

INTRUSION DETECTION SYSTEM FOR RPL BASED INTERNET OF THINGS

BY

HUSSAH ALI ALBINALI

A Dissertation Presented to the
COLLEGE OF GRADUATE AND INTERDISCIPLINARY STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE

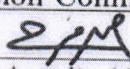
December 2024

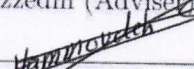
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

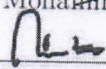
DEANSHIP OF GRADUATE STUDIES

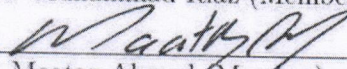
This dissertation, written by **HUSSAH ALI ALBINALI** under the direction of his dissertation adviser and approved by his dissertation committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE**.

Dissertation Committee

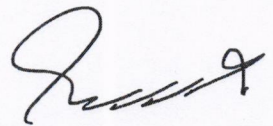

Dr. Farag Azzedin (Adviser)


Prof. Mohammad Hammoudeh (Member)


Prof. Muhammad Riaz (Member)


Dr. Moataz Ahmed (Member)


Dr. Ahcene Bounceur (Member)


Dr. Motaz Alfarraj
Department Chairman

Prof. Suliman Al-Homidan
Dean of Graduate Studies

29- December -2024

Date



©Hussah Albinali
2024

I dedicate this work to my beloved family; Your unlimited love and support help me to achieve my goals..

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to the following: First and foremost, I want to thank Dr. Farag Azzedin, my advisor, for his guidance and knowledge. His direction throughout my Ph.D. studies was truly invaluable. I would also like to extend my gratitude to Prof. Muhammad Riaz for his guidance and direction throughout various chapters of this thesis. His expertise and insights have been influential in shaping this work. Another big thanks are directed to my dissertation committee: Prof. Mohammad Hammoudeh, Prof. Muhammad Riaz, Dr. Moataz Ahmed, and Dr. Ahcene Bounceur, for taking the time to review this dissertation and provide me with valuable comments and inspiring questions. I also want to express my gratitude to my sponsor, Imam Abdulrahman Bin Faisal University, for offering me the opportunity to pursue my Ph.D. Of course, the biggest credit goes to my family, who supported me in every way they could. I am also incredibly thankful to all my friends at KFUPM for their encouragement and support. Finally, thank you to all who helped me with this research, even if I forgot to mention them in this long acknowledgment.

Contents

ACKNOWLEDGMENTS	vi
LIST OF TABLES	xiii
LIST OF FIGURES	xv
LIST OF SYMBOLS	xviii
LIST OF ABBREVIATIONS	xix
ABSTRACT (ENGLISH)	xxi
ABSTRACT (ARABIC)	xxiii
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	4
1.2 Research Questions	5
1.3 Research Methodology	6
1.3.1 Literature Review	7
1.3.2 Attacks Implementation & Taxonomies Building	7
1.3.3 Network Modelling	8
1.3.4 IDS Development	8
1.4 Research Contributions & Outcomes	8
1.5 Dissertation Organization	10
CHAPTER 2 BACKGROUND REVIEW	11
2.1 The Routing Protocol for Low-Power and Lossy Networks	11

2.1.1	Traffic Flows in RPL and Modes of Operations	13
2.1.2	Topology Construction	14
2.1.3	RPL Control Messages	16
2.1.4	Trickle Algorithm	18
2.1.5	Objective Function	19
2.1.6	RPL Mechanisms of Repair & Maintenance	20
2.1.7	RPL Secure Modes	21
2.1.8	Applications built upon RPL	22
2.2	RPL Attacks	23
2.2.1	Blackhole Attack	23
2.2.2	Clone-ID Attack	24
2.2.3	Copypcat Attack	24
2.2.4	DAO Inconsistency Attack	24
2.2.5	DAO Insider Attack	25
2.2.6	Decreased Rank Attack or Sinkhole Attack	25
2.2.7	DIO Suppression Attack	25
2.2.8	DIS Flooding Attack	26
2.2.9	DODAG Inconsistency Attack	26
2.2.10	Hello Flood Attack	27
2.2.11	Increased Rank Attack	27
2.2.12	Local Repair Attack	28
2.2.13	Neighbor Attack	28
2.2.14	Route Table Falsification Attack	29
2.2.15	Sybil Attack	29
2.2.16	Selective Forwarding Attack	29
2.2.17	Version Attack	30
2.2.18	Wormhole Attack	31
2.2.19	Worst Parent Attack	31
2.3	Existing Taxonomies of RPL Attacks	32
2.4	Security Solutions for RPL attacks	36

2.4.1	Prevention Solutions	37
2.4.2	Detection Solutions	37
2.4.3	Mitigation Solutions	45
2.5	Taxonomies of Security Solutions	46
2.6	Descriptive Models of RPL	49
2.6.1	Trust Based Models	50
2.6.2	Game Theory Based Model	50
2.6.3	Finite State Machine Based Model	51
2.6.4	Evidence Theory Based Model	52
CHAPTER 3 TAXONOMY OF RPL ATTACKS		58
3.1	Dropping Packets Attacks	59
3.1.1	Blackhole Attack	60
3.1.2	Selective Forwarding Attack	60
3.1.3	Dropped Destination Advertisement Object Attack	60
3.2	Generating Packets Attacks	61
3.2.1	Packet Flooding Attack	61
3.2.2	DIS Flooding Attack	62
3.2.3	Sybil Attack	62
3.2.4	Clone-ID Attack	62
3.2.5	Hello Flood Attack	63
3.2.6	DAO Insider Attack	63
3.3	Modifying Packets Attacks	63
3.3.1	Decreased Rank Attack or Sinkhole Attack	64
3.3.2	Increased Rank Attack	64
3.3.3	Worst Parent Attack	65
3.3.4	Local Repair Attack	65
3.3.5	Version Attack	65
3.3.6	DODAG Inconsistency Attack	66
3.3.7	DAO Inconsistency Attack	66

3.4	Replaying Packets Attacks	67
3.4.1	Replay Attack	67
3.4.2	Neighbor Attack	67
3.4.3	DIO Suppression Attack	68
3.4.4	Copypcat Attack	68
3.4.5	Wormhole Attack	69
3.5	Analysis	69
3.6	Application of the Proposed Taxonomy	71
CHAPTER 4 MITIGATION SOLUTIONS FOR RPL ATTACKS		79
4.1	Mitigation Solutions Based on Modification	79
4.2	Mitigation Solutions Based on Discarding and Updating	82
4.3	Mitigation Solutions Based on Additions, Authentication, Encryption, and Isolation	84
4.4	Discussion	89
CHAPTER 5 REPLAY ATTACKS IN RPL-BASED INTERNET OF THINGS		99
5.1	Analysis of RPL Replay Attacks	99
5.1.1	Attacks Resulting from Replaying DIO	102
5.1.2	Attacks Resulting from Replaying DAO	109
5.2	Detecting Replay Attacks	113
CHAPTER 6 DEVS-RPL MODEL		117
6.1	Model Objective & Scope	117
6.2	Model Specification	119
6.2.1	Input, Output, State Set	119
6.2.2	Initial State	119
6.2.3	Internal and External Transition Functions	119
6.2.4	Confluent Transition Function	125
6.2.5	Output Function	125

6.2.6	Time Advance Function	125
6.3	Deriving Messages Generation and Sending Time	126
6.3.1	Generating DIS	126
6.3.2	Generating Unicast DIO	127
6.3.3	Generating Multicast DIO	127
6.3.4	Generating DAO	129
6.3.5	Deriving the Number of Received Messages	129
CHAPTER 7	PROPOSED DEVS-RPL IDS	131
7.1	Anomaly Detection using Statistical Process Control	131
7.2	Defining limits in Control Messages	136
7.3	DEVS-RPL IDS Monitoring Implementation	136
CHAPTER 8	EXPERIMENTAL EVALUATION	139
8.1	Replay Attacks Impacts	139
8.1.1	Experimental Setup and Design	140
8.1.2	Evaluation Metrics	141
8.2	DEVS-RPL Model Validation	143
8.2.1	RPL Network Dataset	143
8.2.2	DEVS-RPL Model Dataset	145
8.2.3	Dataset Distribution Validation	145
8.3	Evaluation of DEVS-RPL IDS Detection	146
8.3.1	Evaluation Setup and Assumptions	146
8.3.2	Evaluation Metrics	148
CHAPTER 9	RESULTS AND DISCUSSION	151
9.1	Replay Attacks Evaluation	151
9.1.1	Packet Delivery Ratio (PDR)	151
9.1.2	End-to-End (E2E) Delay	152
9.1.3	Control Message Overhead	154
9.1.4	Power Consumption	155

9.2	DEVS-RPL Validation Results	156
9.2.1	Generated DIS	157
9.2.2	Generated Unicast DIO	158
9.2.3	Generated Multicast DIO	160
9.2.4	Generated DAO	162
9.2.5	Error Evaluation	164
9.3	DEVS-RPL IDS Performance Results	166
9.3.1	Detection Efficiency	168
9.3.2	Detection Delay	172
9.3.3	Power Consumption	173
9.3.4	Memory Consumption	174
CHAPTER 10 CONCLUSIONS AND FUTURE DIRECTIONS		175
10.1	Overview	175
10.2	Limitations	177
10.3	Concluding Remarks	179
10.4	Directions for Future Work	180
REFERENCES		182
VITAE		232

List of Tables

Table 1.1:	Dissertation Outcomes	9
Table 2.1:	Summary of RPL attacks taxonomies	36
Table 2.2:	Summary of detection solutions published in 2013 to 2020 .	44
Table 2.3:	Summary of detection solutions published in 2021 to 2023 .	53
Table 2.4:	Summary of mitigation solutions published in 2013 to 2021	54
Table 2.5:	Summary of countermeasures taxonomies	55
Table 2.6:	Summary of mitigation solutions published in 2021 to 2023	56
Table 2.7:	Summary of RPL models	57
Table 4.1:	Mitigation solutions based on modification	82
Table 4.2:	Modification solutions used in the selected references	83
Table 4.3:	Mitigation solutions: discarding and updating	85
Table 4.4:	Discarding and updating solutions used in the selected ref- erences	86
Table 4.5:	Mitigation solutions: additions, authentication, encryption, isolation	88
Table 4.6:	Additions, authentication, encryption, and isolation solutions	89
Table 4.7:	Evaluation criteria of mitigation solution Efficiency	93
Table 4.8:	Efficiency scores of mitigation solutions	96
Table 4.9:	High efficient mitigation solution per attack	98
Table 5.1:	The classification scheme of potential impact on routing . .	101
Table 5.2:	Evaluation for RPL replay attacks impacts on routing . . .	114
Table 5.3:	Summary of proposed security solutions for replay attacks .	116

Table 6.1:	Participant states of each control message	126
Table 6.2:	DEVS-RPL specification of received messages	128
Table 7.1:	The statistical parameters of RPL messages	135
Table 8.1:	Simulation parameters of replay attacks evaluation	140
Table 8.2:	Configuration parameters in different RPL replay attacks .	141
Table 8.3:	Z1 mote power specification	143
Table 8.4:	RPL simulation settings	144
Table 8.5:	RPL parameters	145
Table 8.6:	DEVS-RPL parameters	145
Table 8.7:	Simulation parameters of DEVS-RPL IDS evaluation	148
Table 9.1:	Control messages distribution in RPL network	157
Table 9.2:	Statistical metrics for error and deviation analysis	164
Table 9.3:	Confusion matrix to evaluate DEVS-RPL IDS	169
Table 9.4:	Detection evaluation metrics	169
Table 9.5:	Memory consumption	174
Table 9.6:	Classes of constrained devices [1]	174

List of Figures

Figure 2.1:	Protocol stacks: (a) IoT IETF, (b) ZigBee IP SEP 2.0. . . .	12
Figure 2.2:	The DODAG construction mechanism	15
Figure 2.3:	The structure of DIS message	17
Figure 2.4:	The structure of DIO message	17
Figure 2.5:	The structure of DAO message	18
Figure 2.6:	The structure of DAO-ACK message	18
Figure 2.7:	RPL attacks taxonomy [2]	33
Figure 2.8:	RPL attacks taxonomy [3]	34
Figure 2.9:	RPL attacks taxonomy [4]	35
Figure 2.10:	Security solutions roles	36
Figure 2.11:	Security solutions taxonomy [5]	48
Figure 3.1:	General classification of RPL attacks	59
Figure 3.2:	Dropping packets attacks	59
Figure 3.3:	Generating packets attacks	61
Figure 3.4:	Modifying packets attacks	64
Figure 3.5:	Replaying packets attacks	67
Figure 3.6:	Taxonomy of RPL attacks	70
Figure 3.7:	Attack detection flowchart using proposed taxonomy for dropping attacks	72
Figure 3.8:	Attack detection flowchart using proposed taxonomy for generating packets attacks	73
Figure 3.9:	Attack detection flowchart using proposed taxonomy for generating DIS attacks	74

Figure 3.10:	Attack detection flowchart using proposed taxonomy for generating DIO attacks	75
Figure 3.11:	Attack detection flowchart using proposed taxonomy for generating DAO attacks	75
Figure 3.12:	Attack detection flowchart using proposed taxonomy for modifying packets attacks	77
Figure 3.13:	Attack detection flowchart using proposed taxonomy for re-playing packets attacks	78
Figure 4.1:	Mitigation solutions for RPL attacks	80
Figure 4.2:	Publications per mitigation solution based on modification .	82
Figure 4.3:	Publications per mitigation solution based on discarding and updating	85
Figure 4.4:	Publications per mitigation solution based on additions, authentication, encryption, and isolation	88
Figure 4.5:	Taxonomy of mitigation solutions for RPL attacks	90
Figure 4.6:	Mitigation solutions percentages for each RPL attack class .	91
Figure 5.1:	RPL replay attacks mechanism	101
Figure 5.2:	DIO replay attack analysis based on relative location: (a) low impact, (b) moderate impact, and (c) critical impact. .	105
Figure 5.3:	Frequent DIO replay attacks analysis based on relative location: (a) low impact, (b) moderate impact, and (c) critical impact.	108
Figure 5.4:	DAO replay attacks analysis based on relative location: (a) moderate impact, (b) critical impact, and (c) DAO replay attack with source forging.	112
Figure 5.5:	Common used features to detect RPL replay attacks	116
Figure 6.1:	State transitions of DEVS-RPL model	120
Figure 8.1:	Network layouts in used dataset	144

Figure 8.2:	Network layouts used for the evaluation	147
Figure 9.1:	Local impact of replay attacks on PDR	153
Figure 9.2:	Local impact of replay attacks on E2E delay	154
Figure 9.3:	Local impact of replay attacks on average control message overhead	156
Figure 9.4:	Replay attacks impact on average power consumption . . .	157
Figure 9.5:	Sending time of DIS messages	158
Figure 9.6:	Distribution of received <i>DIS</i> and <i>DIO_U</i>	159
Figure 9.7:	Average generated <i>DIO_U</i> during the simulation time	160
Figure 9.8:	Average generated <i>DIO_M</i> during the simulation time	160
Figure 9.9:	Distribution of received <i>DIO_M</i> messages	162
Figure 9.10:	Average generated DAO during the simulation time	163
Figure 9.11:	Distribution of received <i>DAO</i> messages	163
Figure 9.12:	EWMA control charts ($\bar{X}$ chart), with vertical lines indi- cating the moments of attacks	167
Figure 9.13:	EWMA control charts (S chart) with vertical lines indicat- ing the moments of attacks	168
Figure 9.14:	Average accuracy, FPR, and TPR over simulation time . . .	170
Figure 9.15:	Comparisons of detection accuracy between DEVS-RPL IDS and specification-based detectors	171
Figure 9.16:	Comparisons of TPR and FPR between DEVS-RPL IDS and specification-based detectors	171
Figure 9.17:	The detection latency of DEVS-RPL IDS in detecting attacks	172
Figure 9.18:	Comparisons of average power consumption between DEVS-RPL IDS and specification-based detectors	173

LIST OF SYMBOLS

D_{DAO}	DAO Delay
DIO_M	Multicast DIO
DIO_M^U	Multicast DIO With Update
DIO_U	Unicast DIO
I_{DAO}	Default Lifetime
I_{DIO}	DIO next delay
I_{DIS}	DIS Interval
$I_{Probing}$	Probing Interval
$Max(I_{DIO})$	$2^{DIOIntervalMin+DIOIntervalDoublings}$
$Min(I_{DIO})$	$2^{DIOIntervalMin}$

LIST OF ABBREVIATIONS

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
DAO	Destination Advertisement Object
DAO-ACK	Destination Advertisement Object-Acknowledgement
DEVS	Discrete Event Systems
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DODAG	Destination Oriented Directed Acyclic Graph
E2E	End-to-End
EWMA	Exponentially Weighted Moving Average
FPR	False Positive Rate
ICMPV6	Internet Control Message Protocol version 6
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
LCL	Lower Control Limit
LLN	Low Power and Lossy Network
MRHOF	Minimum Rank with Hysteresis Objective Function

PDR	Packet Delivery Ratio
RPL	Routing Protocol for Low-Power and Lossy Networks
SLR	Systematic Literature Review
SPC	Statistical Process Control
TPR	True Positive Rate
UCL	Upper Control Limit

DISSERTATION ABSTRACT

NAME: Hussah Ali Albinali

TITLE OF STUDY: Intrusion Detection System for RPL Based Internet of Things

MAJOR FIELD: Computer Science

DATE OF DEGREE: December, 2024

The Internet of Things (IoT) is rapidly evolving and has raised concerns about data security. The Routing Protocol over Low-power and Lossy Networks (RPL) was introduced for IoT environments, but there are currently no official implementations of its secure modes. This leaves RPL vulnerable to attacks, necessitating the development of detection and mitigation systems. In this study, we have created an intrusion detection system to identify replay attacks of RPL control messages on IoT devices. To achieve this goal, we conducted a comprehensive survey of RPL attacks and their mitigation strategies, and we introduced new taxonomies for these attacks and mitigation strategies. We also conducted a comparative and analytical study for replay attacks and their impact on networks. To model nodes' behavior and specify the mean of each RPL message in accordance with the RPL standard, we utilized the discrete

event model. As a result, we created DEVS-RPL IDS, an intrusion detection system that actively monitors the network in a passive manner. It employs a predefined baseline of RPL messages to identify outliers and anomalies in node behavior through statistical process control and an exponentially weighted moving average. As a consequence, the proposed taxonomy for RPL attacks is organized according to attack vectors, whereas the classification for mitigation solutions is structured around mitigation mechanisms. RPL attacks are classified as generating, dropping, modifying, or replaying specific control messages. The mitigation solutions include authenticating, updating, discarding, encrypting, and isolating attacking nodes. Our analysis of replay attacks included five types: neighbor, DIO suppression, DAO replay, copycat, and route table falsification attacks. The analysis reveals that most of these attacks have a severe impact on availability and a serious impact on integrity. DAO replay attacks, DIO suppression, and copycat attacks cause the average local delivery ratio to drop by 60%, and communication latency increases by 50% under certain attack scenarios. In addition, we evaluated the performance of the proposed DEVS-RPL IDS, which shows that the proposed solution successfully detects these attacks with accuracy, reaching 95% with only a 4% false alarm rate. Also, DEVS-RPL IDS added an increase of around 3% in power consumption and 1% in memory consumption. This allows DEVS-RPL to be efficient and lightweight, making it suitable for deployment on constrained nodes.

