inherited attacks, while not heavily taxing the scarce resources of the IoT devices.

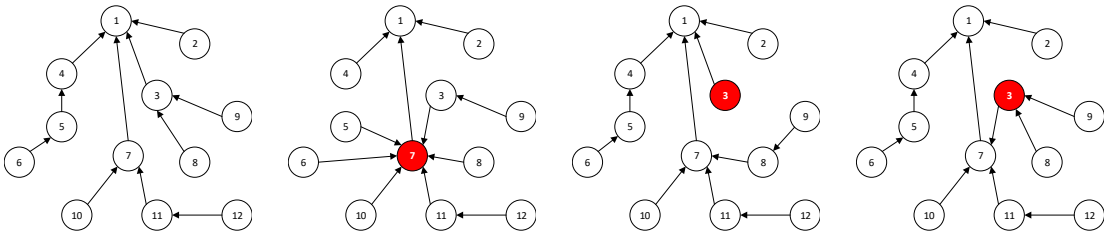
2.7 RPL-Specific Attacks

These attacks are based on characteristics/features of RPL protocol, such as the rank and version fields in RPL's control messages, repair mechanisms, etc. Since these attacks modify control messages, they also known as *Alteration and Spoof Attacks*. The following sections summarize the literature on these attacks.

2.7.1 Rank attacks (RAs)

As mentioned earlier, each node chooses its preferred parent based on two things: the rank of neighboring nodes and the used OF, which are distributed by DIO messages [13]. The rank should increase going downward in the DODAG, and the role of the preferred parent selection is to select the one with the *best* rank. An adversary can manipulate these values, as explained below, and hence severely affect the routing DODAG [14, 24, 52, 76]. It is worth noting that RA, as well as SH attacks, are combined with other types of attacks [14] most of the time, such as SF attack (see Sec. 2.6.1) or IP spoofing [77], to increase the attack impact on the network.

Rank manipulation can be done in two ways: changing the adversary's rank by a specific value, based on the ranks of the adversary's neighbors. This is sufficient if no security measures were applied (such as monitoring IDS). The other way is by manipulating adversary's rank through the use of a different OF (at the adversary)



(a) Normal topology (b) Decreased RA (c) Increased RA (d) Worst parent RA

Figure 2.11: Examples of the three types of RAs, where: (a) is the normal operation of the network without any attacks, (b) a malicious node (node 7) is advertising a lower rank than nodes (2, 3, 4, and 5), resulting in all neighboring nodes to choose the adversary as their preferred parent, (c) a malicious node (node 3) is advertising higher rank than nodes (2, 4, and 7), resulting in all neighboring nodes to select other parents, and (d) a malicious node (node 3) is advertising either its unmodified rank or a lower rank than its neighbors, but forwarding the passing traffic to the adversary's worst parent (assuming that node 7 has the highest rank among node 3's neighbors).

to deceive legitimate nodes into giving the malicious node a better rank. Using this method makes the adversary harder to detect [42, 76] as it can adapt to any changes in the topology dynamically. Beside these two manipulation methods, RAs have the following types [44]:-

- ***Decreased Rank Attacks:*** Malicious node(s) advertises lower rank (with or without changed OF) to other nodes, resulting in many of them selecting the adversary as their preferred parent (see Fig. 2.11(b)). This is basically an SH attack - see Sec. 2.6.2.

- ***Increased Rank Attacks:*** Contrary to the normal behavior of malicious nodes, adversaries here will be near the root node but advertise higher rank and worse routing metrics (OF) - see Fig. 2.11(c). The idea behind this type is to disrupt routing topology and introduce delays, by forcing nodes into choosing further nodes as parents. However, The effect of this type of attack is low compared to the previous type [44].

- ***Worst-Parent Attacks:*** This type is very hard to detect, as the adversary will simply advertise its actual rank but select the worst parent for itself - see Fig. 2.11(d). The results are similar to the previous type, and it can use the decreased rank method to attract more nodes to select it as a parent, while

sending their packets to the worse path [42, 44], resulting in increased delays or even routing loops [42].

Sahay *et al.* in [78] conducted an attack-graph-based assessment on the three types of RAs and supported their assessment with simulations. The results showed that the increased RA introduced more energy consumption and higher overhead traffic, while the decreased RA caused a substantial disruption in the network's traffic and the RPL's DODAG. Finally, the worst-parent RA resulted in network sub-optimization and increased the E2E delays, as expected.

Mitigation techniques: A lot of research was carried out to mitigate rank attacks since it is one of the most effective RPL-specific attacks. A summary is conducted in the following:-

1. *Acknowledgment-based Methods:* Perrey *et al.* in [79,80] proposed *Trust Anchor Interconnection Loop (TRAIL)* as a topology authentication mechanism. The mechanism can be summarized as follows: Whenever a DIO message is received, the receiving node will send a test message that includes a random nonce and the rank of DIO's sender, through that sender toward the root node. It will wait for a signed acknowledgment from the root that includes the DODAG version and the original nonce and the rank of the DIO's sender. All intermediate nodes should verify both the test message and its acknowledgment for the ranks: The DIO's sender rank should be higher than their own, and the previous hop's rank should either lay between the DIO sender's rank and their own (for the test message), or less than theirs (for the acknowledgment). Failing this validation will cause a discard of the message (or the acknowledgment) and a start of a local repair. Once received by the originating node, the acknowledgment's content will be verified and compared to the received DIO message contents. If all fields match, the standard RPL procedures will be carried on. If not, or no acknowledgment was returned within the time-out period, that parent is ignored and another path is selected.

   To minimize the overhead control and enable better scalability for TRAIL, the authors provided an aggregation system based on *Bloom* filters. Their evaluation of the mechanism showed that it is highly effective (at almost 100%) against RAs and VAs, with a max of 20% longer convergence time in the worst-case scenario. However, no energy nor resources consumption were evaluated.

2. *Trust-based Methods:* The mechanisms used for BH mitigation (see Sec. 2.6.1) can also be used to mitigate RAs.

3. *Statistical/Mathematical-based Methods:-*

   - IDSs of this type, such as INTI (see Sec. 2.5.2), InDReS (see Sec. 2.5.3), and game theory IDS (see Sec. 2.5.4), are also valid for RA mitigation, using extensions.
   - Dvir *et al.* in [81] proposed the *Version attack and Rank Authentication (VeRA)* mechanism, a mechanism that uses a one-way hash function to create two related hash chains: one for version numbers and another for associated rank numbers. The authors evaluated their mechanism and showed its ability to detect and mitigate all VAs and decreased RAs. However, it was shown in several reviews [14,44,80,82] that VeRA is still vulnerable to rank forgery by either creating a forged rank hash chain or simply replaying its parent rank. In addition to that, the VeRA's complexity suggests a big increase in resources consumption (energy and others), yet it has to be investigated.
   - Iuchi *et al.* in [82] introduced another mechanism to mitigate RAs, named *Secure Parent Node Selection* Here, each node will compute the *Average Rank* from the DIO messages it receives; then, it uses this average to create a *Rank Threshold* (based on an equation and application-based parameter). Then, the node will exclude any node that has a lower rank than the threshold from its possible parent set, based on the fact that adversaries will announce much lower rank value to attract most of their neighbors. The system showed very interesting results, mitigating 95% of the attack. However, it was found to increase convergence time slightly. In addition, there is no energy nor resources consumption evaluation.

4. *Specification-based Methods:* SVELTE (see Sec. 2.5.1) and RPL specification-based IDS (see Sec. 2.5.5) can mitigate RAs. In addition, the mechanism proposed by Glissa [50] (see Sec. 2.6.1) also can be used for RAs mitigation.

2.7.2 Version attack (VA)

This attack is an exploit of the Global Repair feature of RPL. As mentioned in Sec. 2.3.4, only the root node is supposed to change the version number of the DODAG. However, the current RPL standard does not have any mechanisms to assure the integrity of the communicated version number [58].

If a malicious node sends a DIO message with a higher version number, it will force the global repair procedure to start and create topology inconsistency and routing loops, especially if the malicious node was far from the root node [83].

Aris *et al.* in [83] showed, experimentally, the different effects of VA on an RPL-based IoT network realistic configuration, using a probabilistic adversary scheme and considering mobility: packet delivery was down by more than half the normal value, with the effect getting worse when malicious nodes are near the leaves. Average delays also increased six folds, control overhead was up by 7500% on average, and power consumption was increased by 265%. It is clear that VA's main goal is to exhaust nodes' resources and slow down packet delivery to a crawl.

Mitigation techniques:

1. *Acknowledgment-based Methods:* TRAIL (see Sec. 2.7.1) proved to be very effective against the VA.

2. *Trust-based Methods:* Same trust-based methods used for BH attacks' mitigation (see Sec. 2.6.1) are applicable to mitigate VA.

3. *Statistical/Mathematical-based Methods:* InDReS (see Sec. 2.5.3), Game theory IDS (see Sec. 2.5.4), and VeRA mechanism (see Sec. 2.7.1) are all used for VA mitigation (all these IDSs require extensions).

4. *Specification-based Methods:* SVELTE (see Sec. 2.5.1) and RPL specification-based IDS (see Sec. 2.5.5) can mitigate VA. In addition, the distributed IDS proposed by Mayzaud *et al.* (see Sec. 2.5.6) was designed to mitigate VA.

2.7.3 Local Repair Attacks

In this type of attack, a malicious node will send local repair messages to its neighbors periodically, without actually having any problems. This will cause neighboring nodes

to recalculate their routes that go through the malicious node, ending in constructing the same topology again. Clearly, the main goal of this attack is to create much more control overhead and exhaust nodes' resources [52]. Le *et al.* showed in [84] that this attack could severely affect delivery ratio for the network due to both increased control overhead and that compromised nodes (and its sub-DODAG) will be involved in the local repair process. However, it has less effect on the E2E delay as the routes still have the optimal path as long as the attack is not combined with other types of attacks [42].

Mitigation techniques: Up to the time of writing this chapter, there is no dedicated mitigation method for the local repair attack. However, some proposed solutions for other attacks and some IDSs can be extended/used to mitigate local repair attacks:-

1. *Acknowledgment-based Methods:* Authors of TRAIL (see Sec. 2.7.1) suggested an extension to their mechanism to detect local repair attacks. However, it was not described nor evaluated.

2. *Trust-based Methods:* Mechanisms proposed by Djedjig *et al.* and Airehrour *et al.* (see Sec. 2.6.1) had mentioned a possibility to extend their mechanisms to detect other types of attacks, including the local repair ones. However, no details nor evaluations were conducted.

3. *Specification-based Methods:* Authors of SVELTE (see Sec. 2.5.1) mentioned an extension of their IDS to detect local repair attacks. In addition, RPL specification-based IDS (see Sec. 2.5.5) can be used directly to mitigate these attacks. However, it was shown that it starts with high false-negatives before the true-positive detection rate increases after some time [53].

2.7.4 DIS Attacks

When a new node wants to join a DODAG, a DIS message is sent to inform the surrounding nodes and get the DODAG information from them [13]. An adversary could misuse this feature and periodically send DIS messages to its neighbors. As mentioned in Sec. 2.3.1, the adversary can either:-

- *Broadcast* his DIS messages, forcing several local repair upon neighboring nodes. Depending on the number of affected neighboring nodes and the scale of the

attack, some anomaly- or specification-based IDSs can detect the attack [53].

- *Unicast* the DIS messages, causing an unnecessary exchange of unicasted DIO messages with the neighboring nodes, without resetting their trickle timers [52, 84].

In either case, the adversary can send these DIS messages using its real IP address or a fake one [84].

Le *et al.* [84] studied the effects of DIS attacks, unicasted and broadcasted, in great details. They found that, while DIS attacks do not affect delivery rates (the attack does not change the topology), it does increase the E2E delay significantly. The broadcasted DIS attack specifically can increase the delay by up to 200%, as it forces the neighbors of the adversary to broadcast DIO messages to their neighboring nodes. This process will create more affected routes; hence, larger amounts of control overhead. On the other hand, in the unicasted DIS attack, these neighbors will unicast their DIO messages, thus produce less control overhead than the broadcasting method. This will reduce the possibility of detecting the adversary by anomaly-based or specification-based IDSs.

Mitigation techniques: Up to the time of writing this chapter, there is no dedicated mitigation method for the attack. However, as mentioned in Sec. 2.3.1, IETF proposed some modifications to nodes' response to DIS messages to reduce the effects of such an attack or eliminate it altogether. These modifications are yet to be tested and investigated. In addition, Le *et al.* [53] showed that their specification-based IDS (see Sec. 2.5.5) could provide good mitigation against DIS attacks.

2.7.5 Neighbor attacks (NAs)

In this attack, an adversary will forward any DIO message it gets to its neighboring nodes, without modification, creating the illusion that the original sender is in the range of neighboring nodes [52, 84] while it is not. A worst-case scenario is when the original sender has a good rank and the adversary's neighbors choose it as their preferred parent, even though it is out of their range. It is worth mentioning that while NA also occurs in mobile Ad hoc networks [85], it is usually performed with all exchanged messages, unlike NAs in RPL, which are performed by forwarding unmodified DIO messages only.

The effect of this attack by itself is not significant, as it results only in a slight increase of E2E delay [84], however combining this attack with other types of attacks could make it more dangerous. For example, an adversary could launch a DIS attack to get DIO messages with better metrics, then selecting one of these messages to perform the NA, increasing the effect of such an attack.

Another alteration to this attack is known as *Route Information Replay attacks* [44]. Here, an adversary can store control messages from other nodes, then sends them later to its neighbors, forcing them to update their routing tables with outdated data and creating an inconsistent topology [44].

Mitigation techniques: It is challenging to detect this type of attacks [52, 84], as its adversaries appear transparent to the network. Looking at the available literature, there is very little research on the mitigation of these attacks. Some IDSs can be extended to detect these attacks, such as SVELTE (see Sec. 2.5.1) and RPL specification-based IDS (see Sec. 2.5.5), as mentioned by their authors [53, 56]. In addition, implementing a location-based mitigation method that ties up location and transmission range information of the nodes can theoretically help detect the attack.

2.7.6 RPL Storing Mode Attacks

These attacks require that RPL run in the storing MoP – see Sec. 2.3.3 – as they target stored downward routing tables [43]. In general, there are three types of these attacks:-

- ***Routing Table Overload Attacks:*** Since each node maintains its own routing table, an adversary can send many bogus routes (via DAO messages) to the compromised node(s) until it is saturated. This will prevent the compromised node from accepting legitimate DAO messages and build correct routes [44]. To the time of writing this chapter, there is no detailed study of the effects of this attack, only a few mentions in the literature [86].

- ***DAO Inconsistency Attacks:*** DAO inconsistency and its corresponding repair mechanism have been discussed in Sec. 2.3.4. However, an adversary can misuse this repair mechanism and simply return any packet that it receives with the "F" flag set. This will cause either full isolation for the sub-DODAG of the adversary, or creates sub-optimal routes to the destinations, resulting in longer E2E delays – see Fig. 2.12.

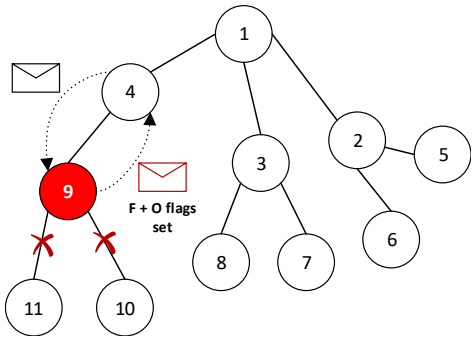


Figure 2.12: DAO inconsistency attack. Node (1) is the root node.

- ***Route Table Falsification Attacks:*** a malicious node(s) will advertise fake routes to other nodes, whether these nodes exist but not in the sub-DODAG of the adversary, or entirely forged [44]. This causes packet losses (in case of forged routes) or longer E2E delays.

In addition to the effects mentioned above, these attacks cause resource exhaustion for compromised nodes due to increased control overhead and repetitive repair attempts [44].

Mitigation techniques: To the time of writing this chapter, there is no proposed method to detect and mitigate RPL storing-mode attacks or a detailed study of the effects of such attacks on RPL networks. However, for DAO inconsistency attacks, RFC6653 [43] proposes limiting the rate of trickle timer resets (due to discarding routes using the recovery method) to a value, suggesting 20 resets per hour, as a method to limit the effect of the attack. This method was challenged by Pu in [87], where the author showed that the adversaries still can initiate the attack while not reaching this fixed limit, and he proposed the *Dynamic Threshold Mechanism (DTM)* to set the rate limit of the trickle timer reset. Simulations of DTM (as carried out by the author) proved its efficiency, as it reduced the control overhead, packet drops, and energy consumption of the network.

2.7.7 DODAG Inconsistency Attacks

In Sec. 2.3.4, DAG inconsistency was discussed with its RPL's repair mechanism. It has been shown in [43,88,89] that an adversary can misuse this mechanism to initiate an attack against the network. In this case, the adversary can:

1. Send inconsistent packets (e.g., for the upward direction, it sets the "O" flag; or for the downward direction, it clears the "O" flag) with the "R" flag also set [43]. Or

2. Manipulate "O" and "R" flags of any packet it receives before forwarding it [88].

The effects of DODAG inconsistency attacks usually are: much higher control overhead, energy/resources exhaustion, and longer E2E delays. If packet manipulation is used to initiate the attack, it creates a BH-like situation that is very hard to detect (legitimate next-hop node will always drop the packet and start a local repair procedure), in addition to the previously mentioned effects.

Mitigation techniques: Currently, there is no method to detect these attacks actively. RFC6553 [43] proposes to use the same limit on the rate of trickle timer resets to mitigate the attack, as was proposed for DAO inconsistency attacks. However, Sehgal and Mayzaud *et al.* showed in [88] that having a fixed approach to this limit will not be efficient to mitigate the attack if the adversary kept its attack rate near the limit. They proposed to use an adaptive approach for the trickle reset limit [88] that uses the ratio between inconsistent and normally forwarded packets. Later, this approach was extended by the authors to become a fully dynamic limit-setting approach [89], based this time on network conditions (neighbors of the node, number of inconsistencies, etc.). Using simulation, both approaches proved to be significantly reducing the control overhead, packet drops, and energy consumption, with a negligible effect on processing resources.

2.7.8 Replay Attacks

The idea behind these attacks is to record legitimate control messages (DIO, DIS, and DAO) and forward them later in the network [44, 86]. Such attacks can be performed even with RPL secure modes active, as the adversaries do not need to know the used cryptography keys. These attacks will result in outdated or false routing entries, degraded routing service, lower packet delivery rate, and in some cases, detachment of the victim's sub-DODAG. It is worth noting that replay attacks also occur in WSNs; however, it is performed with data messages, not control messages, unlike replay attacks in RPL, which are mainly performed by replaying control messages.

A special type of replay attacks, the *DIO Suppression Attack*, was recently discovered and detailed by Perazzo *et al.* in [90]. This attack uses the concept of replay

attacks to exploits the Trickle algorithm governing DODAG creation and maintenance (see Sec. 2.3.2). In its simplest form, the adversary will replay a DIO message it heard from a legitimate node (parent to the victim) several times with a fixed frequency that is enough to convince the victim(s) that there are no changes in the network and double its trickle timer; hence, the attack suppresses the victim's DIO message transmission. This would lead to imperfect routing (degraded QoS) or, in the worst-case scenario, a complete detachment of the victim and its sub-DODAG from the network. The authors described different techniques to launch the attack in more complex situations, such as when encryption is implemented at the link-layer or within RPL. Their evaluation of the attack proved its significant effect on degrading the delivery rates of the network. However, it was also proven that this attack could be mitigated by enabling RPL's optional replay protection mechanism (see Sec. 2.3.5), even though this mechanism would increase [46] network formation delay (3 to 4 times more), power consumption (around 18% increase), and control overhead (around 10% increase).

In general, replay attacks have more damaging effects on dynamic networks than on the static ones: adversaries can record control messages from their neighbors at one end, then move to another part of the network and replay these control messages, forcing the new neighboring nodes to update their routing tables with outdated and false information.

It is worth mentioning that the WH, CA, Sybil, and NA attacks can be considered as types of replay attacks, as they replay RPL control messages in specific ways.

Mitigation techniques: RPL's replay protection mechanism is efficient in mitigating the replay attacks in static networks. However, there are no detection/mitigation proposals in the literature for dynamic networks. In addition, all current implementations of RPL (up to the time of this writing) do not have the optional reply protection mechanism.

2.7.9 Summary and Insights

Unlike WSN-inherited attacks, most of the RPL-specific attacks have a larger impact on the network by themselves. The exception to this rule would be the DIS attacks and NAs, which mainly affect the E2E delay. From a mitigation methods' point of view, specification-based methods are the most effective for the mitigation of the attacks. However, many of the current proposals of these methods do not provide full

details or complete implementations for the mitigation of all RPL-specific attacks.

It can be argued that there is an urgent need for more research on detection and mitigation methods for DIS attack, NA, RPL storing mode attacks, DODAG inconsistency attacks, and replay attacks in dynamic networks or with mobile nodes. There is not much research in this area. In addition, more work should be conducted to incorporate some of the proposed mitigation mechanisms into RPL's standard, with optimization for the different types of networks where RPL would be deployed.

2.8 Issues, Challenges, and Future Directions

There is a great deal of research on mitigating routing attacks in RPL-based networks, which shows clearly from the investigation in the previous sections. However, there are still many issues about how researchers approach the mitigation solution. In the following, the main issues and challenges that face the research community are listed, and a few future directions are suggested to be used to approach the future work on mitigating routing attacks in RPL.

2.8.1 Implementation of RPL's Security Features

As mentioned in Sec. 2.3.5, even though the RPL standard has numerous pages on security features, it marks them as "optional." Hence, and to the best of the author's knowledge, no current implementation of RPL (that has been used in either real applications or research work) has these security features available, except for the partial implementation provided by Perazzo *et al.* in [46]. In essence, this means that the effects of such security features on mitigating routing attacks has not been assessed, which may make some of the added mitigation methods useless or redundant.

The investigation in this chapter suggests that RPL's security features should be one of the first tasks to be investigated in any future research. The emphasis should be on seeking ways of implementing the third mode of security: *Authenticated Secure Mode (ASM)*, or providing alternative methods that improve RPL security in an efficient manner without changing the way it works. Such implementations could be challenging for current platforms in terms of resource consumption; however, with the rapid advancements in hardware, implementing the full list of RPL's security features might be more feasible.

Also, no recent literature, to the best of the author's knowledge, considers these security features in their work. Such negligence may have resulted in less efficient implementations for many of the mitigation methods investigated in this chapter, which results in more resource consumption at the nodes.

2.8.2 Effects of RPL's Routing Attacks on Large Networks

There is a little work on studying the effects of routing attacks on large IoT networks (with more than 250 nodes) [91]. Large IoT networks with many nodes introduce many challenges on several levels, design-wise and security-wise. On the security front, such networks introduce larger delays to DODAG formation and maintenance times, leaving devices exposed to several effective attacks such as Sybil, replay attacks, or any location-based attacks [92]. One suggestion is to consider the work of Abdou *et al.* in [93] as one of the future directions that should be investigated for adaptation in IoT networks, as it focuses on verifying the locations of the nodes in large networks.

In addition, the large number of RPL control messages used would jam these networks for long periods of time (with larger control traffic if the optional replay protection mechanism is implemented [46]), which could be used by some adversaries as a DoS attack. The current approaches [94–97], among many others, are trying to solve this challenge. However, those papers have not shown much detail on how this will affect the security of these networks.

2.8.3 The Large Number of Optional Features in RPL

RPL standard has so many optional features to provide flexibility for many different applications. However, this would lead to many incompatible implementations as each will have a different set of optional features [98]. From the security point of view, having several implementations to communicate with each other could lead to unintentional or even undiscovered attacks. For example, when a network works in non-storing MoP and a new sub-DODAG is joining the network but with an implementation that uses storing MoP, or when nodes use different OFs. It would be argued that streamlining the standard and eliminating the never-used (or outdated features) would lead to more consistent implementations.

2.8.4 Adaptation of RPL to Different Networks Environments (such as M2M, Smart Grids, etc.)

As the IoT world expands, new network environments are introduced; hence, new routing and security requirements arise. For example, there has been a lot of work on Advanced Metering Infrastructure (AMI) and M2M networks recently. With RPL becoming the standard routing protocol for IoT networks, many researchers are looking into adapting RPL to the unique requirements of these networks [99–102]. Aijaz *et al.* in [103, 104] proposed Cognitive and Opportunistic RPL (CORPL) to adapt RPL for the Cognitive Radio (CR)-based smart grid networks. CORPL introduces opportunistic forwarder set (parents) election and coordination scheme for the best forwarder selection. That has been done by using a specially crafted OF and some modifications to RPL and Medium Access Control (MAC) layer protocol. The authors evaluated the proposed protocol using MATLAB simulations, and it showed that CORPL significantly enhanced the packet delivery rate when compared to regular RPL, especially in poor link conditions. However, no evaluation for the performance of the protocol under routing attacks was performed.

Another approach to routing in AMI and smart grid networks, this time proposed by IETF, is the Ad-hoc On-demand Distance Vector (AODV)-RPL [105]. Currently an Internet-draft, the proposed protocol combines the reactive nature of routing on-demand in AODV with the normal RPL operation. The intention is to provide AMI networks with asymmetrical bidirectional paths between the originating and target nodes when the normal RPL path is not available or not reliable enough. Since it is in the early stages of development, no evaluation was conducted on the protocol's performance nor its resilience against security threats.

From this observation, it is clear that there is an urgent need to conduct more research on the security requirements in these new environments and evaluate the effects of routing attacks on them. Also, it is necessary to investigate how RPL (with all the proposed modifications to adapt to the new environments) will react to the attacks that can only be conducted in these environments.

2.8.5 Scalability of IDSs for RPL

Most of the IDSs discussed in this survey have a margin for extensibility, to add detection (and/or mitigation) capability of attacks that were not in the original design

of the IDS. However, the details of such extensions and their evaluation were not provided in most of the reviewed literature. In addition, none of RPL's own security features were considered at the design of all IDSs, as to the best of the author's knowledge. In the previous sections, it has been shown that an extensible IDS which uses RPL in ASM (or a newer secure mode) to mitigate routing attacks would be more efficient. However, no such IDS is currently proposed, making it a suitable research direction.

2.8.6 Advancements in Platform's Hardware

As found from the investigation in this chapter and confirmed in [98], most of the evaluation presented in the literature is carried out with simulation on ContikiOS or TinyOS, and they used either the TelosB [106] or Zolertia Z1 [107] platforms. Both platforms are now considered outdated; TelosB has pre-2010 components, and Zolertia Z1 has been retired by its manufacturer and replaced with more updated motes. This would represent a challenge because the necessary mitigation methods and security features cannot be implemented in a fully functional state (due to higher resources consumption). More recent hardware platforms, such as the ones based on Cortex-M3 [108], should be used for both real-world or simulated evaluation.

The current pace of advancements in electronics provides the basis to develop newer and better-equipped platforms. For example, developing multi-core processors that can handle complex operations and are energy efficient at the same time, investing in research to develop new large-capacity batteries that are small in size, and finally, improving storage mediums to have higher storage sizes in smaller chips formats and with small energy footprint. This would make the future implementation of security features or measures much more feasible and lead to more research in this area (e.g., importing the traditional security measures from the more powerful computer networks world.)

2.9 Summary

In this chapter, the RPL's standard, including the recent IETF proposal of modification to the standard, have been thoroughly reviewed. The recently published attacks on RPL were classified and investigated thoroughly, showing up-to-date results from recent literature. In addition, the mitigation methods of these attacks have been

reviewed, and a novel technique-based classification scheme have been introduced for the mitigation methods. Besides, IDSs were discussed and comprehensively investigated, with the most influential ones highlighted. In addition, RPL was shown to be prone to various vulnerabilities, and it is clear that its security still needs more research. A hybrid-detection IDS with hybrid placement, such as the SVELTE and RPL's Specification-based IDSs, were shown to be the best current solution to mitigate several RPL attacks at the same time. However, more research should be conducted in order to further reduce and optimize resource usage of the IDSs to accommodate the constrained resources of IoT devices. In addition, RPL's security features should be evaluated for attacks' mitigation.

The next chapter provides an in-depth evaluation of RPL's security mechanisms under common routing attacks, which in turn should shed some light on the strengths and weaknesses of these mechanisms.

Chapter 3

Analysis of RPL's Performance under Common Routing Attacks

3.1 Introduction

The previous chapter provided an in-depth review of RPL standard, its "optional" security mechanisms, common routing attacks, and the countermeasures proposed to detect and mitigate these attacks. It was also shown that, interestingly, there has been no research discussing the effects of using RPL's security mechanisms, specifically under routing attacks. This is most probably due to the lack of an official implementation of RPL's security mechanisms in any of the available IoT OSs, such as Contiki OS [8] and TinyOS [45].

However, Perazzo *et al.* in [46, 47] claim to provide the first standard-compliant, partial implementation of RPL security mechanisms. One secure mode, the PSM, and the optional replay protection, the CC mechanism, were introduced to ContikiRPL (Contiki OS version of RPL). The authors provided an evaluation for their implementation and compared RPL's performance between using and not using the PSM. However, It is worth noting that the authors did not evaluate their implementation against actual attacks.

In this chapter, a comprehensive investigation of RPL's performance and security

This chapter is a revised version of the paper "*Secure Routing in IoT: Evaluation of RPL's Secure Mode under Attacks*," published in IEEE GlobeCom 2019 conference [109], and its extended version "*Enhancing Routing Security in IoT: Performance Evaluation of RPL's Secure Mode under Attacks*," published in the IEEE Internet of Things Journal [2], 2020

is conducted (through simulations in Contiki OS via its simulator, Cooja [7]) under four common routing attacks (the BH, SF, NA, and WH). This investigation is conducted using two different, but commonly used, Radio Duty-Cycle (RDC) protocols: the ContikiMAC [110] and NullRDC protocols - see Sec. 3.2.4.