ملخص الرسالة

الاسم: حصه بنت علي البنعلي

عنوان الدراسة: نظام كشف الاختراقات لبروتوكول التوجيه في شبكات إنترنت الأشياء ذات الطاقة المنخفضة

التخصص: علوم الحاسب الآلي

تاريخ الدرجة العلمية: جمادى الأولى 1446هـ

إن إنترنت الأشياء يتطور بسرعة وقد أثار مخاوف بشأن أمن البيانات. تم تقديم بروتوكول التوجيه عبر الشبكات منخفضة الطاقة لبيئات إنترنت الأشياء، ولكن لا توجد حاليًا أي تطبيقات رسمية لاعدادات الأمان الخاصة به. وهذا يجعل بروتوكول التوجيه عبر الشبكات منخفضة الطاقة عرضة للهجمات، مما يستلزم تطوير أنظمة الكشف و تخفيف آثار الهجمات. في هذه الدراسة، قمنا بإنشاء نظام كشف الاختراقات لتحديد هجمات إعادة الإرسال لرسائل التحكم في بروتوكول التوجيه عبر الشبكات منخفضة الطاقة على أجهزة إنترنت الأشياء. وفقًا لذلك، أجرينا مسحًا موسعًا حول هجمات بروتوكول التوجيه عبر الشبكات منخفضة الطاقة وطرق تخفيفها واقترحنا تصنيفات جديدة لها. أجرينا أيضًا دراسة مقارنة وتحليلية لهجمات إعادة الإرسال وتأثيرها على الشبكات. لقد استخدمنا نموذج الحدث المنفصل لتحديد متوسط كل رسالة بروتوكول التوجيه عبر الشبكات منخفضة الطاقة وفقًا لمعيار بروتوكول التوجيه عبر الشبكات منخفضة الطاقة. نتيجة لذلك، قمنا بتطوير نظام كشف الاختراق الذي يراقب الشبكة ويستخدم خط الأساس المحدد لرسائل التحكم للكشف عن القيم المتطرفة والشذوذ في سلوك الأجهزة باستخدام التحكم الإحصائي في العملية ومتوسط متحرك مرجح بشكل كبير. ونتيجة لذلك، فإن التصنيف المقدم للهجمات يعتمد على آلية الهجوم، ويعتمد التصنيف لحلول التخفيف على آليات التخفيف. يتم تصنيف هجمات على أنها توليد أو إسقاط أو تعديل أو إعادة إرسال رسائل تحكم محددة. تتضمن حلول التخفيف المصادقة والتحديث والتخلص من وتشفير وعزل الأجهزة المهاجمة. تتضمن تحليلنا لهجمات إعادة الإرسال خمسة أشكال: هجمات الجار وقمع DIO وإعادة إرسال DAO والتقليد وتزوير جدول التوجيه. وكشف أن معظم هذه الهجمات لها تأثير شديد على التوافر وتأثير خطير على النزاهة. تتسبب هجمات إعادة إرسال DAO وقمع DIO وهجمات

التقليد في انخفاض متوسط نسبة وصول البيانات بنسبة 60%، ويزداد زمن نقل الرسائل بنسبة 50% في سيناريوهات هجومية معينة. لقد قمنا بتقييم أداء حل اكتشاف الاختراقات المقترح، والذي يوضح أن الحل المقترح يكتشف هذه الهجمات بنجاح بدقة تصل إلى 95% مع معدل إنذار كاذب بنسبة 4% فقط. كما أضاف زيادة محدودة بنحو 3% في الطاقة و1% في استهلاك الذاكرة، مما يجعله فعالاً للنشر على الأجهزة منخفضة الطاقة.

CHAPTER 1

INTRODUCTION

The urban development vision to ensure a better quality of life leads to infrastructure construction and integration of the Internet of Things (IoT) in managing assets, including healthcare centers, vehicles, energy utilities, and other community services. A huge investment is promoted using different IoT devices, including sensors, to monitor and control these assets. Sensors connected with real-time monitoring systems open new opportunities for automated services and cutting-edge applications for municipal users. According to S. ODea [6], the number of short-range IoT devices is expected to grow to around 22.4 billion by 2027.

However, these sensors have limitations in computing, storage, energy, and communication capacity. Such networks suffer from many vulnerabilities, including low data rates and packet delivery failure [7]. Low Power and Lossy Network (LLN) is the name given to a network that consists of many constrained nodes with limited processing power and memory. The nature of LLN makes the existing routing protocols not suitable for IoT. As a result, Internet Engineering Task Force (IETF) introduced IPv6 routing protocol over LLNs Routing Protocol for Low-Power and Lossy Networks

(RPL). RPL has emerged as the standard routing protocol for IoT networks, as it was specifically designed to maximize limited resources while delivering efficient routing services [8].

As smart applications collect sensitive data, it is important to consider security and privacy issues at different levels of the architecture while designing and implementing these applications [9]. One of the security challenges involves securing data transmission between IoT devices by ensuring confidentiality, authentication, and integrity in the routing of data [4]. Replay attacks pose a significant threat to security protocols and have been discussed in the literature [10], [3]. Replay attacks deceive legitimate participants into thinking that the protocol execution has been successfully finalized by replaying messages in a setting different from the original context [11], [12]. Replay attacks resist cryptographic communications and are effective without data origin authentication schemes [4].

Consequently, several researchers have identified replay attacks as a security threat in RPL networks. Authors in [2], [3] showcased more than 15 types of RPL attacks. Several attacks exploit control messages by replaying them to accomplish their assaults, including neighbor and route table falsification attacks. Moreover, various studies have highlighted that any control message can serve as a vector for RPL replay attacks [4]. At the same time, various countermeasures have been proposed [13], [4], [3]. The proposed countermeasures are categorized into prevention, detection, and mitigation [14]. The prevention solution offers complete defense against a specific attack. This type of solution aims to use techniques that protect the network from the targeted attack before it happens. However, according to [15], [16], providing pre-

vention solutions against all attacks is almost impossible, especially in wireless and heterogeneous IoT devices. The second countermeasure is detection, which allows systems to identify any suspicious behavior and trigger the system when an attack is present [14]. The last countermeasure solution is mitigation. This solution allows the attack, but it reacts to minimize its impact on the network [14]. Current studies [17], [18] show that detection solutions are given more attention than mitigation solutions, especially in RPL environments where many detection approaches have been explored to detect RPL attacks, including specification-based solutions [5]. The specification-based detection systems identify attacks when compromised nodes lead other nodes to behave differently than the one stated in the protocol. Therefore, the IDS needs to know the rules that define the baseline of the network according to the protocol to categorize nodes into normal and attackers [19]. Specification-based IDS has advantages over other detection approaches, such as signature-based or anomaly-based IDSs. First, specification-based IDS does not need historical network traffic to learn the baseline, unlike anomaly-based systems that require extensive training across different scenarios. In addition, the system does not detect attacks by looking for stored specific attacks' signatures, similar to the design principle of signature-based IDS [20].

Therefore, to design a specification-based intrusion detection system, understanding the behavior of network nodes according to RPL is crucial. Existing research lacks a model for quantifying the generated and exchanged RPL messages. RPL as a network protocol is carefully designed and documented in a standard. In our work, we simplify the specifications to focus on the essential details of RPL. Our work considers the Discrete Event Systems (DEVS) based on systems theory [21]. DEVS formalism

was introduced to provide a hierarchical and modular method of modeling the discrete event systems [21], [22]. In DEVS, the system is viewed as a “black box” that receives and sends messages through its input and output ports.

1.1 Problem Statement and Motivation

In this work, we aim to introduce a new taxonomy that provides a classification of RPL attacks in a way that can be easily extended to future attacks. In addition, to the best of our knowledge, no study has focused on RPL mitigation solutions. Accordingly, we also worked on establishing a new taxonomy of mitigation methods for RPL attacks that is consistent with the introduced taxonomy of RPL attacks.

In addition, the research focusing on replay attacks is quite limited. Based on the systematic literature review by Al-Amiedy et al. [23], only seven studies out of 127 selected references have examined replay attacks (including neighbor attacks as a type of replay attacks). Besides, replay attacks reported impacts vary between limited impacts by causing additional delay and utilizing suboptimal routes [24], [5] to severe impacts such as network partitioning [4], [3]. Consequently, no comprehensive study has been conducted to analyze and evaluate RPL replay attacks. The insufficient research on the various types of RPL replay attacks, and their effects inspired us to undertake a study that comprehensively assesses and analyzes RPL replay attacks while presenting cutting-edge security countermeasures.

On the other hand, the absence of a model that describes the number of exchanged control messages on the RPL protocol motivates us to introduce a model that speci-

fies the number and frequency of exchanged control messages in normal networks by analyzing the RPL protocol. We introduce a model that defines the baseline of exchanged control messages. This model paves the way for designing a detection system that monitors the network and ensuring each node follows normal behavior regarding the number and frequency of the exchanged control messages. Focusing on the placement type of the proposed detection system, we targeted developing this system into a hybrid system with minimal traffic overhead in the network. The hybrid model is needed to allow a dedicated host to perform heavy calculations if required, whereas the network nodes perform the detection and simple calculations. The key issues tackled in this research are:

- The lack of flexible and extensible taxonomy for RPL attacks and corresponding mitigation techniques.
- The limited evaluation of replay attacks on RPL networks.
- The absence of a model defining normal control message exchange in RPL networks hinders the ability to detect anomalies effectively.
- The necessity for an intrusion detection system that balances centralized and distributed detection approaches while minimizing traffic overhead and computation requirements on constrained nodes.

1.2 Research Questions

Specifically, this thesis seeks to address the following questions:

- RQ1: How to classify RPL attacks in extendable taxonomy?
- RQ2: How to classify RPL mitigation solutions?
- RQ3: How can replay attacks be implemented against the RPL protocol, and what are their impacts?
- RQ4: How can establishing and maintaining routes in RPL be modeled to specify the number and frequency of generated RPL control messages?
- RQ5: How to design a lightweight detection system by utilizing the RPL specifications of the number and frequency of exchanged control messages with the following performance metrics:
 1. Accuracy is between 85% and 98%.
 2. False Positive Rate is between 4% and 9%.
 3. True Positive Rate is around 85 % and 95%.
 4. The consumed power in network nodes is around 1.6 mWatt.

1.3 Research Methodology

In order to achieve the proposed contributions, the work was divided into these major steps:

1.3.1 Literature Review

We thoroughly analyzed published works by conducting a Systematic Literature Review (SLR). The primary objective of this SLR is to comprehensively examine RPL attacks and the mitigation mechanisms. By systematically scrutinizing the existing literature on this subject, this SLR aims to provide a detailed analysis of the current state of knowledge in this field. By adopting this, we ensure unbiased research by thoroughly analyzing all available evidence and evaluating each study. This methodology is substantial for producing accurate and reliable results [25]. Additionally, SLRs can be replicated and updated by other researchers.

1.3.2 Attacks Implementation & Taxonomies Building

The studied attacks are implemented to learn exactly how these attacks are launched and to better understand each attack's impact. Moreover, grouping and comparing the mechanisms of attacks and their mitigation solutions helps in developing corresponding taxonomies. To verify the correctness of implementing these attacks, we directly utilized the available algorithms and mechanisms provided in the literature for the attacks. Lastly, for all implemented attacks, we compare the impact of these attacks on the network in terms of the impacted routing metrics with the published values in the literature.

1.3.3 Network Modelling

We simplify the RPL specifications to focus on the essential details for this goal. Our work considers the DEVS based on systems theory [21]. DEVS formalism was introduced to provide a hierarchical and modular method of modeling the discrete event systems [21], [22].

1.3.4 IDS Development

We rely on our network model to develop a detection system. We also use statistical process control to monitor the network and ensure the RPL traffic between nodes is within the specified upper and lower limits.

1.4 Research Contributions & Outcomes

The contributions of this thesis are:

1. Conducting an extensive survey on RPL attacks and their mitigation methods in IoT environments.
2. Proposing a taxonomy for RPL attacks.
3. Proposing a taxonomy for the mitigation methods against RPL attacks.
4. Performing comparative and empirical study on replay attacks in RPL.
5. Modeling RPL normal behavior to define the baseline in IoT environments.
6. Developing an IDS for replay attacks and evaluate its performance.

Table 1.1: *Dissertation Outcomes*

Outcome	Notes	Dissertation Contribution
Albinali, H., Azzedin, F., & Riaz, M. (2024). DEVS-RPL IDS: A Specification-Based Intrusion Detection on RPL-Based Network Using Discrete Event Model & Statistical Process Control	Submitted to Computers & Security. Q1 with IF 4.8. Under revision	#5 and #6
Albinali, H., & Azzedin, F. (2024). Replay Attacks in RPL-Based Internet of Things: Comparative and Empirical Study. Computer Networks [26] .	Published Q1 with IF 4.4.	#1 and #4
Albinali, H., & Azzedin, F. (2024). Towards RPL Attacks and Mitigation Taxonomy: Systematic Literature Review Approach. IEEE Transactions on Network and Service Management [27]	Published. Q1 with IF 4.7	#1, #2, and #3
Alghofaili, R., Albinali, H., & Azzedin, F. (2024). Build–Launch–Consolidate Framework and Toolkit for Impact Analysis on Wireless Sensor Networks. Journal of Sensor and Actuator Networks, 13(1), 17 [28]	Published. Q2 with IF 3.3	#1
Albinali, H., Azzedin, F., & Riaz, M. (2024). DEVS-RPL: Design Formal Discrete Event Model of Routing Protocol over LLN. Proceedings of the 2024 IEEE Cybermatics Congress [29]	Published	#5
Albinali, H., Alghofaili, R., & Azzedin, F. (2023, December). Analytical Study of Local Repair Attack on RPL-Based 6LoWPAN Networks in Internet of Things. In Proceedings of the 7th International Conference on Future Networks and Distributed Systems [30]	Published	#1
Azzedin, F., & Albinali, H. (2023). System And Method to Detect and Countermeasure RPL Attacks In IoT Network (Patent No. 547185US). OBLON, Mclelland, MAIER & Neustadt, L.L.P.	Patent. Status: “Green Light”	#2, #3
Azzedin, F., & Albinali, H. (2021, December). Security in Internet of Things: RPL attacks taxonomy. In Proceedings of the 5th International Conference on Future Networks and Distributed Systems [13]	Published	#2

1.5 Dissertation Organization

Chapter 2 gives an overview of RPL, routing attacks against RPL, security solutions, and descriptive models that demonstrate RPL characteristics. In Chapter 3, we present a comprehensive RPL attacks taxonomy that classifies various attacks in greater detail, including the behavior, while in Chapter 4, we demonstrate the classification of RPL mitigation solutions that can effectively reduce the impact of RPL attacks. Furthermore, Chapter 5 analyzes replay attack implementations using RPL control messages and their impacts. We outline the modeling of the RPL establishment and maintenance to specify the number and frequency of generated RPL messages in Chapter 6. Chapter 7 describes the design of a lightweight detection system that leverages the RPL specifications for the number and frequency of exchanged control messages. Chapter 8 covers implemented attacks and their evaluations, the validation of the DEVS-RPL model, and the implementation and evaluation of our proposed DEVS-RPL IDS detection solution. Chapter 9 details the results of executed attacks and their impact on the network. It also includes the validation results of DEVS-RPL and the evaluation metrics of the IDS, such as accuracy, false positive rate, true positive rate, and memory and power consumption. Finally, the dissertation is concluded in Chapter 10.

CHAPTER 2

BACKGROUND REVIEW

This chapter provides a background of the following: an overview of RPL in Section 2.1, different routing attacks against RPL in Section 2.2, and classification taxonomies in Section 2.3. It also discusses proposed security solutions for these attacks and taxonomies to organize them in Sections 2.4 and 2.5. Lastly, it highlights various descriptive models to illustrate the main characteristics of RPL in Section 2.6.

2.1 The Routing Protocol for Low-Power and Lossy Networks

The usage of IoT applications continues to grow rapidly as an effective solution to the difficulties faced in smart cities, earthquake warning systems, smart industry, and more [31], [32]. Wireless sensor networks (WSNs) are critical in integrating IoT systems for routing and sensing. These networks can be deployed smoothly to supply a wide range of services and applications [31], [33]. WSN has a limited number of

sensor nodes (motes) handled by a multi-layered protocol organization. The IEEE 802.15.4 protocol is the typically used communication system for WSNs as a result of its low power consumption, cost-effectiveness, and its capability of defying physical and connecting layers for wireless short-range transmissions [34], [35]. It operates on 800/900 MHz and 2.4 GHz ISM frequency bands and is the foundation for other standards such as ZigBee, Wireless Hart, WIA-PA, and ISA.100.11a [34].

RPL is IPv6 routing protocol for LLN developed by IETF routing over LLN (ROLL) group as a recommended standard. Given the noteworthy overlapping between LLNs and IoT, and the fact that IPv6 is a crucial feature in IoT environments, RPL has promptly become the routing protocol for IoT. As illustrated in Figure 2.1 (a), IETF puts RPL at the network layer [36]. The success of RPL as an IoT standard is also supported by companies part of ZigBee Alliance [37], [35]. These companies apply industry-standard protocols, including IPv6, 6LoWPAN, RPL, and TCP/UDP/IP, to bring End-to-End (E2E) IPv6 packets without the demands of intermediary gateways. As such, ZigBee IP has favored RPL to easily plug their networks into the IP-based Internet, which was impeding a concrete IoT, as shown in Figure 2.1 (b).

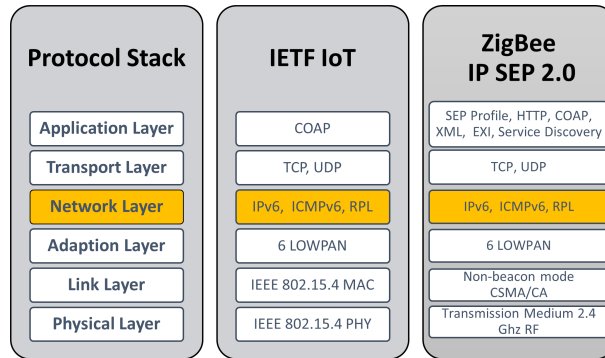


Figure 2.1: *Protocol stacks: (a) IoT IETF, (b) ZigBee IP SEP 2.0.*

The RPL design principle is to create a tree-like topology called directed acyclic graph (DAG) that has a single destination called Destination Oriented Directed Acyclic Graph (DODAG) [38]. This hierarchical design has the benefit of forbidding network traffic loops [39]. Each node in the DODAG has a rank that points out the cost to reach the root; typically, nodes near the root have a lower rank than nodes farther away [40]. RPL uses an objective function to calculate the rank of network nodes [40]. To demonstrate this cost, the objective function uses different metrics such as energy consumption, hop count, or quality of the proposed paths [39].

For clarity and completeness purposes, we provide an overview of RPL protocol. The overview covers RPL traffic flow and modes of operation, topology formation, RPL control messages, trickle timers, and, finally, RPL security.

2.1.1 Traffic Flows in RPL and Modes of Operations

RPL supports various traffic directions; the first and most vital direction is upward routing or multi-point-to-point (MP2P) [40]. In this direction, the packets are sent by network nodes towards the root [41]. The second supported RPL flow is routing downward or point-to-multi-point (P2MP), which is from the root to any node on the network [41], [40]. The last RPL flow is point-to-point (P2P) routing which refers to sending packets between any nodes in the DODAG. This flow is achieved by sending the packet upward to the nearest common ancestor (or to the root in non-storing mode) and then downward to the destination node. This is done to enable traffic from one point to another point in the DODAG. A root must be able to direct packets to a specific location in the network to facilitate P2P activity. Routing tables to

destinations may also exist on network nodes. Until it encounters an ancestor with a known path to the destination, a packet flows in the direction of the root. That common ancestor might be the DODAG root in the most limited scenario (where nodes cannot hold routes) [41].

RPL supports two modes of operation, namely, storing and non-storing modes [41]. In the storing mode, RPL keeps a downward routing table at each node. The routing traffic between two different nodes travels only as far as a common parent [40]. The storing mode has a limitation related to the size of the routing table. Nodes with lower ranks have larger routing tables. In addition, RPL fails when the routing table is full, and a routing entry needs to be appended [42]. In the non-storing mode, all traffic is sent to the root [43]. The root uses source routes to send traffic to leaf nodes [41]. There are many challenges in this mode, including limited memory of DODAG root to handle all network nodes. Another challenge is the overhead, as it requires more compute cycles [40].

2.1.2 Topology Construction

The DODAG construction is shown in Figure 2.2. The root node broadcasts its information using the DODAG information object (DIO) message. Nodes within the communication range of the root will receive the sent DODAG Information Object (DIO) message. Typically, when a node receives a DIO message, it evaluates the routing information, such as RPL instance, version number, object function, and mode of operation that represents the network information. DIO messages also carry information about the sender, including node ID and node rank. Therefore, any node

should add its routing information, including the rank, before sending DIO messages [41], [42], [40].

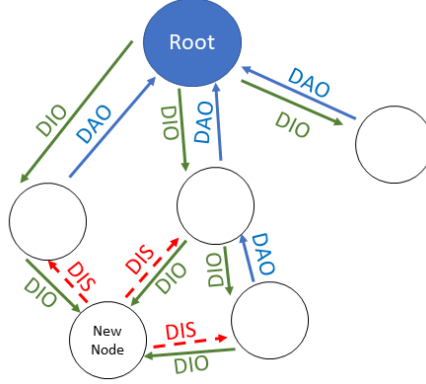


Figure 2.2: *The DODAG construction mechanism*

To join a DODAG, a new node sends the DAG information solicitation (DIS) message. Nodes within the communication range of the new node reply with a DIO message carrying node ID, objective function, and node rank [40]. Until the new node receives a DIO message from one of its neighbor nodes, it constantly broadcasts DODAG Information Solicitation (DIS) messages at a set interval. This interval may vary with different RPL implementations. The new node stops sending DIS messages after receiving DIO messages from neighbor nodes and starts to view all senders as prospective parents. Alternatively, a new node can wait to receive DIO messages from its neighbors without sending a DIS message. The period between two consecutive DIO messages is dynamic, and the trickle timer determines this interval [44].

When a joining node receives a DIO message, it calculates its rank by considering a given objective function that aims to optimize energy consumption, hop count, or quality of the proposed paths [40]. The main purpose of the objective function is to

determine the rank of each node within the DODAG. Therefore, the root node is the sink with the minimum rank [41]. Additionally, the joining node prioritizes the nearby nodes as prospective parents in an ordered list [43]. In DODAG, each node selects the preferred parent, which is the node that offers the lowest cost or the minimum rank for this node [41], [40], [43]. decide whether to join this DODAG [39].

Depending on RPL’s mode of operation, the joining node may participate in downward routing by sending a Destination Advertisement Object (DAO) message to its preferred parent or to the DODAG root [40]. In storing mode, each node maintains a routing table that maps all reachable destinations in its sub-DODAG to their corresponding next-hop nodes, as discovered when receiving DAOs. In non-storing mode, the DAO is delivered directly to the root. When the root receives the DAO, it adds the node to its routing table and stores the parent-child relationship, which is later utilized for source routing. The sender of the DAO message may optionally request their destination to send an acknowledgment. The Destination Advertisement Object-Acknowledgement (DAO-ACK) message is sent back to the DAO sender.

2.1.3 RPL Control Messages

RPL provides the following four primary types of control messages.

- DODAG Information Solicitation DIS: This message is employed to request DIO from an RPL node. Usually, a node probes neighbor nodes in nearby DODAGs using a DIS packet. The structure of the DIS message is shown in Figure 2.3.
- DODAG Information Object DIO: This message contains data that enables a

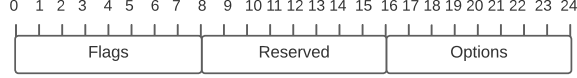


Figure 2.3: *The structure of DIS message*

node to locate an RPL instance, learn about its configuration settings, choose a DODAG parent set, and keep the DODAG up to date. Specifically, DIO packets are used to construct MP2P routing paths as well as assist new nodes in finding neighboring DODAG [40], [45]. Figure 2.4 illustrates the structure of DIO messages. The version number denotes the DODAG version number and keeps all nodes in sync. This number normally increases upon each network information update. The rank details the node's rank that sent the DIO message. The MOP determines the RPL's mode of operation. The DODAG-ID is a unique value for each DODAG identified specifically by a DODAG root.

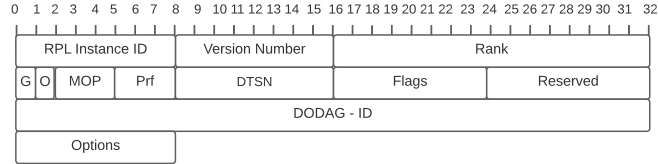


Figure 2.4: *The structure of DIO message*

- Destination Advertisement Object DAO: This message is utilized to spread destination information upwards. In storing mode, the DAO is sent as a unicast message to specific parents, while the DAO is sent as a unicast message to only the DODAG root in non-storing mode. RPL facilitates P2MP traffic by relaying on DAO packets [40], [45]. As modifications are made to the underlying DODAG

topology, destination advertisements may update routing tables [40]. When the root sends messages to a descendant node, it will utilize the routing table based on the received DAO messages [39]. The structure of the DAO message is shown in Figure 2.5.

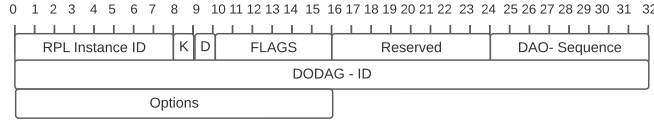


Figure 2.5: *The structure of DAO message*

- Destination Advertisement Object-Acknowledgement DAO-ACK: In response to a unicast DAO message, a DAO recipient (a DAO parent or DODAG root) sends the DAO-ACK message as a unicast packet [41], [40]. The structure of DAO-ACK message is shown in Figure 2.6.

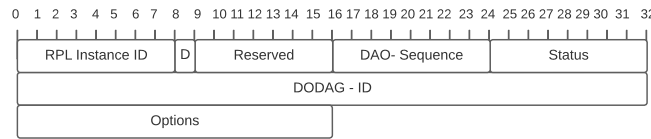


Figure 2.6: *The structure of DAO-ACK message*

2.1.4 Trickle Algorithm

In RPL, network nodes frequently announce their routing information using DIO messages. The frequency between these messages is dynamic, and it is determined by the trickle algorithm [44]. The trickle algorithm is designed to minimize sending redundant

messages, which is achieved by checking the consistency in received DIO messages. As a result, the trickle timer allows nodes to exchange routing information only if an inconsistency is detected. In order to send minimum redundant messages when DODAG exhibits inconsistency, the trickle algorithm sends DIO messages in short intervals. However, when nodes receive consistent information about their DODAG, the trickle algorithm increases the interval between transmissions until a preset maximum interval is reached. This mechanism is implanted using three parameters [44]: minimum and maximum interval times and a consistency threshold. The minimum and maximum intervals define the shortest and longest time between sending consecutive DIO messages. The consistency threshold determines the number of consistent messages before forcing the node to suppress its DIO information. DODAG root determines the values of all these parameters [44], [46]. Accordingly, when a node detects DODAG inconsistency, such as a change in a neighbor rank, the trickle algorithm resets to the minimum interval between two DIO messages. In comparison, when the network is stable, the time between two DIO messages exponentially grows until it hits the maximum set interval [46].

2.1.5 Objective Function

The IETF has officially defined two objective functions: Objective Function Zero (OF0), which considers hop count as the routing metric [47], and the Minimum Rank with Hysteresis Objective Function (MRHOF), which uses paths that minimize Expected transmission count (ETX) as a metric [48]. In (OF0), the objective function is designed to find the nearest root by selecting the preferred parent based on only

the hop count as a routing metric [49]. Therefore, all the upward traffic goes through the preferred parent with no attempt to perform any load balancing or considering the quality of the link [40].

On the contrary, the (MRHOF) objective function selects the route that minimizes the expected transmission count (ETX) values, where each node calculates the probability of successfully delivering a packet by passing it to a specific neighbor [48]. MRHOF relies on the information that DIO messages advertise for metrics values. For example, the use of MRHOF with the latency metric allows RPL to find stable minimum-latency paths from the nodes to a root in the Directed Acyclic Graph (DAG).

2.1.6 RPL Mechanisms of Repair & Maintenance

To ensure consistency in RPL, two maintenance mechanisms are defined. The first one is the local repair mechanism, and the second one is the global repair. The local repair maintenance mechanism is triggered by several cases. The first situation is when a node has no parent in the parents set. The second triggering case is the inconsistency between the rank of two neighbor nodes (parent and child), which indicates a possible loop. To initiate the local repair, the node should send a DIO message that specifies the current parameters as configured from the DODAG root [41].

The second maintenance mechanism is global repair. This maintenance is initiated by the root by announcing a new DODAG version. Consequently, Nodes in the new DODAG version can select a new position without considering their previous rank in

the old DODAG version. Nodes in a DODAG respond to this global repair with a new version number by resetting their own trickle timer, updating the stored version number, and advertising the new version by multicasting a DIO messages [40].

2.1.7 RPL Secure Modes

To ensure the security of RPL messages in terms of confidentiality and integrity, IETF proposed secure modes for RPL. The proposed secure modes are the pre-installed secure mode and the authenticated mode [41], [40]. The pre-installed mode allows nodes to join the DODAG, process, create, and exchange RPL messages only if they have a shared key. Similar to the pre-installed mode, nodes in the authenticated mode can join the DODAG as leaf nodes by utilizing the pre-installed shared keys. However, nodes must get a key from a centralized authentication authority to take part in the forwarding process [41], [40].

Secure modes provide optional services such as encryption using AES128 to ensure confidentiality for RPL messages. Another optional service is replay protection using a consistency check control message [41], [40]. This message is a secure control message with a non-repetitive nonce value [41], [50]. The main limitation regarding these secure modes is that there are no official implementations yet for them [51], [52], [50]. In addition, symmetric encryption algorithms are unable to provide an authenticated mode where RPL only supports symmetric encryption [41].

2.1.8 Applications built upon RPL

RPL serves as a fundamental routing protocol in various IoT applications, particularly those involving resource-constrained devices and unreliable communication links.

Notable applications built upon RPL include:

1. **Smart Grid Advanced Metering Infrastructure:** RPL facilitates efficient communication between utility providers and smart meters, enabling real-time monitoring and management of energy consumption [53]. Its adaptability to low-power and lossy network conditions makes it suitable for the different environments typical of metering infrastructure systems [53].
2. **Industrial Automation:** In industrial settings, RPL supports wireless sensor and actuator networks that oversee machinery and processes [54]. Its design accommodates the stringent reliability and latency supplies essential for industrial control systems [55].
3. **Home and Building Automation:** RPL supports applications that manage lighting, heating, and security systems in smart homes and buildings. Its support for IPv6 allows seamless integration with other IP-based networks and devices [56], [57].
4. **Healthcare Monitoring:** RPL-based networks are utilized in healthcare for patient monitoring systems, where sensors track vital signs and transmit data to healthcare providers [58]. The efficiency of the protocol in low-power scenarios extends the operational life of medical monitoring devices [59].

5. Environment and Agricultural Monitoring: In the agricultural field, RPL enables soil moisture and nutrient sensor networks to communicate data that inform irrigation and fertilization decisions [60], [61]. RPL is also used in sensor networks that monitor environmental parameters such as temperature, humidity, and pollution levels [60]. Its ability to establish reliable multi-hop communication paths in challenging terrains ensures consistent data collection and transmission across a diverse range of IoT scenarios [61].

These applications demonstrate the versatility and effectiveness of RPL in establishing reliable communication networks across a diverse range of IoT scenarios.

2.2 RPL Attacks

This section highlights RPL attacks studied in the literature, showing the attack mechanisms and their impact on networks.

2.2.1 Blackhole Attack

The attacker node drops packets rather than sending or forwarding them [62], [63]. The impact of this attack appears in the form of a noticeable drop in the Packet Delivery Ratio (PDR) [64], [65] and an increase in E2E delay [66], [64]. In addition, it increases the control message overhead as the frequency of DIO messages being exchanged increases [67].

2.2.2 Clone-ID Attack

During this type of attack, the perpetrator obtains the identification of a valid node and replicates it throughout the network to intercept packets intended for that targeted node [68], [69], [70]. A clone-ID attack has a disruptive impact in terms of diverting packets through an unauthorized node [68].

2.2.3 Copycat Attack

This attack is launched when a malicious node intercepts DIO messages that authorized nodes have broadcasted. This is followed by continuous transmission of previously received DIO messages using a set replay interval [71]. There are two types of copycat attacks. The first one is called a spoofed copycat attack, where the attacker replaces the source IP address in the IPv6 packet with the IP address of the original DIO sender. The second type is called a non-spoofed copycat attack, where the attacker replays DIO messages with its own IP address[72]. This attack decreases the PDR and increases E2E network delay [71], [72].

2.2.4 DAO Inconsistency Attack

When a node drops incoming data packets and sends forwarding error packets, it launches a DAO inconsistency attack. This attack aims to create errors in packet forwarding to cause the parent node to discard valid downward paths in the routing table [73], [74]. According to the RPL standard, when a message to a downward destination address includes a forwarding error, the sender node considers the address

as unreachable and sends an appropriate No-Path DAO message[41]. This attack highly decreases the PDR and significantly increases energy consumption [75].

2.2.5 DAO Insider Attack

In this attack, the attacker node frequently sends DAO messages to its parent nodes to force ancestor nodes to flood the network with DAO messages [76], [77]. As a result, this attack harms energy resources and increases E2E latency [78], [77].

2.2.6 Decreased Rank Attack or Sinkhole Attack

The adversary node presents itself with a lower rank than other nodes in an attempt to position itself as a more desirable parent. The attacker node accomplishes this by creating a false route using a different objective function (OF) to lure legitimate nodes or by altering the rank value in DIO messages. Consequently, neighboring nodes are misled into believing this malicious node is a sink[79], [80], [68]. The consequence of this attack is a significant disruption to RPL's DODAG [81], which causes high network traffic toward the attacker node [68].

2.2.7 DIO Suppression Attack

The malicious node receives a DIO message from a legitimate node and replays it repeatedly at a fixed frequency during the attack. This flood of consistent DIO messages causes the victim node to suppress its own DIO transmission due to the trickle algorithm specification. As a result, the victim node stops sending its DIO messages after

receiving enough consistent DIO packets [82], [83]. This attack degrades the quality of routing services by reducing the PDR and network path stretch [84] because the continuous suppression can cause some nodes to be hidden, and consequently, some routes will remain undiscovered [82], [85].

2.2.8 DIS Flooding Attack

The malicious node floods nearby nodes by broadcasting DIS messages during this attack, leading to resetting their trickle timer. Another strategy this attack could involve is targeting a specific neighbor by sending DIS messages and initiating its response in the form of a DIO message [86]. The effect of this attack is to increase the exchange of control packets, leading to network congestion. This attack has more impact when the attacker broadcasts DIS messages. As per RPL standard [41], broadcasting DIS messages leads to resetting the receiver’s trickle timer, which also leads to flooding the network with DIO messages and increases the power consumption of nodes [86].

2.2.9 DODAG Inconsistency Attack

This type of attack occurs when a node alters the packet option header by activating both the “O” and “R” flags [87]. According to RPL standard [41], setting the “O” flag indicates sending the packet to a descendant node. As a result, the attacker sends the packet to an ancestor node. Setting the “R” flag means an error is detected. Consequently, the receiving node drops the packet because of the DODAG inconsistency

forming a blackhole in the next hop [88], [87]. The consequence of this attack is a decrease in the PDR as the attacker forces the next hop to drop all packets of the attacker's descendants. This also leads to an increase in message overhead [88], [87].

2.2.10 Hello Flood Attack

If a node continuously sends a DIO message with strong routing metrics or a powerful signal, it can lead to a hello flood attack [89], [90]. As a result, nodes in the communication range of the attacker reset their trickle timer and could decide to join this network through this malicious node. Meanwhile, the attacker might disappear or reduce its transmission power to normal [91]. This attack decreases the PDR because many legitimate packets sent by other nodes will not be delivered [90]. In addition, this attack exhausts the energy of legitimate nodes and increases control overhead, thereby causing congestion in the network [91], [89].

2.2.11 Increased Rank Attack

An adversary node can execute this attack by artificially elevating its rank to a higher value than its actual rank. This deceptive tactic leads to the node showing up farther along the root than its actual position [82]. Similar to the sinkhole attack, the rank is manipulated by modifying the rank of the attacker node by a specific fake value or by using a different objective function to mislead legitimate nodes [92]. The result of this attack is forcing other nodes to choose further nodes as parents, which leads to consuming more energy and causing additional latency [93], [94].

2.2.12 Local Repair Attack

This attack is performed by exploiting RPL to start the local repair process using the two following methods [95]: The first method is by changing the rank of the attacker node to infinitive and broadcasting this rank to all its neighbors [96] (as a result, the neighbor nodes update the rank information of the attacker node and find a new preferred parent towards the root); the second method is changing the DODAG-ID value of the node. In fact, the DODAG-ID is the address of the root, which is unique to each DODAG. When a node changes its DODAG-ID, it leaves that DODAG and belongs to a different DODAG. Therefore, children nodes have to perform a local repair to select a new preferred parent. According to RPL, the node invokes local repair only if the links to all parent lists are lost [97]. This attack exhausts the energy resources and increases control overhead [95]. As demonstrated [97], another impact is packets dropping because of the temporary loss of valid routes.

2.2.13 Neighbor Attack

To perform this attack, the attacker uses a received DIO message and forwards it without modification. Therefore, neighbor nodes will be misled by information in the sent DIO. [98], [99]. Neighbor attack increases E2E delay and causes more exchanged control messages [100], [98]. It also leads to choosing non-optimized routing paths [99].

2.2.14 Route Table Falsification Attack

This attack happens in a storing mode when the attacker forwards a forged or modified DAO control message. The purpose is to add false routing entries in the routing tables of neighbors and to advertise fake routes to other nodes [3]. This attack causes E2E delay, packet drops, and a depletion of resources [101], [3],[2].

2.2.15 Sybil Attack

Multiple redundant DIS messages with fake identities are being sent out by a malicious node during this attack, which causes the legitimate nodes to reset their trickle algorithm [45], [84]. Because of the attack, the routing topology is disturbed [102], [103]. Also, it leads to the initiation of maintenance procedures [45]. Moreover, this attack causes a swift depletion of limited energy resources because a large number of control messages are being broadcasted [84].

2.2.16 Selective Forwarding Attack

The adversary in this attack forwards only a selection of packets and drops everything else [70], [104]. To accomplish this attack, the malicious node checks incoming packets and then allows sending some of them and dropping others based on the class of packets. According to [104], the class of packets indicates either dropping data packets or control packets but not both. This attack leads to disruption in the routing path [70], especially when the repairing mechanism does not discover or detect this attack [100]. Moreover, the selective forwarding attack destabilizes the data flow in the

network [70].

2.2.17 Version Attack

In this attack, the malicious node changes the version number in the DIO message to a higher number [89]. According to RPL, only the root can update the version number to control the global repair mechanism and to ensure that all nodes are within the same routing state. However, there is no mechanism to check the integrity of the version number in received DIO messages [105]. Therefore, when nodes receive a malicious DIO with a new version value, they respond to global repair by performing the following processes: resetting the trickle timer, updating the version to the new value, and advertising this new version by sending DIO messages to their neighbor nodes [106], [107]. Even though the global repair mechanism guarantees a loop-free topology, it is costly in terms of the number of exchanged messages. Nodes that send DIO messages with the old value of the version indicate the node will not be a preferred parent by other nodes. However, data packets from nodes with the new version are not allowed to transit through nodes with the old version to avoid loop creation [106]. This attack leads to inconsistent topology with routing loops. It also has a negative impact on the energy resources of the nodes and channel availability [89]. Furthermore, the location of the attacker node determines the impact of this attack, as it critically increases when the attacker is a leaf node [108].

2.2.18 Wormhole Attack

In DODAG networks, manipulating the traffic between various segments is often accomplished through a wormhole attack. This attack involves a group of malicious nodes working together to achieve this attack [109], [110]. According to [111], this attack occurs in multiple ways, including the following: packet encapsulation, where two attacker nodes collaborate to pretend they are nearby and form a tunnel to exchange packets between them directly (as a result, other nearby legitimate neighbors transmit their packets through them, which may cause a delay in transmission or loss of packets). Another way is called packet relay, where attacker nodes connect two legitimate nodes that are not neighbors (as a result, the victim nodes, which are unreachable by each other, are connected through a malicious node by relaying the packet between two out-of-range legitimate nodes) [112]. This form of wormhole attack is similar to a neighbor attack. Moreover, an out-of-band channel is another technique used for launching a wormhole attack [111], [112]. In this way, launching the attack depends on the link layer, which is out of the scope of RPL attacks. The effect of this attack is causing a larger E2E delay [110] in addition to high packet losses, according to [113].