The chapter's contributions can be summarized in the following:

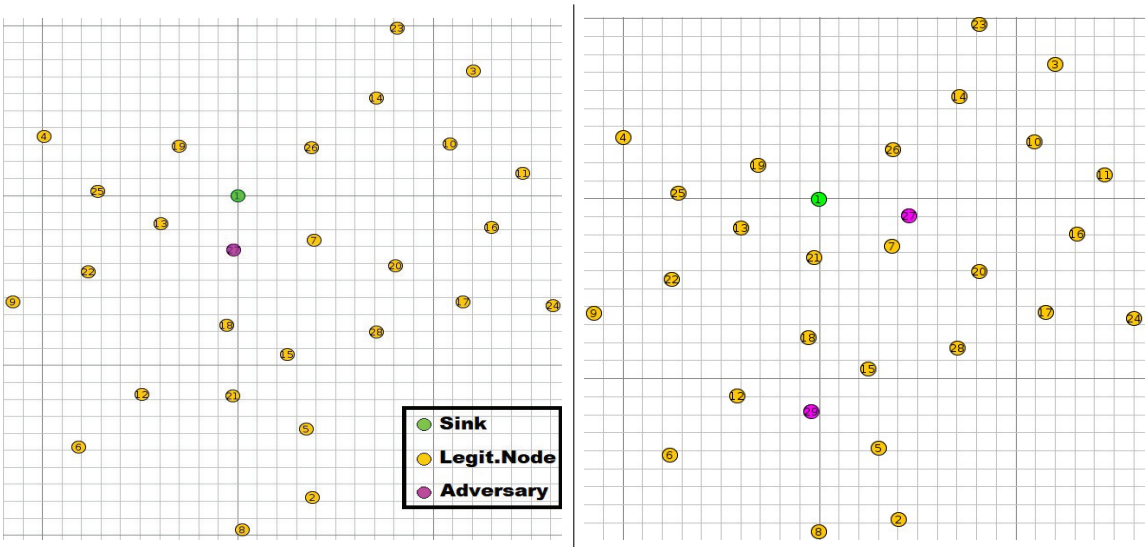
- Through more than a thousand experiments, a performance comparison for RPL was provided between the UM and PSM; the latter is examined with and without the optional replay protection. It was shown that running RPL in PSM (without replay protection) does not use more resources than UM, even under an attack.

- The investigation verified that RPL in PSM can stop external adversaries from joining the IoT network for the investigated attacks, except for the WH attack. Furthermore, it showed that the optional replay protection also provides excellent mitigation against the NA. However, it requires further optimization to reduce its effect on energy consumption.

- The investigated attacks on the routing topology were observed and analyzed, then two simple techniques were proposed that could help reduce the effects of the investigated attacks, without using external security measures such as IDSs or added security mechanisms.

- Another performance evaluation for the implementation of the proposed techniques was conducted. The results showed improved performance of RPL under the BH and SF attacks, in terms of PDR and E2E latency.

The rest of this chapter goes as follows: Section 3.2 discusses the evaluation methodology, setup, assumptions, adversary model, and attack scenarios. Evaluation results are analyzed in section 3.3. Section 3.4 discuses the observations from the results and proposes two techniques to be used when designing RPL-based IoT networks. In addition, an implementation of the proposed techniques is evaluated and the results are discussed. Finally, the chapter is concluded in 3.5.



(a) BH, SF, and NA scenarios. (b) WH attack scenario.

Figure 3.1: Network topologies used for the different simulation scenarios (better viewed in colors.)

3.2 Evaluation of RPL's Security Mechanisms under Attacks

In order to inspect the strengths and weaknesses of RPL's secure modes, RPL performance and security are evaluated against four attacks: the BH, SF, NA, and the WH attacks [10, 14] (see Chap. 2). Experiments were conducted with RPL in both UM (vanilla ContikiRPL) and PSM (as in Perazzo *et al.* [46] implementation). For the latter, RPL was evaluated with and without the optional replay protection mechanism.

3.2.1 Evaluation Setup

Cooja [7], the simulator for Contiki OS, was used for all the simulations (with simulated motes). Fig. 3.1(a) shows the topology used in the evaluation for the BH, SF, and NA attacks, while Fig. 3.1(b) shows the one used to evaluate the WH attack (as two adversaries are needed). A list of simulation parameters is provided in Table 3.1.

Both topologies represent a single DODAG network with one root or sink node (the green node). The minimum number of adversaries required for each attack was

Table 3.1: List of Simulation Parameters

Description	Value
No. of sim. sets	Two: one for each RDC protocol (See Sec. 3.2.4)
No. of experiments per set	Four: one for each attack (See Sec. 3.2.3)
No. of scenarios per experiment	4 (ContikiMAC) / 5 (NullRDC)
No. of sim. rounds per scenario / time	10 rounds / 20 min. per round
Node Positioning	Random (three clusters)
Deployment area	290m W x 310m L
Number of nodes (adversary included)	28 (ContikiMAC) / 29 (NullRDC)
Sensor nodes type	Arago Sys. Wismote mote
Propagation model	Unit Disk Graph Model (UDGM)
DATA transmission rate	$\simeq$ 1 packet/minute per legitimate node

used to reduce the complexity of the observed metrics. For the BH, SF, and NA attacks, node (27) was used as an adversary and positioned near the sink node, which would introduce the most prominent effect of the three attacks [14, 111, 112]. For the WH attack, two adversaries (nodes 27 and 29) were used and positioned to create a wormhole between the node cluster (1, 7, 20, and 26) and the targeted nodes. The targeted nodes for all the attacks are (2, 5, 6, 8, 12, 15, 18, 21, and 28), with node (28) providing an alternative path for the targeted nodes to send their packets toward the sink. Having an alternative path is crucial to the evaluation to examine how the self-healing mechanisms (see Sec. 2.3.4) of RPL will respond to the attacks.

It is worth mentioning that an implementation of the simulations using Zolertia Z1 motes [107] (8KB RAM and 92KB Flash memory) was conducted to compare the results to that of [113]. However, enabling RPL's replay protection mechanism caused the Z1 motes to always run out of RAM due to increased size of the OS binaries, rendering the simulation impossible. Hence, the more powerful Wismote motes (16KB RAM and a 256KB Flash memory [114]) were used in the simulations.

3.2.2 Assumptions

The following assumptions were considered in the evaluation: all the legitimate nodes send data packets toward the root at a rate of 1 packet/minute per node, while the adversary does not send any data packets. For all the evaluated secure modes, RPL is set up with the default OF, namely the MRHOF [35] – see Sec. 2.3. Contiki OS is using the following settings for its uIP stack: IEEE 802.15.4 [20] for the Physical Layer and MAC sub-layer, ContikiMAC [110] and NullRDC [8] for the RDC sub-layer (see Sec. 3.2.4), IPv6, 6LoWPAN, and RPL at the Network Layer, and User Datagram Protocol (UDP) for the Transport Layer. To keep the focus on RPL at the Network Layer, it was assumed that neither security measures nor encryption were enabled at the Link layer. All the attacks were implemented at the Network layer.

The data traffic model used is a deterministic one that mimics a typical sensing-IoT network, where nodes send their sensor readings toward the root node at pre-determined periods. Here, only the legitimate nodes send data packets toward the root, each sending one packet per minute, while the adversaries only participate in the DODAG formation without sending any data packets.

The results obtained from the simulations were averaged over ten rounds for each scenario with a 95% confidence level.

3.2.3 Adversary Model and Attack Scenarios

For each RDC protocol (see Sec. 3.2.4), a *set* of four experiments was conducted: the first three experiments (RPL in UM, RPL in PSM, and RPL in PSMrp) have an *internal* adversary[1], who participates in the creation of the topology from the beginning (and has the preinstalled encryption key in the 2^{nd} and 3^{rd} experiments). The fourth experiment (RPL in PSM) uses an *external* adversary[2] who runs RPL in UM and does not have the knowledge of the secure versions of RPL's control messages, while the legitimate nodes run RPL in PSM. Table 3.2 lists the settings for these experiments.

For the attacks themselves, there are five scenarios:

1. *No Attack*: the adversary works as a fully legitimate node.

[1] An internal adversary is an adversary who is part of the network, e.g., it has the encryption keys used by the legitimate nodes for RPL in PSM or ASM.

[2] An external adversary is an adversary who is not part of the network, e.g., it does not have the encryption key used by the legitimate nodes for RPL in PSM, or runs RPL in UM.

2. *Blackhole (BH) Attack*: The adversary drops all types of traffic coming through, including RPL control messages and data packets [10] – see Sec. 2.6.1. In the conducted evaluation, the BH adversary will keep its radio operational according to the RDC protocol, but it will simply discard any incoming or outgoing frames. The main objectives of this attack are the disruption of data packets transfers toward the root node and disconnecting the routing topology.

3. *Selective-Forward (SF) Attack*: The adversary drops any non-RPL packets, including data. However, only RPL control messages will be processed as normal and passed [111] – see Sec. 2.6.1. Similarly to BH, the simulation of the adversary will keep its radio operational, but it will check the "*Type*" field of any incoming or outgoing ICMPv6 packet to see if it holds an RPL control message (see Sec. 2.3.1). If an RPL control message is detected, it will be processed as usual and passed. All other types of packets will be discarded. An SF attack aims to create a DoS attack in the network that is hard to detect.

4. *Neighbor attack (NA)*: The adversary will pass any DIO message it receives from its neighbors without any processing or modification [84] – see Sec. 2.7.5. This will create the illusion of having the original sender in the range of the victim nodes. The implemented simulation of this adversary is a simple one: while operating as a legitimate node, whenever the adversary receives a DIO message (even if it was not addressed to it), it will multicast an exact copy of the received message to its sub-DODAG before processing the message as usual. The adversary in this attack aims to increase data packet latency and exhausting the nearby nodes resources.

5. *Out-of-Band Wormhole (WH) Attack*: Two adversaries use an out-of-band link to forward RPL control messages from legitimate nodes between the two locations where the adversaries reside [5, 10] – see Sec. 2.6.3. This scenario is available only in the NullRDC set of experiments, see Sec. 3.2.4. This attack tries to be a jack of all trades. It aims to significantly disrupts the routing topology, increase data packets' latency, exhausts the victim nodes' resources, and disconnects parts of the network, all without being detected.

In the BH, SF, and NA scenarios, the adversary always starts as a legitimate node, tries to join the network, and actively participates in the creation and maintenance of the DODAG. Then, it works as a legitimate node for two minutes (to assure full

Table 3.2: Experiments summary

Experiment	Secure Mode	Replay Protection	Adversary Type
UM-I	×	×	Internal (I)
PSM-I	✓	×	Internal (I)
PSMrp-I	✓	✓	Internal (I)
PSM-E	✓	×	External (E)

integration with the network) before launching the attack afterward. For the WH attack, the two adversaries are always in promiscuous mode and never participate in the DODAG.

The choice of these attacks was due to having a minimum cost for the adversary to launch them, as they require little or no processing of RPL's messages. At the same time, the effect of these attacks can be significant on the network [10].

It is worth mentioning that the simulation of the WH attack is based upon the work in [68]. The authors implemented an out-of-band WH on a real testbed, with a wired link between the adversaries. Each adversary operates in the *promiscuous mode*, sniffs all types of frames, sends the sniffed frames through the wired link, and replays the frames it received from the wired link. However, the implementation in this dissertation differs from theirs in a few points:

1. The implementation is simulation-based and is conducted in Cooja [7]. The host computer was used to emulate a fast link between the adversaries.

2. The wormhole is implemented at the Network Layer level in order to detect and replay RPL's control messages only. In addition, the adversaries can identify the secure versions of RPL's control messages.

3. The adversaries use a multi-buffer approach for the packets received from the radio and for the packets awaiting the replay. This approach accelerates the operation of the adversaries when there are many neighbors, and makes sure that all forwarded packets are replayed without dropping any of them.

3.2.4 Implementation Challenges

Contiki OS [8] divides the Link layer into three sub-layers: the *MAC* sub-layer, which is responsible for addressing, sequencing, and retransmissions; the *FRAMER* sub-layer that is responsible for creating and parsing of frames; and the *RDC* sub-layer that controls the radio component. Currently, Contiki OS comes with several RDC protocols, with the most used ones are the ContikiMAC [110] and NullRDC.

ContikiMAC is the default setting for RDC protocol in Contiki OS. Here, the radio is kept off most of the time, with the protocol waking up the radio periodically to check for transmissions. If a transmission is detected, the radio will be kept on long enough to receive the frame, send an ACK to the sender (if it was accepted) [110,115], then the radio is turned off. Similarly, the sender will turn on the radio, probe the channel, perform several attempts to transmit a frame, and wait for either an ACK (which dictates a successful transmission) or reach a threshold that means a failed transmission [110]. Either way, the radio is turned off afterward. ContikiMAC protocol is proved to be very efficient with power consumption, at the expense of having longer E2E latency [115].

On the other hand, the NullRDC protocol keeps the radio always on and does not perform frequent channel probing, which means lower E2E latency and a smaller number of retransmissions at the expense of higher power consumption [115].

During the implementation of the WH attack using the ContikiMAC protocol, it was found that the messages forwarded through the wormhole were replayed very late by the adversaries; hence, those messages got ignored by the legitimate nodes. A further investigation showed that a mix of simulation artifacts and the lengthy sending procedure of ContikiMAC are the culprits for such late replay. Several trials were made to reduce simulation latency (e.g., reducing output text, using faster host, etc.) and accelerate ContikiMAC sending procedure; all have failed due to the simulation artifacts. However, this situation is not expected to occur in a real-world implementation [68].

However, implementing the WH attack using the NullRDC protocol proved to be working perfectly. Since the sending procedure is much simpler than that of ContikiMAC, the WH attack performed as expected, without any added latency and resulting in full disruption to the routing topology (as explained later). Since the power consumption, in this case, is dominated by the high usage of the always-on radio (almost fixed at 122 milliwatts), it was not possible to evaluate the effect of the

investigated attacks on power consumption using the NullRDC protocol.

For that reason, only the NullRDC experiment set evaluates the WH attack, omitting the power consumption metric.

3.3 Results and Analysis

The results for ContikiMAC and NullRDC sets of experiments are shown in Fig. 3.2 and Fig. 3.5, respectively. Tables 3.3 and 3.4 show the numerical values of all results obtained in this paper.

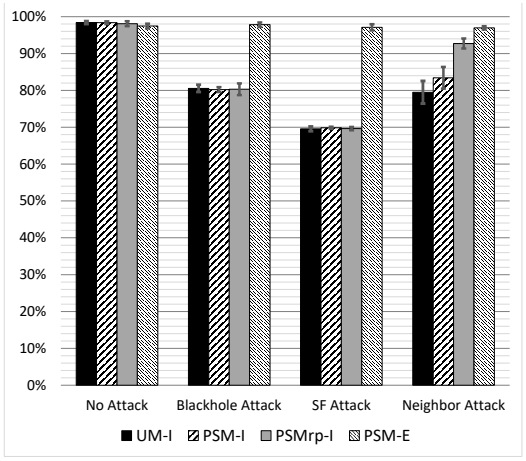
These results are expressed as the average PDR, average E2E latency, the number of exchanged RPL control messages (per legitimate node), and average network power consumption (per received packet). Fig. 3.3 and Fig. 3.4 show the routing DODAG for each scenario that was formed in 90% of the time in all experiments.

3.3.1 ContikiMAC Set Results

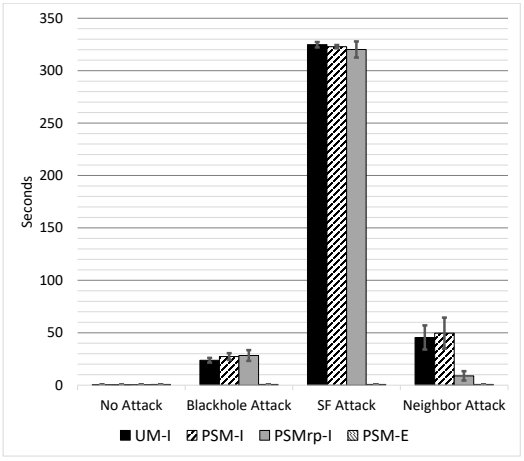
Effects on data packet delivery rate (PDR): Looking at Fig. 3.2(a), it is clear that the RPL in PSM successfully mitigated the BH, SF, and NA when the adversary is external with the PDR hovering around 98%.

On the other hand, when the adversary is internal, the SF attack has the most effect (in all experiments) on the PDR, decreasing it to a low of 70%. The main reason behind the success is that the adversary, due to being an active participant in the DODAG maintenance, is always chosen as the preferred parent for its sub-DODAG. However, none of their data packets are passed to the sink node. Fig. 3.3(a) shows the routing DODAG during the SF attack.

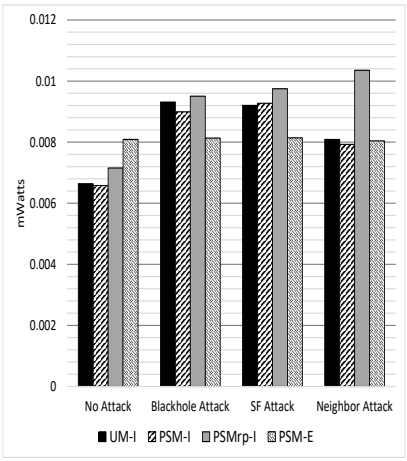
For the BH attack, the self-healing mechanisms of RPL were always able to detect the unresponsive adversary after approximately ten minutes from the attack launch time (which is the default setting for "dead parent" timeouts in the Contiki OS) and initiated a local repair for the affected sub-DODAG to switch to an alternative path. Hence, not all data packets got dropped, which explains why PDR is in the range of 80%. Fig. 3.3(b) shows the routing DODAG after ten minutes from the BH attack launch time and the isolated adversary.



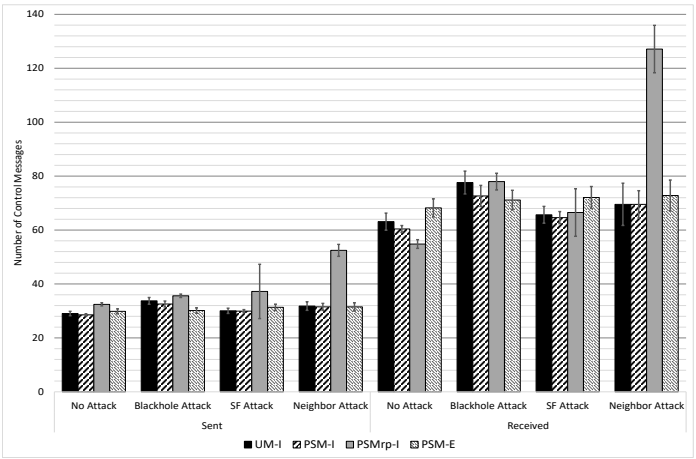
(a) Average data packet delivery rate (PDR).



(b) Average network E2E latency for data packets.

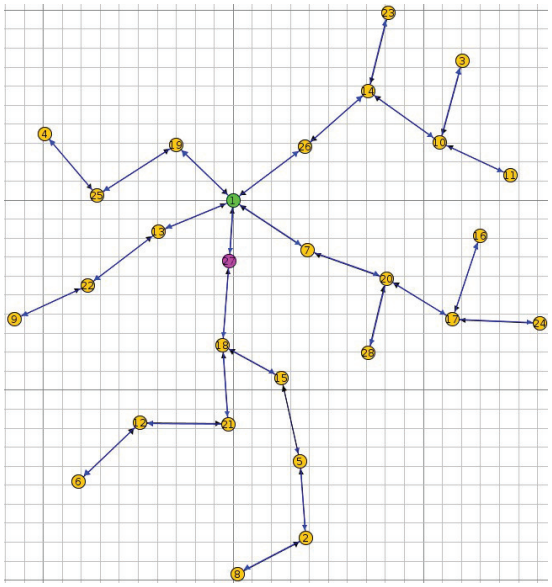


(c) Average network power consumption, per received data packet.

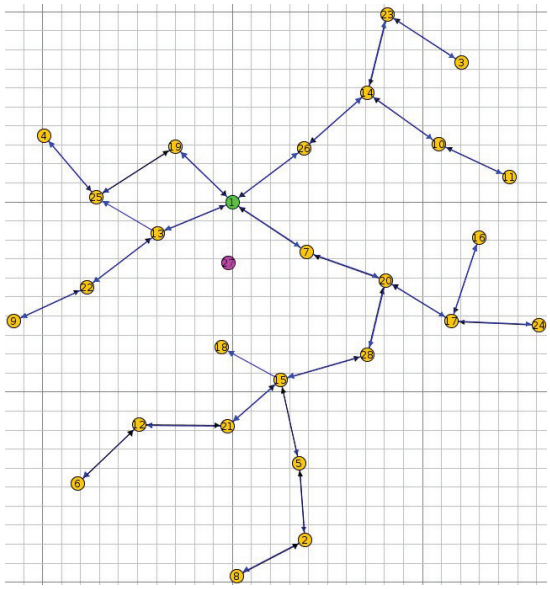


(d) Exchanged RPL control messages, per legitimate node.

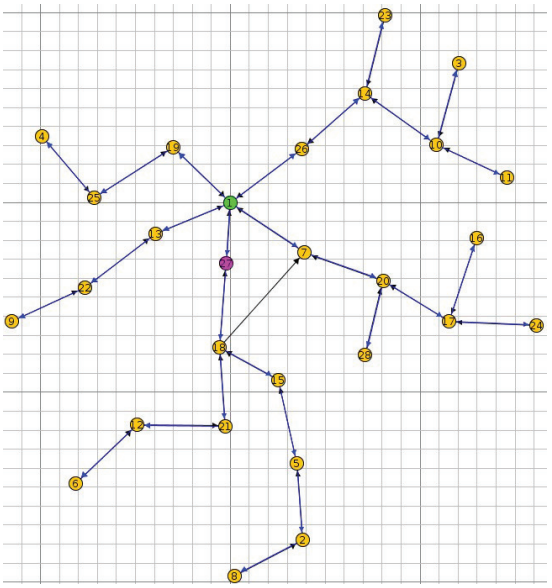
Figure 3.2: Simulation results for the four experiments (three attacks scenarios), using ContikiMAC RDC protocol. (UM: unsecured mode, PSM: preinstalled secure mode, PSMrp: preinstalled secure mode with replay protection, I: internal adversary, E: external adversary.)



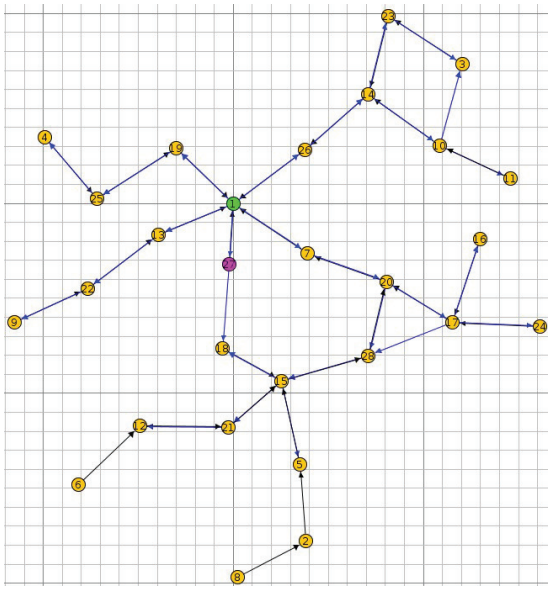
(a) No Attack scenario and SF Attack scenario (all experiments except PSM-E)



(b) BH Attack scenario (all experiments), and all scenarios for PSM-E.



(c) NA scenario (UM-I and PSM-I).



(d) NA scenario (PSMrp-I).

Figure 3.3: Routing DODAGs during the investigated scenarios.

Table 3.3: Simulation results for the four experiments (three attacks scenarios), using ContikiMAC RDC protocol.

Scenario	Average PDR				Average E2E Latency (in seconds)				Average Power Consumption (per received packet - in mWatt)				Exchanged RPL Control Messages (per legitimate node)							
													Sent				Received			
	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E
Default Setting - (Fig. 3.2)																				
No Attack	98.40%	98.36%	98.10%	97.45%	0.480	0.374	0.575	0.656	0.00664	0.00658	0.00715	0.00809	29	29	32	30	63	60	55	68
BH Attack	80.55%	80.21%	80.31%	97.82%	23.792	27.307	28.277	0.672	0.00931	0.00900	0.00951	0.00813	34	33	36	30	78	73	78	71
SF Attack	69.56%	69.85%	69.63%	97.11%	324.754	322.885	320.194	0.725	0.00920	0.00927	0.00975	0.00814	30	30	37	31	66	65	67	72
NA Attack	79.49%	83.43%	92.72%	96.97%	45.506	49.610	8.885	0.680	0.00809	0.00793	0.01035	0.00804	32	32	52	32	70	70	127	73
First Suggestion - (Fig. 3.7)																				
No Attack	98.04%	98.08%	98.19%	97.55%	0.806	0.529	0.533	1.038	0.00650	0.00670	0.00696	0.00732	31	31	34	32	74	76	81	86
BH Attack	85.32%	84.94%	85.50%	97.93%	17.783	17.998	17.695	0.562	0.00781	0.00804	0.00824	0.00733	33	34	37	32	83	85	90	85
SF Attack	77.82%	75.79%	76.92%	98.04%	213.223	231.112	214.871	0.485	0.00779	0.00825	0.00885	0.00724	31	31	36	32	75	76	85	85
NA Attack	86.00%	86.50%	94.74%	97.70%	44.424	33.717	4.486	0.631	0.00716	0.00708	0.00930	0.00714	32	32	51	32	77	76	132	81
Second Suggestion - (Fig. 3.8)																				
No Attack	98.32%	97.71%	98.02%	97.02%	0.425	1.076	0.889	1.990	0.00696	0.00750	0.00741	0.00895	29	33	36	34	53	59	60	61
BH Attack	88.78%	87.81%	87.74%	97.08%	8.213	10.367	10.591	1.896	0.00892	0.00922	0.00968	0.00878	35	36	41	34	66	67	71	61
SF Attack	69.41%	69.33%	69.59%	97.02%	312.429	312.945	307.448	1.843	0.00993	0.00974	0.01037	0.00864	33	32	37	33	55	53	57	61
NA Attack	79.93%	83.05%	93.84%	96.74%	52.318	48.455	4.596	1.960	0.00888	0.00847	0.01124	0.00895	35	34	58	34	61	59	104	62

Finally, for the NA, the adversary was able to reduce the PDR for the UM-I and PSM-I experiments, as node 18 always chose either node 7 or 13 as its preferred parent (Fig. 3.3(c) shows that node 18 selected node 7 as its preferred parent), due to receiving their DIO messages through the adversary. Since nodes 7 and 13 are actually out of node 18's range, all packets sent toward them from node 18 and its sub-DODAG are lost. Hence, the PDR is in the same range as in the BH attack scenario. However, activating the replay protection mechanism results in much better PDR as the mechanism verifies each DIO message's original sender before processing its contents. Fig. 3.3(d) demonstrate how the network (in PSMrp-I experiment) opted for the alternative path after a few minutes from launching the NA.

Effects on the data packets' E2E latency: Confirming the findings mentioned above, Fig. 3.2(b) shows that the RPL in PSM mitigated the BH, SF, and NA when they were launched by an external adversary, keeping the E2E latency at a minimum.

Due to the large number of undelivered data packets for the affected nodes, the SF attack had the longest E2E latency among all the internal attacks. This effect is, again, due to the adversary's active participation in the DODAG maintenance.

For the same reason, the BH attack introduced some latency to the network. However, since the affected nodes were able to find an alternative path and were successful in delivering the rest of their data packets, the latency was much lower than in the SF attack scenario.

The situation is more complicated for the NA scenario, as self-healing mechanisms were triggered several times to recover the affected nodes from the attack, which led to even higher E2E latency than the BH attack scenario. In general, whenever node 18 switches its preferred parent to node 7 or 13, the sub-DODAG suffers from BH-like conditions resulting in losing several data packets. In addition, node 18 will either switch its preferred parent back to the adversary when it does not receive DIO messages from the "ghost parent" (node 7 or node 13), or initiate a local repair procedure (if DODAG inconsistencies were detected) that results in the whole sub-DODAG choosing the alternative path to deliver their packets. Either way, it will add more latency to the network. Using the replay protection will significantly reduce the latency from the NA, as node 18 will not switch its preferred parent as long as it does not receive the correct CC response from nodes 7 and 13.

Effects on the exchanged number of RPL's control messages: As seen in Fig. 3.2(d), the number of control messages exchanged in the network is almost the

same for all experiments and all the scenarios, with the replay protection mechanisms adding a bit more control messages. The exception of this conclusion is the NA scenario with RPL in PSMrp. In this particular case, the replay protection mechanism introduced a much higher number of control messages, due to the exchange of the CC messages whenever a "ghost" DIO message is received by nodes 7, 13, or 18.

It is worth noting that the number of received control messages is always higher than the sent one because many of the sent control messages are multicast messages which will be received by all neighboring nodes of the sender.

Effects on power consumption: Fig. 3.2(c) shows the average network power consumption per received packet, as it gives a more accurate look into the effect of the attacks on the power consumption than just using the regular average power consumption readings.

Looking at the results of the external adversary experiment in the No Attack scenario, it can be seen that the power consumption is a bit higher than the same scenario in the other experiments. The reason is that the data packets from the affected nodes are taking the alternative and longer path, i.e., more power is used by the nodes on that path. However, the power consumption pattern is identical in all the scenarios of the external adversary experiment, which indicates no effect from the attacks; hence, successful mitigation of the attacks.

For all internal-adversary experiments, the power consumption patterns (per scenario) are very similar between RPL in UM and PSM for the No Attack, BH, and SF attacks scenarios, with the replay protection mechanism having a bit more power consumption than the rest. This is because many data packets were not delivered, and the power consumed for their unsuccessful deliveries is entirely wasted.

Now, it is clear from Fig. 3.2(c) that using the replay protection significantly increases the average power consumption when the NA is launched, even if almost all of the sent data packets were delivered successfully. This time, the reason behind this behavior is the increased number of control messages exchanged to mitigate the attack, as seen in Fig. 3.2(d).

3.3.2 NullRDC Set Results

Comparing Fig. 3.5 to Fig. 3.2, it is clear that the first four scenarios (No Attack, BH, SF, and NA) have similar results in both RDC protocols, i.e., NullRDC and

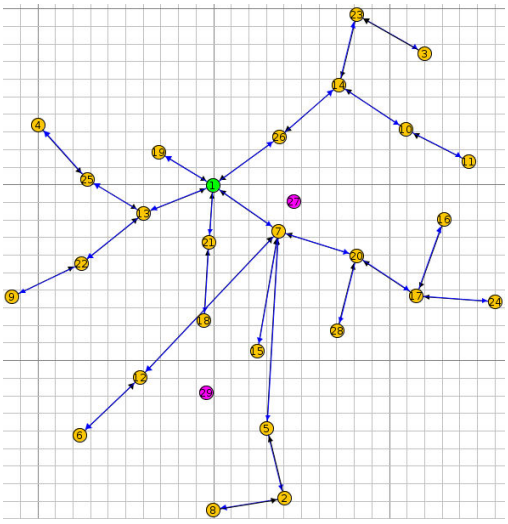


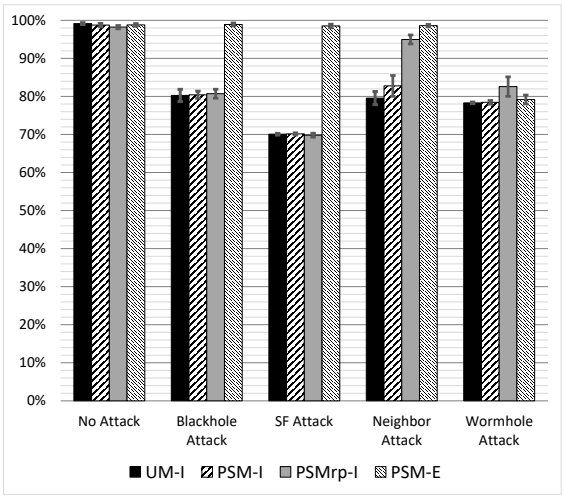
Figure 3.4: Routing DODAG during WH Attack scenario (all experiments)

ContikiMAC. Hence, the focus of this analysis will be on the WH attack scenario. The effects of the WH attack on the routing DODAG can be seen in Fig. 3.4.