2.2.19 Worst Parent Attack

The adversary, in this attack, selects a node as a parent that maximizes its rank in DODAG [93], [19]. This attack leads to an increase in the expected transmission count not only for the attacker node but also for all the nodes in the sub-DODAG of the malicious node [114]. Also, it leads to additional delays [115], and in terms of the

topology, this attack could create routing loops [115].

2.3 Existing Taxonomies of RPL Attacks

This section discusses several proposed taxonomies to classify RPL attacks. The first taxonomy to classify routing attacks was proposed by [2], where the basis of classification considers the attack target and attack impact. Work done [2] introduced three different categories as shown in Figure 2.7. The first category is network resources draining attacks. These attacks carry redundant processing to deplete the network resources. Direct and indirect attacks are the two subcategories that have been created within this category. Direct attacks included hello flood, DIS flooding, and routing table overflow attacks. On the other hand, the subcategory of indirect attacks involved DAG inconsistency, version, and increasing rank assaults.

The second category discusses attacks targeting the network topology by manipulating the DODAG structure. Topology attacks are classified into sub-optimization and isolation attacks. Attacks utilizing suboptimal routes are sinkhole, worst parent, wormhole, routing information replay, and routing table falsification attacks. In isolation attacks, blackhole and DAO Inconsistency attacks are examples of this type of attack.

The third category represents attacks that target the traffic of the network by either misusing or eavesdropping. Sniffing and traffic analysis attacks are instances of eavesdropping attacks. Sybil and decreased rank attacks are illustrations of misappropriation attacks.

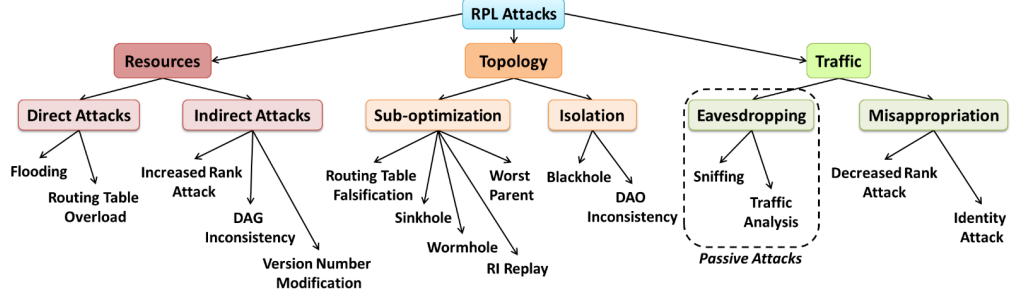


Figure 2.7: *RPL attacks taxonomy [2]*

Authors in [24] extended the work done by [2] by adding more attacks to the same taxonomy. For resource attacks, the local repair attack was added under the direct attack subcategory. Authors in [24] also added attacks in the topology attacks category, including selective forwarding and neighbor attacks are added as a sub-optimization attack [24].

In spite of the role of this taxonomy in highlighting the impact of each RPL attack and the damage that this attack aims, it has some gaps. For instance, the taxonomy places decreased rank attacks in traffic attacks, whereas the sinkhole attack is placed as a topology attack. In fact, these two attacks are identical. Another placement gap is placing the DAG inconsistency attack under indirect attack, though it has an isolation role similar to the blackhole attack [116]. An additional gap in this taxonomy is placing increased parent attack as a resource attack despite its topology impact [92], [10].

Authors in [3] introduced a classification of RPL attacks and mitigation methods. This paper distinguishes attacks inherited from WSNs different from those attacks that utilize RPL vulnerabilities as shown in Figure 2.8. WSN-inherited attacks include

blackhole, selective forwarding, sinkhole, wormhole, hello flood, sybil, and clone-Id attacks. In comparison, RPL-specific attacks include rank, version, local repair, DIS flooding, neighbor, RPL-storing mode, DODAG inconsistency, as well as replay attacks [3]. Additionally, authors in [10] extend the classification proposed in [3] by adding attacks that target 6LoWPAN attacks as well.

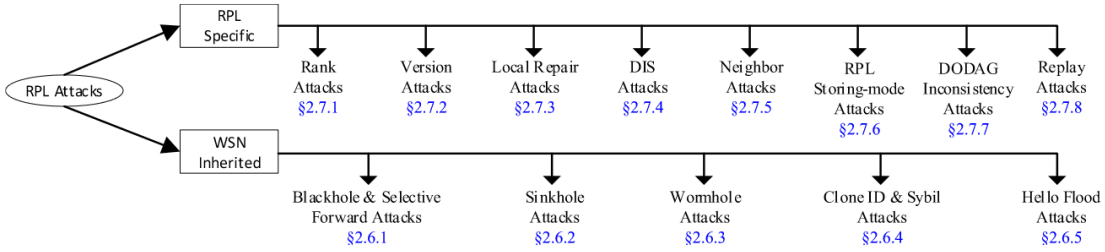


Figure 2.8: *RPL attacks taxonomy [3]*

Authors [117] presented RPL attacks taxonomy based on threats to confidentiality, integrity, and availability. Threats to confidentiality include attacks such as sniffing and traffic analysis and neighbor attacks. Threats to integrity cause network damage, such as inconsistency, sub-optimality, or isolation. Such attacks include blackhole, DAO Inconsistency, replay, spoofing, clone-ID, rank, version, and routing table falsification attacks. Lastly, threats to network availability include DIS, flooding, identity, replay, and denial of service attack [117]. In fact, the classification introduced by [117] was inspired by [118] that introduced a threat analysis based on the missed security service in RPL. According to this threat analysis, [118] classified RPL attacks based on missed security services. Authentication failure-related attacks are the first category. Node impersonation, dummy nodes, and node resource spam belong to this category. Attacks prompted by a breach of confidentiality are the second attack cat-

egory. This class includes routing exchange exposure as well as routing information exposure. Integrity attacks are the third category of RPL attacks. Routing information manipulation and node identity theft are two types of integrity attacks. Four subcategories compose the last RPL attack category on availability: node resource exhaustion, routing disruption, network traffic forwarding disruption, and communications resource disruption. Later, the taxonomy provided by [4] introduced a classification of RPL attacks based on control messages being used, as presented in Figure 2.9. Accordingly, the taxonomy relates each known attack with the four RPL control messages. Under DIS control message, DIS and Sybil attacks. DIO control messages include the most attacks that are rank, version, neighbor, DODAG inconsistency, hello flood, selective forwarding, sinkhole, wormhole, and routing choice intrusion attacks. DAO packets are susceptible to routing table flooding, routing table falsification, local repair attacks, and DAO attacks. The DAO-ACK control message is also susceptible to fake ACK attacks. In addition, this taxonomy has placed replay attacks under all control messages [4].

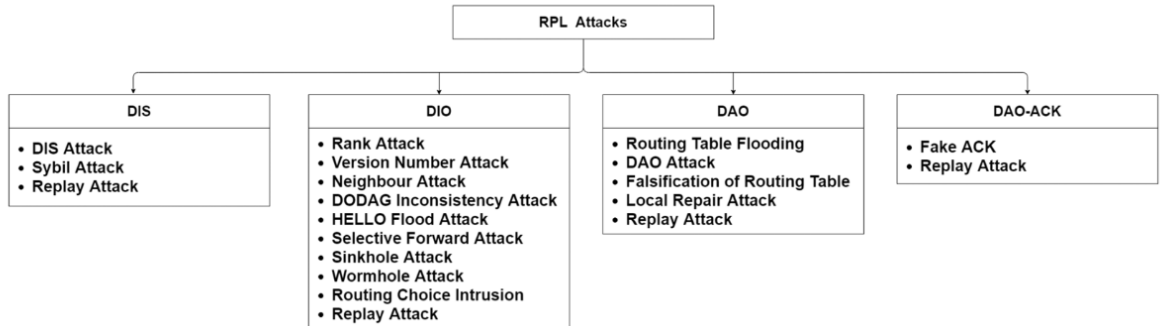


Figure 2.9: *RPL attacks taxonomy [4]*

For a thorough rundown of RPL attacks taxonomies, Table 2.1 presents the tax-

onomies and their corresponding classification base.

Table 2.1: *Summary of RPL attacks taxonomies*

Reference	Classification base
[2]	Attack target and impact
[3]	Attacks origin
[117]	Threatened security service
[4]	Used control messages

2.4 Security Solutions for RPL attacks

In this section, we show various solutions to security attacks against networks, including prevention methods, detection methods, and mitigation methods [14], which are illustrated in Figure 2.10.

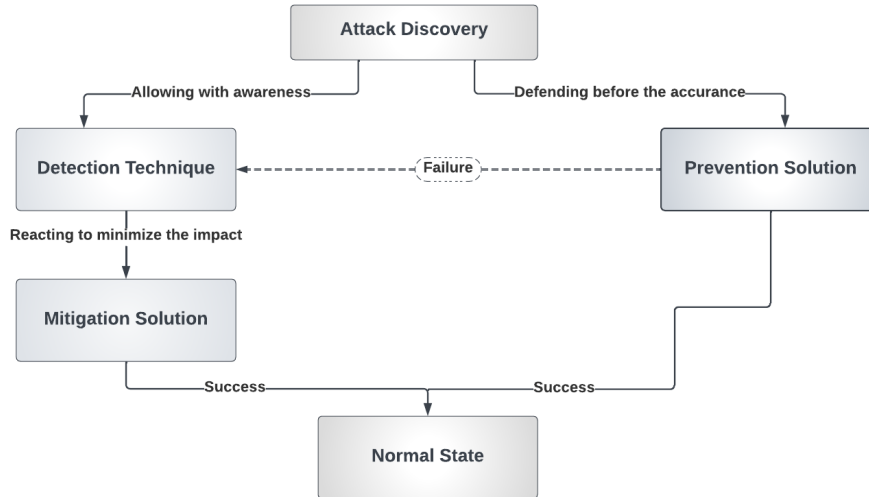


Figure 2.10: *Security solutions roles*

2.4.1 Prevention Solutions

The first is a prevention solution, which offers complete defense against a specific attack. This type of solution aims to use techniques that protect the network from the targeted attack before it happens. For instance, RPL defines two secure modes, namely “pre-installed” and “authenticated” modes, where nodes are equipped with pre-installed keys to apply hashing in a Message Authentication Code (MAC) or add a digital signature [41]. Another prevention solution was introduced in [119], where the hash function (SHA-2) is used to ensure source authentication and prevent replay attacks. However, offering prevention solutions against all attacks is almost impossible, especially in wireless environments and heterogeneous devices like IoT [9].

2.4.2 Detection Solutions

The second solution is the detection system, which offers a system with the ability to distinguish any suspicious behavior in a network and trigger an alarm when an attack is present [14]. Typically, the detection solution appears when it becomes impossible to fully prevent attacks either by the failure of the prevention solution or, in some cases, applying a protection solution becomes a burden on the network in terms of resources [11]. Current studies reveal that IDSs got more attention in the research community than mitigation or prevention systems [120], [17], [18]. For instance, Simoglou et al. [18] focused only on IDSs and the design requirements for IDSs based on the nature of the attack on the protocol and its impacts like control overhead and network performance. Another study was introduced by Pasikhani et al.

[17], where a systematic literature review explores attacks against RPL protocol and the related IDSs to identify these attacks on networks. Likewise, Nandhini et al. [120] performed a comparative study on recent IDSs that provide solutions against RPL-based routing attacks. We will highlight some classifications of detection solutions for RPL attacks where these systems are classified based on either placement type or detection approach.

Placement Based

- **Centralized Placement:** In this type of detection system, all control processes are managed in a central location. It can be located in the root node, or it can be configured at a dedicated host [11], [3]. One of the main benefits of this detection solution is its access to unlimited computational resources. However, a major drawback is the system's complete failure in the event of dedicated device loss [11]. Another drawback appears in adding additional traffic in the network to feed the detection system with the required data [17]. We will demonstrate some centralized-based detection solutions that have been proposed. Aydogan et al. [90] introduced a centralized method using genetic programming to detect hello flood and version attacks. Similarly, Jmaiel et al. [121] developed a centralized anomaly-based rank attack detection system to safeguard an IoT network, employing a one-class support vector machine algorithm. In addition, Napiah et al. [122] proposed a centralized-based system utilizing a IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) compression header as the feature for the machine learning algorithms. Then, the rule or signature

created by the machine learning algorithm is placed at the border router device for detection of potential attacks. The evaluated attacks are hello flood, sinkhole, and wormhole attacks [122].

- **Distributed Placement:** This type of system relies on network nodes to play the role of monitoring the network to detect malicious nodes [3]. The advantage of this approach is that the detection is faster because the decision is local, and it does not wait for a response from a centralized facility [11]. To illustrate this approach, Pu et al. [45] apply a bloom filter array to detect sybil attacks by hashing each network node's identifier and then distributing them in a new packet called BF-DAO. Another solution was introduced by Mayzaud et al. [123], which uses a passive monitoring system. In this system, monitoring nodes use simple statistics of many features to evaluate the network like Instance ID, DODAG ID, DODAG version, node's rank, and the objective function of each node [123]. This method is evaluated under DODAG inconsistency attack. Moving to another distributed detection method that is proposed by [124]. The foundation of Müller's system is Kernel Density Estimation, which is used to obtain the density function f and then deploy the pre-trained models to network nodes. The considered features are the number of control messages, DODAG versions over a specific time window, and UDP forward Ratio. This method detected version, hello flood, and blackhole attacks.
- **Hybrid Placement:** The system is placed in multiple locations looking to benefit from the previous placement approaches [3]. For example, the Heartbeat Intru-

sion Detection System (IDS) [67] falls into this category. In this system, a root or dedicated node periodically sends out UDP requests, and the other nodes are required to respond to these requests. To detect blackhole and selective forwarding attacks, the detection method utilizes UDP messages and RPL Internet Control Message Protocol version 6 (ICMPV6) ECHO requests and replies. Another hybrid placement system has been introduced by Verma et al. [125]. In this system, sniffers are directly connected to the sensor for traffic collection. The collected packets are stored in a centralized repository and analyzed by an engine responsible for classifying traffic as either normal or part of an attack. The analysis engine employs various ensemble-based machine learning classifiers such as Boosted Trees, Bagged Trees, Subspace Discriminant, and RUS Boosted Trees to detect RPL attacks, including sinkhole, blackhole, sybil, clone ID, selective forwarding, hello flooding, and local repair attacks [125]. Moreover, Nikam et al. [126] developed a hybrid placement opinion metric intrusion detection mechanism against sybil, packet flooding, and selective forwarding attacks. In the developed mechanism, each node uses opinion metrics to compute the average belief value for every neighbor in the network, and the border router detects the malicious node based on the degree of disbelief.

Detection Method

- Anomaly Based: Anomaly-based IDS is used for monitoring nodes to ensure that they have normal behavior, and any abnormal activity is classified as malicious behavior. Therefore, the IDS needs to know the baseline of the network

to categorize nodes into normal and attackers [20]. A distinguishing feature of this system as an anomaly-based IDS is its ability to detect new types of attacks, which enhance the network’s security [11]. To illustrate this approach, an anomaly-based detection system was introduced by Verma and Ranga [71] that tackles the copycat attack using an outlier detection. The feature set considered in this study is the number of DIO packets and the time interval between successive DIOs. Moreover, Thamilarasu and Chawla [127] used deep learning in their solution. For this model, a deep belief network is used to fabricate the feed-forward deep neural network to detect blackhole, sinkhole, and wormhole attacks [127]. Another anomaly-based IDS is proposed by Budania et al. [128]. The introduced solution utilized unsupervised explainable artificial intelligence to detect sinkhole and black hole attacks with a high F1-score of over 95%.

- **Signature Based:** This type of system detects attacks considering the attacks’ signatures. Several researchers have introduced signature-based solutions. Yet this detection approach consumes fewer resources than other approaches of detection systems; it does not have the capability to detect new attacks [3]. For instance, Sahay et al. [129] proposed a detection system that detects blackhole and selective forwarding attacks based on the drop rate as a signature of these attacks. Another signature-based detection method is proposed by Iqbal et al. [130] that detects sinkhole and increased rank attacks. The signature used in identifying the mentioned attacks is the relation between the node’s rank and its parent’s rank. Additionally, Mayzaud et al. [131] developed a detection so-

lution that identifies version attacks based on the difference between the node's version and the root's version.

- **Specification Based:** The specification-based detection systems detect attacks when compromised nodes behave differently than the one stated in the protocol. This type of IDS depends on the possibility of obtaining adequate specifications from the standard and the available traces of the traffic. For instance, Deshmukh-Bhosale et al. [112] developed a specification-based IDS to tackle the wormhole attack utilizing link quality between nodes. The proposed method calculated the distance between nodes and identified attackers based on the received signal strength RSSI value [112]. Another specification detection method is introduced by Le et al. [19]. In their technique, the monitoring nodes track node ranks, parent relationships, and topology changes. If a node and its parent violate the ranks relationship and the number of changes exceeds a threshold, a suspicious topology attack is detected. This method detects rank, sinkhole, local repair, neighbor, and DIS attacks [19]. Additionally, Kiran et al. [115] proposed a specification-based mechanism to detect worst parent attacks utilizing the minimum hop rank increase stated in RPL specification. According to [115], If the parent's rank exceeds the minimum hop rank increase of neighboring nodes, the IDS raises a potential attack alarm.
- **Hybrid Based:** This detection system combines multiple detection approaches in one system. Bostani et al. [132] introduced a hybrid method in terms of combining the specification-based IDS with anomaly-based IDS. The goal of

the proposed model is to detect outlier behaviors (as attacks) and identify the malicious nodes that cause these behaviors to detect sinkhole, selective forwarding, and wormhole attacks. The first stage was a specification-based IDS. The second stage was the anomaly detection method that considers the following features: the rate of receiving packets, the rate of dropping packets, the delay, and the maximum hop count. In this stage, the decision is based on a voting classifier [132]. Another hybrid method is presented by Tyagi et al. [96]. The proposed DETONAR system combines signature-based and anomaly-based methods to identify malicious activities in the network by sniffing the traffic. For the features, DETONAR uses both data and control packets. For identifying the attacks, DETONAR relies on historical data of nodes to feed the Auto-Regressive Integrated Moving Average model [96]. It is worth mentioning that this system has been evaluated against 14 RPL attacks [96]. These attacks are blackhole, selective forwarding, sinkhole, hello flood, sybil, wormhole, version, replay, rank, worst parent, DIS, local repair, DODAG inconsistency, and storing modes attacks [96].

To sum up, detection systems are classified based on their placement, which is centralized, distributed, or hybrid placement type. Another classification is based on the detection approach, which can be anomaly-based, signature-based, specification-based, or hybrid-based detection systems. Table 2.2 and Table 2.3 summarize the studies that introduced detection systems for RPL attacks in Scopus, Web of Science (WoS), and ACM, between 2013 and 2023.