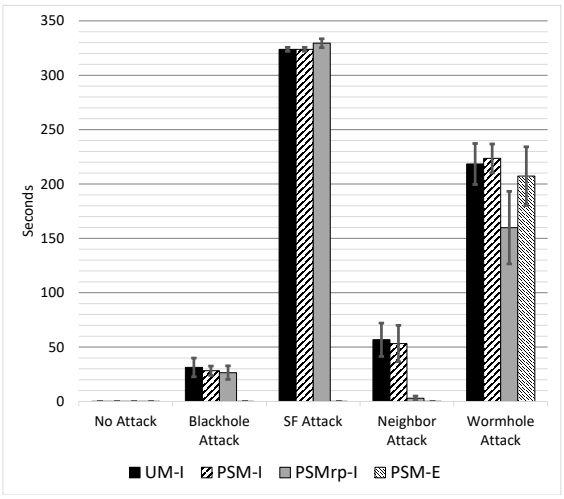
Effects on the data packet delivery rate (PDR): The WH attack successfully lowered the PDR to the low 80th percentile in all scenarios, regardless of the used RPL's secure mode or the adversary type. The observation shows that the reason behind such behavior is because the adversaries are transparent to the network and that all control messages (from both sides of the wormhole) are forwarded and received within their time-windows, deceiving the legitimate nodes to think they are in close proximity.

Effects on the data packets' E2E latency: Since most of the affected nodes were unable to deliver their data packets successfully, the average E2E latency of the network rose to 200 seconds - see Fig. 3.5(b). RPL's replay protection mechanism slightly reduced the effect of the WH attack. However, this is due to having slight delays with the CC message exchanges.

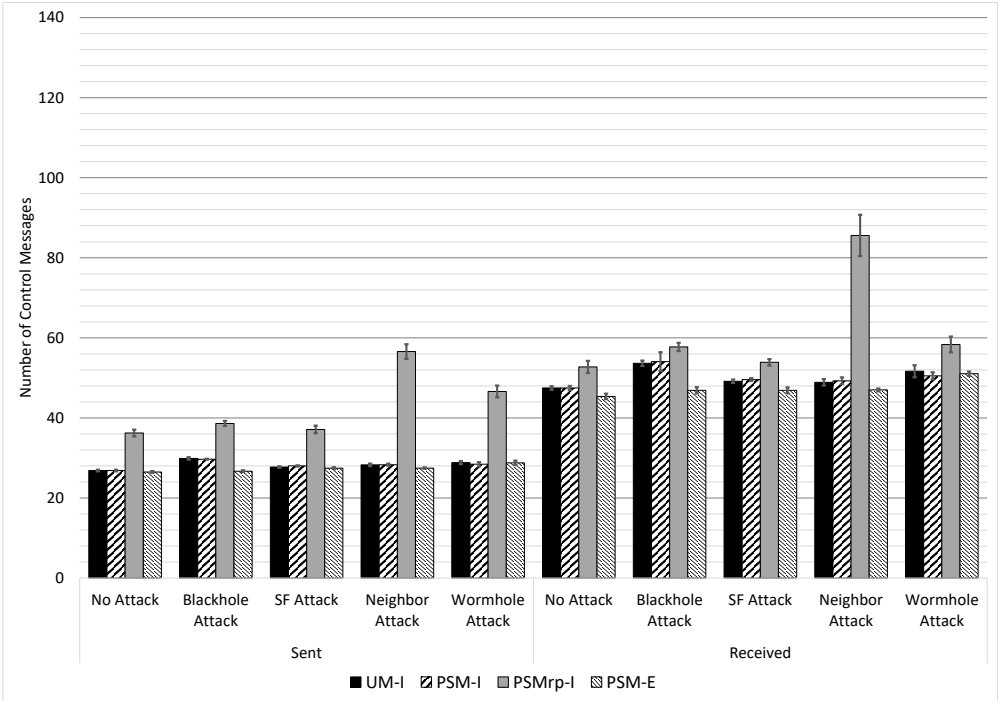
Effects on the exchanged number of RPL's control messages: At a first look, it is evident that using RPL over NullRDC protocol reduces the number of exchanged control packets compared to using the ContikiMAC protocol, which has been documented in [115]. Besides that, the WH attack exchanged a similar number of control messages as in the other attacks, with the replay protection mechanism in PSMrp slightly increasing that number over the other experiments.



(a) Average data packet delivery rate (PDR).



(b) Average network E2E latency for data packets.



(c) Exchanged RPL control messages, per legitimate node.

Figure 3.5: Simulation results for the four experiments (four attacks scenarios), using NullRDC RDC protocol.

Table 3.4: Simulation results for the four experiments (four attacks scenarios), using NullRDC RDC protocol.

Scenario	Average PDR				Average E2E Latency (in seconds)				Exchanged RPL Control Messages (per legitimate node)							
									Sent				Received			
	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E	UM-I	PSM-I	PSMrp-I	PSM-E
Default Setting - (Fig. 3.5)																
No Attack	99.15%	98.74%	98.22%	98.80%	0.033	0.034	0.047	0.036	27	27	36	26	47	47	53	45
BH Attack	80.28%	80.44%	80.72%	98.93%	31.302	28.266	26.500	0.036	30	30	39	27	54	54	58	47
SF Attack	70.07%	70.16%	69.83%	98.52%	323.857	323.858	329.444	0.036	28	28	37	27	49	50	54	47
NA Attack	79.57%	82.77%	94.98%	98.60%	56.762	53.242	2.923	0.036	28	28	57	27	49	49	86	47
WH Attack	78.32%	78.46%	82.59%	79.19%	218.413	223.518	159.881	207.210	29	28	47	29	52	51	58	51
First Suggestion - (Fig. 3.9)																
No Attack	98.84%	98.69%	98.02%	98.89%	0.052	0.038	0.052	0.040	27	27	38	27	51	51	60	50
BH Attack	82.61%	84.64%	83.86%	98.78%	23.728	19.043	16.321	0.036	29	29	40	27	55	55	63	52
SF Attack	73.73%	72.93%	73.44%	98.48%	284.978	291.616	277.439	0.036	27	27	38	27	48	49	57	52
NA Attack	85.24%	85.49%	97.24%	98.86%	41.850	47.155	0.566	0.036	27	28	52	27	51	51	80	53
WH Attack	80.08%	80.74%	85.07%	80.99%	199.385	200.145	135.046	190.021	29	28	51	29	60	58	69	58
Second Suggestion - (Fig. 3.10)																
No Attack	98.66%	98.72%	98.97%	98.50%	0.362	0.164	0.036	0.691	30	30	35	29	53	53	64	52
BH Attack	88.90%	88.68%	86.09%	98.46%	8.976	9.161	10.962	0.479	32	33	38	30	59	61	70	55
SF Attack	70.18%	70.11%	66.94%	98.58%	309.822	311.586	323.208	0.426	30	30	34	30	49	49	58	55
NA Attack	81.19%	85.66%	97.45%	98.42%	45.938	46.157	0.823	0.503	30	31	42	30	51	54	75	55
WH Attack	82.91%	82.62%	83.27%	83.25%	134.065	134.374	135.079	135.804	32	33	49	32	58	59	63	58

3.4 Discussion

Based on the analysis of the obtained results, the following observations were noted and, as a result, two designing techniques are suggested to improve RPL's response to the investigated attacks.

3.4.1 Observations

- Using RPL in PSM (and by extension, the ASM) can mitigate the external adversaries of the Blackhole, Selective-Forward, and Neighbor attacks, as long as the adversary does not run RPL in any secure mode.
- RPL's performance using PSM (without the replay protection mechanism) is similar to that when using UM, but with the added benefit of mitigating the external adversaries of the BH, SF, and NA as investigated in this chapter.
- RPL's secure modes cannot mitigate out-of-band WH attacks (with the Null-RDC protocol at the Link layer) as their adversaries can operate external to the network.
- It is worth mentioning that another experiment (using ContikiMAC) was conducted that had the external adversary running RPL in PSM while not knowing the encryption key used by the legitimate nodes. The results from that experiment were identical to the PSM-E experiment except for the NA scenario, which was successfully launched. Since each type of RPL control messages has its unique ICMPv6 "Code" value, with the secure versions having different values than the unsecure ones, only a node that runs RPL in PSM/ASM could identify the secure versions of RPL control messages. Hence, the adversary was able to identify RPL's secure DIO messages and replay them.
- Enabling RPL's replay protection mechanism will significantly reduce the effect of NA on PDR and E2E latency. However, in its current implementation, it will increase the power consumption as well, which can lead to energy depletion of the devices. In theory, an adversary can replay DIO messages regularly to keep the affected nodes always busy with the consistency checks, leading to depletion of their energy and shutdown.
- RPL's secure modes require more memory and storage spaces than the unsecured mode, which means not all IoT devices can use them – see Sec. 3.2.1.

3.4.2 Suggestions to Reduce the Effects of the Investigated Routing Attacks on RPL's Performance

Based on the observations mentioned above, the following suggestions are proposed to help reduce the effects of routing attacks on RPL's performance, without introducing any additional security mechanisms or systems.

1. Designing the network topology in a way where there are more alternative paths toward the root node and more neighbors per node. This would decrease the recovery time required for nodes to overcome a BH attack and reduce the effects from the other investigated attacks on PDR and E2E latency.

2. Reducing the timeout duration after which an RPL router should declare a preferred parent as "dead". Currently, ContikiRPL uses fixed timeout values for the upward (UIP_CONF_ND6_REACHABLE_TIME) and downward routes (RPL_CONF_DEFAULT_LIFETIME), both set to 10 minutes. Reducing these values could decrease the E2E latency and increase the PDR of the network under some attack situations. However, static decrements may also increase power consumption when there are no attacks. It is recommended to use a dynamic approach for adapting these timeout values to the network's changing conditions. For example, randomizing the timeout values after each expiration, or using the IPv6 over Low-powered Wireless Personal Area Network-Neighbor Discovery (6LoWPAN-ND) protocol [116–118], which aids RPL to detect node's neighbors and checks their status in a resource-friendly way.

3.4.3 Evaluation of the Proposed Suggestions

For the first suggestion, having more routes toward the root node means adding more routing nodes. Hence, three routing nodes (29, 30, and 31) were added to the topology - see Fig. 3.6. In the WH attack scenario, three routing nodes (30, 31, and 32) were also added, which are located at the same positions as in Fig. 3.6 but with the topology in Fig. 3.1(b).

To evaluate the effect of the second suggestion on RPL's performance under the investigated attacks, both "dead parent" timeouts (see Sec. 3.4.2) were set to five

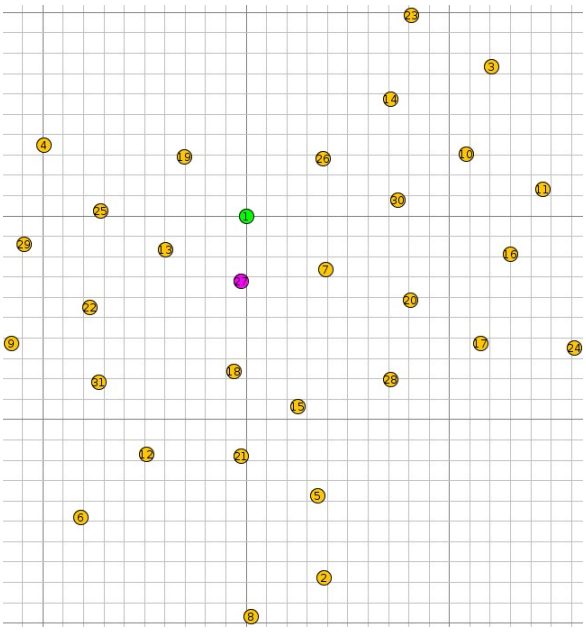


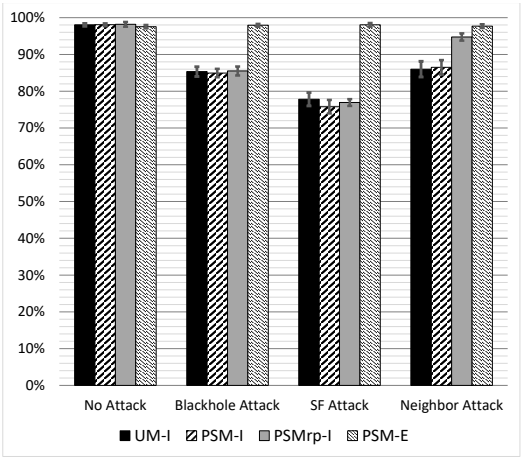
Figure 3.6: Network topology for the first suggestion.

minutes. The use of a fixed value instead of a dynamic approach was used to examine the effect of the reduced "dead parent" timeouts only. The topologies for the evaluation are the same ones used in Fig. 3.1. The whole evaluation was conducted using the same metrics and methodology as in Sec. 3.2.

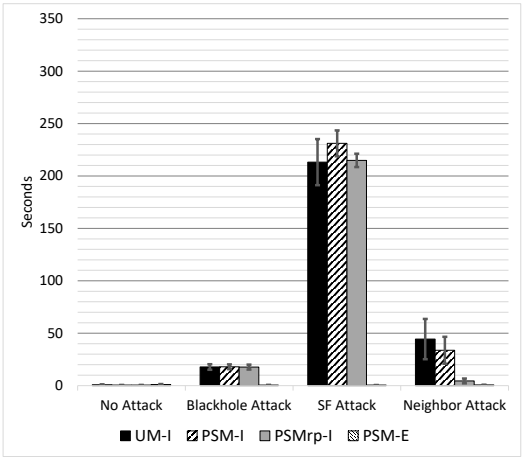
Effects on data packet delivery rate (PDR): Comparing Fig. 3.7(a) to Fig. 3.2(a) (ContikiMAC) and Fig. 3.9(a) to Fig. 3.5(a) (NullRDC), it can be seen that the first suggestion slightly enhanced the network's PDR for the BH, SF, and NA scenarios, adding about 6% more delivered packets. From the observation, the reason behind this improvement is that some of the affected nodes chose the new alternative routes, minimizing the effect of the investigated attacks.

On the other hand, the second suggestion affected only the BH scenario, increasing the PDR to a respected 88% - this is clear from comparing Fig. 3.8(a) to Fig. 3.2(a) (ContikiMAC) and Fig. 3.10(a) to Fig. 3.5(a) (NullRDC). The reduced timeouts caused the effected nodes to detect the adversary parent faster and switch to a different parent. However, this suggestion does not have any effect in the case of the other attacks, since their adversary reacts to received messages, unlike the BH adversary.

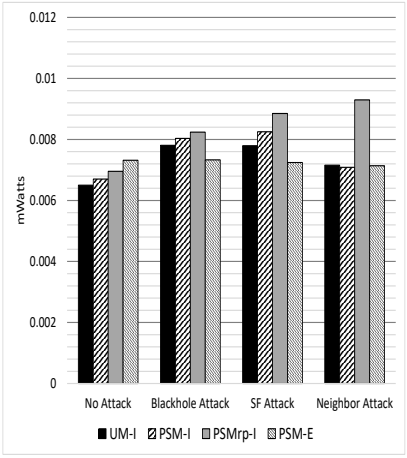
However, neither suggestion had any effect on RPL's performance in the WH attack scenario - compare Fig. 3.9 and 3.10 to Fig. 3.5. This is mainly due to the nature of the WH attack, which forwards all RPL control messages, i.e., the messages and their responses.



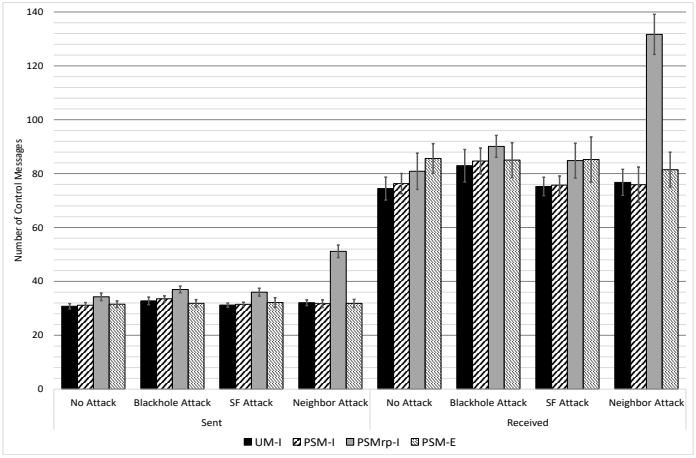
(a) Average data packet delivery rate (PDR).



(b) Average network E2E latency for data packets.

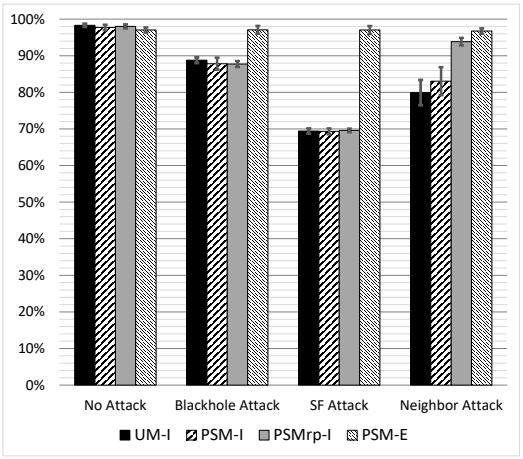


(c) Average network power consumption, per received data packet.

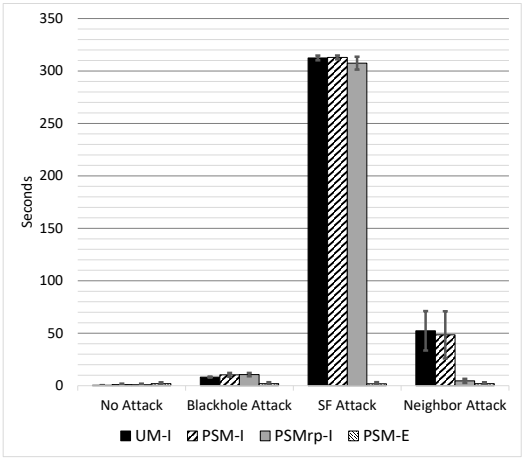


(d) Exchanged RPL control messages, per legitimate node.

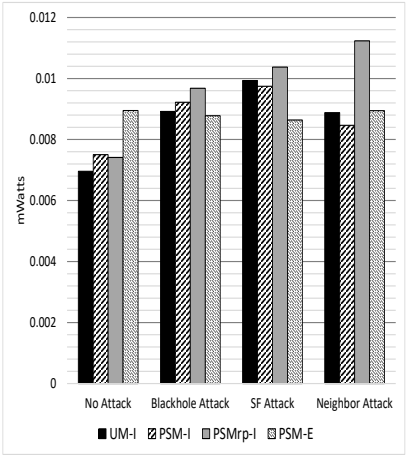
Figure 3.7: Simulation results for the first suggestion (having more routes toward the root node), using ContikiMAC RDC protocol.



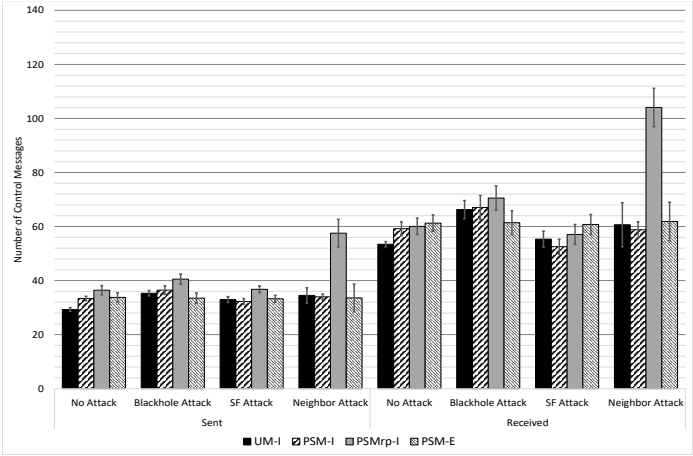
(a) Average data packet delivery rate (PDR).



(b) Average network E2E latency for data packets.

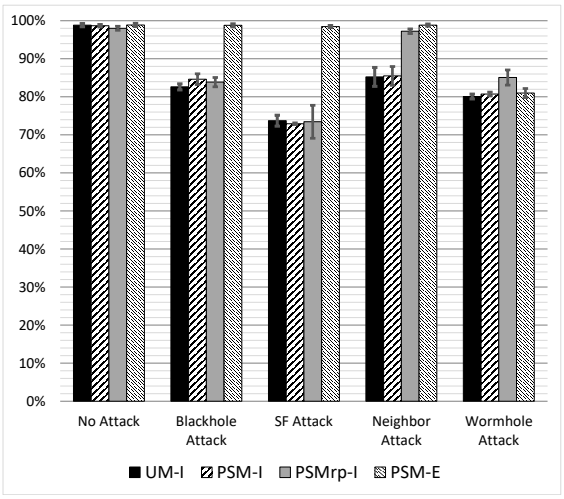


(c) Average network power consumption, per received data packet.

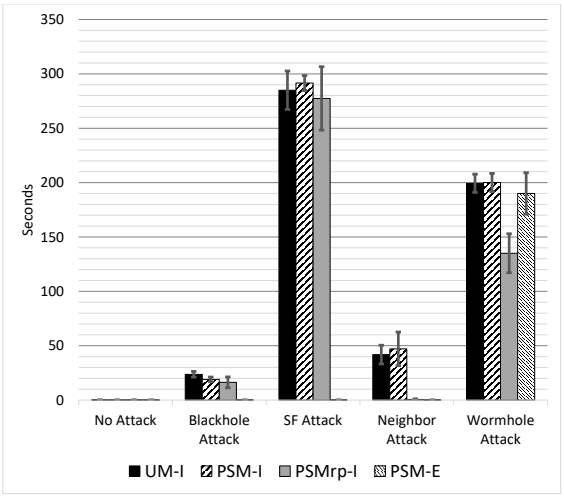


(d) Exchanged RPL control messages, per legitimate node.

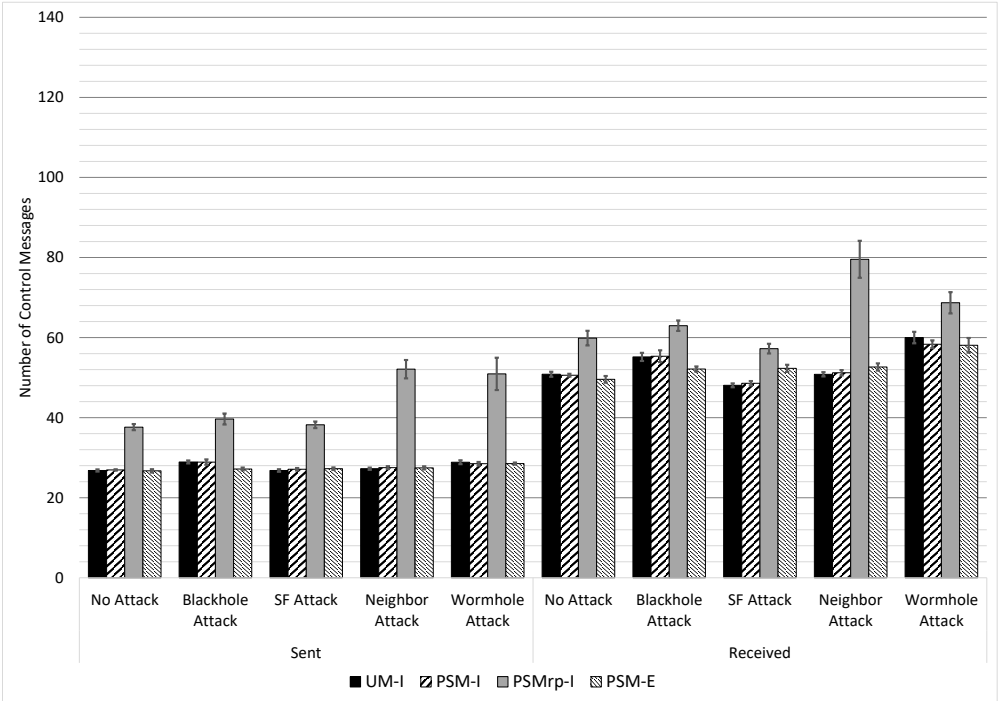
Figure 3.8: Simulation results for the second suggestion (reducing the timeout value for declaring a parent as dead), using ContikiMAC RDC protocol.



(a) Average data packet delivery rate (PDR).

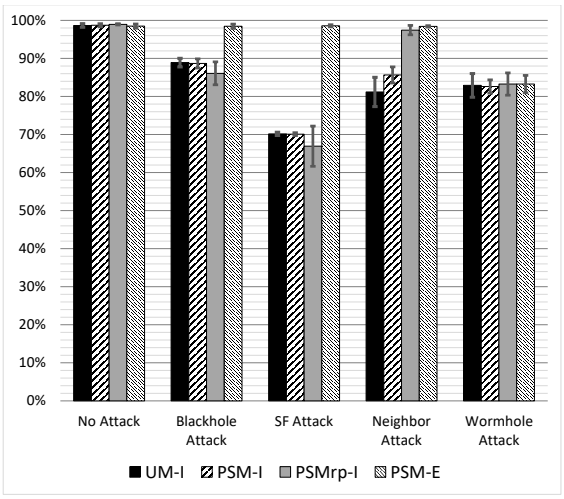


(b) Average network E2E latency for data packets.

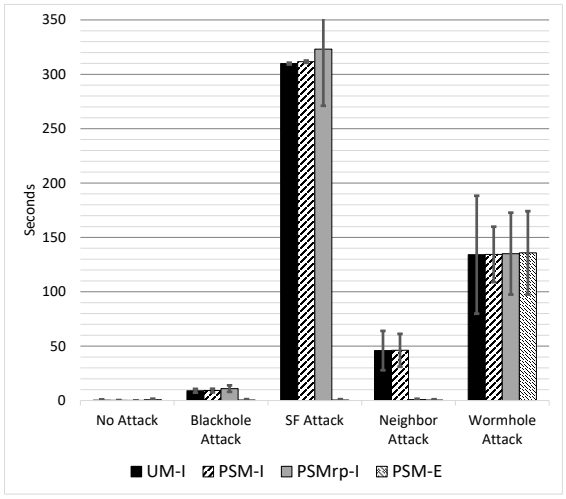


(c) Exchanged RPL control messages, per legitimate node.

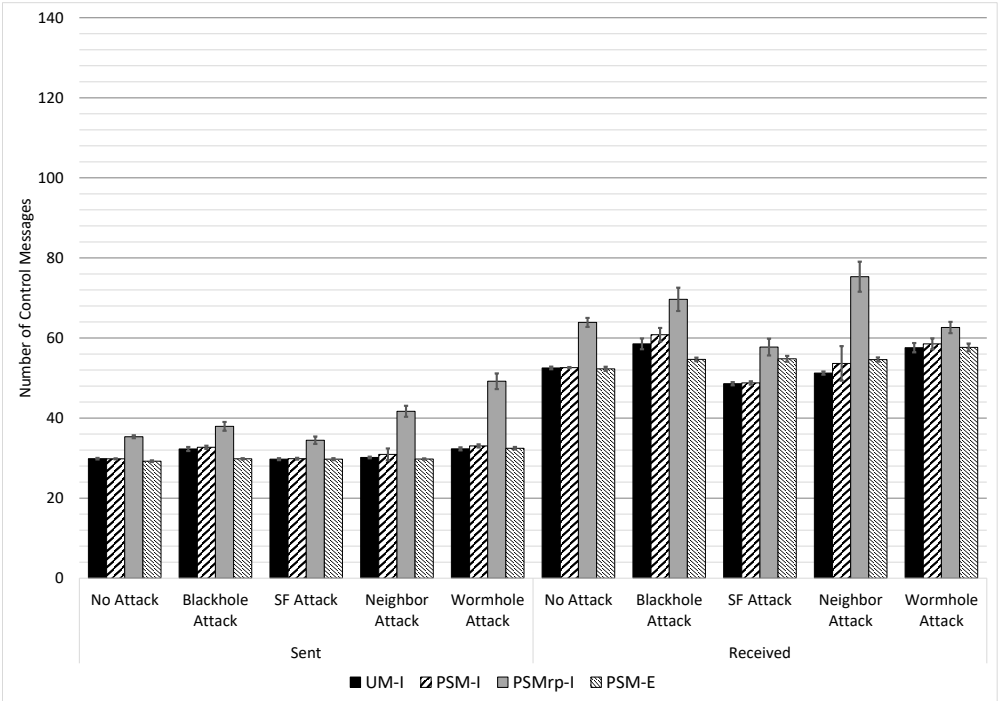
Figure 3.9: Simulation results for all four attacks with the first suggestion implemented, using NullRDC RDC protocol.



(a) Average data packet delivery rate (PDR).



(b) Average network E2E latency for data packets.



(c) Exchanged RPL control messages, per legitimate node.

Figure 3.10: Simulation results for all four attacks with the second suggestion implemented, using NullRDC RDC protocol.

Effects on the data packets' E2E latency: Fig. 3.7(b) and Fig. 3.9(b) show that the first suggestion decreased the E2E in the case of the SF scenario, especially for the ContikiMAC protocol (∼220 seconds, down from ∼320 seconds) compared to NullRDC (∼270 seconds, down from ∼ 330 seconds). Again, this is because some of the affected nodes chose the alternative routes away from the adversary and more data packets are delivered.

As for the second suggestion (Fig. 3.8(b) and Fig. 3.10(b)), the main enhancement occurred is in the case of the BH scenario (10 seconds down from 30 seconds). As the affected nodes were able to detect the dead adversary parent much faster, the total E2E was reduced by more than 50%.

Effects on the exchanged number of RPL's control messages: As seen in Fig. 3.7(d), the first suggestion increased the number of received control messages for the ContikiMAC set. However, the reason this time is the added routing nodes and not the attacks. On the other hand, the second suggestion (Fig. 3.8(d)) slightly reduced the number of received RPL control messages, especially in the NA scenario.

From Fig. 3.9(c) and Fig. 3.10(c), it can be seen that both suggestions do not have any effect on the exchanged control messages when NullRDC is used. This is due to the always-on radio and the simpler sending mechanism.

Effects on power consumption: Fig. 3.7(c) and Fig. 3.8(c) show that the average network power consumption (per received packet) has been reduced for the first suggestion while increased for the second one. The reason behind the reduction for the first suggestion is that more data packets are delivered successfully. However, the power consumption increase in the second suggestion experiments is because more probing is performed for parent's freshness (due to the shorter timeouts). It is worth mentioning that this analysis is only valid for ContikiMAC set and not NullRDC, as it was not possible to collect usable power readings for the latter - see Sec. 3.2.4.

From the discussion above, it can be concluded that, individually, the two suggested techniques have mostly a positive effect on the network when under an attack. Hence, combining both of the suggestions with a dynamic timeout setup would further enhance RPL's performance without taxing the scarce resources of the nodes. This, however, is still being investigated.

3.5 Summary

In this chapter, the performance of RPL and its security mechanisms were evaluated under the presence of four common routing attacks (the BH, SF, NA, and WH attacks). The evaluation was carried using two widely used RDC protocols, ContikiMAC and NullRDC. The analysis of the obtained results showed that using RPL in PSM can mitigate external adversaries of the investigated attacks (except for the WH attack) as long as the adversaries do not run RPL in PSM/ASM. It also showed that using RPL in PSM without the replay protection does not consume more energy than RPL in UM. It has been confirmed that enabling RPL's replay protection mechanism reduces the effect of the Neighbor attack at the expense of consuming more energy.

Two techniques were proposed to be considered when designing RPL-based IoT networks: (i) having more routes toward the root node, and (ii) reducing the "dead parent" timeouts. Evaluating each of these suggestions showed improved performance of RPL under the investigated attacks. A further investigation should be conducted on implementing both of the suggested techniques at the same time while having a dynamic approach for the "dead parent" timeouts optimization.

The work in this chapter showed that the secure modes of RPL are still vulnerable to routing attacks, both internal and external. For example, the authentication-based replay attacks, such as the WH and NA, do not require the processing of RPL's control messages, so their adversaries can operate internally or externally. Hence, they still pose a legitimate threat to the network. In addition, it was found that RPL's secure modes do not provide authentication that the control messages are sent from their original sender and not replayed (except for DIO messages, which can be verified using the CC mechanism). Such vulnerability could be abused by an adversary to launch a "two-way" replay attack (e.g., out-of-band WH attack) that is not detectable by RPL's standard procedures and very hard to mitigate even with external security systems (e.g., IDSs).

The next chapter discusses a similar situation at the Network Layer. Fragmentation attacks at the 6LoWPAN adaptation layer exploit the same vulnerability found in this chapter for RPL: the nodes cannot authenticate that the received fragments are coming from a legitimate sender. An inspecting look into this matter and the proposed solutions, including a one proposed by this dissertation, are provided within the next chapter.

Chapter 4

Common Attacks on Forwarding at 6LoWPAN Layer

4.1 Introduction

The 6LoWPAN Adaptation Layer is expected to be an essential component of many IoT and vehicular networking systems [120, 121]. IoT devices range from embedded sensors in machinery, to RFID tags and readers, to smart devices (smartwatches, smartphones, smart sensors, etc.) [11]. Hence, IoT covers many concepts: RFID systems, WSN, M2M Communications, Low-powered Wireless Personal Area Network (LoWPAN), and even Wireless LANs [22]. In this chapter, as well as this dissertation, the focus will be on LoWPAN implementation, as it is one of the fastest developing technologies nowadays.

It is understood that all IoT should use the TCP/IP protocol stack, and that most of these devices work on low energy (e.g. batteries) with constrained resources (memory and processing powers). This is considered troublesome for such devices when moving/dealing with large packets such as IPv6 packets (the minimum of IPv6 Maximum Transmission Unit (MTU) is 1280 bytes)

For that, IETF proposed an additional layer to be inserted between Network and Link Layers, forming a new standardized protocol stack for IoT. This new layer is called the "*6LoWPAN Adaptation*" Layer, and it is mainly responsible for adapting

This chapter is a revised version of the paper "*The Effect of Buffer Management Strategies on 6LoWPAN's Response to Buffer Reservation Attacks*," published in IEEE International Conference on Communications (ICC) 2017 conference [119], 2017

IPv6 packets to IoT Link Layer frames (IEEE 802.15.4 MAC Protocol Data Units (PDUs)) [122]. The details of the protocol stack are out of this dissertation's scope, but can be viewed in detail in [15].

Two of the main functions of the new layer are the *fragmentation* and *reassembly* of IPv6 packets. Since sending fragments through the network means there is no guarantee they will arrive in order, there is a need for a buffer strategy to store the fragments in the buffer until all of them are received. There are two major points to be covered by such a buffer strategy: the way fragments are stored in the buffer, and the packets dropping criteria for buffer overload situations. According to the current draft of the IoT framework [11, 15], the 6LoWPAN Adaptation Layer does not have a specific implementation for buffer management, and this was left to IoT devices' manufacturers and OS developers [17]. Many of these manufacturers and OSs restrict the buffer to receive only one or two fragmented packets, due to the restricted memory available on the device. For example, the default setting in Contiki OS [8] is to reserve enough buffer for only one fragmented packet. This introduces an opportunity for several types of DoS attacks to be initiated at the 6LoWPAN Layer level, as described in the following paragraph.

6LoWPAN Adaptation Layer, as specified in the current draft [11], relies mainly on the security measures at the Link and Network Layers, i.e. encryption at Link Layer and IPSec protocol at the Network Layer [17]. These security measures could be compromised when subjected to fragmentation attacks, as neither measure works on fragments: the encryption at the Link Layer would work only for exchanged frames, and IPSec would work only after receiving the whole IP packets. The fragments exist in-between at the 6LoWPAN Adaptation Layer only, and this is where fragmentation attacks take place: there are no security measures nor message authenticity for the fragments [123, 124].

Fragmentation Distributed Denial-of-Service (DDoS) attacks at the 6LoWPAN Adaptation Layer pose a non-trivial challenge. In this chapter, the current and proposed [125] buffer management strategies for the 6LoWPAN Adaptation Layer were studied, with the following contributions in mind:

1. The effect of buffer management strategies on the 6LoWPAN Adaptation Layer's response to fragmentation-based buffer reservation attacks was investigated.

2. To provide a better comparison, some modifications to the proposed strategy

[125] were proposed to further mitigate the buffer reservation attacks' effects.

3. All the strategies were evaluated in the presence of a buffer reservation DoS attack, in order to determine how each strategy will perform.

The rest of this chapter is organized as follows: in Sec. 4.2, a look at the buffer management strategies in the 6LoWPAN Adaptation Layer and the common types of fragmentation attacks is presented. In Sec. 4.3, a quick review on *Split-Buffer* defence (as proposed in [125]) is conducted. In Sec. 4.4, a detailed discussion of a few proposed modifications is introduced. Sec. 4.5 handles the evaluation and simulation setup. In Sec. 4.6, the results from the evaluation are discussed and analyzed. Finally, the chapter concludes in Sec. 4.7.

4.2 Background Review of 6LoWPAN Adaptation Layer

4.2.1 Fragmentation at 6LoWPAN Adaptation Layer

In order to adapt IPv6 packets to IEEE 802.15.4 MAC PDUs, the 6LoWPAN Adaptation Layer will first compress the IPv6 and upper layers headers into smaller versions [17]. Then, if needed, it divides the packets into Link-Layer-suitably-sized fragments, each starts with a 6LoWPAN header that helps the receiving node to reassemble the original packet from its fragments [17], followed by the fragment's payload. The first fragment of the packet, contains the compressed higher layers header, is called ***FRAG1***, while the rest of fragments are called ***FRAGN*** [17]. Details on the fragmentation process and overlapping-prevention mechanisms in the 6LoWPAN protocol are out of this dissertation's scope, but are described in [15, 17, 122–125]. For convenience, the process is illustrated in Fig. 4.1, and a short description of fragmentation headers (FRAG1/N) goes as follows:-

- **Fragmentation header flag (5 bits):** This is used to inform the nodes that this is a fragmented packet. It always starts with (11) bits, followed by either (000) for the first fragment or (100) for the subsequent fragments [122].
- **Datagram_Size (11 bits):** the size of the entire uncompressed IPv6 packet before 6LoWPAN fragmentation, but after any IP fragmentation at the Network

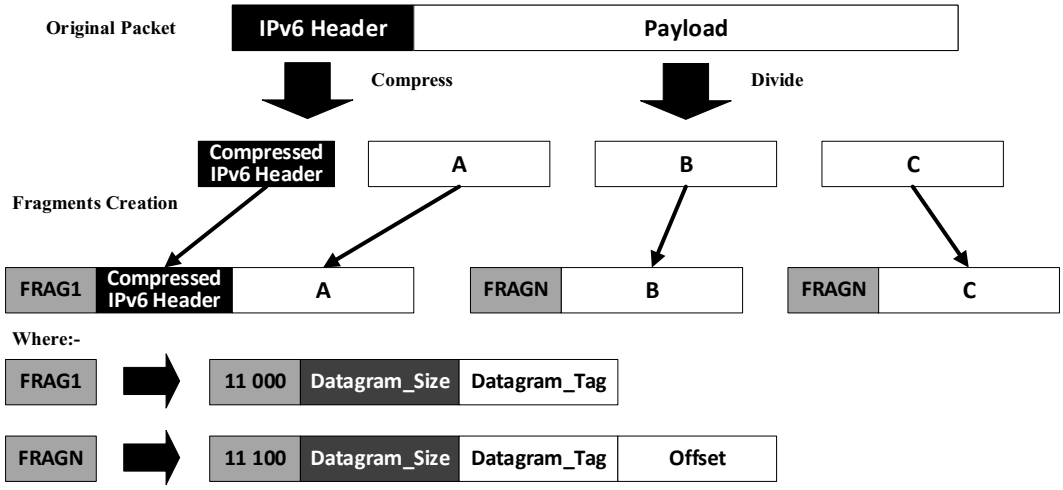


Figure 4.1: Fragmentation at the 6LoWPAN Adaptation Layer [15, 122, 123, 125]. This example shows a payload divided into three parts (A, B, and C), then fragments are created using these parts and the compressed IPv6 header, in addition to the 6LoWPAN fragmentation headers, FRAG1 and FRAGN.

Layer, if available. This enables the receiving node to reserve enough buffer space reassemble the whole packet [123, 125].

- **Datagram_Tag (16 bits):** This field is used to identify all the fragments that belong to the same IPv6 packet; thus, they all should have the same value, but at the same time, it should be unique for both the sender and the IP packet. Since this is a 16 bits field, the tag can range from zero to 65535 [125].

- **Offset (8 bits):** This field only shows up in the second and subsequent fragments, and it specifies the offset (in increments of eight octets) of the beginning of the payload with respect to the beginning of the original IP packet [125].

4.2.2 Forwarding Fragments in 6LoWPAN Networks

According to the 6LoWPAN standard [17, 18, 126], fragments of a packet can be forwarded by intermediate nodes in one of three different ways:

1. **Route-Over**: Here, every node on the routing path must reassemble the fragmented packet and send the reassembled packet to the Network Layer where the routing decision is made [125, 127]. Once the next hop is determined, the

packet is fragmented again and is sent to the next hop. This method is the default option for RPL-based networks.

2. **Mesh-Under**: In this method, a 6LoWPAN mesh header is added to all the fragments, which includes the *originator* and *final-destination* Link Layer addresses. The 6LoWPAN Adaptation Layer will use these information and a mesh routing protocol (e.g., AODV protocol [128, 129]) to decide on the next hop. The fragments are forwarded individually and only reassembled at the final destination [130].

3. **Enhanced-Route-Over**: Introduced in [126], only the first fragment (i.e., FRAG1 - see Sec. 4.2.1) is sent to the Network Layer to make a routing decision on the next hop, then this fragment is forwarded to the next hop [127, 130]. The 6LoWPAN Adaptation Layer will store the datagram tag and the next hop in a forwarding table to be used for forwarding the following fragments without reassembly or sending them to the Network Layer first. If properly implemented, this method may provide lower E2E delays for the data packets and less power consumption for the routing nodes.

4.2.3 Fragments Buffer Management Strategies

Introducing fragmentation to packets in a connectionless system means that there is no guarantee fragments will be received in-order, making it necessary to have a buffer to store the received fragments of the packet until all of them arrive, before reassembling them back to the original packet. A buffer management strategy is used to maintain this buffer and get as many as possible of the fragmented packets completed.

As mentioned earlier in Sec. 4.1, any receiving buffer management strategy has to answer two questions:

1. How to store the fragments in the buffer, and

2. Which packet(s) to drop at buffer-overload situations.

The answers to these questions define how effective the system is: the more completed packets (at the receiver), the better the system is. In many IoT devices, memory (as well as the processing power and energy) is a scarce resource, resulting in

limited buffer space available to the 6LoWPAN Adaptation Layer. This makes it hard for a buffer management system to do its job efficiently. In general, the next sections describe what many IoT manufacturers have implemented for the two questions of buffer management systems.

How to Store the Fragments (Storing Method)

What almost all the current implementations in IoT OSs do is as follows: once a fragment is received, the original size of the packet is extracted from fragment's header, and that size (in bytes) is reserved in the buffer for the fragmented packet. Usually, IoT devices limit this procedure to store either one or two packets only [8, 131].

Hummen *et al.* [125] proposed another way to store fragmented packets, in which: the buffer space is "*slotted*" into several slots, each with a size of one fragment (usually 102 bytes, since each fragment is made to fit into the payload portion of Link Layer frame). Whenever a fragment is received, it will be stored in a slot, regardless if its packet's fragments are in the buffer or not. The buffer management strategy keeps track of the fragmented packets in the buffer, by using the (Datagram_Tag) field in the fragments' headers. This method allows more than two fragmented packets to be stored in the buffer, but all are competing to win most of the buffer in order to be completed.

An important aspect of the fragments' buffers is that every received fragmented packet (regardless of the storing method) has a reassembly timeout, after which all its fragments will be dropped from the buffer if not completed. This is stated in the current 6LoWPAN draft [17] in compliance with the IPv6 standard [16].

For the rest of this chapter, the term *"Slotted Buffer"* will refer to divided buffer space. In contrast, *"Non-Slotted Buffer"* will be used for the traditional implementation (i.e., reserving the whole buffer for one or two fragmented packets).

What Packet(s) to Drop at Buffer-Overload Situation (Dropping Criteria)

Buffer-overload occurs in two different ways, depending on how the system stores the fragments:

1. In a ***non-slotted buffer***, buffer-overload occurs when a new fragment arrives for a new packet that does not have fragments in the buffer and the limit of stored packets is reached.

2. In a ***slotted buffer***, buffer-overload occurs when a new fragment arrives, and all the slots are used.

In general, and to free some buffer space, there are two ways to deal with an overload situation:

- The most common implementation is to collect fragments of one of the buffered packets and drop them from the buffer. The choice of the packet is determined based on a selection criterion. Currently, the most implemented criterion is to drop fragments of the oldest packet in the buffer [8, 9, 131]. The problem with this approach is that it does not consider the sending behaviour or the completion percentage of the buffered packet, resulting (in many times) in dropping legitimate, almost completed packets. Therefore, a strategy was proposed in [125] that uses *"Scoring"* to rate each packet, depending on the completion percentage of the packet in the buffer and the sending behaviour of the sender. The packet with the least score will be dropped at buffer overload situations. The details of this strategy are discussed in Sec. 4.3.

- Another way is to drop any incoming fragments of packets that are not already in the buffer, and only allowing the reception of fragments that belong to already buffered packets. This method is more suitable for the non-slotted buffer, as it contradicts the main concept of slotted-buffer: allowing the reception of any received fragments. That is why it is rarely implemented.

4.2.4 Common Fragmentation Attacks

Buffer management strategies do not check if the fragments' contents are consistent with each other or if they come from the original sender. An adversary could use these vulnerabilities to launch a DoS attack. For example, the adversary could modify or reconstruct fragmentation fields in the fragments' header, which may overwhelm receiving nodes with big amounts of uncompleted sets of fragments, causing buffer overflow and resources exhausting, resulting in node stalling or shutdown [125].

There are other types of fragmentation attacks, such as *Fragments Gap*, *Ping of death*, and *Teardrop* [123, 125]. However, those attacks were already addressed in 6LoWPAN stack as the adaptation layer always checks *"Datagram_Size"* and *"Offset"* fields in fragments' header and detect any overlapping and/or oversizing.

Still, two types of fragmentation attacks could happen at this layer [22, 123–125]:

- *Fragment-Duplication attack*: shortly put, a malicious node will eavesdrop on the legitimate node(s) to check when a fragmented packet is being sent and inject a fake replica of one of the following FRAGNs before the legitimate one is sent. This will lead the receiver to drop the whole packet at the Network Layer as it will be considered corrupted. Hummen *et al.* proposed a solution to this attack [125], which is discussed in Sec. 4.3.

- *Buffer-Reservation attack*: due to the constrained nature of many IoT nodes, the original 6LoWPAN implementation dictates that when a fragment is received, the receiving node will reserve some of its buffer (and resources) for this fragmented packet. Most IoT OSs limit this buffer to only one or two fragmented packets [8, 125, 131]. In a buffer-reservation attack, the adversary will observe the sending behaviour of legitimate node(s) to get an estimation of how frequently they send a fragmented packet (by sniffing for FRAG1s), and the time-out value for the receiving node(s). The adversary will then precede all legitimate nodes by sending a FRAG1 to the receiver and reserve the buffer for himself. Afterward, it will wait for the estimated time-out to occur at the receiving node and sends another FRAG1 to repeat the reservation procedure, causing a DoS.

  In a more sophisticated attack, the adversary can send several fragments (besides the first FRAG1) in each iteration of the attack, or send all the fragments (of a fake payload) spread over time-out period. However, the latter attack is not as efficient as the other two methods, but it could deceive some security systems.

4.3 Review of Split-Buffer Management Strategy

Hummen *et al.* [125] proposed a scheme and a defence mechanism (presented as a buffer management strategy) to detect and mitigate fragmentation attacks: the ***Content Chain*** scheme to work against *Fragments Duplication* attacks, and ***Split-Buffer*** management strategy against *Buffer Reservation* attacks. Their solution's main features were its efficiency and the low overhead it introduced. Their work [125] showed that the *Content Chain* scheme is very efficient, but *Split-Buffer* defence could benefit from some additional improvements in managing the buffer, as will be described later in this section.

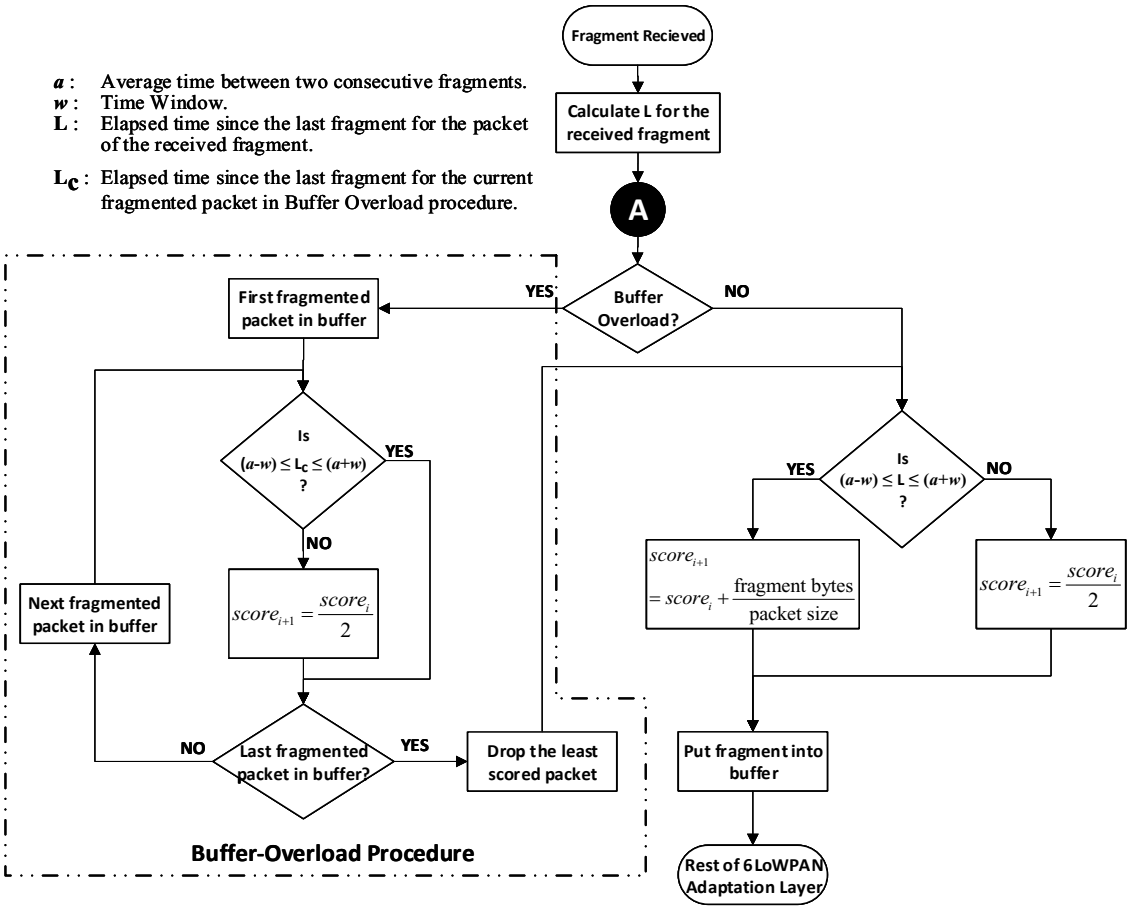


Figure 4.2: *Split-Buffer* management strategy proposed by Hummen *et al* [125]. (This flowchart reflects the implementation of the strategy as used in this chapter's evaluation.)

The *Split-Buffer* management strategy [125] is illustrated in Fig. 4.2, and the main working principles are summarized as follows:-

- The node's reassembly buffer is split into fragment-sized buffer slots. Each slot has the maximum size of a 6LoWPAN fragment for a given Link Layer. These slots are filled until either a packet has been fully received (afterward, the reassembling node assembles the fragments of the packet in-order inside the buffer and processes the packet as usual), or an overload situation is reached. In the latter case, the node has to decide which packet to discard, based on their ***"Packet Score"*** .

- ***Packet Score*** is calculated based on two parameters: *Completion Percentage*, where each successfully-received fragment adds points to its packet's score; and

Sending Behaviour, where each node will penalize packets with a significant change in transmission behaviour. This is done by considering the average time between two consecutive fragments for each packet ($\boldsymbol{a}$), the elapsed time since the last fragment for each packet ($\boldsymbol{L}$), and a global time window ($\boldsymbol{w}$), to give some margin for routing paths changes.

It was noted in [125] that this implementation does not treat legitimate nodes with slow transmissions or those whose fragments went through slower paths fairly as they will suffer from penalties. For example, a short burst of fragments or slow-arriving large fragments will result in higher scores. Increasing the time window ($\boldsymbol{w}$) to overcome this problem will result in a higher probability of admitting malicious packets [125], due to being a global variable. In other words, the strategy is not very tolerant of sudden, long-term changes in network topology and/or conditions.

In addition, it is apparent that when a fragmented packet, whether legitimate or malicious, is dropped (because of the strategy or timeouts), any late fragments from that packet will challenge existing fragmented packets in the buffer, causing a drop for one of them. This happens due to the lack of a trace mechanism for the previously dropped packets. If implemented, such a trace mechanism could filter the late fragments (of previously dropped packets) and prevent them from challenging the currently buffered fragments.

4.4 Proposed modifications to Split-Buffer

Based on observations in the previous section, the following changes are proposed to be added to the current *Split-Buffer* management strategy implementation:-

- The average time between two consecutive fragments ($\boldsymbol{a}$) will be dynamically recalculated (per packet) in the event of a significant change in the reception rate. A counter, ***Trig*** (defined per fragmented packet and is initialized to zero), will track the number of times the inter-fragment time ($\boldsymbol{L}$) goes outside the allowed window ($a \pm w$). Once the counter reaches a predefined value, ***Threshold***, it will trigger a recalculation of ($\boldsymbol{a}$). In addition, the counter is reset whenever ($\boldsymbol{L}$) stays inside the allowed window for three consecutive fragments to reduce the number of ($\boldsymbol{a}$) recalculations. This procedure provides flexible handling of network changes and enhance the network's tolerance to reception rate fluctuations.

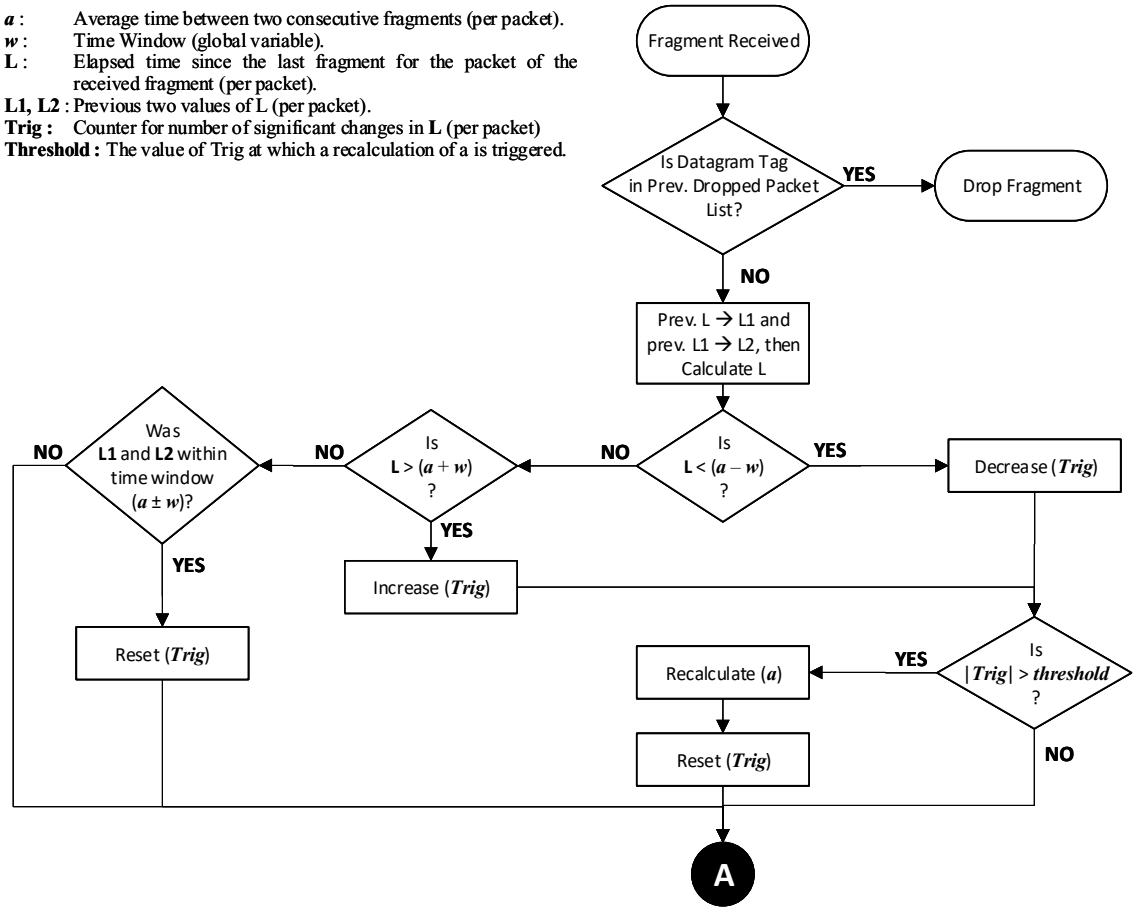


Figure 4.3: Flowchart of proposed modification to the *Split-Buffer* strategy.

- Whenever a fragmented packet is dropped from the buffer, the *Split-Buffer* management strategy will add its *Datagram_Tag* to a list of previously dropped packets (not shown in Fig. 4.3). This list will be maintained and checked every time a fragment is received in order to prevent them from challenging the currently buffered packets. To minimize the effect on the constrained nodes' memory, each entry on this list will be removed after a specific timeout value.

The flowchart of the approach mentioned above is shown in Fig. 4.3. After performing these changes, the execution will continue from point (A) in Fig. 4.2.

4.5 Evaluation Setup

All simulation work was carried out using NS3 [9], as it provides a working module for the 6LoWPAN Adaptation Layer that conforms to the standard in many ways. The *Network model* used for the evaluation is based on a typical 6LoWPAN stack: UDP at

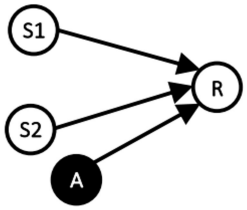


Figure 4.4: Network topology.

Table 4.1: Simulation scenarios

	Adversary sends...		
Adversary Timing	**Full Packets (Normal DoS)**	**1st Fragment Only (Basic buffer resrv. attack)**	**All except last fragment (Soph. buffer resrv. attack)**
Before legitimate node(s)	a-1	a-2	a-3
Simultaneously with legitimate node(s)	b-1	b-2	b-3
After legitimate node(s)	c-1	c-2	c-3

the Transport Layer, IPv6 at the Network Layer, 6LoWPAN Adaptation Layer, and Carrier-Sense Multiple Access (CSMA) protocol for the Logical Link Control (LLC) sub-layer, and IEEE 802.15.4 [20] for the MAC sub-layer. Also, a manual routing entry (pointing toward the receiver node) was used at each sender (legitimate and malicious) instead of enabling a routing protocol due to the simple topology used for the evaluation. The network topology consists of four nodes: two legitimate senders (S1 and S2), one legitimate receiver (R), and one adversary (A). Fig. 4.4 shows the network topology. This topology is widely used for 6LoWPAN protocol evaluations [125, 132] as it puts the focus on the fragmentation/reassembly process of 6LoWPAN Adaptation Layer. Hence, results from this topology can be generalized [125, 132].

In the evaluation, the adversary node is assumed to be a powerful one: unlimited power source, capable processor, and large memory/storage capacity. It will simulate the process of eavesdropping on node (S2) transmissions and initiate the buffer reservation attacks. The aim of the adversary is to create a DoS attack which leads to energy-exhaustion of the legitimate nodes. Finally, to be noted that all the nodes are static, not mobile.