Table 2.2: *Summary of detection solutions published in 2013 to 2020*

Year	Ref.	Placement	Method	Attacks
2013	[104]	Distributed	Specification	Selective forwarding
2016	[19]	Hybrid	Specification	Decreased rank, local repair, neighbor, DIS flooding
	[123]	Distributed	Anomaly	DODAG inconsistency
	[131]	Hybrid	Signature	Version
	[133]	Distributed	Signature	Wormhole
	[134]	Centralized	Signature	Version, decreased rank
2017	[135]	Specification	Centralized	Decreased rank
	[136]	Distributed	Specification	Decreased rank, selective forwarding
	[137]	Distributed	Signature	Decreased rank
2018	[122]	Centralized	Hybrid	Hello flood, decreased rank, wormhole
	[132]	Hybrid	Hybrid	Selective forwarding, wormhole, sinkhole
	[138]	Centralized	Hybrid	Decreased rank
	[62]	Centralized	Anomaly	Blackhole
	[81]	Distributed	Signature	Decreased rank
	[99]	Distributed	Specification	Neighbor, sybil
	[126]	Hybrid	Specification	Sybil, flooding, selective forwarding
2019	[139]	Centralized	Anomaly	Sybil, decreased rank
	[91]	Centralized	Anomaly	Hello flood, version, decreased rank
	[140]	Distributed	Anomaly	Neighbor, DIS flooding
	[90]	Centralized	Anomaly	Hello flood, version
	[141]	Distributed	Specification	Decreased rank
	[129]	Centralized	Signature	Blackhole, selective forwarding
	[142]	NA	Specification	Blackhole
	[143]	Distributed	Anomaly	Blackhole
	[124]	Distributed	Anomaly	Version, hello flood, blackhole
	[125]	Hybrid	Signature	Decreased rank, blackhole, sybil, clone-ID, selective forwarding, hello flood, local repair
	[144]	Centralized	Anomaly	Decreased rank, wormhole
	[112]	Hybrid	Specification	Wormhole
	[68]	Centralized	Specification	Clone-ID, decreased rank
2020	[89]	Centralized	Anomaly	Version, worst parent, hello flood
	[66]	Distributed	Specification	Blackhole
	[145]	Distributed	Specification	Sybil
	[95]	NA	Specification	Local repair
	[146]	Distributed	Hybrid	Hello flood, version, decreased rank, blackhole
	[71]	Centralized	Anomaly	Copypcat
	[147]	Hybrid	Specification	Decreased rank
	[148]	Distributed	Anomaly	Decreased rank, local repair, neighbor, DIS Flooding
	[149]	Distributed	Specification	Selective forwarding
	[150]	Hybrid	Specification	DIS flooding
	[151]	Hybrid	Specification	Wormhole
	[152]	Centralized	Specification	Decreased rank, increased rank, worst parent
	[153]	Hybrid	Specification	Decreased rank
	[107]	Centralized	Anomaly	Version
	[154]	Centralized	Anomaly	Decreased rank, version, blackhole, sybil
	[67]	Hybrid	Specification	Blackhole, selective forwarding
	[110]	Distributed	Specification	Sybil, wormhole
	[121]	Centralized	Anomaly	Decreased rank
	[155]	Centralized	Anomaly	Selective forward, decreased rank, version, wormhole, hello flooding
	[156]	Hybrid	Specification	Wormhole
	[157]	Centralized	Anomaly	Wormhole, decreased rank, increased rank, worst parent
	[158]	Centralized	Anomaly	Decreased rank
	[130]	Centralized	Signature	Decreased rank, increased rank

2.4.3 Mitigation Solutions

The last security solution is the mitigation solution. This solution allows the occurrence of the attack, but it reacts to the attack to minimize the impact of this attack on the network [14]. In practice, the mitigation begins with a detection phase of an occurring attack, as shown in Figure 2.10, followed by a quick and effective response [11]. Considering that mitigation solutions accept the existence of attacks and their threat, they focus on how to reduce their harm to the system. In mitigating RPL attacks, many solutions are proposed; one of the first methods is introduced by Raza et al. [195]. The designed system consists of three components. The first and the main one is the 6LoWPAN Mapper, which has the role of reconstructing RPL DODAG to ensure that all nodes are complying with RPL protocol. The second component is the detector component, which detects forged information, any inconsistency in the DODAG graph, and the relationship between the rank of neighbor nodes. The last component is the mini firewall that protects the network against global attackers that launch the network gateway [195]. The RPL attacks evaluated in this method are sinkhole, selective forwarding, and wormhole attacks. Another illustration is the system proposed by Farooq et al. [196] that mitigated Sybil, rank, and sinkhole attacks in IoT by proposing a trust framework that utilizes fog layer capabilities. According to [196], the proposed solution minimizes the energy consumption and messages overheads by 25% to 30%. Additionally, [197], [116], and [88] introduce a dynamic threshold to mitigate inconsistency attacks instead of the static value which is 20 as has been stated in the RPL standard [198]. In their work, the dynamic threshold is set

by each parent node dynamically to adjust the threshold of accepting error packets within a time period based on the number of received error packets as well as the estimated normal error rate [197], [116], [88].

To the best of our knowledge, no study has focused on the mitigation solutions that have been developed to minimize the impact of RPL attacks on the network. References that proposed mitigation solutions for RPL attacks in Scopus, Web of Science (WoS), and ACM, between 2013 and 2023 are shown in Table 2.4 and Table 2.6.

2.5 Taxonomies of Security Solutions

This section points out several taxonomies to categorize countermeasures of RPL attacks. Classifying security countermeasures against RPL attacks has gotten less attention in the research. Authors [3] introduced the first taxonomy of RPL security solutions, which classifies the mitigation methods into five categories based on the adopted technique. These categories are acknowledgment-based, trust-based, location-based, statistical and mathematical-based, and specification-based methods.

The first category is the acknowledgment-based method, which stands on sending a message and receiving an acknowledgment to prevent alternation attacks like using UDP messages and RPL ICMPV6 ECHO requests and replies to detect attacks. The second mitigation category, according to [3], is using trust-based methods where each node monitors its neighbors to create trust relations based on their routing behavior. The third countermeasure class is location-based methods which correlate

nodes to their locations within the network. This method is specifically effective with wormhole and sybil attacks. The fourth category relies on statistical & mathematical methods, such as measuring the disparity of the identities in the received control messages to identify the existence of attacks. The last category of mitigation methods is specification-based methods that tackle some specifications in RPL.

As this taxonomy was the first work that classifies security solutions for RPL attacks based on the implemented method. It is noticeable that it does not distinguish the role of the countermeasure i.e., detection, mitigation, or prevention.

Simultaneously, another taxonomy that classifies the countermeasure methods against RPL attacks is introduced [5]. This taxonomy mainly includes two categories: secure protocols and intrusion detection systems, as shown in Figure 2.11. For the secure protocol, [5] considers three types, namely cryptography-based, trust-based, and threshold-based methods. On the other hand, the intrusion detection systems class includes four types that are signature-based, anomaly-based, specification-based, and hybrid methods.

For secure protocol, [5] considers cryptographic methods like using the hash chain method to authenticate the control packet to prevent various attacks. For a trust-based mechanism, the trustworthiness of each node is evaluated to determine the path selection in the routing process. The last type of secure protocol class is a threshold-based method, where the mitigation solution is built using an adaptive threshold to control RPL features and mitigate RPL attacks.

Alternately, intrusion detection systems (IDS) are the second category of defense solutions in the mentioned taxonomy [5]. IDS is also classified into four types. The

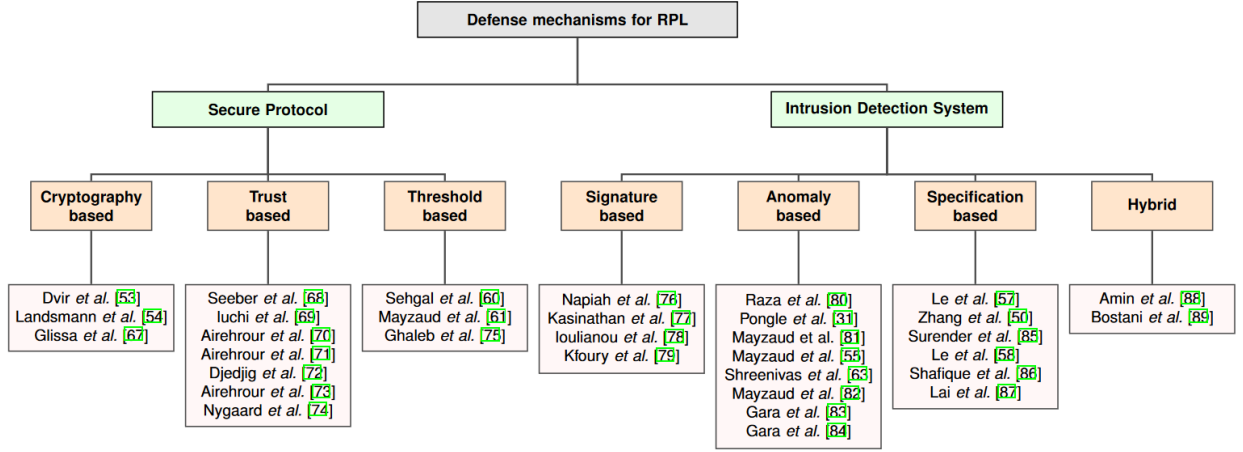


Figure 2.11: *Security solutions taxonomy [5]*

first one is signature-based methods, where the signature of previous attacks on the network traffic is used to identify malicious nodes in the network. The second IDS type is anomaly-based, which detects local anomalies using statistics of many features like Instance ID, DODAG ID, DODAG version, and node's rank to evaluate and detect attacks in the network. The third type of IDS is specification-based IDS which depends on the possibility of obtaining adequate specifications based on the available traces of the traffic. The last IDS type, according to [5], is hybrid IDS where multiple approaches are adopted, like signature and anomaly-based or anomaly and specification-based, are combined. This classification clearly distinguishes between the detection solutions and the mitigation or prevention solutions. For each category, the taxonomy considers the designing approach of the suggested solution. Secure solutions, however, overlook some mitigation techniques. For instance, [83] proposed reducing the sending rate of DIO messages to mitigate sybil and DIO suppression attacks. Also, the study introduced by [82] considered modifying the process of joining

new nodes to include unicasting a DAO message to the root node and waiting for its DAO acknowledgment packet to mitigate DIO suppression attack.

It is worthwhile to mention that some researchers focus on classifying countermeasures against specific types of attacks. For instance, [227] introduces a taxonomy to classify current security techniques against rank attacks. This classification includes cryptographic tools, IDS-based, trust-based, and identification-scheme. On the other hand, multiple researchers have developed several taxonomies that focus on classifying IDSs against RPL attacks. As an illustration, authors [226] introduced a taxonomy for IDSs based on the used machine learning technique. These categories are supervised learning, unsupervised learning, deep learning, and reinforcement learning.

Table 2.5 compiles the taxonomies that classified countermeasures against RPL attacks and provides information on the number of discussed categories and whether they differentiate their role, such as mitigation, or detection.

2.6 Descriptive Models of RPL

Various models have been developed to define the behavior of nodes in RPL according to the standard specification and traffic traces. In this section, we present relevant research that has introduced descriptive models of nodes. Our focus will be on studies that detail the routing behavior of individual nodes in RPL.

2.6.1 Trust Based Models

Trust models have been widely introduced to define the legitimate routing behavior of nodes. The trust value of a node measures its reputation and is defined as a range of values that express the node's behavior over time during communication with its neighbor [253]. Quality of services also plays a role in selecting the secure and optimal route [39]. In fact, multiple proposed trust models often focus on defining certain routing behaviors [10]. For instance, [102] introduced a trust model that uses the received signal strength indicator and a centralized trust scheme to calculate the trust and detect sybil attacks. In another illustration of trust definition, [66] defined trust based on integrating the data delivery in each hop and calculating the link hop value to detect blackhole attacks. One of the drawbacks of trust models appears to be altering native RPL and adding trust as a routing metric. As an illustration of this approach, [239] integrated the trust into RPL by specifying a new objective function called SecTrust. SecTrust-RPL evaluated the trustworthiness of each node to determine the path selection in the routing process instead of utilizing the implemented objective functions [239]. This model has been used to mitigate rank and sybil attacks.

2.6.2 Game Theory Based Model

Other RPL models were developed based on game theory. As an illustration, [148] introduced GAIDS, which consists of two models: a stochastic game-based learning model and an evolutionary game-based learning model. The stochastic game model enables local learning on each node to monitor neighboring nodes based on eight states

that reflect traffic patterns of DIS and DIO messages [148]. The evolutionary game model uses a clustering structure and leverages the state transitions observed using the stochastic game model. Eventually, GAIDS was applied to detect decreased rank, local repair, neighbor, and DIS flooding attacks. It is important to emphasize that the statistical features in this model are dependent on observing the routing states and transitions in the dataset of the trained data [148].

Additionally, [254] proposed another security game model that involves IoT devices, including a potential attacker. The players' strategies revolve around different attack signatures detected by the IDS agent, and probability distribution vectors are defined for both the players and the IDS. These signatures are defined using various features such as packet dropping rate, signal strength intensity, packet send rate, number of requests, and packet's round trip time. The game's payoff matrix between the IDS agent and the attacker is determined, making it a complete information game as the IDS agent knows the attacker player's payoff within its radio range. The study by [254] examines Hello flood, sybil, sinkhole, and blackhole attacks. It is worth noting that this model defines the signature of known attacks rather than the normal behavior of nodes in RPL.

2.6.3 Finite State Machine Based Model

Other models of RPL were based on automata theory, specifically finite state machines. For instance, [19] developed an extended finite state machine for RPL by using a semi-automatic specifying technique to profile RPL operations. The proposed model utilized simulation trace files to profile RPL operations and then converted them into a

set of rules for monitoring data from the network nodes related to relevant properties. The proposed model was used to identify rank, sinkhole, local repair, neighbor, and DIS attacks. It's worth noting that this model relies on simulation results to define the topology changes and ranks' specifications.

The study [74] introduced another finite state machine model called RAIDER to effectively determine attack behavior using a finite number of states and transitions. Monitoring nodes scrutinize network device behavior and translate monitoring actions into transitions. RAIDER verifies device behavior based on a pre-estimated threshold value of rank variances and error packet count. This model is utilized to detect DAO inconsistency, denial of service attacks, and rank-based attacks.

2.6.4 Evidence Theory Based Model

Surendar et al. [255] introduced an estimation model for the compliance degree of a node using Dempster-Shaffer evidence theory. The compliance degree is calculated based on the Beta probability density function using the history of node ranks and received signal strength to rank each adjacent node. Authors in [255] utilized this model to detect sinkhole attacks. The proposed model relies on the specification of rank values and the minimum rank increase between the child and the parent node [255]. To sum up, Table 2.7 summarizes the above highlighted studies, including the modeling approaches, the specification obtained from RPL, and the detected attacks.

Table 2.3: *Summary of detection solutions published in 2021 to 2023*

Year	Ref.	Placement	Method	Attacks
2021	[159]	Centralized	Anomaly	Clone-ID
	[160]	Centralized	Anomaly	Wormhole
	[161]	Distributed	Specification	Selective forwarding
	[162]	Centralized	Anomaly	Decreased rank, DIS flooding, version, worst parent
	[76]	Centralized	Anomaly	Version, DIS flooding, DAO insider
	[163]	Centralized	Anomaly	Hello flood, DIS flooding, decreased rank, version
	[164]	Centralized	Anomaly	Decreased rank
	[165]	Centralized	Anomaly	Sinkhole, version, flooding
	[166]	Distributed	Anomaly	Selective forwarding, version, rank
	[167]	Distributed	Anomaly	Hello flood, decreased rank, blackhole, version
2021	[96]	Centralized	Hybrid	Blackhole, selective forwarding, decreased rank, hello flood, sybil, wormhole, version, replay, increased rank, worst parent, DIS flooding, local repair, DODAG inconsistency
	[168]	Centralized	Anomaly	Version, decreased rank, flooding
	[169]	Centralized	Anomaly	Blackhole, selective forwarding, decreased rank, version, hello flood
	[170]	Distributed	Specification	Decreased rank, increased rank, worst parent
	[171]	Centralized	Anomaly	Decreased rank, increased rank
	[172]	Distributed	Specification	Selective forwarding
	[173]	Hybrid	Specification	Blackhole, selective forwarding
	[174]	Centralized	Hybrid	Sybil
	[175]	Centralized	Anomaly	Version, decreased rank, flooding
	[176]	Hybrid	Specification	Blackhole
	[177]	Centralized	Anomaly	Version
	[178]	Hybrid	Specification	Selective forwarding
	[179]	Centralized	Anomaly	Hello flood, DIS flooding, version, decreased rank
	[180]	Hybrid	Specification	Sybil, wormhole, decreased rank
2022	[181]	Centralized	Anomaly	DIS flooding
	[182]	Distributed	Signature	Blackhole
	[183]	Hybrid	Anomaly	Blackhole
	[184]	NA	Hybrid	Version, blackhole, selective forwarding, hello flood, DIS flooding, DAO insider
	[185]	Hybrid	Anomaly	Decreased rank, version, hello flood, blackhole, sybil, selective forwarding, wormhole
	[186]	Centralized	Anomaly	Wormhole
	[187]	Centralized	Specification	Decreased rank
	[115]	Centralized	Specification	Worst parent
	[45]	Distributed	Anomaly	Sybil
	[188]	Distributed	Signature	Blackhole
	[189]	Hybrid	Specification	Decreased rank, increased rank, worst parent, wormhole
	[102]	Hybrid	Specification	Sybil
	[190]	Centralized	Anomaly	Increased, decreased, worst parent, wormhole
	[191]	Centralized	Anomaly	Decreased rank, wormhole
2023	[98]	Distributed	Specification	Version, neighbor, worst parent, decreased rank
	[103]	Hybrid	Specification	Sybil
	[192]	Hybrid	Anomaly	Wormhole
	[193]	Hybrid	Anomaly	Sybil, blackhole, decreased rank, clone-ID, local repair, hello flood, selective forwarding
	[194]	Hybrid	Hybrid	Sybil, decreased rank, replay

Table 2.4: *Summary of mitigation solutions published in 2013 to 2021*

Year	Ref.	Mitigation Mechanism	Attack
2013	[109]	Node authentication	Wormhole
	[195]	Deleting inactive nodes, Discarding previous information, Recalculating node's rank	Blackhole, Selective forwarding, Clone-ID, sybil, sinkhole
2014	[88]	Modifying options header	DODAG inconsistency
	[199]	Disabling trickle timer reset, Discarding control packets	DODAG inconsistency
2016	[64]	Deleting malicious node entry, selecting another parent	Blackhole
	[200]	Authenticating network information, encrypting data packets, deleting malicious nodes entries	Increased rank, replay, selective forwarding, version, sinkhole
	[63]	Modifying objective function	Blackhole
	[201]	DODAG reconstruction	Sinkhole
	[113]	Discarding DIO message	Packet flooding, wormhole
2017	[202]	Discarding DIO message	Sinkhole
	[203]	Invoking global repair	Selective forwarding
	[204]	DODAG reconstruction	Sybil
2018	[205]	Discarding incoming packets	Clone-ID, packet flooding, selective forwarding
	[206]	Deleting malicious node entry, selecting another parent	Blackhole
	[85]	Discarding control packets	Blackhole, selective forwarding
	[207]	Modifying objective function	Blackhole, sinkhole, hello flood
2019	[208]	Modifying objective function	Packet flooding, version
	[209]	Invoking local and global repairs	Blackhole
	[210]	Node authentication	Selective forwarding, sinkhole
	[211]	Deleting malicious node entry	Wormhole
	[212]	DODAG reconstruction	Sinkhole
	[213]	Modifying objective function	Sinkhole, blackhole, increased rank, selective forwarding, sinkhole, sybil
	[108]	Discarding version update	Version
	[214]	Discarding control packets	Packet flooding
	[75]	Discarding control packets	DIO suppression, packet flooding
	[215]	Modifying objective function	Sinkhole, sybil
2020	[216]	Discarding DIS	DIS flooding
	[217]	Recalculating node's rank	Sinkhole
	[218]	Discarding control packets	Hello flood
	[219]	Adding virtual clusters, mandating storing mode	Blackhole, sinkhole, wormhole
	[220]	Modifying parent selection, discarding version update, node authentication	Sinkhole, version, selective forwarding
	[221]	DODAG reconstruction	Hello flood
	[222]	Discarding DIO message, invoking local repair	Increased rank, sinkhole, worst parent
2021	[223]	DODAG reconstruction	Blackhole
	[69]	Discarding control packets	Clone-ID
	[83]	Modifying RPL timers	DIO suppression, sybil
	[93]	Disabling global repair, local repair, and trickle timer reset. Modifying parent selection.	Version, DODAG inconsistency, neighbor, packet flooding, local repair, replay, blackhole, clone ID, selective forwarding, sinkhole, sybil
	[224]	Node authentication	DIS flooding, version, increased rank, sinkhole
	[225]	Modifying objective function	Sybil

Table 2.5: *Summary of countermeasures taxonomies*

Reference	# Categories	Countermeasure Solution		
		Mitigation	Detection	Not specified
[3]	5			✓
[5]	7	✓	✓	
[17]	4		✓	
[23]	10			✓
[226]	4		✓	
[227]	4			✓