Table 4.2: Simulated buffer management strategies

		Dropping Criteria		
Buffer Storing Method	**Non-Slotted**	*Drop Oldest Packet*	*Split-Buffer Scoring*	*Mod. Split Buffer Scoring*
	Slotted	*Drop Oldest Packet*	*Split-Buffer Scoring*	*Mod. Split Buffer Scoring*

Table 4.3: List of Simulation Parameters

Description	**Value**
Packet size	1280 bytes (the minimum IPv6 MTU)
Data Transmission Rate	100 kbps
Channel Delay	5 milliseconds
Sending Interval	S1 = 250 milliseconds (enough to send 40 packets)
	S2 = 150 milliseconds (enough to send 67 packets)
	A = 150 milliseconds (enough to send 67 packets)
Split-Buffer Window Size ($\boldsymbol{w}$)	45 milliseconds
Modified Split-Buffer Settings	Timeout for the Already-Dropped Packet List Entries = 200 milliseconds
	Trig Threshold = 3

Nine simulation scenarios were used to simulate different cases that an adversary could use to initiate buffer reservation attacks. These scenarios are summarized in Table 4.1, where the notations point to the part of the subfigures representing that scenario, e.g., (a-1) points to part (1) of subfigure (a). In each scenario, six buffer management strategies were used for the comparison. Table 4.2 summarizes the setup of these strategies. In each part of the subfigures, Non-slotted buffer strategies are the three bars to the left of that part, while the slotted buffer strategies are the three to the right. The settings for all the scenarios are listed in Table 4.3

4.6 Results and Discussion

Two experiments were conducted, with a few differences: The first experiment uses one legitimate sender (S2) only, with the receiver (R) having a limited buffer (the non-slotted buffer = 1 packet, slotted buffer = 11 slots). The second experiment uses both senders (S1 and S2), with the receiver having slightly bigger buffer (the non-slotted buffer = 2 packets, slotted buffer = 20 slots). Performing the two experiments provides a better insight into the network's behaviour and whether the nodes will recognize legitimate packets more than malicious ones.

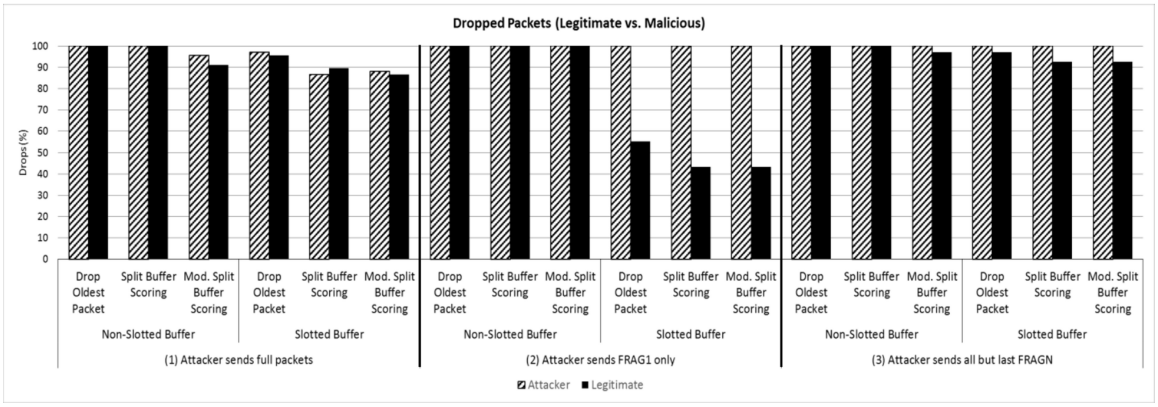
First Experiment Results

Fig. 4.5 shows these results. By looking at non-slotted buffer systems' results, it is clear that most of the packets (legitimate or malicious alike) were dropped, with the modified *Split-Buffer* strategy allowing a bit more packets in (3-10%). From the experiment logs, it appears that preventing the previously-dropped packets from re-entering the buffer is the reason for such difference. When or what adversary is sending does not change the results, nor does changing the dropping criteria as the devices have a small buffer, it will always drop the packet in the buffer.

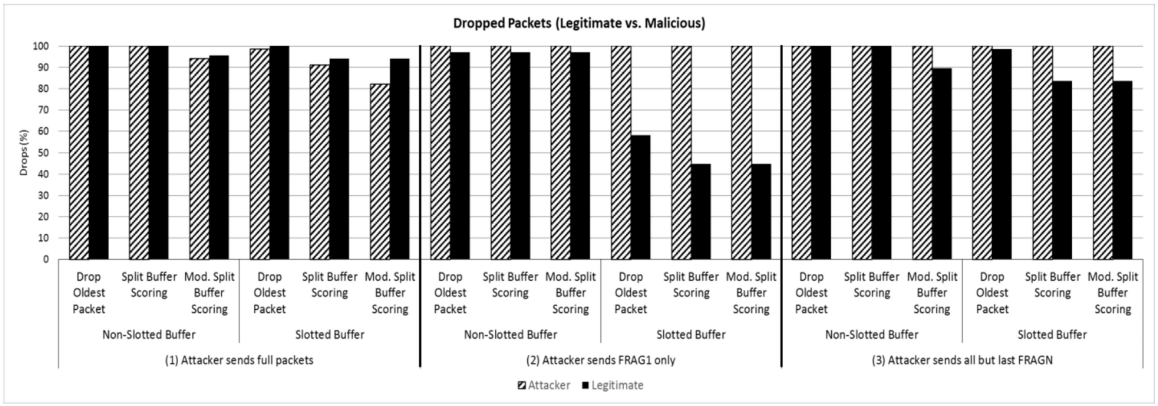
Introducing slotted buffer to the system changed the results drastically: the number of dropped packets reduced between (3-54%) depending on the scenario (the average improvement is 20%), with the biggest improvement happened when the adversary sends only one fragment. Another note is that the scoring system improves the results but by a small margin, as it is clear from comparing *Split-Buffer* mechanisms to the regular 6LoWPAN implementation (with slotted buffer).

Second Experiment Results:

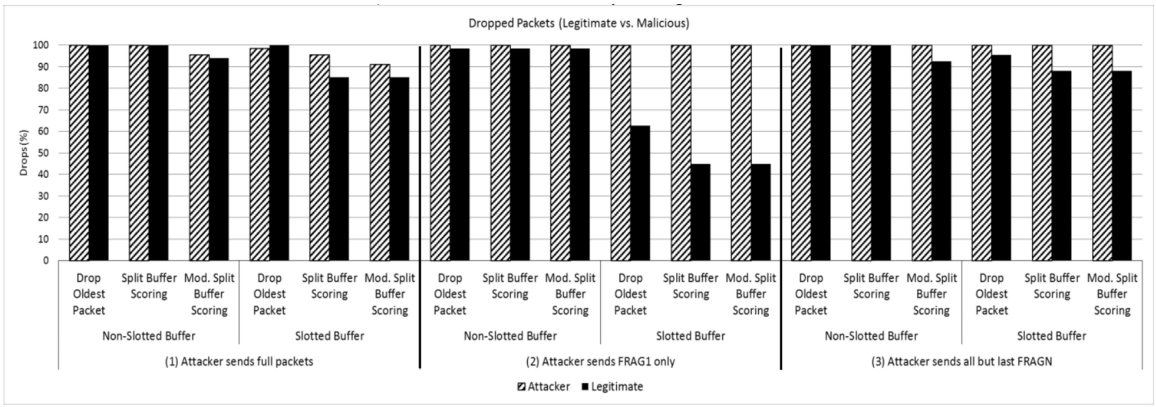
Results are shown in Fig. 4.6, which demonstrate a similar behaviour to the first experiment. In the case of the non-slotted buffer, the *Split-Buffer* scoring strategy provided a better differentiation between legitimate and malicious packets, with the modified version performing slightly better in the other scenarios. The observation is that preventing fragments of previously dropped packets from re-entering the buffer helps more than recalculating the *Split-Buffer*'s scoring parameters.



(a) Adversary sends before legitimate nodes.

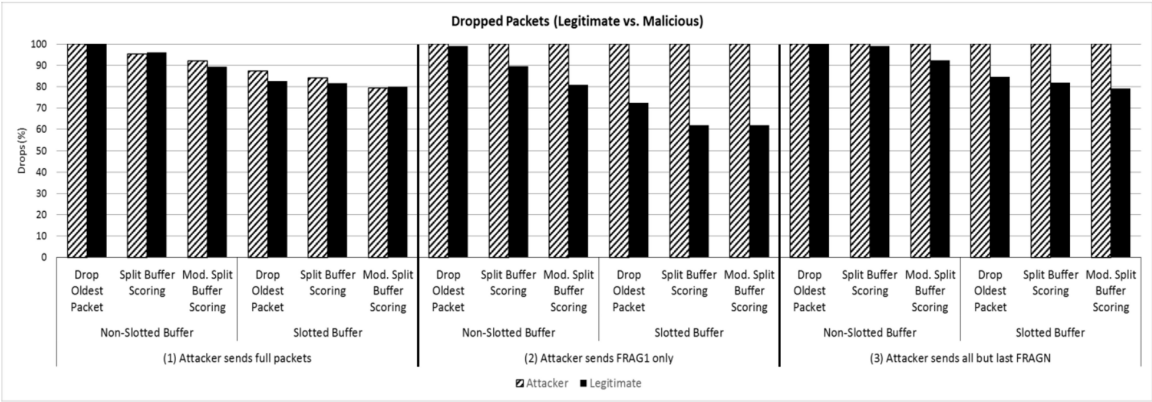


(b) Adversary sends simultaneously with legitimate nodes.

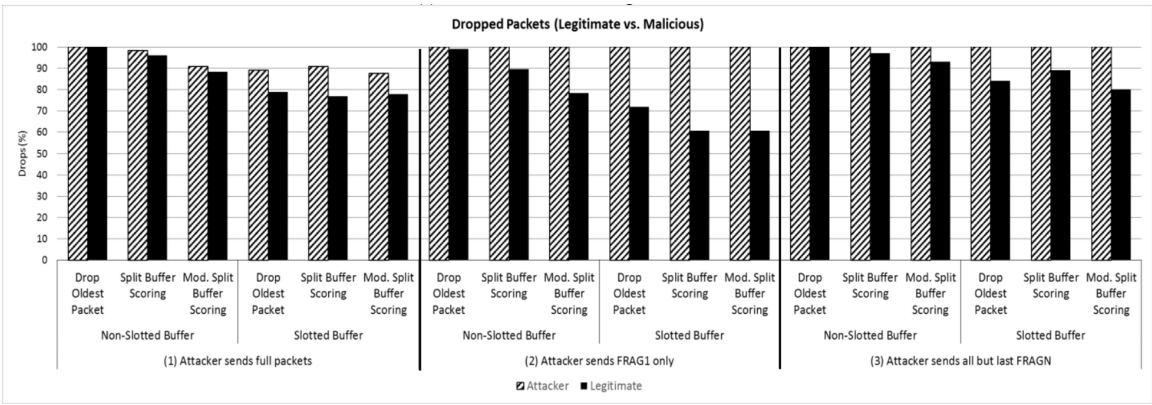


(c) Adversary sends after legitimate nodes.

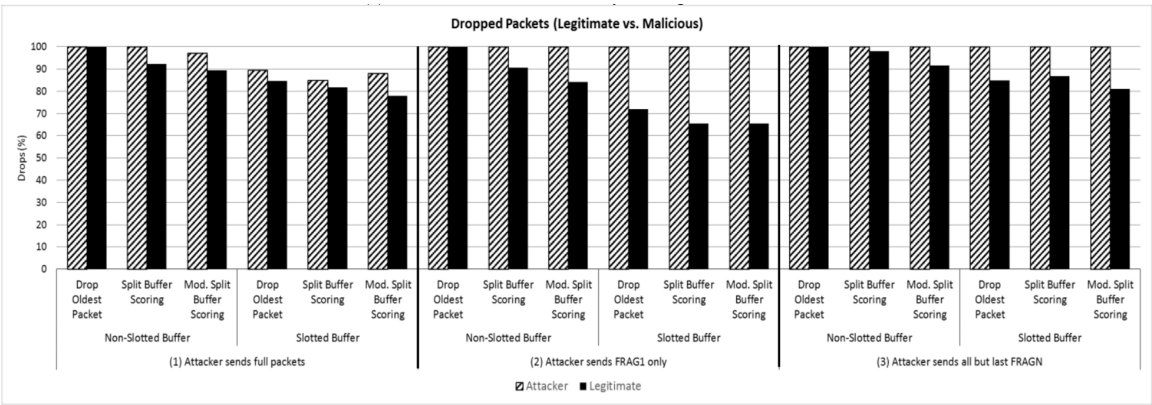
Figure 4.5: Experiment (1): Two senders (one adversary and one legitimate node), one receiving node (buffer size = 1 packet "non-slotted buffer" or 11 slots "slotted buffer").



(a) Adversary sends before legitimate nodes.



(b) Adversary sends simultaneously with legitimate nodes.



(c) Adversary sends after legitimate nodes.

Figure 4.6: Experiment (2): Three senders (one adversary and two legitimate nodes), one receiving node (buffer size = 2 packet "non-slotted buffer" or 20 slots "slotted buffer").

Again, introducing slotted buffer improved the results: the dropping rate of legitimate packets has decreased (9–28%) depending on the scenario (the average is 17.25%). The biggest improvement occurs when the adversary sends only one fragment, with *Split-Buffer* mechanisms having a slight advantage over regular 6LoWPAN implementation. This is due to the scoring system used in the mechanisms, which penalizes heavily malicious nodes due to their sender behaviour and their incompletion.

On a side note, the same experiments were performed using a non-slotted buffer with a dropping criterion that, in case of buffer overload, drops any new incoming fragmented packets until one of the buffered packets are either completed or timed out. The results (not shown in the figures) were as expected: slightly better than the regular 6LoWPAN's drop oldest packet(s) strategy, but still worse than *Split-Buffer* mechanisms. This is because the system is still accepting fragments that belong to the buffered packets.

4.7 Summary

From the above discussion, it can be concluded that using a slotted buffer provides a noticeable improvement for the 6LoWPAN Layer's response against fragmentation-based buffer reservation attacks. Also, it has been shown that using a scoring strategy to clean the buffer enhances the results, by differentiating the malicious and legitimate packets through their sender's behaviour and completion percentage. However, buffer slotting is the major contributor to the improvement in the 6LoWPAN response.

It was also shown that using a scoring system (with the proposed modifications) with non-slotted buffer reduces the dropping rates, which make it comparable to the traditional 6LoWPAN drop-oldest-packet strategy with a slotted buffer in some scenarios. This could be helpful in case it is difficult to implement a slotted buffer.

In the conducted simulations, the values for *Split-Buffer* variables were fixed to check the response of the strategies. However, changing these values could affect such responses For example, several values were investigated for ($\boldsymbol{w}$), and it was found that changing this variable could affect the dropping rate significantly depending on the sender's behaviour. Hence, a further investigation of the optimization of *Split-Buffer* variables is required.

Finally, it is clear that, even with the use of the Split-Buffer strategies, the 6LoWPAN Layer is still susceptible to fragmentation attacks, due to not being able to authenticate the source of the fragments, similar to what RPL suffers from (see Chap. 3). NC comes as a possible solution. The next chapter looks into a solution to the source-authentication problem in RPL using intra-flow NC concepts, presented as a framework. A use case of the framework proposes a security integration with 6LoWPAN will be looked over for in Chap. 6.

Chapter 5

Network Coding-based Security Framework for 6LoWPAN Internet of Things

5.1 Introduction

In Chap. 3, it has been shown that RPL's secure modes, while providing reasonable mitigation of some external attacks, are still vulnerable to many routing attacks (both internal and external) - see Sec. 5.2 - especially the authentication-based replay attacks such as NA and WH attacks. In this chapter, a novel security framework is proposed as a new secure mode for RPL - the Chained Secure Mode (CSM) - which can integrate with any other security measure, whether external mechanisms, IDSs, or other protocols. CSM is designed using the principle of intra-flow NC [135, 136] to introduce an extra layer of security for RPL control communications, and to provide RPL with mitigation capabilities against authentication-based attacks, e.g., replay attacks, while keeping the same working principles of RPL - see Sec. 5.3.

The contributions of this chapter can be summarized as follows:

- A novel framework for RPL, the CSM, was designed with recovery mechanism

This chapter is a revised version of the paper "*Introducing Network Coding to RPL: The Chained Secure Mode (CSM),*" published in IEEE International Symposium on Network Computing and Applications (NCA 2020) [133], and its extended version "*Securing RPL using Network Coding: The Chained Secure Mode (CSM),*" submitted for review to the IEEE Internet of Things Journal, and currently available at arXiv.org [134]

and integration capability with external security mechanisms in mind. The new secure mode uses the principle of intra-flow NC to create a linked chain of coded RPL control messages between every two neighboring nodes. The chaining effect can limit adversaries' ability to launch some routing attacks, mainly the authentication-based attacks (e.g., WH and NA attacks) [10].

- A prototype of CSM's proposed framework was implemented in Contiki OS [8].
- To demonstrate the capabilities of the CSM framework, a security and performance comparison between RPL in CSM and PSM (with and without the optional replay protection) against the NA, CA, and WH attack was conducted using PDR, E2E latency, and network average power consumption (per received packet) as metrics.
- For the internal adversary cases, the results showed that CSM is capable of mitigating both the NA and WH attacks with less latency (≈95% less) and power consumption (≈13-28% less) than PSM with replay protection. In addition, CSM showed enhanced security and was able to significantly reduce the impact of CA on the network, compared to all other secure modes. See Sec. 5.6.
- For the external adversary cases, the results showed that CSM is the only secure mode capable of mitigating the WH attack with PDR≈95-99%, E2E latency between 5-10 milliseconds, and power consumption similar to the normal operation of RPL in UM. See Sec. 5.7.

The rest of the chapter goes as follows: The motivations behind developing CSM is discussed in section 5.2. A brief review on NC is presented in section 5.3. In section 5.4, the details of CSM operation are described. The evaluation setup, assumptions, and adversary model are explained in section 5.5. Evaluation results are analyzed in sections 5.6 and 5.7, with their discussion in section 5.8. Finally, the chapter is concluded in section 5.9.

5.2 Motivations Behind CSM

As mentioned earlier in Chap. 3, RPL secure modes (PSM, and by extension, ASM) are able to mitigate most of the external attacks[1], while it does not enhance RPL's

[1] An external attack refers to an attack that is launched by an adversary who is not part of the network, e.g., it does not have the encryption key used by the legitimate nodes for RPL in PSM, or runs RPL in UM.

security against the internal attacks[2]. However, it has been confirmed that external adversaries still can launch replay attacks, even when PSMrp is used (e.g., in the case of the WH attack - see Sec. 3.3.)

A further investigation of RPL standard [13] shows that it only provides confidentiality and integrity of its control messages, without any verification of their authenticity. This opens the door wide open for attacks such as the Rank, Version, Sinkhole, Sybil, identity cloning, eavesdropping, and replay attacks [10] (see Chap. 2) to be launched regardless of the secure mode RPL is running in. For example, an external adversary can launch a *Neighbor attack* (see Sec. 2.7.5) by merely monitoring the "*Type*" and "*Code*" header fields in any ICMPv6 message to identify RPL's DIO messages[3], without the need to decrypt the actual message [2] – see Sec. 3.4.1.

The lack of message authenticity in RPL provided a main motivation for this dissertation's work, to devise an innovative method to overcome this problem, and NC came into the light as a possible solution. Incorporating the intra-flow NC into RPL would provide any receiving node with proof of immediate-sender authenticity, assuming that the first message came from the legitimate sender. This is true for most attacks as the adversaries usually join the network after it has been initiated and stabilized.

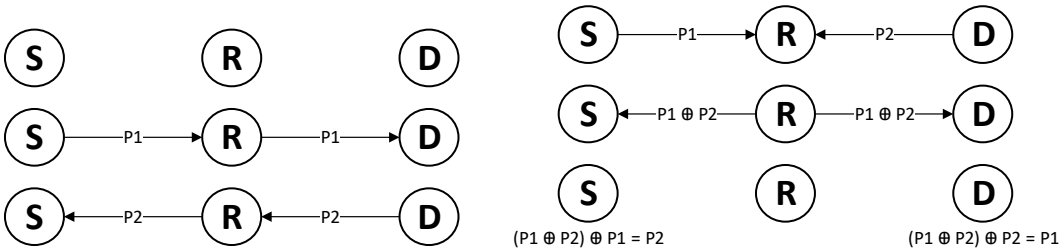
5.3 A Brief Review on Network Coding

NC has received a great deal of attention since it was first proposed by Ahlswede *et al.* [3]. Many researchers have investigated NC schemes (e.g., *XOR*, *Random Linear NC*, etc.) to improve network efficiency (e.g., throughput, reliability, and E2E delay) using different communication technologies (wired, wireless, or ad hoc networks) [137].

The basic idea of NC is that a source combines multiple pieces of information or packets using a coding scheme and forwards the coded information to the next network device. The receiver then, upon receiving enough information, decodes the combined information to recover the original data. The simplest NC scheme is XOR; e.g., a node can perform bit-by-bit XOR operations of two packets in sequence and forward the XOR-ed packet to the next hop to reduce the number of transmissions [3].

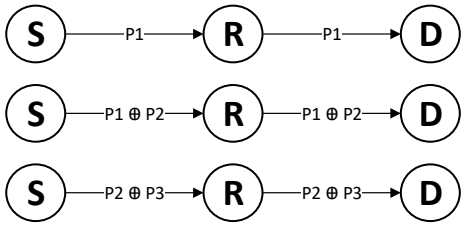
[2] An internal attack is launched by an adversary who is part of the network, e.g., it has the encryption key used by the legitimate nodes for RPL in PSM.

[3] (Type = 155) means this is an RPL message. (Code = 1 or 129) means it is a regular or secure DIO message, respectively.



(a) Normal network communication

(b) Inter-flow NC



(c) Intra-flow NC

Figure 5.1: Examples of NC communication. The ⊕ sign represents XOR as a simple NC operation.

NC can be applied to either *inter-flow* or *intra-flow* traffic. Inter-flow NC applies coding to packets from different traffic flows (see Fig. 5.1(b)), whereas intra-flow NC uses coding for packets of the same traffic flow [6, 138] (see Fig. 5.1(c)), creating a *chain* of messages. Inter-flow NC requires more complex operations, such as buffering and synchronization of packets from multiple flows or different sources. Intra-flow NC, on the other hand, is much easier as it only considers the sequence of packets within the same flow, which makes it suitable to the resource-constrained IoT.

In this chapter, a security framework, presented as an innovative secure mode for RPL (the CSM), is proposed using the intra-flow NC. The *chaining* effect from this method adds the missing immediate-sender authenticity to RPL (assuming that the first message came from the legitimate sender) and increases its resilience against several authentication-based routing attacks (e.g. replay attacks). For concept demonstration only, the simplest NC scheme, the ***XOR***, was used. However, more sophisticated NC schemes can be used for higher levels of security.

Throughout this chapter and the following one, a *flow* is defined as the stream of RPL control messages from a node toward a specific IPv6 address. This can be for a Unicast (UC) transmission (a unicast IPv6 address of a certain neighbor of the node) or a Multicast (MC) transmission (a multicast IPv6 address.)

5.4 How CSM Framework Operates

The following points constitute the main design considerations of the CSM framework:

- Adhering to the RPL standard by maintaining the same procedures used for PSM, which is important for operational simplicity and compatibility.
- Using intra-flow NC to provide a broad authentication-based replay attacks mitigation capability as part of RPL standard.
- Allowing external security measures to integrate with CSM by controlling how RPL trusts nodes and using this information to make security decisions.
- CSM's primary focus currently is on protecting static networks, as they constitute the majority of current IoT applications.

The simplest implementation of intra-flow NC is to *encode* the current packet with the previous one from the same flow using a simple XOR NC scheme – as in Fig. 5.1(c). Here, the receiver node should always keep the previous packet so it can *decode* the incoming message and retrieve the new packet.

A problem that arises when implementing the aforementioned concept in an IoT network is the limited resources available for the nodes, which renders such implementation impractical. As an example, if a node has 30 neighbors, it should store the last 30 messages from these neighbors. Assuming the average size for RPL control messages is 80 bytes [13], the receiving node has to reserve at least 2400 bytes (≈2.4 KB) from its limited memory so it can decode any received message properly.

To overcome this problem, CSM uses Secret Chaining (SC) values instead of the entire previous packet for the encoding/decoding process. These SC values are 4 bytes long (as currently in the prototype design), randomly generated unsigned integers (for each sent control message), and are locally unique for each of the node's neighbor. Compared to the example mentioned above, the receiving node will store 120 bytes only of the SC values instead of 2400 bytes for all the thirty neighbors. This is a huge saving for the resource-constrained IoT nodes.

Since RPL sends its control messages as either Multicast (MC) or Unicast (UC) messages, CSM considers them two independent flows: an MC-flow and a UC-flow. Hence, every node in the network should maintain a table (the *SC table*) of the following SC values for each neighbor, in order to successfully encode and decode their control messages:

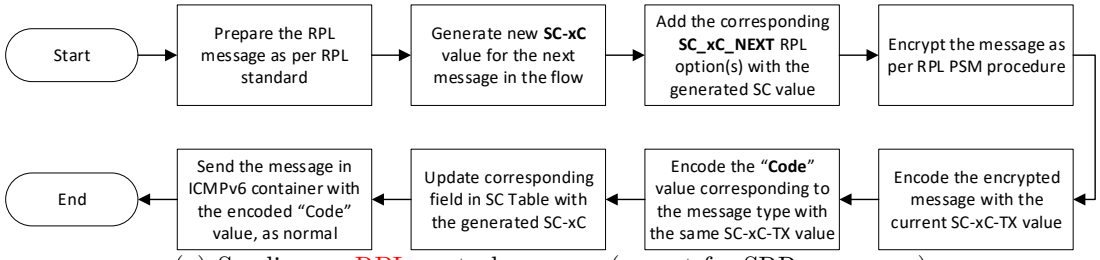
- **SC_UC_RX:** The SC value used to decode the next incoming UC-flow message from the neighbor.
- **SC_MC_RX:** The SC value used to decode the next incoming MC-flow message from the neighbor.
- **SC_UC_TX:** The SC value used to encode the next outgoing UC-flow message to the neighbor.
- **SC_ER_TX:** The SC value used to encode the next outgoing Emergency (ER)-flow message to the neighbor - see Sec. 5.4.1.

In addition, each node should maintain the next SC value for its next MC-flow transmission (**SC_MC_TX**) and ER-flow reception (**SC_ER_RX**) – see Sec. 5.4.1. For simplicity, the current CSM design uses *zero* as a value for the SC used for the first transmission in each flow.

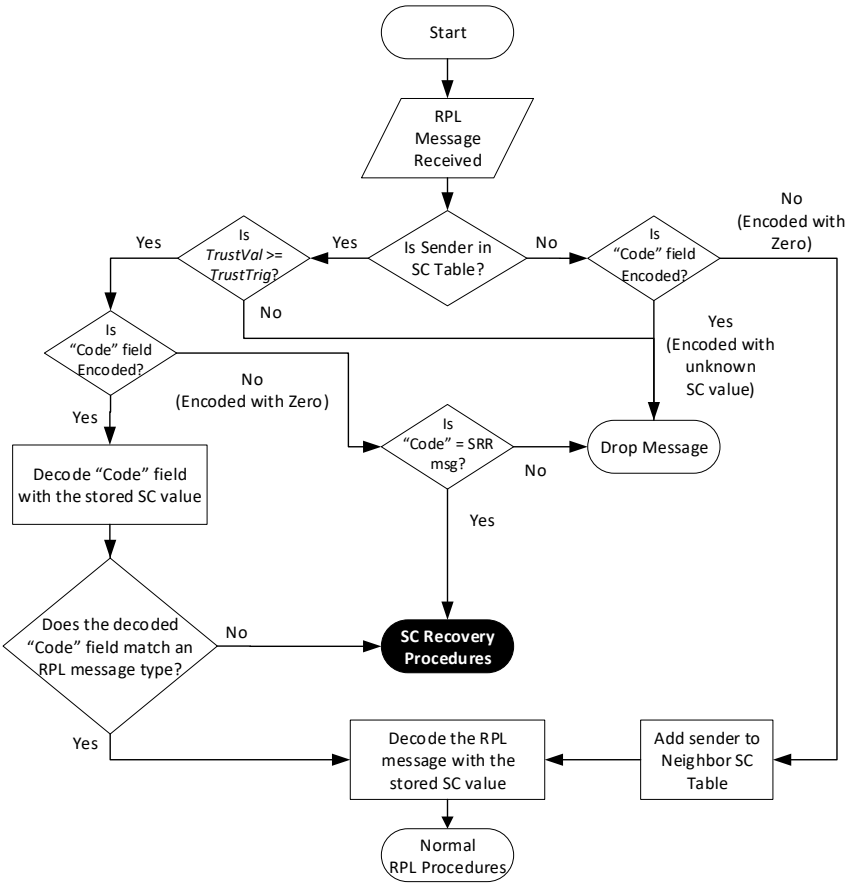
To exchange the SC values used to encode the next control message, CSM employs the *RPL Control Message Options* from the standard [13]. These optional add-ons are used to provide (or request) information to (or from) the receiver. CSM adds three new options to accommodate the transmission of the next SC used for each flow: the (**SC_UC_NEXT**) option includes the SC value to be used for the next UC-flow message, (**SC_MC_NEXT**) is for the SC value to be used for the next MC-flow message, and (**SC_ER**) is for the SC value to be used for the next ER-flow message – see Sec. 5.4.1.

When a node wants to send an RPL control message (whether for the UC-, MC-, or ER-flow), it will follow the following steps - see Fig. 5.2(a):

1. Prepare the message as per the standard PSM procedures.
2. Generate a new SC value to be sent within the corresponding RPL option. The generation process also ensures that the generated SC values, when used to encode the ICMPv6 *Code* field (see step 4), will not result in one of the valid values of the RPL message type identifier. This step is not performed for SC Recovery Request/Response (SRR) messages - see Sec. 5.4.1.
3. Adding the (SC_xC_NEXT) and (SC_ER) new control message options, as per the RPL standard. CSM should add both the (SC_UC_NEXT) and



(a) Sending an RPL control message (except for SRR messages.)



(b) Receiving an RPL control message.

Figure 5.2: Flowcharts represent the sending and reception procedure of an RPL message in the current CSM prototype.

(SC_MC_NEXT) for UC-flow messages and only the (SC_MC_NEXT) for the MC-flow messages. Sending both options for the UC-flow allows for quicker recovery from message chain-breakage in the MC-flow.

4. The *Code* field of the ICMPv6 header is encoded using the corresponding SC_UC_TX or SC_MC_TX value to mitigate the security vulnerability addressed

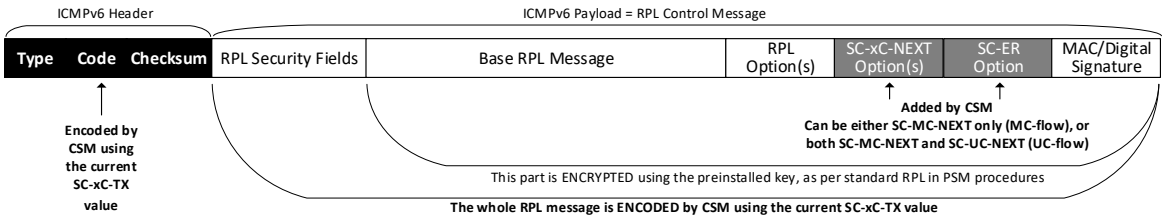


Figure 5.3: Format of an RPL control message, as constructed by the proposed CSM. The black parts represents ICMPv6 header, the white parts are standard RPL in PSM fields, and the grey parts are added by CSM.

in Sec. 5.2. The only exception is the *SRR* messages, which keep their designated ICMPv6 "Code" value without encoding (see Sec. 5.4.1). Since the *Code* field is one byte long, its encoding will be a sequential one; i.e., it is encoded first with the first byte of the SC value, then the result is encoded with the second byte of the SC value, and so on.

After encrypting the message (according to standard PSM procedures), CSM will encode the whole message using the corresponding SC value then send it as usual. Fig. 5.3 depicts how CSM constructs an RPL message.

At the receiving node, the decoding SC value is found from the SC table using the sender IP address. The found SC value is used to decode the *Code* field of the ICMPv6 header to identify the type of RPL message, then the whole message is decoded using the same SC value and is processed as per PSM procedures. Any message with a non-decodable *Code* field will be discarded without processing. Fig. 5.2(b) shows a flowchart for message reception in CSM.

Fig. 5.4 shows few examples of CSM normal operation. The parts (a) and (b) represent the first MC transmissions, where the sender node generates new SC_MC_TX value (to be used for its next MC transmission) and SC_ER_RX. Then, it includes the generated values within the sent message before encoding it with zero (this is the first MC transmission). At the receiver side, once the message is decoded successfully, the receiver adds an entry in its SC Table for the sender with the SC values it extracts from the received message. A similar situation is shown in parts (c) and (d) for the first UC transmission. However, the difference here is the inclusion of both the UC and MC SC values in the sent message. Finally, the parts (e) and (f) show how the subsequent UC and MC transmissions will update the SC Table at the nodes.

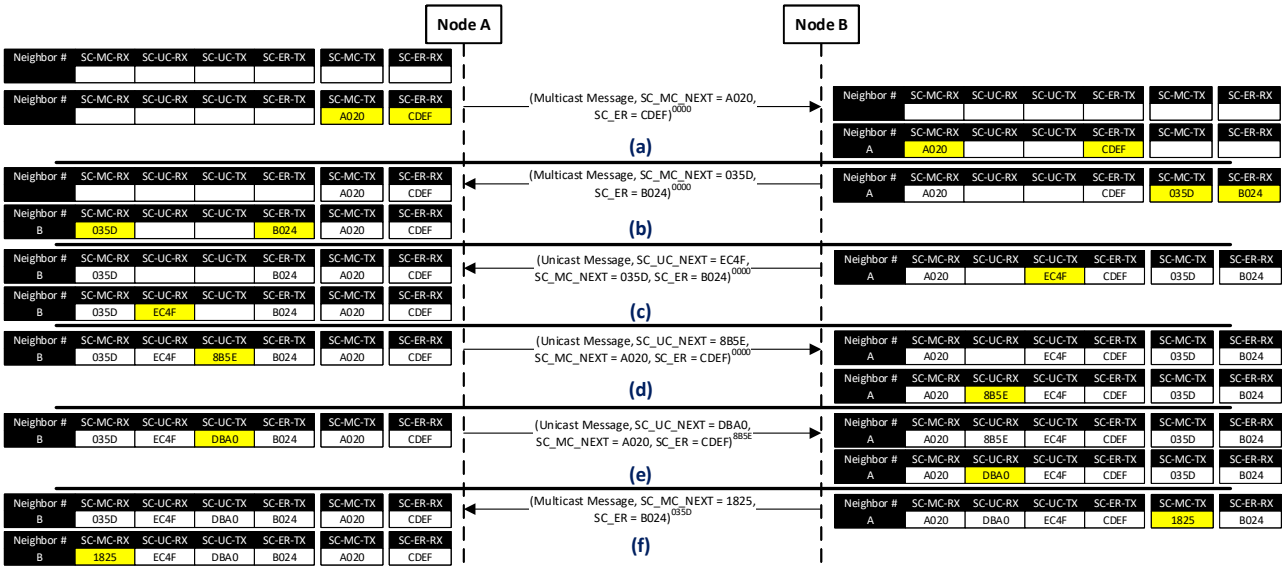


Figure 5.4: Examples of normal CSM operation in chronological order (the number on the top-right of the brackets represents the SC value used to encode that message): (a and b) the first message in the MC-flow, (c and d) the first message in the UC-flow, (e) subsequent messages of the UC-flow, and (f) subsequent messages of the MC-flow. The yellow color highlights a creation or a change of an SC value in the SC table.

5.4.1 SC Recovery Mechanism

It is clear that CSM requires a proper recovery mechanism for when a control message from any NC flow is missed or lost, otherwise all subsequent communications in that flow will be discarded due to not having the correct SC value to decode it.

As CSM is designed for the static networks, there are two cases where nodes can lose track of the SC values:-

1. **Node Reset:** When a node is reset for any reason (e.g., battery replacement, firmware upgrades, etc.), it will lose all stored SC values and start from scratch. For this, CSM assumes that the OS will periodically save all SC values (node's own and SC table) to the filesystem, and loads them at boot-up time, so the node can resume from the point just before the reset.

2. **Lost or Corrupt Messages:** Missing a control message is normal for lossy networks such as the ones RPL is designed for [13]. However, in CSM this means breaking the message-chain for one of (or both) flows, resulting in discarding all subsequent messages of the broken flow.

To recover from the second case mentioned above, CSM implements a special recovery mechanism, dubbed the *SC Recovery*. This recovery mechanism applies only to the neighbors that are already in the SC table. To secure the recovery process, the SC values are not sent as clear text. Instead, a challenge/response exchange is performed based on the concept of solving Linear Algebra equations that involves the missing SC values. In general, assuming that node (A) is the receiver of the "non-decodable" message and node (B) is the original sender, node (A) will send an SC Recovery Request (SRReq) message to node (B) containing the coefficients of a system of linear equations, to which node (B) will use the coefficients and its next SC values to calculate the results of the linear equations. These results are replied to node (A) inside an SC Recovery Response (SRRes) message. Now, node (A) will use the provided information to solve the linear equations and extracts the missing SC values for node (B). The reason behind this procedure is to raise the bar for the adversaries, in terms of processing powers and other resources, to launch attacks against (or abuse) the recovery mechanism.

Based on the above, the SC recovery mechanism goes as follows:

- A new NC flow is added to CSM, the *Emergency (ER) flow*, to be used when exchanging the SRR messages. Each node will maintain an SC value (**SC_ER**) for this flow, and exchanges it through the (**SC_ER**) option in every message of the other flows – see Sec. 5.4. However, the SC_ER only updates after a successful recovery.

- When a message is received, if the decoded "Code" field at the ICMPv6 header of the received message does not represent an RPL control message, the following methods are performed to recover the missing SC value:

  - If the received message was from the MC-flow, a regular UC-DIS message is sent to the sender. As per RPL standard, the sender must reply with a regular UC-DIO, which will have all the correct SC values for the next message in all the flows.

  - If the received message was from the UC-flow, the SC recovery mechanism is conducted:

    1. The received message will be discarded.
    2. The receiver sends a UC-SRReq message to the original sender, containing the randomly-generated coefficient values to be used by the

original sender as explained above. The SRReq is encoded with the original sender's **SC_ER** value.

3. Once receiving the SRReq message, the original sender will calculate the results of the linear equations, then it sends them within a MC-SRRes message to the receiver, encoded with the receiver's **SC_ER** value. Since this is a multicast message, the SRRes message contains the IPv6 address of the node that is supposed to process it. In addition, the original sender will update his SC_ER and send it within the corresponding RPL option.
4. Now, the receiver will use the coefficients and the received results to solve the linear equations and update its SC table with the extracted SC values.

5.4.2 The *CSM-Trust* Integration Interface

To allow integration with external security measures, CSM provides a trust-based control interface for such security systems called the *CSM-Trust* interface, which uses the *TrustVal* value (as part of the SC table) to define the trust-worthiness of each of the node's neighbors. Hence, the acceptance or rejection of RPL's control messages from that neighbor.

The *CSM-Trust* interface allows external security measures to set and use the TrustVal value according to their needs. For example, an external security mechanism (such as an IDS) would set the boundaries for *TrustVal*; the maximum (*TrustValMax*) and minimum (*TrustValMin*), and the trigger value (*TrustTrig*) that, if *TrustVal* went below it, CSM will drop any RPL control messages from that neighbor. the conceptual diagram of *CSM-Trust* interface is shown in Fig. 5.5.

In addition, the calculation of *TrustVal* is left to the external mechanism and can be dynamic to allow for larger flexibility to different applications and scenarios. For example, an IDS can use its own methods (e.g., special messages, monitoring the traffic, etc.) to determine the neighbor's trustworthiness; then it updates *TrustVal* in the SC table, which tells CSM if it should accept or reject control messages from that neighbor.

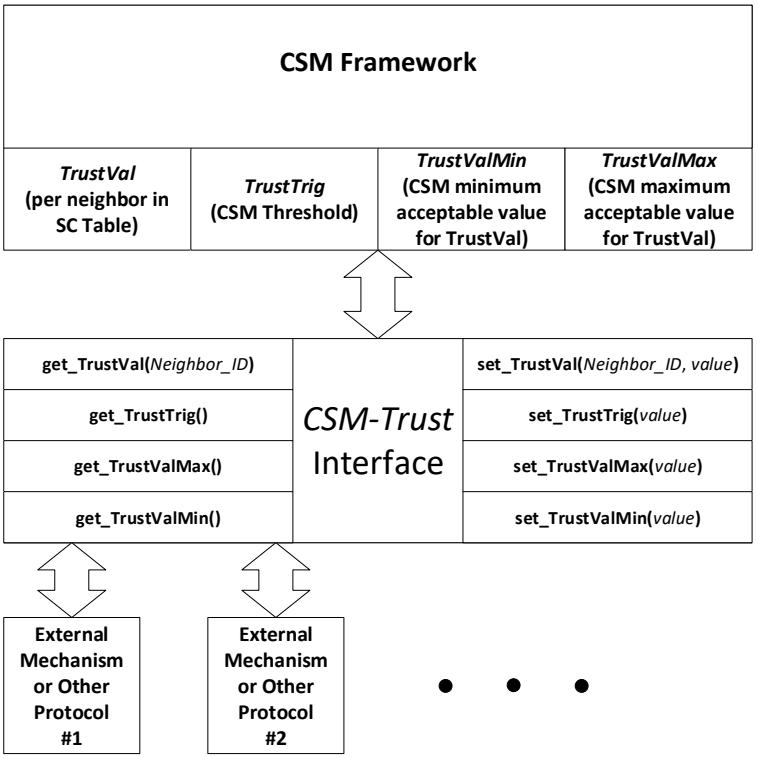


Figure 5.5: *CSM-Trust* interface concept diagram.

5.5 Evaluation of the CSM Framework

To evaluate the proposed CSM framework, a comparison of security and performance was conducted between the devised prototype of CSM and the currently implemented secure modes: RPL in UM (vanilla ContikiRPL), PSM, and PSMrp (both according to Perazzo *et al.* [46] implementation). All secure modes were evaluated against three routing attacks (NA, CA, and WH).

5.5.1 Evaluation Setup and Assumption

Cooja, the simulator for Contiki OS [8], was used for all the simulations (with simulated motes). Fig. 5.6 shows the topology used in the evaluation. Also, a list of simulation parameters is provided in Table 5.1.

For the purpose of the evaluation, the following metrics were used: the average data PDR, average data E2E latency, and the average network power consumption per received data packet.

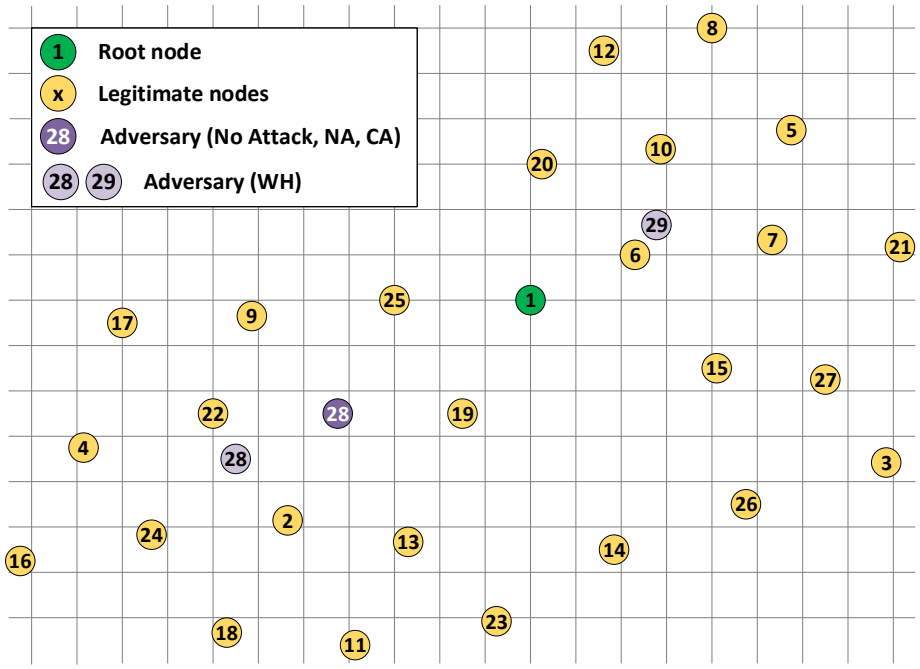


Figure 5.6: Network topology used for the evaluation. The adversaries' locations are represented by a dark-purple circle (No Attack, NA, and CA scenarios) or the light-purple ones (WH scenario.)

Table 5.1: List of Simulation Parameters (per RDC protocol)

Description	**Value**
No. of simulation sets	Two: one for each adversary type (**I**nternal and **E**xternal) (See Sec. 5.5.2)
No. of experiments per set	Four: one for each secure mode (UM, PSM, PSMrp, and CSM)
No. of scenarios per experiment	3 (ContikiMAC) / 4 (NullRDC) - See Sec. 5.5.1
Sim. rounds per scenario / time	10 rounds / 20 min. per round
Node positioning	Random distribution
Deployment area	210m W x 150m L
Number of nodes	28 (29 for the WH scenario) including 1 adversary (2 for WH)
Sensor nodes type	Arago Sys. Wismote mote

The following assumptions were considered in the evaluation: all the legitimate nodes send data packets toward the root at a rate of 1 packet/minute per node, while the adversary does not send any data packets. For all the evaluated secure modes, RPL is set up with the default OF, namely the MRHOF [35]. Contiki OS is using the following settings for its uIP stack: IEEE 802.15.4 [20] for the Physical Layer and MAC sub-layer, ContikiMAC [110] and NullRDC [8] for the RDC sub-layer (see Sec. 3.2.4), IPv6, 6LoWPAN and RPL at the Network Layer, and UDP for the Transport Layer. To keep the focus on RPL, all security measures and encryption at the Link Layer were assumed to be disabled.

The data traffic model used for the evaluation, as described above, is a deterministic one that mimics a typical sensing-IoT network, where nodes send their sensor readings toward the root node at predetermined periods ($\approxeq$ 1 packet every minute).

To test the CSM-Trust integration interface, the following simple, proof-of-concept, mechanism was implemented (only in CSM experiments):

- *TrustValMin*, *TrustValMax*, and *TrustTrig* were set to 0, 100, and 50, respectively.
- For the first RPL message from a neighbor, a successful reception will set *TrustVal* to *TrustValMax*.
- Afterward, TrustVal will increase or decrease based on the successful (or unsuccessful) decoding of the received RPL control messages. The increment (and decrement) amount was set to 10.

The results obtained from the simulations were averaged over ten rounds per experiment with a 95% confidence level.

5.5.2 Adversary Model and Attack Scenarios

The authentication-based attacks used for the evaluation were chosen because of the low cost for the adversary to launch them, as they require little or no processing of RPL's messages. At the same time, the effect of these attacks can be significant on the network. The location of the adversary(ies) was chosen to present the most prominent effect of the investigated attacks [14, 111, 112] – see Sec. 3.2.

All the secure modes were evaluated in both normal operation (*No Attack*) and against three replay routing attacks [10, 14], see Chap. 2, (the following describes the attacks as implemented in this chapter):-

1. **Neighbor attack (NA):** Whenever the adversary hears a DIO message from any neighbor (regardless of its destination), it will replay it (as a multicast) to all its neighbors without modifications or processing, deluding them to think that the original sender is within their range. If the original sender has a better rank (see Sec. 2.3), the receivers of the replayed message will select it as their preferred parent, which may result in lost data packets and longer E2E delays [2]. As in Sec. 3.2.3, the objectives of this attack are the disruption of data packets transmission and disconnecting the routing topology.

2. **CloneID attack (CA):** The adversary will clone the identity of another node, in our case node (25), by changing its IPv6 and MAC addresses to match that of the cloned node (by monitoring its frames and RPL messages). In addition, it will follow and copy the cloned node's rank by reading its DIO messages. The implementation in this dissertation combines CA with a SF attack (that only drops data packets passing through the adversary - see Sec. 2.6.1) to better show how the CA changed the DODAG around the adversary. As mentioned in Sec. 2.6.4, the main goals of this attack are to disrupt the routing topology and manipulate any reputation-based IDS.

3. **Out-of-Band Wormhole (WH) attack:** Two adversaries (connected by an out-of-band link) will forward and replay all RPL control messages they hear from their neighbors between the two locations where they reside. Due to simulation limitations [2], this attack scenario is only available in the NullRDC set of experiments - see Sec. 3.2.4. Similar to Sec. 3.2.3, the attack aims to increase data packets' latency, disrupts the routing topology, exhausts the victim nodes' energy, and disconnects parts of the network.

For the adversaries, they run in the same RPL secure mode as the legitimate nodes and they are assumed to have unlimited energy sources, capable processing powers, and large memory/storage capacities. However, they have two types: *Internal* adversaries, which have the proper preinstalled encryption key for PSM, PSMrp, and CSM experiments; and the *External* adversaries, which do not have the required encryption key. Also, it is worth mentioning that the external and internal versions of the adversaries in UM scenario are the same.

In all cases, the adversary starts as a legitimate node, tries to join the network, then launches the attack after two minutes. For the WH attack, the two adversaries

are always in promiscuous mode and never participate in the DODAG.

5.6 Results for Internal Adversary Sets

5.6.1 Effects on the Data PDR

Looking at Fig. 5.7(a) (ContikiMAC) and Fig. 5.8(a) (NullRDC), it is clear that PSMrp and CSM (for both ContikiMAC and NullRDC) successfully eliminated the NA effect, with both of them having almost 100% PDR. UM and PSM suffered more (PDR≈ 80-90%) as the adversary actually was able to become part of the network.

For the CA scenario, it is noticeable that the attack was able to confuse the surrounding nodes and in many cases it successfully switched their preferred parent to the adversary for the UM, PSM, and PSMrp. This shows in the lower PDR (PDR≈ 80-85%) for all secure modes except CSM. On the other hand, CSM was able to reduce the effect of the attack (PDR≈ 85-95%), albeit the reduction here is caused by CSM's integration with the external security mechanism, which acted as a dynamic blacklist and "*untrusted*" both the legitimate and cloned nodes after several unsuccessful recovery attempts.

Finally, CSM outperformed the other secure modes in the WH attack scenario, where it was able to mitigate the attack (PDR≈ 95-99% compared to 75-80% for the other secure modes). This is mainly due to CSM policy of dropping RPL control messages (without processing) from new neighbors if they were encoded with unknown SC values. Hence, the routing topology will not change due to the WH attack.

5.6.2 Effects on the Data E2E Latency

Looking at Fig. 5.7(b) (ContikiMAC) and Fig. 5.8(b) (NullRDC), it can be seen that NA has been mitigated by both PSMrp and CSM (latency ≈ a few milliseconds), and that it introduced higher E2E latency to the network for the other secure modes (≈ 40-70 seconds), confirming the findings in Chap. 3.

The results for the CA shows the significant effect of the attack on the network, where the latency is in the (100-200) seconds range for UM and PSM, while PSMrp varies between 20 seconds (NullRDC) to 100 seconds (ContikiMAC). This variation is caused by ContikiMAC as its energy-conservative mechanism amplifies any latencies due to transceiver extended *sleep* times [2, 115].

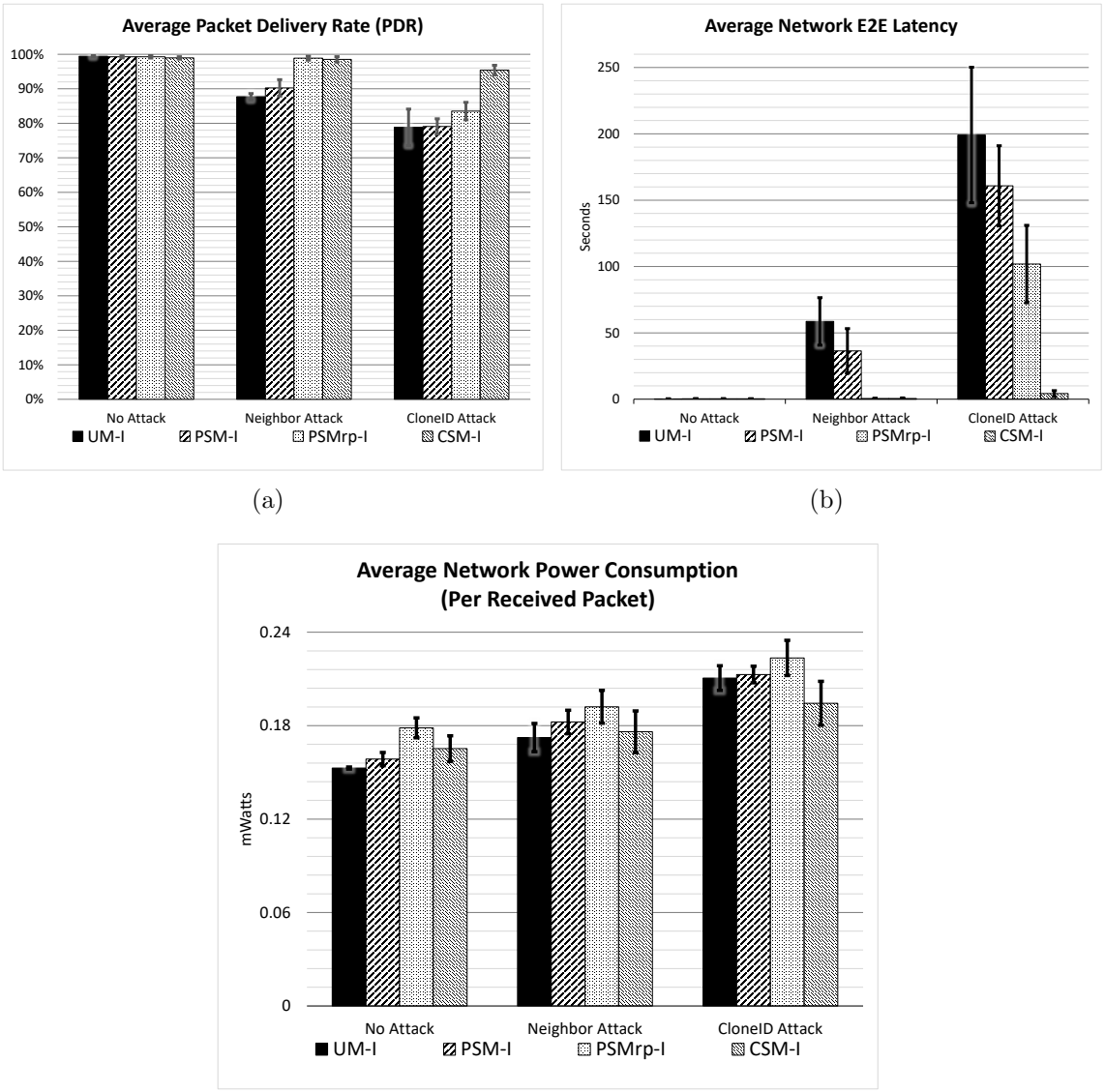


Figure 5.7: Simulation results for the four experiments (three attacks scenarios - internal adversary), using ContikiMAC RDC protocol.

Further proofing CSM mitigation of WH attack, latency is kept to minimum (<5 seconds) compared to other secure modes (≈80-200 seconds) for both RDC protocols.

5.6.3 Effects on Power Consumption

Comparing Fig. 5.7(c) (ContikiMAC) and Fig. 5.8(c) (NullRDC), it can be seen that all secure modes have similar patterns, where the CA scenario has the higher power

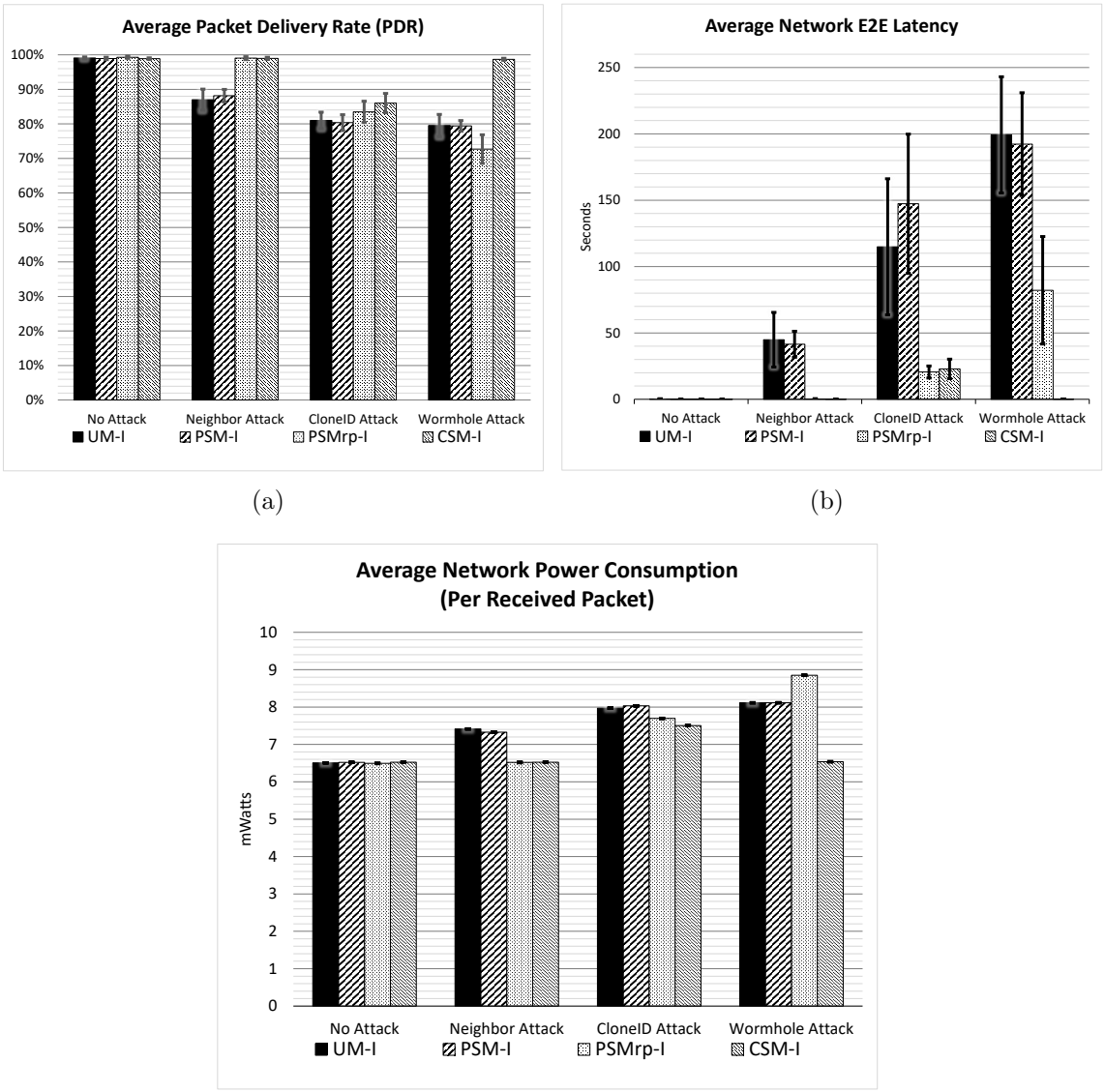


Figure 5.8: Simulation results for the four experiments (four attacks scenarios - internal adversary), using NullRDC RDC protocol.

consumption in the ContikiMAC set, and the WH attack scenario has the highest power consumption in the NullRDC set. However, CSM has the lowest power consumption in every situation when compared to other secure modes. This is more prominent in the WH attack scenario. It is worth mentioning that power readings for NullRDC are higher than the ones for ContikiMAC, as the transceiver is always on for the former while it is off most of the time for the latter [2, 109].

5.7 Results for External Adversary Sets

5.7.1 Effects on the Data PDR

Fig. 5.9(a) (ContikiMAC) and Fig. 5.10(a) (NullRDC) show that PSM, PSMrp, and CSM are capable of mitigating both NA and CA, with the latter having a slightly more effect in the case of ContikiMAC set (PDR $\approx$ 90-99%). However, CSM was the only secure mode that mitigated the WH attack with (PDR $\approx$ 98%) compared to ($\approx$70-75%) for the other secure modes. To explain the effect of the CA attack on the network when ContikiMAC is used, the way RPL decides on selecting the preferred parent or switching to another parent must be understood. In general, RPL depends on neighbors' statistics, which are provided by RPL itself, IPv6 Network Discovery protocol [116], and Link-layer protocols. In ContikiMAC, the Link-layer statistics depends on the successfully-received frames, probing frames, and acknowledgment frames from these neighbors, among other factors [139]. Now, when RPL operates on top of ContikiMAC, it will use these statistics, besides its own, to decide if the neighbor is still "*fresh*" or not. In other words, if it should keep the neighbor in its possible-parents list or even as the preferred parent [13]. However, NullRDC does not provide such statistics, leaving RPL only to use its statistics.