Table 2.6: *Summary of mitigation solutions published in 2021 to 2023*

Year	Ref.	Mitigation Mechanism	Attack
2021	[228]	Selecting another parent	Sinkhole, increased rank, worst parent
	[72]	Deleting malicious node entry	Copycat
	[229]	Discarding control packets	DIS flooding, hello flood
	[230]	Saving RPL state	Local repair , version
	[231]	DODAG reconstruction	Increased rank, sinkhole
	[232]	Modifying RPL timers	DIS flooding
	[86]	Modifying RPL timers	DIS flooding
	[233]	Discarding DIO message	Version
	[234]	DODAG reconstruction	Sinkhole
	[235]	Modifying objective function	Blackhole
	[236]	Deleting malicious node entry	Sinkhole
	[237]	Adding a registration process, encrypting data packets, modifying objective function	Blackhole, neighbor, sinkhole, sybil
	[82]	Mandating DAO-ACK	DIO suppression
	[238]	Modifying objective function	Blackhole
	[239]	Modifying objective function	Increased rank, sybil, sinkhole
	[240]	Discarding control packets	Clone-ID, neighbor, wormhole
	[241]	Modifying objective function	Sinkhole, increased rank, worst parent
	[242]	Mandating DAO-ACK	Neighbor
	[73]	DODAG reconstruction	Dropped Destination Advertisement Object
	[243]	Modifying objective function	Blackhole, sinkhole, increased rank
	[80]	Discarding DIO message	Sinkhole
	[84]	Modifying RPL timers	DIO suppression, sybil
	[78]	Discarding DAO	DAO insider
2022	[111]	Deleting malicious node entry, Node authentication	Wormhole
	[244]	DODAG reconstruction	Sinkhole
	[245]	Discarding DIS	DIS flooding
	[246]	Discarding control packets	Version
	[247]	Discarding DIO message	Sinkhole
	[248]	DODAG reconstruction	DIS flooding
	[249]	Deleting malicious node entry	Sinkhole
	[196]	DODAG reconstruction	Sinkhole, sybil
	[94]	Modifying objective function, selecting another parent	Blackhole, sinkhole
	[250]	DODAG reconstruction, invoking local and global repairs	Sinkhole, increased rank
2023	[114]	Modifying parent selection	Increased rank, sinkhole, worst parent
	[251]	Node authentication	DIS flooding, sybil
	[252]	Discarding DIS	DIS flooding
	[74]	DODAG reconstruction	Sinkhole, DAO inconsistency, packet flooding

Table 2.7: *Summary of RPL models*

Modeling Approach	Reference	Specification	Detected attacks
Trust	[102]	Link quality	Sybil
	[66]	Data delivery per hop	Blackhole
	[239]	PDR	Rank and sybil
Game theory	[148]	Traffic patterns of RPL packets	Sinkhole, local repair, neighbor, DIS flooding
	[254]	Packets exchanging rate and link quality	Hello flood, sybil, sinkhole and blackhole
Finite state machine	[19]	Node's rank, Number of topology changes	Rank, sinkhole, local repair, neighbor, and DIS flooding
	[74]	Forwarding error packet count and rank variance	Sinkhole, DAO inconsistency, and flooding
Evidence theory	[255]	Node ranks and the minimum rank increase	Sinkhole

CHAPTER 3

TAXONOMY OF RPL ATTACKS

This chapter addresses RQ1 by categorizing RPL attacks within an extendable taxonomy to facilitate the classification of future attacks. We categorize RPL attacks based on the similarities of attack vectors and launching methods. Attack vectors refer to the method that attackers use to deliver destructive outcomes and take advantage of vulnerabilities. Recognizing and tracking an adversary’s attack vectors is crucial for designing better countermeasures for current or future attacks [256].

As shown in Figure 3.1, we categorize attack behaviors into four distinct classes: dropping packets attacks discussed in Section 3.1, generating packets attacks illustrated in Section 3.2, modifying packets attacks presented in Section 3.3, and replaying packets attacks shown in Section 3.4. In addition, our analysis in Section 3.5 outlines a comprehensive explanation of RPL vulnerabilities utilized in associated attack vectors. Section 3.6 illustrates an application of the proposed classification to create a detection system capable of detecting RPL attacks based on attack traces.

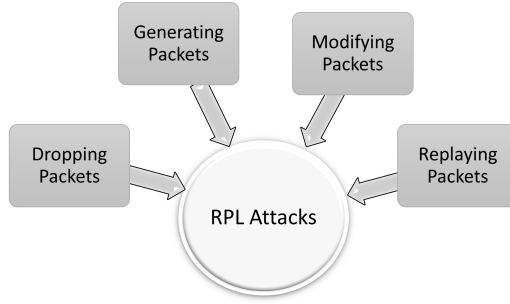


Figure 3.1: *General classification of RPL attacks*

3.1 Dropping Packets Attacks

Any attack carried out by dropping packets falls under this category. Rather than sending the packets to their intended destination, the attacker node drops all or some of the packets in order to successfully launch these attacks. This class, as Figure 3.2 shows, includes blackhole, selective forwarding, and (DDAO) attacks.

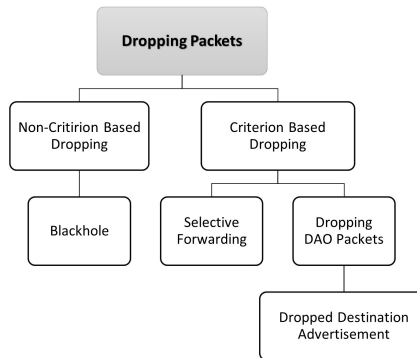


Figure 3.2: *Dropping packets attacks*

3.1.1 Blackhole Attack

The blackhole node maliciously drops packets instead of sending or forwarding them, causing disruptions and potential security breaches [62], [63]. According to [238], [64], blackhole attacks occur when a node drops packets forwarded to it for routing. Blackhole attack is implemented in [257], where all packets that need to be forwarded are dropped.

3.1.2 Selective Forwarding Attack

The attacker in this scenario forwards only specific packets and drops all others [70], [104]. In order to achieve this, the attacker examines incoming packets and forwards only those that meet specific conditions [65], [238]. As a result, the method of dropping packets in selective forwarding attacks is similar to that of a blackhole attack. The difference is that in selective forwarding, the dropping arises based on criteria set by the attacker rather than dropping all received packets.

3.1.3 Dropped Destination Advertisement Object Attack

Another dropping packet attack is (DDAO). To launch this attack, the node attempts to mislead the victim by (a) dropping DAO packets sent by its child node or any of its successors and (b) tricking the victim by pretending to transmit a DAO-ACK message [73]. The differentiation of this attack from other packet dropping attacks lies in the generation of a counterfeit DAO-ACK packet by the attacker and its transmission to the victim node [73].

3.2 Generating Packets Attacks

All attacks involving the creation of packets fall under the category of packet generation attacks. In this type of attack, the attacker generates packets to overwhelm the network and deplete its resources. This category is illustrated in Figure. 3.3, encompasses packet flooding, DIS flooding, clone-ID, sybil, hello flood, and DAO insider attacks.

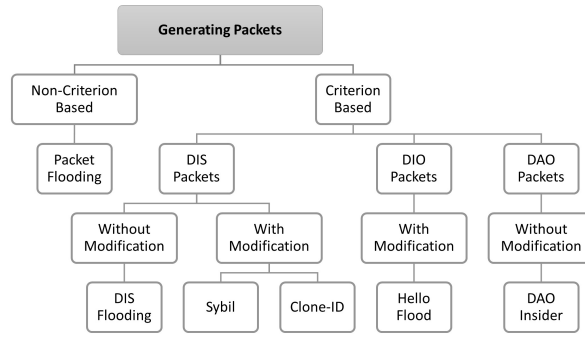


Figure 3.3: *Generating packets attacks*

3.2.1 Packet Flooding Attack

The packet flooding attack entails the generation and transmission of a substantial volume of packets in order to deplete network resources [214], [205], [208], [74]. As described by the authors in [93], the attack is executed by repeatedly sending falsified packets.

3.2.2 DIS Flooding Attack

The DIS flooding attack involves overwhelming nearby nodes by broadcasting DIS packets. This is achieved by sending DIS packets to nodes within its range at regular intervals, prompting neighboring nodes to reset their trickle timer [86], [251]. Another tactic employed in this attack is to direct DIS packets at a specific neighbor, eliciting a response in the form of a DIO packet [86].

3.2.3 Sybil Attack

The malicious node in this attack produces multiple DIS packets with bogus identities. The surrounding nodes reset their trickling algorithm as a result of this illegitimate packet creation [45], [84]. Sybil attackers are modeled as nodes that multicast several DIS packets with false IDs to all their nearby node [83] and [84]. Furthermore, any moving node that transmits multiple DIS packets using a new IPv6 address may be perceived as sybil nodes by its neighbors, according to [204].

3.2.4 Clone-ID Attack

This attack starts similarly to Sybil attacks. [195], [205] indicates that the distinction lies in the attacker's method of obtaining physical IoT devices through device ID extraction. Authors in [68], [70] clarify that clone-ID attackers duplicate the identity of a valid node throughout the network. It has been explicitly shown by the authors in [159] that this attack is dependent on fake DIS packets that carry the cloned identity of a real node.

3.2.5 Hello Flood Attack

Attacking nodes manipulate the generation of hello packets, which is a DIO packet, to launch this attack [91], [10]. A hello flood attack will occur if an attacker sends DIO packets repeatedly with strong routing metrics or a potent signal [89], [90], [221]. As a result, nodes within the attacker’s communication range reset their trickle timer and may choose to connect to the network via the attacker node. These victim nodes must seek out new neighbors because the attacker may vanish or scale back its transmission strength to normal in the meantime [91], [122].

3.2.6 DAO Insider Attack

In order to carry out this attack, the attacking node continuously sends unmodified DAO packets to its parent nodes, causing the ancestor nodes to flood the network with DAO packets [78], [76], [77]. When RPL is in storing mode, the impact of the attack extends beyond its immediate area, and the transmission of DAO packets moves towards the root node [78], [77].

3.3 Modifying Packets Attacks

Any manipulation of RPL control packets creates the modification of packet attacks. This includes alterations to the option header or the DIO packet. As depicted in Figure 3.4, attacks that modify DIO packets by changing the node’s rank include increased rank, sinkhole, local repair, and worst parent attacks. Likewise, modifying network information in DIO messages causes local repair and version attacks. In addition,

attacks using DAO and DODAG inconsistencies are initiated by altering the options header of RPL control packets.

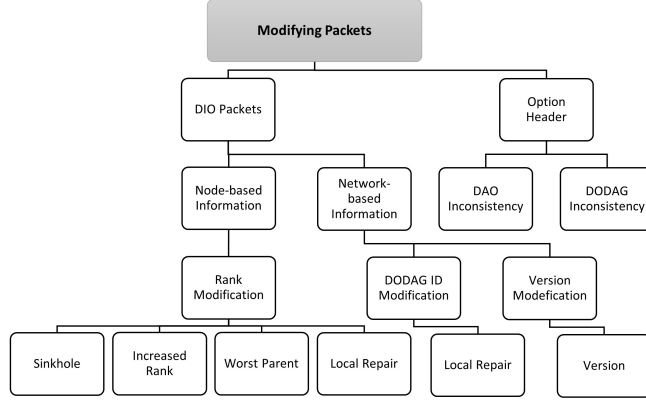


Figure 3.4: *Modifying packets attacks*

3.3.1 Decreased Rank Attack or Sinkhole Attack

This type of attack involves an adversary node manipulating DIO packets to present itself as having a lower rank than other nodes. The goal is to position itself as a more appealing parent [215], [228]. This can be achieved by creating a false route using a different objective function or by altering the rank value in DIO packets [68], [81]. As a result, neighboring nodes may be misled into believing that this attacker is a sink [153], [80], [68].

3.3.2 Increased Rank Attack

Similar to the reduced rank attack, an adversary node increases its rank to deceive other nodes [228], [82]. The difference is that in this attack, the node appears farther from the root than its actual position [82].

3.3.3 Worst Parent Attack

The adversary node in this attack selects the parent node with the highest rank to maximize its own rank value [93], [19]. This is done by manipulating the process of selecting preferred parents [228], [250], [220].

3.3.4 Local Repair Attack

The attacker misuses RPL by changing its rank to infinitive and broadcasting this rank to all neighbors to trigger local repair mechanisms [95], [19], [232]. As a result, neighbor nodes try to find another preferred parent by exchanging a large number of DIO messages. This attack is classified as modifying node-based information as it involves changing the node's rank. Another way to launch this attack is by changing the DODAG-ID value of the attacker node, where DODAG-ID refers to the unique address of the root. According to this change in the DODAG-ID, the attacker leaves that DODAG and joins a different DODAG [95]. Consequently, the attacker's children have to select a new preferred parent [97]. This form of local repair attack is considered a modification of network-based information because the DODAG-ID has been altered.

3.3.5 Version Attack

Version attacker launches this attack by changing the DIO packet's version number to a higher number [108], [220], [224]. When nodes receive a DIO packet with a new version number, they respond to the global repair by resetting their trickle timers to update the version to the new value. They also advertise the received malicious

version by sending DIO packets to their neighboring nodes [106], [107].

3.3.6 DODAG Inconsistency Attack

To launch this attack, the attacker alters the option header in an IPv6 packet by activating both the 'O' and 'R' flags, then it forwards it to its parent [199], [88], [93]. As the attacker sends the altered packet to an ancestor node. Setting the 'O' flag indicates sending the packet to a descendant node, and setting the 'R' flag means an error is detected according to RPL standard [41]. Consequently, the receiving node identifies an inconsistency, drops the packet, and henceforth forms a blackhole in the next hop [88], [87].

3.3.7 DAO Inconsistency Attack

A node initiates a DAO inconsistency attack when it discards incoming data packets and transmits forwarding error packets instead. The attacker alters the IPv6 packet option header by changing the Forwarding-Error flag in order to carry out this attack. Then, the adversary sends the forwarding error packet to its parent to remove valid entries from the routing table [197], [74], [73]. In accordance with the RPL standard, when a node receives a message that includes a forwarding error in the downward destination address, it considers the address unreachable. It sends a suitable No-Path DAO [41].

3.4 Replaying Packets Attacks

The attacks resulting from replaying RPL packets are shown in Figure 3.5. These attacks include replay, neighbor, DIO suppression, copycat, and wormhole attacks.

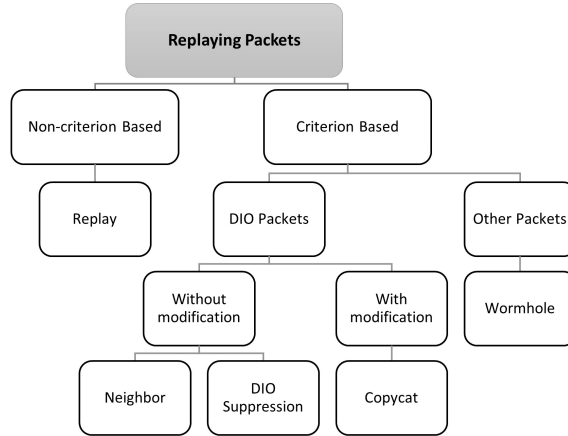


Figure 3.5: *Replaying packets attacks*

3.4.1 Replay Attack

Here, the attacker continuously replays all received RPL packets [93]. According to [200], replay attacks can be launched using two techniques. The first one is when the attacker postpones delivering the received packet and delivers it over a past action [200]. The second method is to replay the received packet immediately but in a repetitive way [93].

3.4.2 Neighbor Attack

In this attack, DIO packets are replayed being altered. In other words, the attacking node repeats an exact duplicate of the received DIO packet Without appending any

new data. [100], [3]. Hence, neighboring nodes receive misleading information from the distributed DIO packets [98], [99].

3.4.3 DIO Suppression Attack

The adversary node in this attack intercepts a DIO packet from a legitimate neighbor and then replays it multiple times at a fixed interval. This flood of repeated DIO packets can cause victim nodes to stop sending their own DIO packets due to the trickle algorithm specification [82], [83]. The neighbor and the DIO suppression attacks employ the same mechanism. But in DIO suppression attacks, the received DIO packet is replayed at a set frequency [258], [82].

3.4.4 Copycat Attack

This type of attack occurs when a malicious node intercepts DIO packets that have been broadcasted by authorized nodes. Then, the attacker node continuously replays these received DIO packets in specific replay intervals [71], [72]. The spoofed copycat attack is the first form of copycat attack in which the attacker substitutes its IP address with the original DIO sender's IP address. The DIO suppression attack is exactly the same as this one. The second type of copycat attack is known as non-spoofed copycat, in which the attacker replicates DIO packets using its IP address [72].

3.4.5 Wormhole Attack

In DODAG networks, the manipulation of traffic between various segments often involves a wormhole attack. This type of attack entails a coordinated effort among a group of malicious nodes to achieve its objectives [109], [110]. According to [111], two attackers can launch this attack via packet encapsulation, i.e., by working together to appear to be nearby and create a tunnel so they can exchange packets directly. As a result, through these attackers, victim nodes that are inaccessible to one another are connected [112]. Moreover, an Out-of-the-band channel is another technique to launch a wormhole attack [111], [112]. In this form of wormhole, a wide bandwidth out-of-band channel is set up between the endpoints to establish a wormhole link. We classify a wormhole attack as a replay attack because it involves leaking network information to a second attacker by replaying received packets.

To summarize, the comprehensive RPL attacks taxonomy is depicted in Figure 3.6. This taxonomy is structured according to the methods employed to execute the attack. It is important to emphasize that this classification is not indicative of the attacks' intended goals or outcomes. The primary focus is to ensure that attack patterns within each category consistently utilize a specific technique.

3.5 Analysis

After showing various launching methods of RPL attacks, we highlight several vulnerabilities and flaws that need to be addressed in RPL. First, RPL lacks mechanisms to verify the validity of the advertised rank value [247]. Nodes can advertise ranks

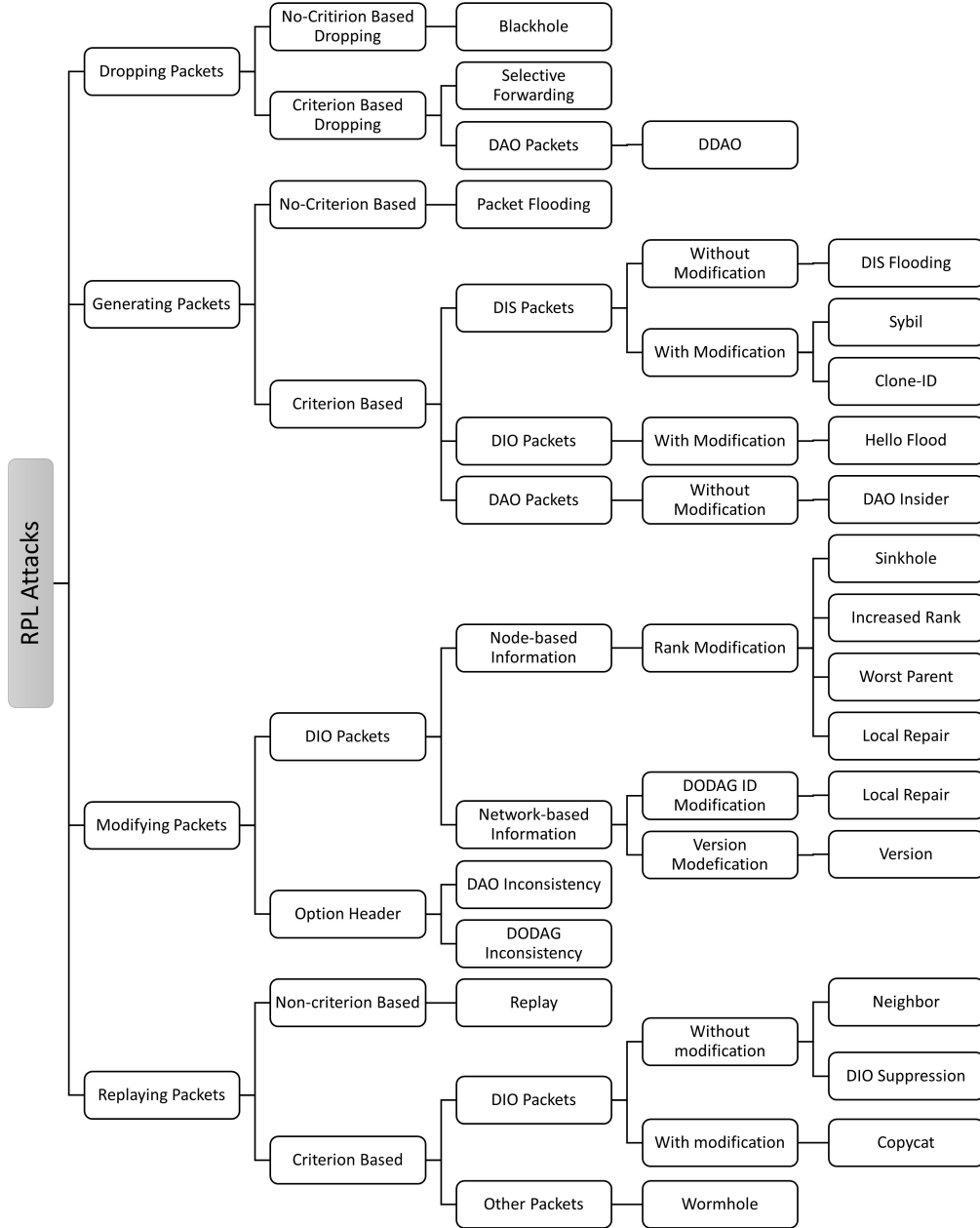


Figure 3.6: *Taxonomy of RPL attacks*

that are incompatible with the objective functions specified by the root. This vulnerability leads to variant rank modification attacks. Similarly, for version attacks, there is no mechanism to check the integrity of the version number in received DIO messages [105]. As per RPL, only the root can update the version number to control the

global repair mechanism [105]. In addition, the current implementation of RPL lacks measures to ensure the freshness and validity of the received information, making it vulnerable to replay attacks [118]. Moreover, while RPL establishes a default lifetime for each route, it lacks a mechanism to monitor the minimum number of control messages received from each neighbor. This susceptibility opens the door for potential adversaries to execute dropping packet attacks [41]. Last, despite that RPL includes a suppression feature to minimize the exchange of control messages [44], RPL does not provide specific counters for tracking received DIS or DAO messages, leaving it vulnerable to various types of flooding attacks [77].

3.6 Application of the Proposed Taxonomy

The proposed classification aims to create a system and method capable of detecting RPL attacks based on attack traces without relying on a training-based solution. This eliminates the need for prior knowledge of normal traffic and traffic with attacks. The objective is to design a system or method that can easily identify and classify almost all types of RPL network attacks without putting a significant computational burden on the detection system for training and classifying traffic. An exemplary embodiment describes IDS configured to detect attacks based on received ICMPV6 packets from IoT devices and to output multiple packets. The taxonomy is a tree structure with branches to matching signatures based on which part of an ICMPV6 packet is misused.

In the proposed taxonomy, a method suggested to detect drooping packet attacks is presented in Figure 3.7. The algorithm examines whether the ratio of incoming

packets to outgoing packets of each node exceeds the dropping threshold and if the number of outgoing packets is zero. If this is the case, then the node is considered a black hole. On the other hand, if the number of outgoing packets is not zero but the ratio of incoming packets to outgoing packets exceeds the dropping threshold, then the receiver node is considered a selective forwarder.

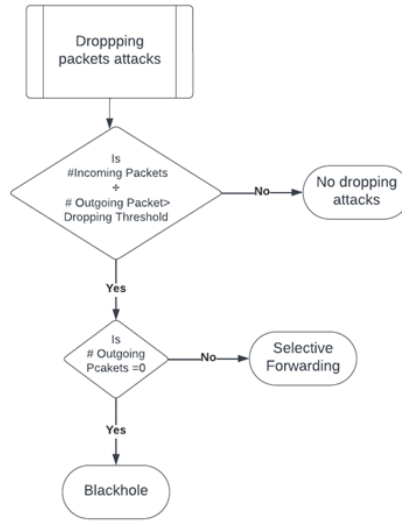


Figure 3.7: *Attack detection flowchart using proposed taxonomy for dropping attacks*

The algorithm for detecting generating packet attacks involves several steps based on the RPL message type, as shown in Figure 3.8. Upon receiving a DIS packet, the algorithm that identifies the related attacks is illustrated in Figure 3.9; the algorithm checks if the sender's ID and Media Access Control (MAC) match the stored values in the probing list. If they don't match, an attack is suspected. To determine the type of attack, the algorithm verifies if the DIS sender ID exists with a different MAC. If so, the node is flagged as a clone ID. If the MAC exists but with a different DIS sender ID, it is flagged as a Sybil. Additionally, if the average time between the received

DIS messages is less than the minimum DIS interval, the sender is classified as DIS flooding attacker.

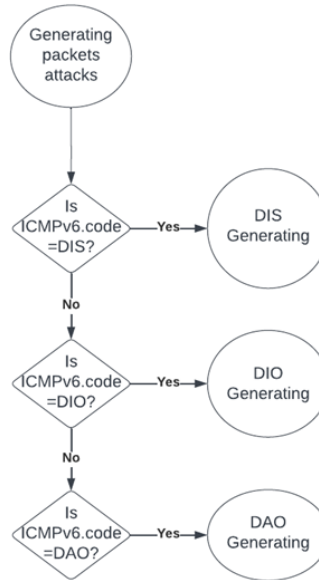


Figure 3.8: *Attack detection flowchart using proposed taxonomy for generating packets attacks*

On the other hand, Figure 3.10 shows the detection method when the RPL packet is DIO. The detection algorithm verifies whether the DIO interval between the received DIO packets is less than the minimum set DIO interval. If this condition is satisfied, the sender node is identified as a hello flood.

Additionally, for generated attacks of DAO packet, Figure 3.11 demonstrates the corresponding flowchart. The detection method determines whether a node has sent more DAO than the maximum allowed per kid. The sender gets labeled as a DAO insider if this is confirmed.

To detect modifying packet attacks, Figure 3.12 shows the IDS is designed based on the proposed taxonomy. The algorithm conducts the following checks: if the RPL

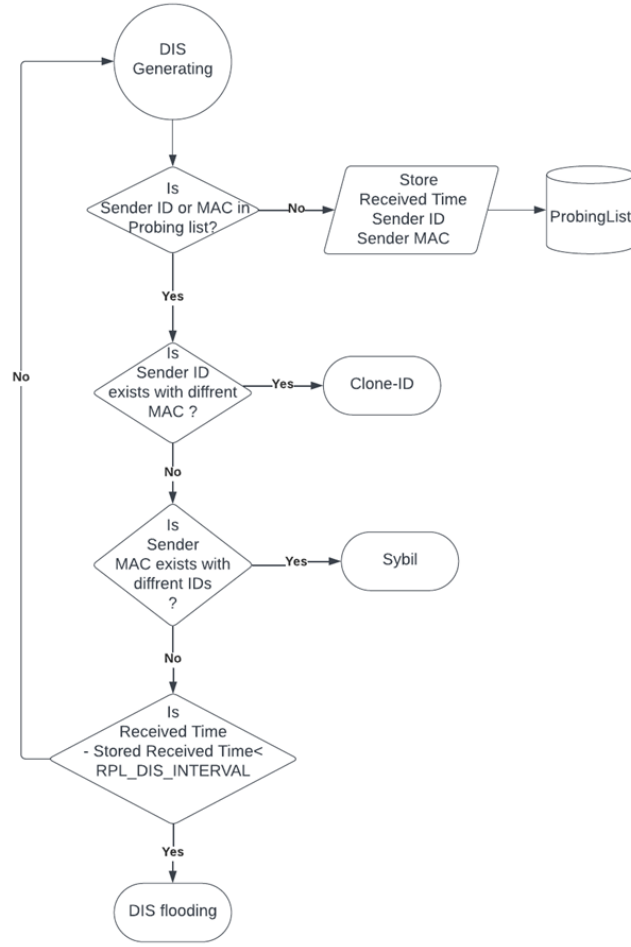


Figure 3.9: Attack detection flowchart using proposed taxonomy for generating DIS attacks

packet has the 'F' flag set, it increments the Inconsistent packet counter. When the Inconsistent packet counter surpasses the maximum number of Inconsistent packet requests, it designates the sender as having a DAO inconsistency. To identify a DODAG inconsistency attack, the algorithm verifies if the RPL message is directed to the parent node with the 'O' flag set or if the 'O' flag equals 0 and is transmitted to descendant nodes. In either scenario, the algorithm raises the Inconsistent packet counter. If the Inconsistent packet counter exceeds the maximum allowed Inconsistent packet

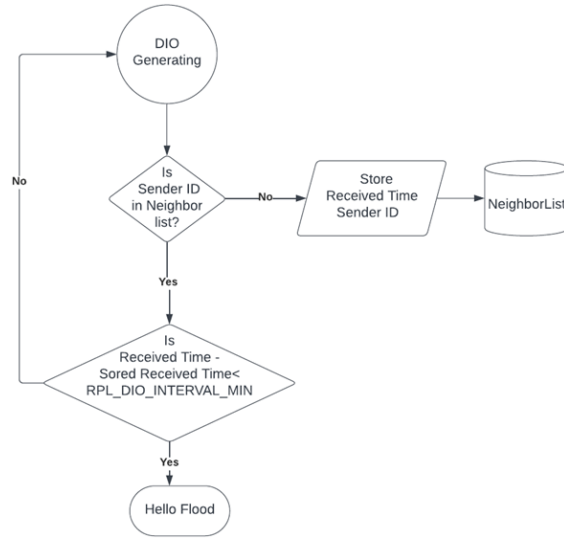


Figure 3.10: Attack detection flowchart using proposed taxonomy for generating DIO attacks

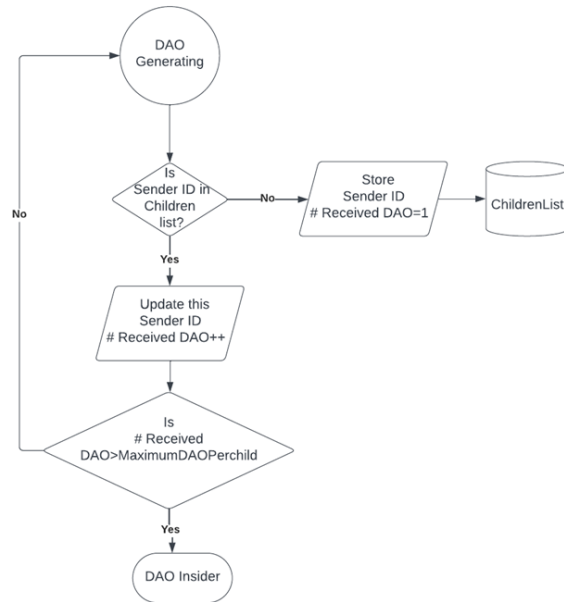


Figure 3.11: Attack detection flowchart using proposed taxonomy for generating DAO attacks

requests, it designates the sender as having a DODAG inconsistency.

The algorithm then checks the version value in the received DIO packet. If the version does not equal the version value of the root, the sender is marked as a version attack. Likewise, if the DODAG-ID in the received DIO message differs from the DODAG-ID of the root, the algorithm raises the local repair counter. If the local repair counter exceeds the maximum local repair counter, the sender node is marked as a local repair.

The detection algorithm also examines the received DIO's rank value. If the node rank changes, the rank is recorded and checked to ensure the adjustment does not indicate an assault. The local repair counter is raised if the node rank is infinite. It is identified as a local repair if the local repair counter exceeds the maximum local repair counter. A sinkhole is identified when the sender's rank is lower than its parent's rank. To identify a potential increased rank attack, the system compares the rank of the sender with the ranks of all its children. If any child possesses a rank lower than the sender, it is flagged as exhibiting an increased rank. To detect the worst parent attack, we evaluate the rank of each of the sender's neighbors. If any neighbor has a rank lower than the parent's rank, we mark that sender as the worst parent attacker.

To detect replay attacks on packets, the algorithm, as presented in Figure 3.13, first checks the type of RPL message. For DIO packets, the algorithm compares the sender node's rank with its neighbors. A node is indicative of a potential attack if its rank matches any of its neighbors. The node is labeled a copycat if it keeps its identity while repeatedly replaying and sending its neighbor's rank. A node is classified as a DIO suppression if it repeatedly broadcasts its neighbor's identification and rank

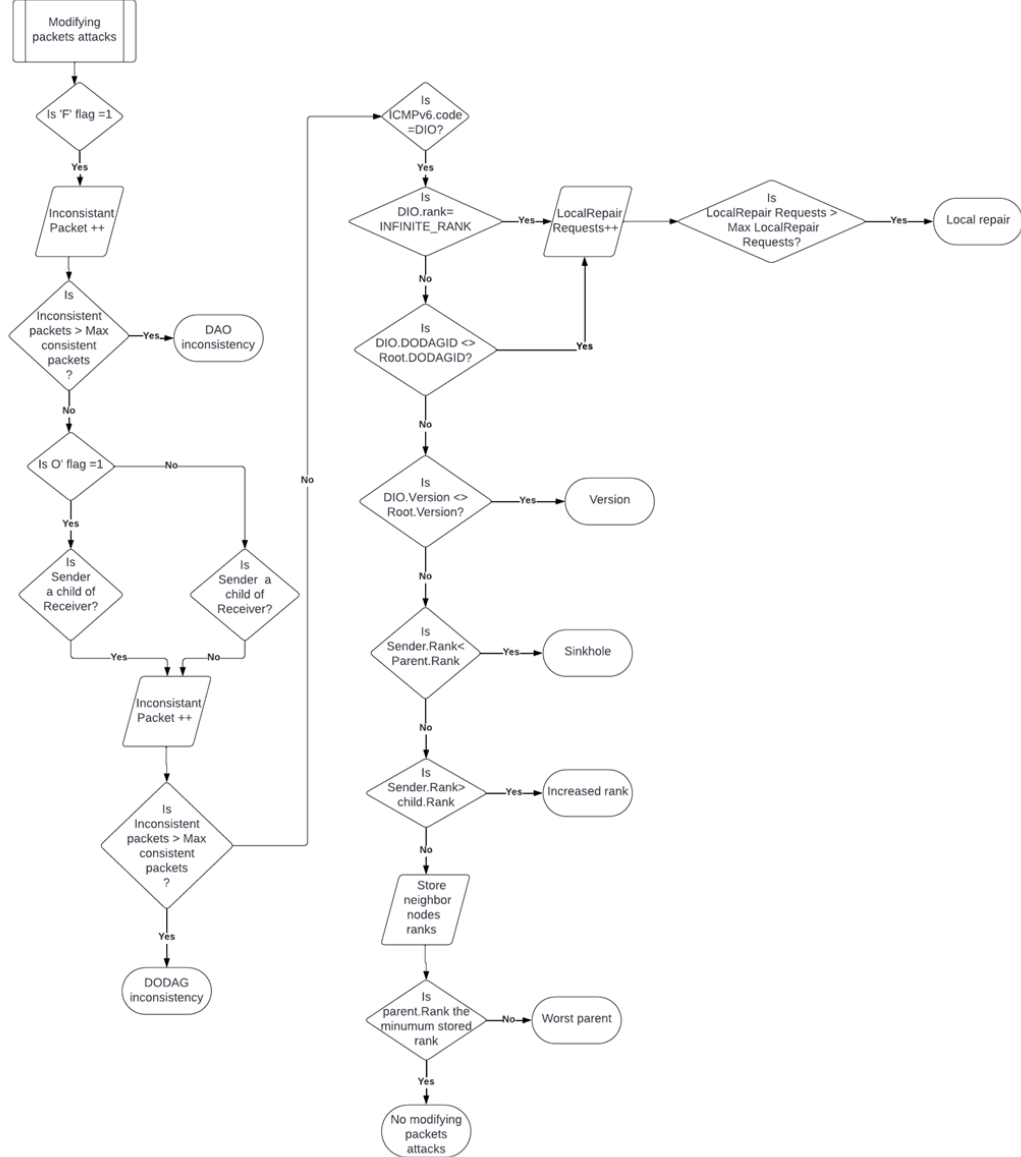


Figure 3.12: Attack detection flowchart using proposed taxonomy for modifying packets attacks

replays. Finally, the node is labeled as a neighbor attacker if it periodically transmits its neighbor's identity and rank while replaying it. For DAO packet, the detection algorithm checks whether the received DAO comes from a child in its routing table. If the sender ID doesn't exist in the neighbor list, the sender is marked as an adversary

that launches a rout table falsification attack.

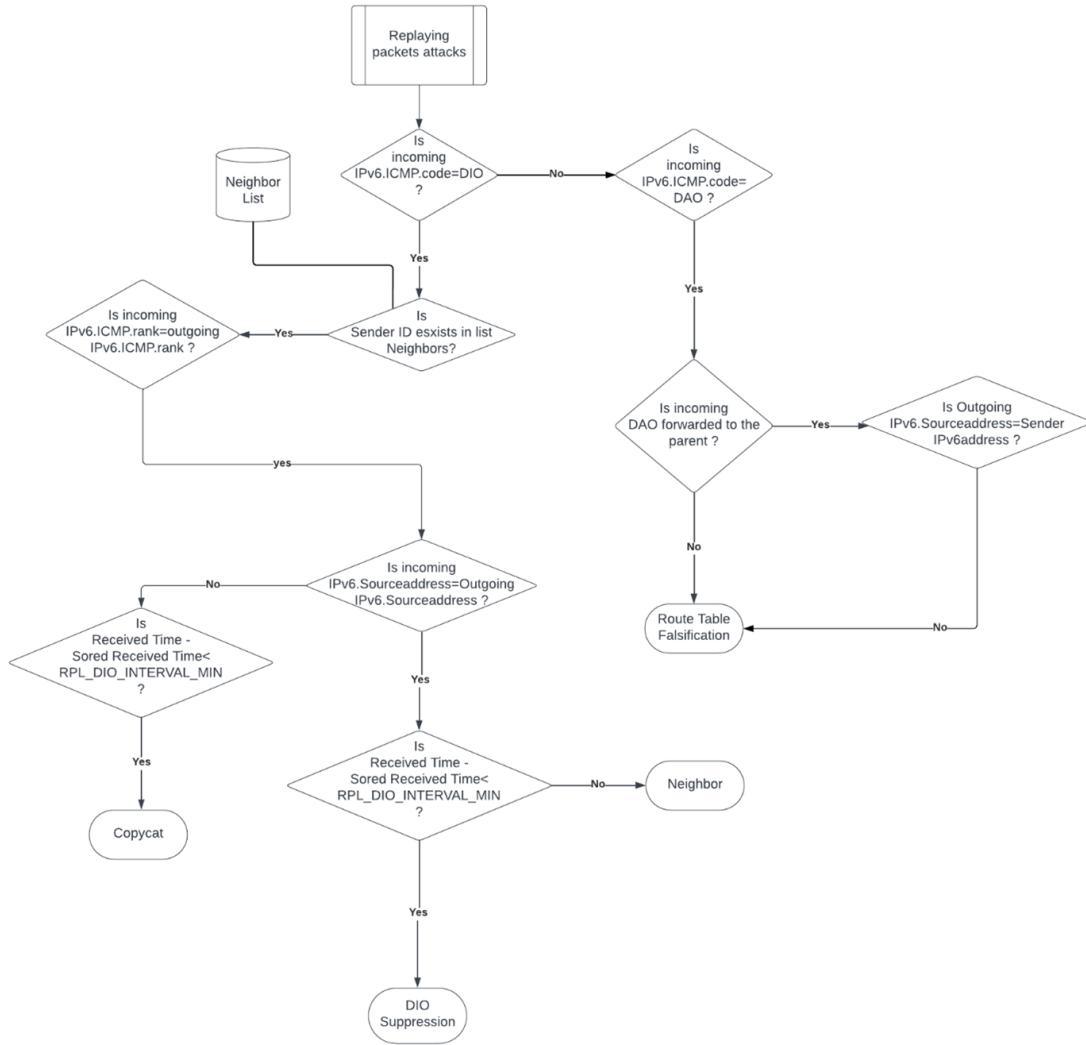


Figure 3.13: Attack detection flowchart using proposed taxonomy for replaying packets attacks

CHAPTER 4

MITIGATION SOLUTIONS FOR RPL ATTACKS

This chapter answers RQ2, which is about classifying RPL mitigation solutions. Generally, mitigation solutions include additions to RPL packets, authenticating nodes or network information, discarding malicious packets, encrypting packets, isolating forged nodes, updating routing information, modifying RPL functions, or changing some RPL settings. Figure 4.1 summarizes mitigation solutions, and we will shed more light on these solutions in the following sections.