Since the dropped RPL control messages are still successfully received by ContikiMAC, it will affect RPL's decision on keeping the victim node (and the cloned node, i.e., the adversary) as the preferred parent at the neighboring nodes. In many cases, it will prolong the time required to switch to another parent until RPL's statistics point to a communication failure and force the switch, unlike the case for NullRDC where RPL quickly detects the communication failure and switch to another parent.

5.7.2 Effects on the Data E2E Latency

similar to the analysis of PDR, Fig. 5.9(b) (ContikiMAC) and Fig. 5.10(b) (NullRDC) show that the three secure modes have successfully mitigated the NA, while the CA still has an effect in the case of ContikiMAC for the same reason mentioned above. Again, CSM shows as the only secure mode capable of mitigating the WH attack, with minimum E2E latency.

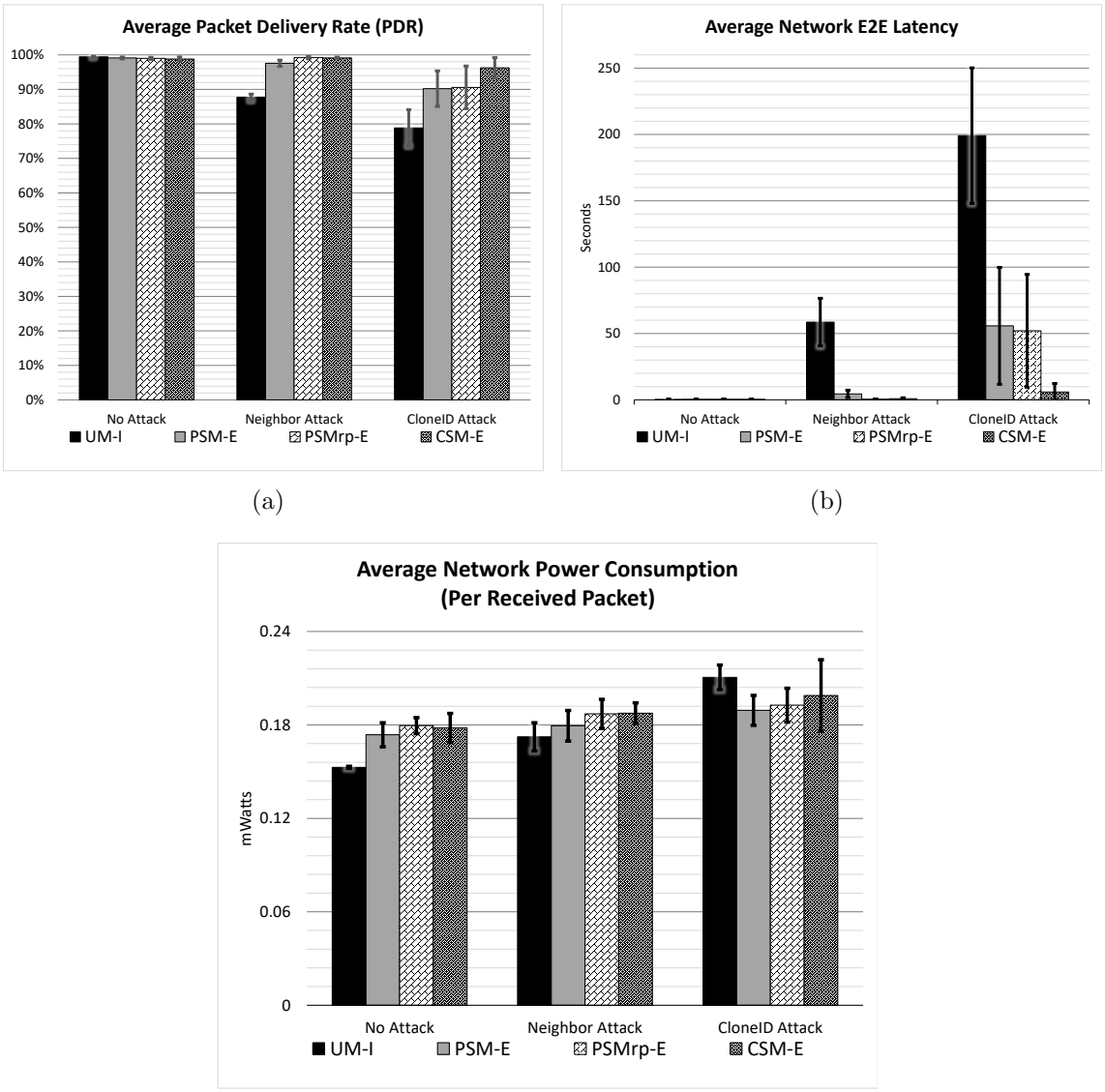


Figure 5.9: Simulation results for the four experiments (three attacks scenarios - external adversary), using ContikiMAC RDC protocol.

5.7.3 Effects on Power Consumption

The results for ContikiMAC (Fig. 5.9(c)) shows similar patterns among all the secure modes and for all attacks as a proof of significant attacks' mitigation. However, PSM, PSMrp, and CSM do have a slightly more power consumption than UM in the No Attack scenario, due to the additional security measures. This case is reversed for the CA attack scenario, as the UM was the only mode fully affected by the attack.

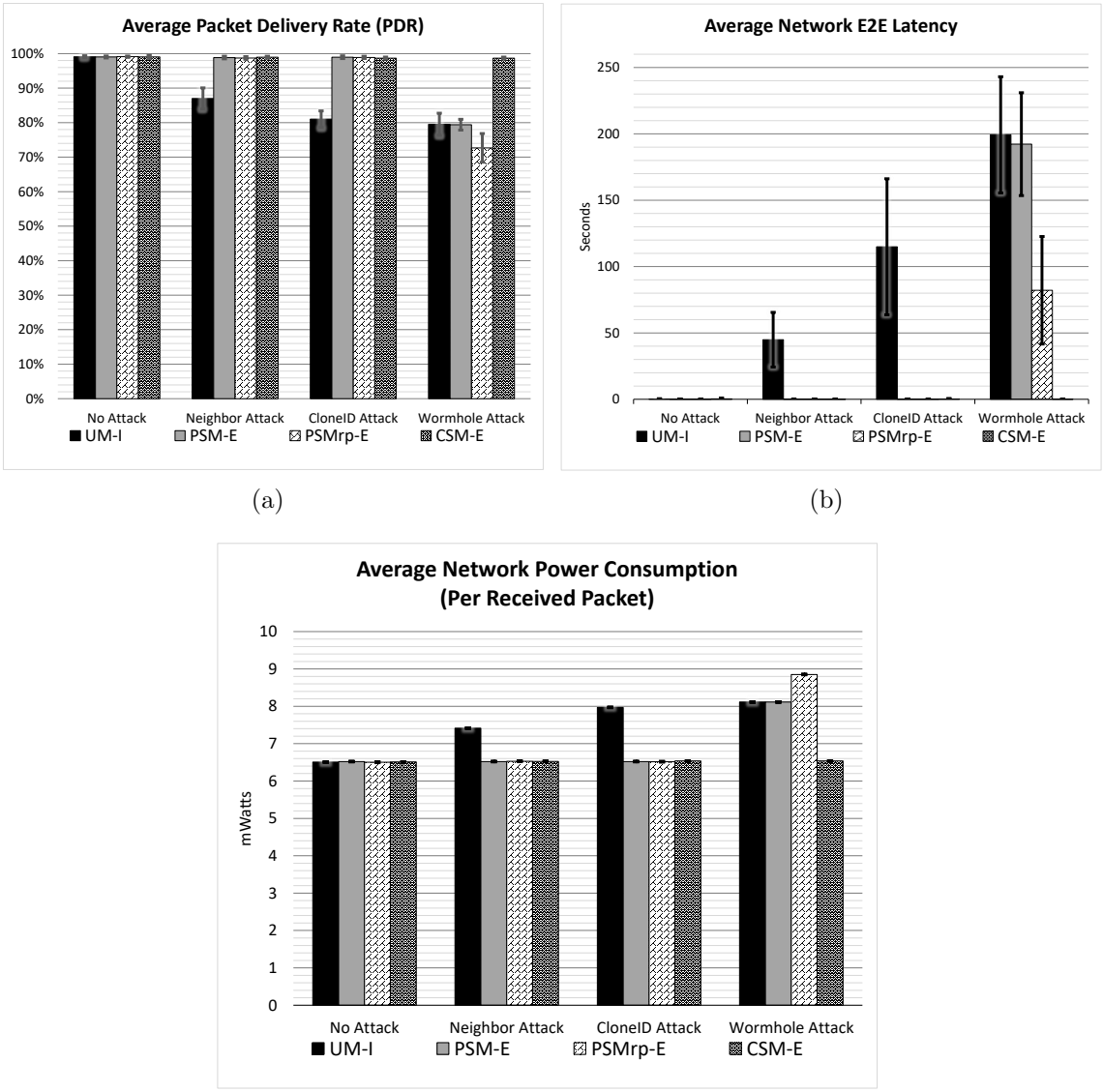


Figure 5.10: Simulation results for the four experiments (four attacks scenarios - external adversary), using NulRDC RDC protocol.

Fig. 5.10(c) show that all three secure modes have similar patterns for the NA and CA scenarios (significant mitigation of the attacks) under NullRDC. While, for the WH attack, CSM has the lowest power consumption (matching the No Attack scenario), due to the mitigation of the attack.

5.8 Discussion

The observations from the evaluation experiments can be summarized in the following:

5.8.1 Enhanced Security Features of CSM

Those can be summarized as follows:

i) CSM adds an extra layer of security by encoding the control messages and chaining them with the SC values, providing a means of immediate-sender authentication, which limits the adversaries' ability to eavesdrop on, manipulate, forge, and replay RPL control messages.

ii) Encoding of the *Code* field of the ICMPv6 header in CSM means that external adversaries cannot identify the type of RPL control messages by reading the ICMPv6 header, except for the first message of each message flow as it is currently encoded with zero - see Fig.5.4c. Hence, external replay attacks that target specific RPL control messages (e.g., NA) can be mitigated by using CSM.

iii) Unlike PSMrp, CSM provides mitigation to both "one-way" and "two-way" replay attacks (e.g., the NA and WH, respectively), due to the chaining of the control messages (by the SC values) that acts as an immediate-sender authentication mechanism, without the need for a challenge/response mechanism as in PSMrp.

Due to the characteristics of the intra-flow NC, there is one case where an internal adversary can launch one of the investigated attacks, which is if it was able to extract and track all the SC values from the exchanged RPL control messages between the adversary's neighbors. However, depending on the IoT network size, the location of the adversary, and the type of attack desired, tracking all the SC values for all the neighboring nodes communications (and in other parts of the network for a WH attack) would require a tremendous amount of resources (e.g., processing power, memory/storage capacity, and fast transceiver) to be available to the adversary.

5.8.2 CSM Reduction of the In-threat Period

The in-threat period for a replay-attack adversary can be defined as "*the time duration in which an adversary can overhear and understand the whole (or a part of)*

the exchanged RPL control messages and launch replay attacks". This period ranges between **zero** (*the adversary cannot launch attacks successfully*) to **infinity** (*the adversary can launch attacks at any time*), depending on the secure mode used, the adversary type, and the attack.

For UM, the in-threat period is always **infinity** as the adversary can understand RPL messages and launch attacks at any time. On the other hand, the in-threat period for PSM can be either:

- **Infinity** for all internal adversaries, or external adversaries of replay/identity-cloning attacks. The former can decrypt the whole control message with the preinstalled encryption key at any time, while the latter can identify RPL control messages from the "Type" and "Code" fields of the ICMPv6 header, then replay them at any other time without the need to decrypt the message contents.

- **Zero** for external adversaries of attacks that require a full understanding of RPL control messages, due to the lack of the used encryption key.

As CSM enhances RPL security through the intra-flow NC, it limits the adversaries' ability to launch several internal and external authentication-based attacks that require identifying and understanding RPL control messages. Hence, CSM significantly reduces the in-threat period to either:

- **The time between receiving the first MC and the first UC messages** for all internal adversaries. During this period, the adversary will try to intercept the first UC control message (which is encoded with zeros and has the SC values for both UC and MC flows), so it can use the included SC values to decode (then decrypt) the following message from any flow. However, the adversary needs to continuously intercept and decode all control messages from its victims in order to keep up with used SC values, which significantly raises the cost of any attack's launch.

- **Zero** for all external adversaries, due to the lack of both the used encryption key and the correct SC values. In addition, CSM's encoding of the "Code" field of the ICMPv6 header makes it harder to the adversary to identify the type of RPL control message; hence, more difficult to launch message-specific-targeted replay attacks.

To further reduce the in-threat period for CSM, RPL should be forced to send the first UC message as soon as it finishes processing the first MC message, which was done for the current prototype.

5.9 Summary

In this chapter, a novel secure mode for RPL was proposed as a framework, the Chained Secure Mode, to enhance RPL security and to build a mitigation capability against several authentication-based replay attacks into the protocol itself, without significantly changing the way RPL works. A prototype of CSM was devised, and its security and performance were evaluated against the currently implemented secure modes of RPL (UM, PSM, and PSMrp) under three replay routing attacks (NA, CA, and WH attack). It was shown that CSM successfully mitigated the replay attacks (NA and WH attack) while significantly reduced the effect of CA, all with latency and power consumption less than the other secure modes. Also, it was shown that CSM has a significantly smaller in-threat period than all other secure modes.

In addition, the ability to integrate external security mechanism opens the door to further enhance RPL security through future expansions. To demonstrate one of the possibilities to use this integration capability, the next chapter shows a use case of the CSM framework where it is used to mitigate the external adversaries of *buffer-reservation* attacks (see Chap. 4) against the 6LoWPAN adaptation layer, providing a mean of chained trust effect as another method of immediate-sender authentication for the 6LoWPAN layer.

Chapter 6

Using *CSM-Trust* Interface: Integrating 6LoWPAN's security with RPL in CSM

6.1 Introduction

The CSM framework introduced in Chap. 5 was proved to have excellent mitigation capabilities against authentication-based replay attacks (e.g., NA, CA, and WH attacks) - see Sec. 5.5. In addition, its integration capability with external security mechanisms was also put to test using a simple trust-based external mechanism, see Sec. 5.5.1, showing great possibilities for more use cases.

In this chapter, a security integration case for 6LoWPAN Adaptation Layer is proposed. As has been seen in Chap. 4, the 6LoWPAN protocol suffers from the same vulnerability as RPL, the lack of the sender's authentication for the fragments. By using the CSM framework and integrating a suitable trust-based external security mechanism (to provide a root of trust), the 6LoWPAN protocol can use the generated chain of trust to accept (or drop) fragments into the node's assembly buffer. The focus here is on mitigating the *buffer-reservation* attacks - see Sec. 4.2.4.

The rest of the chapter goes as follows: The concept behind the security integration case and its demonstration are discussed in section 6.2. The evaluation setup, assumptions, and adversary model are explained in section 6.3. Evaluation results are analyzed and discussed in section 6.4. Finally, the chapter is concluded in section 6.5.

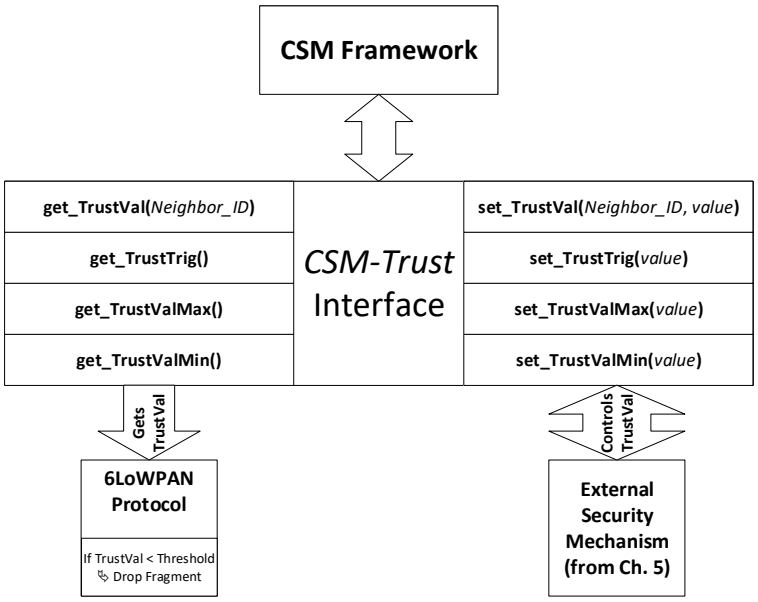


Figure 6.1: CSM-6LoWPAN integration concept diagram.

6.2 The Concept Behind The Security Integration Case

The CSM framework provides the external security mechanisms with an integration option through the *CSM-Trust* interface (see Sec. 5.4.2). The external mechanism can use its own methods to define and calculate the trustworthiness of the node's neighbors and set the *TrustVal* value accordingly; hence, controlling the acceptance of control messages from the node's immediate neighbors. In general, the *TrustVal* is used by RPL in CSM to secure the control plane of the routing topology.

On the other hand, the same interface can also be used, by other protocols, to read the *TrustVal* value and employ it for decision making, and this is what the work in this chapter is intended for: to use the CSM framework *CSM-Trust* interface to control the fragments traffic at the 6LoWPAN Adaptation Layer, i.e., the data plane of the Network Layer.

The general concept for the security integration case (see Fig. 6.1) is that if each node trusts its neighbors through the CSM framework at the control plane, then the nodes on the routing path for a fragmented data packet (at the data plane) can also be trusted due to the *chain of trust*. In other words, if the first routing node trusts the original sender of the fragments, and the second routing node trusts the first routing

node, and so for the remaining routing nodes until the destination node, then the whole path is considered secure.

Based on Sec. 4.2.2, it is clear that the *chain-of-trust* concept used in this security integration case is only applicable for the Route-Over and Enhanced-Route-Over methods, as CSM can be only used wherever RPL is available. The following subsection describes the implementation of the security integration case.

6.2.1 The Security Integration Case Implementation

As can be seen from Sec. 4.2.4; the buffer-reservation attack still presents a significant threat to 6LoWPAN networks due to the lack of immediate sender authentication. Hence, the goal of this security integration case is to mitigate such attacks from external adversaries[1] on the 6LoWPAN Adaptation Layer.

As a proof-of-concept demonstration for the security integration case, the implementation in this chapter will use the same simple external security measure from the previous chapter (see Sec. 5.5.1). In other words, the external security mechanism will control the *TrustVal* value, while the 6LoWPAN protocol will use the *TrustVal* value to decide on admitting the received fragment to the assembly buffer or not, all by using the *CSM-Trust* interface of the CSM framework.

Other worth-mentioning implementation points are:

- The decisions of the 6LoWPAN protocol are based on the trust of the immediate sender of the fragment, not the original sender, fulfilling the concept of *chain-of-trust*. The protocol will use the Link Layer sender address from the frame containing the fragment and find the associated IPv6 address. Then, it will use the found IPv6 address to get the corresponding TrustVal from the *CSM-Trust* interface of the CSM framework.

- Similar to CSM, the 6LoWPAN protocol will have its own trust threshold to make its decisions, the (6LOWPAN_TRUST_THRESHOLD). For demonstration purposes and to differentiate it from the CSM threshold (*TrustTrig*), it was set to 60.

- Currently, the Contiki OS implementation of 6LoWPAN supports only the Route-Over method [8].

[1] For more details about external and internal types of adversaries, see Sec. 5.2.

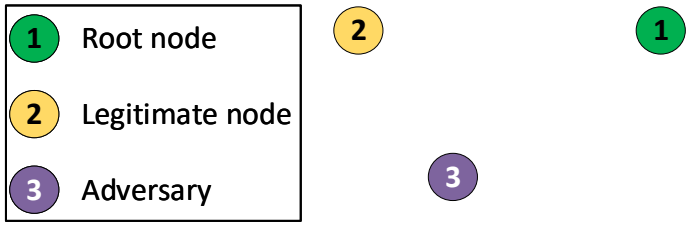


Figure 6.2: Network topology used for the evaluation.

Table 6.1: List of Simulation Parameters

Description	Value
No. of experiments	Two: vanilla 6LoWPAN and CSM-6LoWPAN
No. of scenarios per experiment	Nine - See Sec. 6.3.2
Sim. rounds per scenario / time	10 rounds / 20 min. per round
Sensor nodes type	Arago Sys. Wismote mote

6.3 Evaluation of The Security Integration Case

To evaluate the proposed security integration case, a comparison of security and performance was conducted between the Contiki OS implementation of 6LoWPAN protocol (vanilla 6LoWPAN) and the 6LoWPAN with CSM framework integration (CSM-6LoWPAN for short). Both protocols were tested against an external adversary of the buffer-reservation attack in several scenarios.

6.3.1 Evaluation Setup and Assumptions

As with most of the evaluations in this dissertation (see Chap. 3 and 5), Cooja, the simulator for Contiki OS [8], was used for all the simulations (with simulated motes). Fig. 6.2 shows the topology used in the evaluation, which is a widely used topology for 6LoWPAN Adaptation Layer evaluation [119, 125, 127, 132]. A list of simulation parameters is provided in Table 6.1.

For the purpose of the evaluation, two metrics were used: the average data PDR and the average power consumption for the legitimate sending node (per received data packet).

The following assumptions were considered in the evaluation: both the legitimate

Table 6.2: Summary of the simulation scenarios

	Adversary sends...			
	Full Packets (Normal DoS)	1st Fragment Only (Basic Buffer-Resrv. Attack)	All Except Last Fragment (Soph. Buffer-Resrv. Attack)	
Attack Launch Timing	Before Legitimate Node	Before Legitimate Node	Before Legitimate Node	No Attack Scenario
	Simultaneously with Legitimate Node	Simultaneously with Legitimate Node	Simultaneously with Legitimate Node	
	After Legitimate Node	After Legitimate Node	After Legitimate Node	

node and the adversary sends (512 bytes) data packets toward the root at a rate of 1 packet/minute per node. However, the adversary follows the attack scenarios as described in Sec. 6.3.2.

For the legitimate node, RPL is set to operate in UM for the vanilla 6LoWPAN experiment and in CSM for the CSM-6LoWPAN experiment. In both cases, RPL is set with the default OF, namely the MRHOF [35]. Contiki OS is using the following settings for its uIP stack: IEEE 802.15.4 [20] for the Physical Layer and MAC sub-layer, ContikiMAC [110] for the RDC sub-layer, IPv6, 6LoWPAN, and RPL at the Network Layer, and UDP for the Transport Layer. In addition, all security measures and encryption at the Link Layer were assumed to be disabled.

For the 6LoWPAN protocol, the default Contiki reassembly timeout was used (20 seconds) and the max size for the fragment payload is set to 102 bytes (as per the standard [17]). In addition, fragment forwarding follows the Route-Over approach, as it is the only currently available option in Contiki OS.

Finally, the results obtained from the simulations were averaged over ten rounds per experiment with a 95% confidence level.

6.3.2 Adversary Model and Attack Scenarios

For the adversary, it runs in the same RPL secure mode as the legitimate nodes. However, as an external adversary, it does not have the required encryption key (for the CSM-6LoWPAN experiment). Also, it is worth mentioning that there are no differences between an external and internal adversary from the UM point of view (the vanilla 6LoWPAN experiment).

In all cases, the adversary starts as a legitimate node, tries to join the network, then launches the attack after 50 seconds. This is to allow the network to reach the steady-state situation.

For the buffer-reservation attack, the same scenarios in Sec. 4.5 are used for this evaluation in addition to one *No Attack* scenario for each experiment. In total, there are ten scenarios for each experiment as summarized in Table 6.2.

It is worth mentioning that the adversary and the legitimate node send their fragments at their designated time ± 2 seconds, due to the randomness nature of the simulation, which mimics the real-life situation.

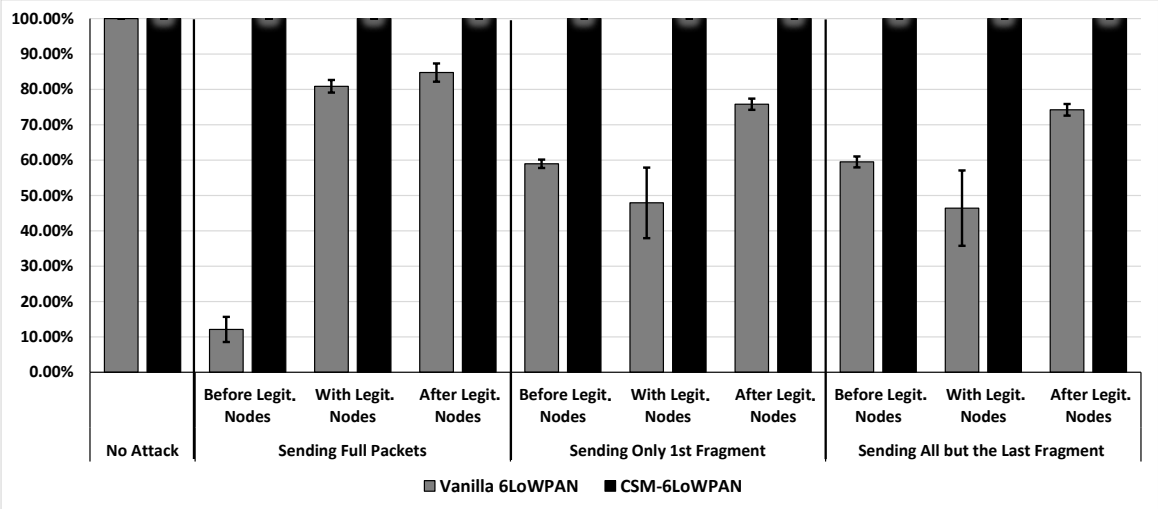
6.4 Results Analysis and Observations

The simulations results are shown in Fig. 6.3, expressed as the average PDR and average power consumption for the legitimate sending node (per received data packet).

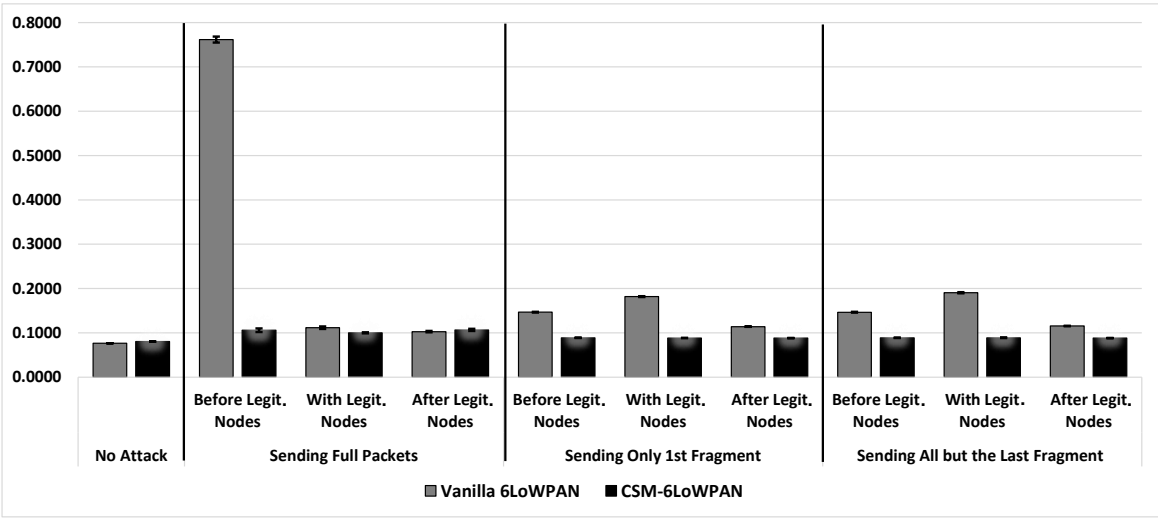
6.4.1 Results Analysis

Looking at Fig. 6.3(a) for the PDR, it is clear that CSM-6LoWPAN mitigated the attack successfully in all the different scenarios (PDR ≈ 100%). For the vanilla 6LoWPAN, the attack was more successful when the adversary preceded the legitimate node (PDR ≈10–60%) as the adversary reserved the buffer before the legitimate node. In addition, sending one or a few fragments was slightly more effective than sending full packets (PDR ≈60–75%, compared to ≈80–85%), except for the case of the adversary sending fragments before the legitimate node, where the attack was more successful when the adversary sent full packets (PDR ≈13%). However, the reason for the later case seems to be mostly due to the simulation randomness irregularities; as the logs of the ten rounds suggests that the adversary always preceded the legitimate node by enough time to guarantee the buffer reservation.

Moving to the average power consumption readings in Fig. 6.3(b), it can be seen that CSM-6LoWPAN did not introduce any additional power consumption to the legitimate node while maintaining a higher level of security (successful mitigation of the buffer-reservation attack from an external adversary), compared to the vanilla 6LoWPAN experiment (around 0.1 milliwatt for CSM-6LoWPAN, compared t 0.15–0.20 milliwatt for the vanilla 6LoWPAN.)



(a) Average Data Packet Delivery rate (PDR). The less the PDR, the more successful the attack.



(b) Average power consumption for the legitimate sending node (per received data packet). Readings are in milliwatts

Figure 6.3: Simulation results for the two experiments (vanilla 6LoWPAN and CSM-6LoWPAN.)

6.4.2 Discussion

The observations from the evaluation experiments can be summarized in the following:

- When the *CSM-Trust* interface is used to provide trust measurements from RPL to other protocols (in this demonstration, the 6LoWPAN protocol), the mitigation depends on implementing a suitable external security mechanism that integrates into the framework and provide the required trust metric.

- In this chapter's demonstration, the integration of the simple external security mechanism we used for the CSM framework (see Chap. 5) was able to mitigate the external adversaries of the simple *buffer-reservation* attacks at 6LoWPAN Adaptation Layer.
  - However, since the mechanism does not have global view of the network nor consider the behavior patterns of the neighbors, it doesn't provide mitigation capability against the internal adversaries of the *buffer-reservation* attacks.
  - In addition, a more sophisticated adversary can still launch a *buffer-reservation* attack if the adversary used the Link-Layer address of the victim node(s).
- The security integration case showed in this chapter demonstrated the many possibilities CSM frame brings to IoT networks through the *CSM-Trust* interface, allowing integration not only between one external security measure and RPL, but among several security measures at once. In this demonstration, the integration was between RPL, the external security mechanism, and 6LoWPAN.
- The security integration case also shows that the immediate-sender authentication vulnerability in 6LoWPAN Adaptation Layer can be patched, through the integration with CSM framework, without heavily taxing the limited resources of the IoT devices, unlike many of the currently proposed solutions [127, 130, 140].

6.5 Summary

In this chapter, a security integration case of the CSM framework was demonstrated, where the security integration between the 6LoWPAN protocol, RPL, and an external security mechanism, through the *CSM-Trust* interface of the CSM framework, could provide better security to the whole IoT network. The demonstrated case proposed a security integration between 6LoWPAN, the simple external routing-security mechanism used in Chap. 5, and CSM framework as a solution to the immediate-sender authentication vulnerability discussed in Chap. 4. The evaluation of this security integration case showed the potentials of the CSM framework, as the simple security

integration between the three components (6LoWPAN protocol, RPL, and the external routing-security mechanism) was able to mitigate the external adversaries of a simple buffer-reservation attacks at 6LoWPAN Adaptation Layer.

Finally, this security integration case also shows that, depending on the integrated security mechanism and using CSM framework, a solution to the above-mentioned vulnerability is possible without implementing resource-exhausting techniques, such as the public-private keys or new protocols. However, the requirements and the design of such a security mechanism would be a subject of further research, as it depends on the security requirements of the deployed network and the types of attacks it is susceptible to.

Chapter 7

Conclusions and Future Work

As the IoT enters new areas every day, security becomes important and an essential requirement in their deployment. As was discussed through this dissertation, the security of the routing and forwarding processes in IoT is a fundamental part of IoT security. Many current IoT networks use the uIP protocol stack, in which the two protocols under consideration in this dissertation operates: the RPL for the routing process at the Network layer and 6LoWPAN protocol at the 6LoWPAN Adaptation sub-layer for the forwarding and adaptation of IPv6 packets to the smaller IEEE 802.15.4 frames.

An in-depth thorough exploration of RPL and its common attacks was conducted, in which a novel classification for the mitigation methods for the surveyed attacks was introduced, and an extensive survey on the current IDS proposals for RPL was provided. 6LoWPAN was also investigated with great details along with its common attacks, and some of the proposed solutions for these attacks were examined.

Next, this dissertation moved to evaluate the security and performance of both protocols under several common attacks. In the case of RPL, the work presented a first-of-its-kind evaluation of RPL's implemented secure modes, the UM, PSM, and PSMrp, under common routing attacks. For the 6LoWPAN protocol, the work evaluated the security and performance of the protocol under the *buffer-reservation* attacks since it is still an active area of research. The results of both evaluations confirmed a significant vulnerability in both protocols: the lack of immediate-sender authentication. As proved by this dissertation, this vulnerability may allow adversaries to lunch authentication-based attacks, such as replay attacks in RPL or *buffer-reservation* attacks in the 6LoWPAN Adaptation Layer, even with external adversaries. Such attacks can have devastating effects on the IoT network, e.g., a DoS attack, long

E2E delays for the data packets, routing topology disruptions, resource exhaustion attacks, and much more.

Finding a suitable solution for this vulnerability was the main motivation behind the work in this dissertation, and NC has been proposed a suitable solution. Based on the intra-flow NC concepts, a novel security framework was proposed as a new secure mode for RPL, the CSM framework. The new secure mode divides RPL's control messages into two intra-flows: UC-flow and MC-flow. Then, it *encodes* the already-encrypted RPL control messages with SC values that change with every transmission per flow. The SC values are sent as part of the encoded (and encrypted) control messages through the standard-compliant RPL options. To recover from lost-control-message situations, CSM has a recovery mechanism that uses challenge-response message exchange through another intra-flow, the ER-flow, which uses its own SC values. In addition, a trust-based integration interface, the *CSM-Trust* interface, allows external security measures to control the acceptance (or rejection) of RPL control messages from a node's neighbors. The same interface also allows other protocols or mechanisms to read this trust value of the neighbors and use them to make security decisions. This integration interface is also the first-of-its-kind in such a security proposal, to the best of my knowledge.

A comprehensive security and performance evaluation of the new framework was conducted against several types of replay attacks, and the results were compared with the ones from the currently implemented secure modes of RPL. The framework showed excellent mitigation capabilities against the internal and external adversaries of investigated attacks (i.e., the NA, CA, and WH attacks) with either less or similar power consumption to the one from PSM, which do not provide mitigation against the investigated attacks. Also, a simple trust-based security mechanism was used to test the *CSM-Trust* interface. The evaluation proved, through the results of integrating a simple proof-of-concept, the countless possibilities of such integration between RPL and external security mechanisms such as IDSs.

As a demonstration of the *CSM-Trust* interface, an integration between the CSM framework, the proof-of-concept security mechanism used to evaluate CSM, and the 6LoWPAN protocol was proposed to protect the 6LoWPAN Adaptation Layer from the external adversaries of the *buffer-reservation* attacks. The concept is to have the external security mechanism control the trust of the node's neighbors through the *CSM-Trust* interface of the CSM framework. Simultaneously, the 6LoWPAN protocol

uses the same interface to get the trust values of the neighbors and make decisions on admitting (or dropping) the received fragments from that neighbor (regardless of the original sender). Since each node is responsible for trusting its neighbors through the Network Layer's control plane, the process will create a *chain-of-trust* along the routing path of the fragments on the Network Layer's data plane. The evaluation of this security integration case showed a mitigation of the investigated attack when an external adversary is used, with equivalent power consumption to the No Attack scenario for the vanilla 6LoWPAN protocol and using RPL in UM.

Future Work

Based on the evaluations of the CSM framework (in Chap. 5 and Chap. 6), the work presented in this dissertation can be considered beneficial for the real world. However, as with any new proposals, more testing and scrutiny for its security and performance are needed before announced ready for real-world deployments. In addition, the following lists few open opportunities to enhance the work or build upon its proposals:

- **Adding support for mobility in the CSM framework:** Currently, the CSM supports only static networks due to the way intra-flow NC works. One way to support mobile nodes in CSM would be to use of the *CSM-Trust* interface and a suitable external mechanism that tracks the mobile nodes and automatically populate the required SC Table entries.

- **Supplement CSM with enhanced compatibility mode:** In its current implementation, CSM does not allow nodes operating in other secure modes of RPL to stay in the network. A proposal would add a compatibility mode similar to the one stated in the RPL standard [13] for the ASM. In this compatibility mode, nodes operating in PSM (and possibly, ASM) can join the network as leaf nodes only. This will maintain the enhanced security of CSM while allowing the nodes to send their data to the root.

- **Developing a behavioral-based security mechanism to integrate with the CSM framework:** The goal of such a mechanism would be to provide more accurate and network-global trust measurements to RPL, which will be accessible to the other protocols (e.g., 6LoWPAN) through the CSM-Trust interface. Hence, the mechanism will allow CSM to mitigate different types of attacks and across several layers.

Bibliography

[1] Cisco, "Cisco Annual Internet Report (2018–2023)," *Cisco*, pp. 1–41, 2020.

[2] A. Raoof, A. Matrawy, and C.-H. Lung, "Enhancing Routing Security in IoT: Performance Evaluation of RPL Secure Mode under Attacks," *IEEE Internet of Things Journal*, vol. 7, pp. 11536 – 11546, Dec 2020.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Info. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[4] R. H. Patel, "Locating the Attacker of Wormhole Attack on Rpl in Iot," Master's thesis, Gujarat Technological University, Ahmedabad, India, 2016.

[5] P. Pongle and G. Chavan, "Real Time Intrusion and Wormhole Attack Detection in Internet of Things," *International Journal of Computer Applications*, vol. 121, no. 9, pp. 1–9, 2015.

[6] J. Hansen *et al.*, "Bridging Inter-Flow and Intra-Flow Network Coding in Wireless Mesh Networks: From Theory to Implementation," *Computer Networks*, vol. 145, pp. 1–12, 2018.

[7] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proc. of 31st IEEE Conference on Local Computer Networks*, pp. 641–648, IEEE, 2006.

[8] A. Dunkels *et al.*, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," in *29th IEEE Int'l Conf' on Local Computer Networks*, Nov 2004.

[9] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*, pp. 15–34. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[10] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing Attacks and Mitigation Methods for RPL-Based Internet of Things," *IEEE Comm. Surveys and Tutorials*, vol. 21, pp. 1582–1606, 2nd Quarter 2019.

[11] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," May 2015.

[12] M. Sain, Y. J. Kang, and H. J. Lee, "Survey on security in Internet of Things: State of the art and challenges," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 699–704, 2017.

[13] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks," RFC 6550, RFC Editor, March 2012.

[14] L. Wallgren *et al.*, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *Int'l Journal of Dist. Sensor Net.*, vol. 2013, 2013.

[15] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[16] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." RFC 8200, July 2017.

[17] G. Montenegro, J. Hui, D. Culler, and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks." RFC 4944, September 2007.

[18] P. Thubert and J. Hui, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks." RFC 6282, September 2011.

[19] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)." RFC 7252, June 2014.

[20] IEEE, *IEEE Standard for Low-Rate Wireless Networks (802.15.4-2015)*, April 2015.

[21] H. Singh and D. Singh, "Taxonomy of Routing Protocols in Wireless Sensor Networks: A Survey," in *2016 2nd International Conference on Contemporary Computing and Informatics*, pp. 822–830, 2016.

[22] J. Granjal *et al.*, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Comm. Surveys and Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[23] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[24] D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure Routing for Internet of Things: A Survey," *Journal of Network and Computer Applications*, vol. 66, pp. 198–213, 2016.

[25] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *Journal of Network and Computer Applications*, vol. 88, no. December 2016, pp. 10–28, 2017.

[26] Cenk Gündoğan and Dominique Barthel and Emmanuel Baccelli, "DIS Modifications," Internet-Draft draft-ietf-roll-dis-modifications-00, Internet Engineering Task Force, March 2017. Work in Progress-Expired.

[27] J. Kumari and Prachi, "A Comprehensive Survey of Routing Protocols in Wireless Sensor Networks," in *2nd International Conference on Computing for Sustainable Global Development (INDIACom 2015)*, pp. 325 – 330, IEEE, 2015.

[28] N. Janicijevic *et al.*, "Routing Protocol for Low-power and Lossy Wireless Sensor Networks," in *19th TELFOR 2011*, 2011.

[29] J. Vasseur, D. Barthel, K. Pister, M. Kim, and N. Dejean, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks." RFC 6551, March 2012.

[30] T. Watteyne *et al.*, "Routing Requirements for Urban Low-Power and Lossy Networks." RFC 5548, May 2009.

[31] J. Martocci *et al.*, "Building Automation Routing Requirements in Low-Power and Lossy Networks." RFC 5867, June 2010.

[32] K. Pister *et al.*, "Industrial Routing Requirements in Low-Power and Lossy Networks." RFC 5673, October 2009.

[33] G. Porcu *et al.*, "Home Automation Routing Requirements in Low-Power and Lossy Networks." RFC 5826, April 2010.

[34] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)." RFC 6552, March 2012.

[35] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function." RFC 6719, September 2012.

[36] P. Levis and O. Gnawali, "The ETX Objective Function for RPL," Internet-Draft draft-gnawali-roll-etxof-01, Internet Engineering Task Force, May 2010. Work in Progress (Expired on Nov 2010).

[37] M. Qasem, A. Al-Dubai, I. Romdhani, B. Ghaleb, J. Hou, and R. A. JADHAV, "Load Balancing Objective Function in RPL," Internet-Draft draft-qasem-roll-rpl-load-balancing-02, Internet Engineering Task Force, October 2017. Work in Progress.

[38] M. Qasem, A. Al-Dubai, I. Romdhani, B. Ghaleb, and W. Gharibi, "A New Efficient Objective Function for Routing in Internet of Things Paradigm," in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–6, IEEE, October 2016.

[39] R.-A. Koutsiamanis *et al.*, "Traffic-aware Objective Function," internet-draft, IETF, Mar 2019. Work in Progress.

[40] P. Levis, T. H. Clausen, O. Gnawali, J. Hui, and J. Ko, "The Trickle Algorithm." RFC 6206, March 2011.

[41] D. Culler *et al.*, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)." RFC 6554, March 2012.

[42] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The Impact of Rank Attack on Network Topology of Routing Protocol for Low-power and Lossy Networks," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3685–3692, 2013.

[43] J. Hui and J. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams." RFC 6553, March 2012.

[44] A. Kamble, V. S. Malemath, and D. Patil, "Security Attacks and Secure Routing Protocols in RPL-based Internet of Things:Survey," in *International Conference on Emerging Trends & Innovation in ICT (ICEI)*, (Pune, India), pp. 33–39, 2017.

[45] P. Levis *et al.*, "Tinyos: An operating system for sensor networks," in *Ambient Intelligence*, pp. 115–148, Berlin, Germany: Springer, 2005.

[46] P. Perazzo *et al.*, "An Implementation and Evaluation of the Security Features of RPL," in *16th Int'l Conf. on Ad Hoc Networks and Wireless*, vol. 10517, pp. 63–76, Springer Int'l Pub., 2017.

[47] A. Arena, P. Perazzo, C. Vallati, G. Dini, and G. Anastasi, "Evaluating and Improving the Scalability of RPL Security in the Internet of Things," *Computer Commun.*, vol. 151, 2020.

[48] Y.-C. Hu, a. Perrig, and D. Johnson, "Packet Leashes: A Defense Against Wormhole Attacks in Wireless Networks," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1976–1986, March 2003.

[49] M. Surendar and A. Umamakeswari, "InDReS: An Intrusion Detection and response system for Internet of Things with 6LoWPAN," in *2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, pp. 1903–1908, 2016.

[50] G. Glissa, A. Rachedi, and A. Meddeb, "A Secure Routing Protocol Based on RPL for Internet of Things," in *IEEE Global Commun. Conf. (GLOBECOM 2016)*, 2016.

[51] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A Survey of Intrusion Detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, April 2017.

[52] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *International Conference on Pervasive Computing (ICPC 2015)*, pp. 1–6, January 2015.

[53] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A Specification-based IDS for Detecting Attacks on RPL-based Network Topology," *Information (Switzerland)*, vol. 7, no. 2, 2016.

[54] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of Sinkhole Attacks for Supporting Secure Routing on 6LoWPAN for Internet of Things," in *2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, pp. 606–611, 2015.

[55] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time Intrusion Detection in The Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[56] D. Shreenivas and E. Ab, "Intrusion Detection in the RPL-connected 6LoWPAN Networks," in *3rd ACM International Workshop on IoT Privacy, Trust, and Security - IoTPTS '17*, pp. 31–38, 2017.

[57] M. Alzubaidi, M. Anbar, and S. M. Hanshi, "Neighbor-Passive Monitoring Technique for Detecting Sinkhole Attacks in RPL Networks," in *Proc. of the Int'l Conf. on Computer Science and Artif. Intell. (CSAI 2017)*, ACM Press, 2017.

[58] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Distributed Monitoring Strategy for Detecting Version Number Attacks in RPL-based Networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 472–486, 2017.

[59] H. Sedjelmaci, S.-m. Senouci, and T. Taleb, "An Accurate Security Game for Low-Resource IoT Devices," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 9381 – 9393, October 2017.

[60] D. Siegmund, *The Sequential Probability Ratio Test*, pp. 8–33. New York, NY: Springer New York, 1985.

[61] F. Gara, L. Ben Saad, and R. Ben Ayed, "An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs," *13th Int'l Wireless Commun. and Mobile Computing Conf. (IWCMC 2017)*, 2017.

[62] A. Kumar *et al.*, "Impact of Packet Dropping Attacks on RPL," in *4th Int'l Conf. on Parallel, Dist. and Grid Computing*, pp. 694–698, 2016.

[63] K. Weekly and K. Pister, "Evaluating Sinkhole Defense Techniques in RPL Networks," in *International Conference on Network Protocols, ICNP*, 2012.

[64] D. Airehrour *et al.*, "Securing RPL Routing Protocol From Blackhole Attacks Using A Trust-based Mechanism," in *26th Int'l Telecomm. Networks and Applications Conference, ITNAC 2016*, 2016.

[65] Z. A. Khan and P. Herrmann, "A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things," in *IEEE 31st International Conference on Advanced Information Networking and Applications 2017*, pp. 1169–1176, 2017.

[66] N. Djedjig, D. Tandjaoui, and F. Medjek, "Trust-based RPL for the Internet of Things," in *IEEE Symposium on Computers and Communications*, pp. 962–967, February 2016.

[67] N. Djedjig *et al.*, "New Trust Metric for the RPL Routing Protocol," in *8th Int'l Conf. on Inform. and Comm. Sys. (ICICS 2017)*, IEEE, 2017.

[68] P. Perazzo *et al.*, "Implementation of a Wormhole Attack against a RPL Network: Challenges and Effects," in *14th Conf. on Wireless On-demand Net. Sys. and Services (WONS 2018)*, pp. 95–102, IEEE, Feb 2018.

[69] S. Mukherjee, M. Chattopadhyay, S. Chattopadhyay, and P. Kar, "Wormhole Detection Based on Ordinal MDS Using RTT in Wireless Sensor Network," *Journal of Computer Networks and Communications*, vol. 2016, p. 15, 2016.

[70] V. K. Raju and K. V. Kumar, "A Simple and Efficient Mechanism to Detect and Avoid Wormhole Attacks in Mobile ad hoc Networks," in *International Conference on Computing Sciences, ICCS 2012*, pp. 271–275, 2012.

[71] F. I. Khan, T. Shon, T. Lee, and K. Kim, "Wormhole Attack Prevention Mechanism for RPL Based LLN Network," in *International Conference on Ubiquitous and Future Networks, ICUFN*, pp. 149–154, 2013.

[72] G.-H. Lai, "Detection of Wormhole Attacks on IPv6 Mobility-based Wireless Sensor Network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, p. 11, November 2016.

[73] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil Attacks and Their Defenses in the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[74] F. Medjek, D. Tandjaoui, M. R. Abdmeziem, and N. Djedjig, "Analytical Evaluation of The Impacts of Sybil Attacks Against RPL Under Mobility," in *12th International Symposium on Programming and Systems, ISPS 2015*, pp. 13–21, 2015.

[75] L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," *IEEE Communications Letters*, vol. 15, pp. 1193–1195, November 2011.

[76] A. Rehman, M. M. Khan, M. A. Lodhi, and F. B. Hussain, "Rank Attack Using Objective Function in RPL for Low Power and Lossy Networks," in *International Conference on Industrial Informatics and Computer Systems, CIICS 2016*, 2016.

[77] K. K. Rai and K. Asawa, "Impact analysis of rank attack with spoofed IP on routing in 6LoWPAN network," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, pp. 1–5, IEEE, August 2017.

[78] R. Sahay, G. Geethakumari, and K. Modugu, "Attack graph - Based vulnerability assessment of rank property in RPL-6LOWPAN in IoT," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 308–313, IEEE, February 2018.

[79] H. Perrey, M. Landsmann, O. Ugus, T. C. Schmidt, and M. Wählisch, "TRAIL: Topology Authentication in RPL," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 73–74, 2013.

[80] H. Perrey, O. Ugus, W. Matthias, and T. C. Schmidt, "TRAIL : Topology Authentication in RPL," in *Int'l Conf. on Embedded Wireless Sys. and Networks*, 2016.

[81] A. Dvir, T. Holczer, and L. Buttyan, "VeRA - Version Number and Rank Authentication in RPL," in *8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011*, pp. 709–714, 2011.

[82] K. Iuchi, T. Matsunaga, K. Toyoda, and I. Sasase, "Secure Parent Node Selection Scheme in Route Construction to Exclude Attacking Nodes from RPL Network," in *2015 21st Asia-Pacific Conference on Communications, APCC 2015*, pp. 299–303, 2015.

[83] A. Aris, S. F. Oktug, and S. Berna Ors Yalcin, "RPL Version Number Attacks: In-depth Study," in *IEEE/IFIP Netw. Oper. and Manage. Symp. (NOMS-2016)*, 2016.

[84] A. Le *et al.*, "The Impacts of Internal Threats Towards Routing Protocol for Low power and lossy network Performance," in *Int'l Symp. on Comp. and Comm. (ICC 2013)*, pp. 789–794, Jul 2013.

[85] H. L. Nguyen and U. T. Nguyen, "A study of different types of attacks in mobile ad hoc networks," in *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–6, April 2012.

[86] D. Sharma, I. Mishra, and S. Jain, "A Detailed Classification of Routing Attacks against RPL in Internet of Things," *International Journal of Advance Research Ideas and Innovations in Technology*, vol. 3, no. 1, pp. 692–703, 2017.

[87] C. Pu, "Mitigating DAO inconsistency attack in RPL-based low power and lossy networks," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 570–574, IEEE, January 2018.

[88] A. Sehgal, A. Mayzaud, R. Badonnel, I. Chrisment, and J. Schönwälder, "Addressing DODAG inconsistency attacks in RPL networks," in *2014 Global Information Infrastructure and Networking Symposium, GIIS 2014*, 2014.

[89] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "Mitigation of Topological Inconsistency Attacks in RPL-based Low-power Lossy Networks," *International Journal of Network Management*, vol. 25, pp. 320–339, September 2015.

[90] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO Suppression Attack Against Routing in The Internet of Things," *IEEE Communications Letters*, vol. 21, pp. 2524–2527, November 2017.

[91] X. Liu, Z. Sheng, C. Yin, F. Ali, and D. Roggen, "Performance Analysis of Routing Protocol for Low Power and Lossy Networks (RPL) in Large Scale Networks," *IEEE Internet of Things Journal*, vol. 4, pp. 2172–2185, December 2017.

[92] A. Abdou, A. Matrawy, and P. C. van Oorschot, "Accurate Manipulation of Delay-based Internet Geolocation," in *ACM on Asia Conference on Computer and Communications Security 2017*, ASIA CCS '17, pp. 887–898, ACM, April 2017.

[93] A. Abdou, A. Matrawy, and P. C. van Oorschot, "CPV: Delay-Based Location Verification for the Internet," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, pp. 130–144, March 2017.

[94] M. Conti, P. Kaliyar, and C. Lal, "REMI: A Reliable and Secure Multicast Routing Protocol for IoT Networks," in *12th International Conference on Availability, Reliability and Security - ARES '17*, pp. 1–8, ACM Press, 2017.

[95] P. Thubert, "RPL-BIER," Internet-Draft draft-thubert-roll-bier-01, Internet Engineering Task Force, January 2018. Work in Progress.

[96] O. Bergmann, C. Bormann, S. Gerdes, and H. Chen, "Constrained-Cast: Source-Routed Multicast for RPL," Internet Draft draft-ietf-roll-ccast-01, IETF, October 2017. Work in Progress.

[97] J. Hui and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)." RFC 7731, February 2016.

[98] H. S. Kim, J. G. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.

[99] A. I. Sabbah, A. El-Mougy, and M. Ibnkahla, "A Survey of Networking Challenges and Routing Protocols in Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 210–221, February 2014.

[100] A. A. Khan, M. H. Rehmani, and M. Reisslein, "Requirements, Design Challenges, and Review of Routing and MAC Protocols for CR-Based Smart Grid Systems," *IEEE Communications Magazine*, vol. 55, pp. 206–215, May 2017.

[101] J. R. R. Renofio, M. E. Pellenz, E. Jamhour, A. Santin, M. C. Penna, and R. D. Souza, "On the dynamics of the RPL protocol in AMI networks under jamming attacks," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, May 2016.

[102] R. Bruzgiene, L. Narbutaite, and T. Adomkus, "MANET Network in Internet of Things System," in *Ad Hoc Networks*, vol. 3, pp. 89–114, InTech, May 2017.

[103] A. Aijaz, H. Su, and A. H. Aghvami, "Enhancing RPL for cognitive radio enabled machine-to-machine networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2090–2095, IEEE, April 2014.

[104] A. Aijaz, H. Su, and A. H. Aghvami, "CORPL: A routing protocol for cognitive radio enabled AMI networks," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 477–485, 2015.

[105] S. Anamalamudi, M. Zhang, A. R. Sangi, C. E. Perkins, S. Anand, and B. L. (Remy), "Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)," Internet-Draft draft-ietf-roll-aodv-rpl-03, Internet Engineering Task Force, March 2018. Work in Progress.

[106] AdvanticSys, "TelosB CM5000 Mote Module." https://www.advanticsys.com/shop/mtmcm5000msp-p-14.html. Accessed on March 3^{rd}, 2018.

[107] Zolertia, "Zolertia Z1 Mote Module." https://zolertia.io. Accessed on March 3^{rd}, 2018.

[108] ARM Limited, "Cortex-M3 Processor." https://developer.arm.com/products/processors/corte m/cortex-m3. Accessed on March 3^{rd}, 2018.

[109] A. Raoof, A. Matrawy, and C.-H. Lung, "Secure Routing in IoT: Evaluation of RPL's Secure Mode under Attacks," in *IEEE Global Commun. Conf. (GLOBECOM 2019)*, 2019.

[110] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," technical report, Swedish Institute of Computer Science, Dec 2011.

[111] C. Pu and S. Hajjar, "Mitigating Forwarding Misbehaviors in RPL-based Low Power and Lossy Networks," in *15th IEEE CCNC 2018*, 2018.

[112] A. Mayzaud *et al.*, "A Taxonomy of Attacks in RPL-based Internet of Things," *Int'l Journal of Net. Security*, vol. 18, no. 3, pp. 459–473, 2016.

[113] A. Raoof, A. Matrawy, and C.-H. Lung, "POSTER: Evaluation of RPL Preinstalled Secure Mode Under Common Routing Attacks," in *7th IEEE Conf. on Comm. and Net. Sec. (CNS 2019)*, pp. 1–2, June 2019.

[114] A. Systems, "Wismote Mote Module." http://www.aragosystems.fr/produits/wisnet/wismote Accessed on April 21st, 2019.

[115] A. Y. Barnawi, G. A. Mohsen, and E. Q. Shahra, "Performance Analysis of RPL Protocol for Data Gathering Applications in Wireless Sensor Networks," *Procedia Computer Science*, vol. 151, no. 2018, pp. 185–193, 2019.

[116] C. Bormann *et al.*, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)." RFC 6775, Nov 2012.

[117] P. Thubert *et al.*, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery." RFC 8505.

[118] P. Thubert *et al.*, "Address Protected Neighbor Discovery for Low-power and Lossy Networks," internet-draft, IETF, Apr 2019. Work in Progress.

[119] A. Raoof and A. Matrawy, "The effect of buffer management strategies on 6lowpan's response to buffer reservation attacks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2017.

[120] M. Carignani, S. Ferrini, M. Petracca, M. Falcitelli, and P. Pagano, "A prototype bridge between automotive and the IoT," in *IEEE 2nd World Forum on Internet of Things (WF-IoT 2015)*, IEEE, December 2015.

[121] X. Wang, H. Cheng, and Y. Yao, "Addressing-Based Routing Optimization for 6LoWPAN WSN in Vehicular Scenario," *IEEE Sensors Journal*, vol. 16, pp. 3939–3947, May 2016.

[122] B. Pediredla, K. I. K. Wang, Z. Salcic, and A. Ivoghlian, "A 6LoWPAN Implementation for Memory Constrained and Power Efficient Wireless Sensor Nodes," in *39th Annual Conf. of the IEEE IECON (IECON 2013)*, IEEE, 2013.

[123] H. Kim, "Protection Against Packet Fragmentation Attacks at 6LoWPAN Adaptation Layer," in *2008 International Conference on Convergence and Hybrid Information Technology*, IEEE, 2008.

[124] C. Hennebert and J. D. Santos, "Security Protocols and Privacy Issues into 6LoWPAN Stack: A Synthesis," *IEEE Internet of Things Journal*, vol. 1, pp. 384–398, Oct 2014.

[125] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6LoWPAN Fragmentation Attacks and Mitigation Mechanisms," in *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2013)*, ACM Press, 2013.

[126] T. Watteyne, P. Thubert, and C. Bormann, "On Forwarding 6LoWPAN Fragments over a Multi-Hop IPv6 Network." RFC 8930, November 2020.

[127] M. Hossain, Y. Karim, and R. Hasan, "SecuPAN: A Security Scheme to Mitigate Fragmentation-Based Network Attacks in 6LoWPAN," in *Proceedings of the 8th ACM Conference on Data and Application Security and Privacy (CODASPY)*, vol. 2018-Janua, pp. 307–318, ACM, Mar 2018.

[128] S. R. Das, C. E. Perkins, and E. M. Belding-Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing." RFC 3561, July 2003.

[129] C. E. Perkins, S. Ratliff, J. Dowdell, L. Steenbrink, and V. Pritchard, "Ad Hoc On-demand Distance Vector Version 2 (AODVv2) Routing," Internet-Draft draft-perkins-manet-aodvv2-03, IETF, Feb 2019. Work in Progress.

[130] G. Glissa and A. Meddeb, "6LowPSec: An End-to-End Security Protocol for 6LoWPAN," *Ad Hoc Networks*, vol. 82, pp. 100–112, Jan 2019.

[131] "RIOT: 6LoWPAN Fragmentation." http://riot-os.org/api/rbuf_8h.html, 2013. Accessed on Aug 4^{th}, 2016.

[132] A. K. Bediya and R. Kumar, "Real Time DDoS Intrusion Detection and Monitoring Framework in 6LoWPAN for Internet of Things," in *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 824–828, IEEE, Oct 2020.

[133] A. Raoof, C.-H. Lung, and A. Matrawy, "Introducing Network Coding to RPL: The Chained Secure Mode (CSM)," in *The 19th IEEE Int. Symp. on Netw. Comput. and App. (NCA)*, IEEE, 2020.

[134] A. Raoof, C.-H. Lung, and A. Matrawy, "Securing RPL using Network Coding: The Chained Secure Mode (CSM)." Submitted for review at IEEE IoT Journal, arXiv:2102.06254 [cs.NI], 2021.

[135] D. Zhu, X. Yang, P. Zhao, and W. Yu, "Towards Effective Intra-Flow Network Coding in Software Defined Wireless Mesh Networks," in *24th Int. Conf. on Comput. Commun. and Netw. (ICCCN)*, IEEE, Aug 2015.

[136] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure network coding for wireless mesh networks: Threats, challenges, and directions," *Computer Communications*, vol. 32, no. 17, pp. 1790 – 1801, 2009.

[137] M. Hay, B. Saeed, C.-H. Lung, T. Kunz, and A. Srinivasan, "Network Coding and Quality of Service for Mobile Ad Hoc Networks," *Int. Journal of Commun., Network and System Sciences*, vol. 07, no. 10, pp. 409–422, 2014.

[138] B. Saeed, C.-H. Lung, T. Kunz, and A. Srinivasan, "Multimedia Streaming for Ad Hoc Wireless Mesh Networks Using Network Coding," *Int. Journal of Commun., Network and System Sciences*, vol. 06, no. 05, pp. 204–220, 2013.

[139] M. P. Uwase *et al.*, "Experimental Comparison of Radio Duty Cycling Protocols for Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 17, no. 19, pp. 6474–6482, 2017.

[140] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service Detection in 6LoWPAN based Internet of Things," in *9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 600–607, IEEE, Oct 2013.