4.1 Mitigation Solutions Based on Modification

The first mitigation solution used in the literature relies on modifying RPL packets to mitigate various attacks. Figure 4.1 reveals that 33% of the selected references discuss or use this mitigation solution. For instance, [225] introduced a distributed technique to monitor and countermeasure sybil attacks. In this solution, the parent selection

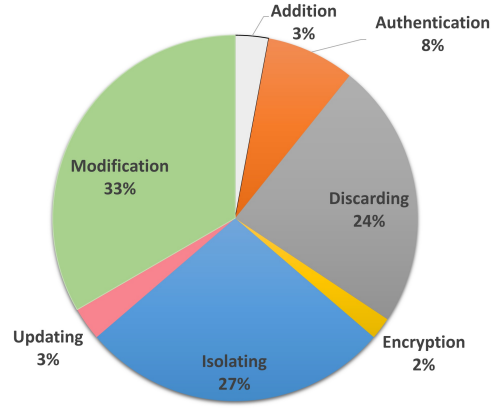


Figure 4.1: *Mitigation solutions for RPL attacks*

process is modified to consider both the nodes' trust value besides the rank value. This modification leads to avoiding nodes with a lower trust value. Table 4.1 lists the modifications to RPL that are adopted in the literature. As shown in Figure 4.2, the first and most common alteration targets the RPL objective function. To illustrate this approach, authors in [235] developed a lightweight trust-based mitigation solution embedded in RPL protocol to mitigate blackhole attacks. The proposed solution periodically computes and embeds trust values by creating a feedback chain between nodes to monitor nodes' behavior. Similarly, the mitigation solution provided by [114] mitigates rank attacks. The authors utilized the petal method to build the best path between the source and root by choosing the next-hop nodes. The Moth-Flame technique is then used to mitigate RPL rank attacks to prevent rogue nodes from being chosen as the preferred parent. Another technique proposed in [208] is to mitigate flooding and version attacks by computing the score factor to evaluate the node's trustworthiness in the parent selection. Besides the scoring factor, the means of version number and residual energy are considered in the optimal route selection. In

another illustration, authors in [220] overcome attacker nodes by adding them to the blacklist table and alerting relevant nodes to skip them. According to this solution that mitigates sinkhole attacks, blacklisted nodes are excluded from the parent selection process of the objective function. In addition, authors in [237] adopt sail fish-based in the objective function to secure routing against sinkhole, sybil, blackhole, and neighbor attacks.

Moreover, other mitigation solutions invoke or disable healing operations, which are repair techniques used by RPL to dynamically update routing decisions and adapt network topology to mitigate link failures. For instance, [209] mitigates blackhole attacks at local and global levels. The local module performs a local repair operation of RPL protocol and selects a different route. Then, the global module notifies the border router to initiate global repair and form a new DODAG version. Additionally, modifying RPL timers by stopping trickle timer resets has been suggested in several studies. For example, authors in [86] mitigate DIS flooding attacks by disabling trickle timer reset as a way to conserve node resources. A mandatory storage mode is also suggested as a mitigation solution. As outlined in [219], the wormhole attack was mitigated by incorporating virtual clusters and mandating storing mode operation. RPL timer modification is another RPL-based modification mitigation solution. For instance, to counteract sybil and DIO suppression attacks, the solution suggested in [84] decreases DIO packet responding rate. Additional RPL-based solutions include altering RPL device joining procedure by requiring sending DAO-ACK. For example, authors in [82] modify the process of joining new nodes to include unicasting a DAO message to the root node and waiting for its DAO acknowledgment packet to mit-

igate DIO suppression attack. The attacks that have been countered by each RPL modification solution are listed in Table 4.2.

Table 4.1: *Mitigation solutions based on modification*

Mitigation Solution	Abbr.
Modifying objective function	RPL1
Disabling global repair	RPL2
Disabling local repair	RPL3
Disabling trickle timer reset	RPL4
Invoking global repair	RPL5
Invoking local repair	RPL6
Mandating storing mode operation	RPL7
Modifying RPL timers	RPL8
Mandating DAO-ACK	RPL9

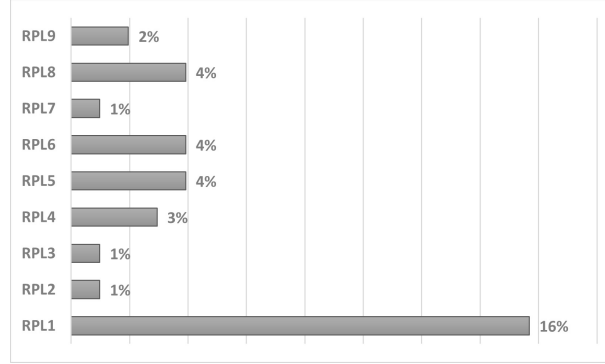


Figure 4.2: *Publications per mitigation solution based on modification*

4.2 Mitigation Solutions Based on Discarding and Updating

Discarding bogus packets is the second category of RPL attack mitigation solutions.

According to the literature, this method is used in about 24% of the selected studies.

Table 4.3 lists the discarded information utilized as mitigation solutions. Researchers

Table 4.2: *Modification solutions used in the selected references*

Attack Mitigated	RPL1	RPL2	RPL3	RPL4	RPL5	RPL6	RPL7	RPL8	RPL9
Blackhole	[93], [235], [238], [207], [213], [237]				[209]	[209], [63], [243], [94]	[219]		
Clone-ID	[93]				[70]				
DIO sup- pression								[83], [84]	[82]
DIS flood- ing				[86]				[232]	
DODAG in- consis- tency		[93]	[93]	[93], [199], [88]					
Hello flood	[237]								
Increased rank	[114], [213], [239], [241], [243]				[250]				
Local repair			[93]	[93]					
Neighbor	[237]	[93]	[93]	[93]					[242]
Packet flooding	[208]	[93], [208]	[93]	[93]					
Replay		[93]	[93]	[93]					
Selective forwarding	[93], [213]				[70], [203]				
Sinkhole	[93], [220], [114], [207], [213], [237], [239], [241], [243], [94], [215]				[250]	[250], [202]	[219]		
Sybil	[93], [213], [237], [239], [215], [225]							[83], [84]	
Version	[208]	[93]	[93]	[93]					
Wormhole							[219]		
Worst par- ent	[114], [241]					[222]			

suggest discarding all incoming packets, discarding only control packets, or discarding specific RPL packets. Figure 4.3 suggests that the commonly utilized approach is discarding all control packets from rogue nodes. For instance, authors in [218] reduced the impact of the hello flood attack by removing extraneous control packets. Akin to this, authors in [214] restrict flooding attacks by forbidding one-hop neighbors of the suspected node from accepting any control packets. As an illustration of discarding specific packets, [252] mitigated the DIS flooding attack by discarding excessive DIS packets when the number of received DIS exceeds a particular threshold. Other researchers [202] discard DIO packets and suggest that nodes validate the

received DIO packets by requesting the sink to oppose sinkhole attacks. If the sink’s response suggests discarding this DIO, the verifier should blacklist the sender as an intruder. Discarding DAO packets is another solution [78] to mitigate DAO insider attacks, where authors suggested discarding redundant incoming DAO packets when the number of sent DAO packets reach a certain threshold. On the other hand, [205] proposed discarding incoming packets at the link layer to mitigate packet flooding, selective forwarding, and clone-ID attacks. The last mitigation solution proposed is discarding version number updates [108].

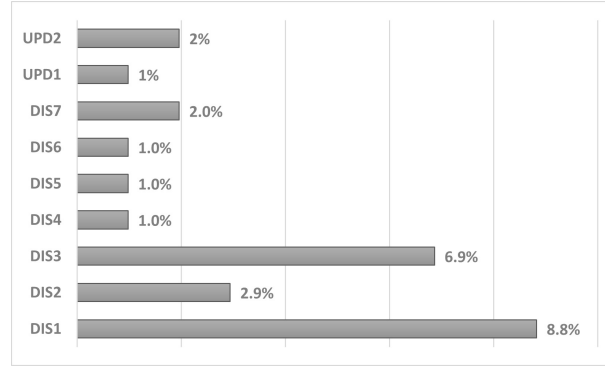
Another mitigation solution is updating network information by updating various field values in the control packets. The solution counts for only 3% of the chosen references. In [88], the suggested updating solution involves changing the RPL packet options header to mitigate DODAG inconsistency attack. Before sending the packets, researchers [88] proposed modifying the options header by resetting the “O” and “R” flags. Moreover, updating the node rank through DIO packets was utilized to counter-act sinkhole attack [217], [195]. Table 4.4 shows the mitigated attacks by discarding or updating along with their using studies.

4.3 Mitigation Solutions Based on Additions, Authentication, Encryption, and Isolation

RPL additions utilize adding registration process [237] offered a solution to mitigate neighbor, sinkhole, sybil, and blackhole attacks. In this solution, each network must register with the security system, which enforces all registered nodes to use strong

Table 4.3: *Mitigation solutions: discarding and updating*

Mitigation Solution	Abbr.
Discarding control packets	DIS1
Discarding DIS packets	DIS2
Discarding DIO packets	DIS3
Discarding DAO packets	DIS4
Discarding incoming packets	DIS5
Discarding previous information	DIS6
Discarding version number update	DIS7
Updating options header in RPL packets	UPD1
Updating node rank in DIO packets	UPD2

**Figure 4.3:** *Publications per mitigation solution based on discarding and updating*

credentials. If a node wants to transmit its data, it must undergo an authentication process. Secure routing is adopted in RPL by utilizing the sail fish optimization algorithm. Each node encrypts its sensed data using the prince algorithm before transmitting it. Another addition [219] consists of virtual clusters on top of the DODAG topology to mitigate decreased rank, blackhole, and wormhole attacks. The task of these virtual clusters is to monitor the network for mitigation purposes. Furthermore, [230] suggested hardening RPL to counter node reset attacks including local repair and version attacks.

Table 4.4: *Discarding and updating solutions used in the selected references*

Attack Mitigated	DIS1	DIS2	DIS3	DIS4	DIS5	DIS6	DIS7	UPD1	UPD2
Blackhole	[85]								
Clone-ID	[69], [100]				[205]	[195]			
Copycat					[72]				
DAO insider				[78]					
DIO suppression	[75]								
DIS flooding	[229]	[252], [216], [245]							
DODAG inconsistency	[199]							[88]	
Hello flood	[229], [218]								
Increased rank			[222]						
Neighbor	[100]								
Packet flooding	[75], [214]		[113]		[205]				
Selective forwarding	[85]		[70]		[205]				
Sinkhole			[202], [247], [80], [222]						[217], [195]
Sybil						[195]			
Version	[246]		[233]				[220], [108]		
Wormhole	[100]		[113]						
Worst parent			[222]					[93]	

Authentication, whether authenticating nodes or network information changes, can be used to thwart multiple RPL attacks. An authentication solution is utilized by [210] to countermeasure sinkhole and selective forwarding attacks. In solution, a group key is assigned for each node in the transmission path. The border router possesses all the keys associated with each group. Hence, if a node tries to change its rank value, the neighbor table would notice the change. Node authentication is done in [111], and the wormhole attack was mitigated by completing neighbor authentication after network initialization. Consequently, nodes with erroneous or inappropriate node IDS were disregarded. Another illustration, the authors of [224] provided a thorough architecture and implementation of mutual authentication. The findings of [224] demonstrated that the suggested authentication approach prevents the admission of unauthenticated nodes, shielding the network from attacks, including DIS flooding,

version, increased rank, and sinkholes. Encrypting data packets is another mitigation solution. Each node encrypts data using the prince algorithm before relaying it to the group head node [237]. This is done to mitigate sinkhole, sybil, blackhole, and neighbor attacks. Similarly, authors in [200] used the AES algorithm as symmetrical encryption to mitigate version, rank, selective forwarding, and replay attacks.

Finally, isolating the attacker is the second most common mitigation solution with 27%, as shown in Figure 4.1. Several isolation solutions have been proposed, as demonstrated in Table 4.5, including deleting inactive nodes, deleting malicious nodes from the neighbor’s list, rebuilding DODAG without malicious nodes, or selecting another parent from the potential parent’s list. Deleting inactive nodes to counteract blackhole and selective forwarding attacks is proposed in [195]. This solution deletes nodes not sharing their routing status within a specific threshold. Another isolating solution is deleting malicious nodes from legitimate nodes’ neighbor lists. Authors in [211] mitigated wormhole attacks by deleting attackers from the routing table. DODAG reconstruction without malicious nodes was widely considered in approximately 13% of the studies. Authors in [74] identify distinctive characteristics of sinkhole, DAO inconsistency, and flooding attackers. A node was removed from the DODAG when its behavior matched these characteristics. Similarly, [201] performs node monitoring to identify sinkhole attacks. The observer node isolates the forging node and rebuilds the network when a node is discovered as malicious. The last isolation solution is selecting another parent. As an illustration, [206] mitigated blackhole attacks by monitoring whether the reported node is a parent node. In that case, the monitoring node would remove the parent node from its list of parents and choose a new parent. Then, the

Table 4.5: *Mitigation solutions: additions, authentication, encryption, isolation*

Mitigation Solution	Abbreviation
Adding registration process	ADD1
Adding virtual clusters	ADD2
Saving RPL state in non-volatile memory	ADD3
Authenticating change in network information	AUT1
Node authentication	AUT2
Encrypting data packets	ENC1
Deleting inactive nodes within specific period	ISL1
Deleting malicious node from neighbors list	ISL2
Rebuilding DODAG without malicious node	ISL3
Selecting another parent from potential parents list	ISL4

monitoring node broadcasts an ICMP6 control packet to its neighbors. All mitigation solutions utilizing additions, authentication, encryption, or isolation are shown in Table 4.5. Figure 4.4 reveals the ratio of adapting each solution. In summary, Figure 4.5 shows a taxonomy of mitigation solutions for RPL attacks based on the countermeasure mechanism.

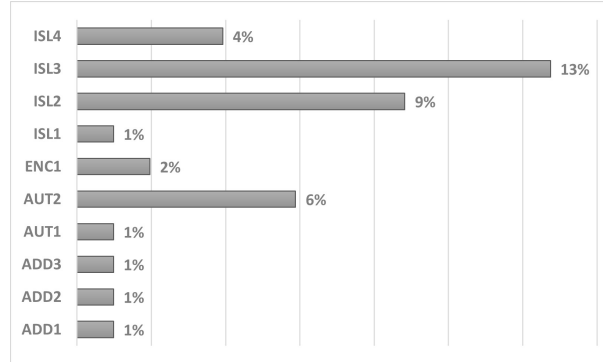


Figure 4.4: *Publications per mitigation solution based on additions, authentication, encryption, and isolation*

Table 4.6: *Additions, authentication, encryption, and isolation solutions*

Mitigated At- tack	ADD1	ADD2	ADD3	AUT1	AUT2	ENC1	ISL1	ISL2	ISL3	ISL4
Blackhole	[237]	[219]				[237]	[195]	[206], [64]	[223]	[206],[94],[64]
Clone-ID										[70]
Copycat										
DAO incon- sistency									[74]	
DIS flooding					[224], [251]				[248]	
DDAO									[73]	
Hello flood									[221]	
Increased rank				[200]	[224]	[200]		[200]		[228]
Local repair			[230]							
Neighbor	[237]					[237]				
Packet flood- ing									[74]	
Replay				[200]		[200]		[200]		
Selective for- warding				[200]	[210], [259]	[200]	[195]	[200]		[70]
Sinkhole	[237]	[219]		[200]	[224], [210], [259]	[237], [200]		[236], [249], [200], [202]	[250], [74], [234], [212], [231], [196], [244], [201]	[94], [228]
Sybil	[237]				[251]	[237]			[204], [196]	
Version			[230]	[200]	[224]	[200]		[200]		
Wormhole		[219]			[111], [109]			[211], [111]		
Worst parent										[228]

4.4 Discussion

According to the illustrated mitigation solutions and the proposed taxonomy, we mapped the used mitigation solution with the corresponding class of RPL attack as shown in Figure 4.6. This figure provides valuable insight into the percentage of each mitigation solution utilized to combat each RPL attack class. In general, RPL modification is the most used countermeasure to mitigate discarding, modifying, and replaying packet attacks. On the other hand, for generating packet attacks, the most used solution is discarding the malicious packets to maintain the RPL performance.

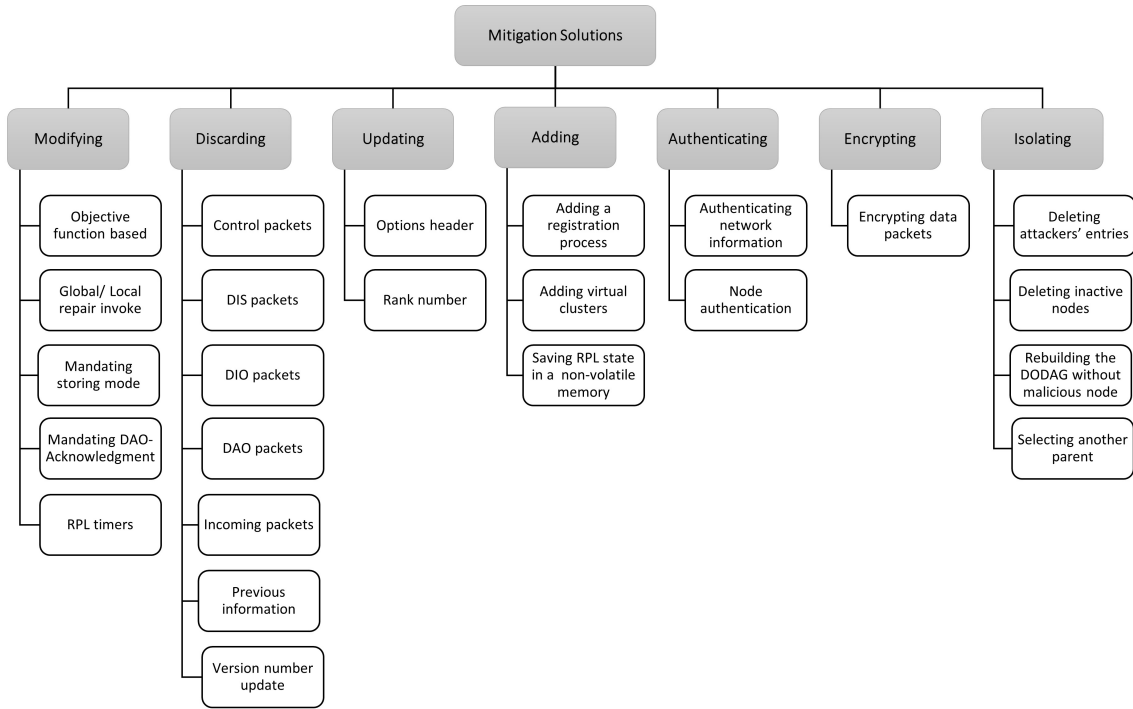


Figure 4.5: *Taxonomy of mitigation solutions for RPL attacks*

Discarding packets is also used to mitigate replay attacks to avoid impacting the network with replayed packets. At the same time, dropping and modifying packet attacks used discarding solutions in a limited number of studies. In addition, it is noticeable that updating mitigation solutions has only been used for modifying packet attacks.

Besides, we want to emphasize the following observations: First, isolating attacking nodes was widely applied in all RPL attacks. For dropping packet attacks, this countermeasure solution is suitable as dropping packets severely impacts network availability. For other attacks, this solution might be destructive. Even though the malicious behavior is minimized, the network lifetime and performance after isolating attacking nodes could be worse, especially when isolating attacking nodes leads to network partitions. Second, it is worth mentioning the difference in the implemen-

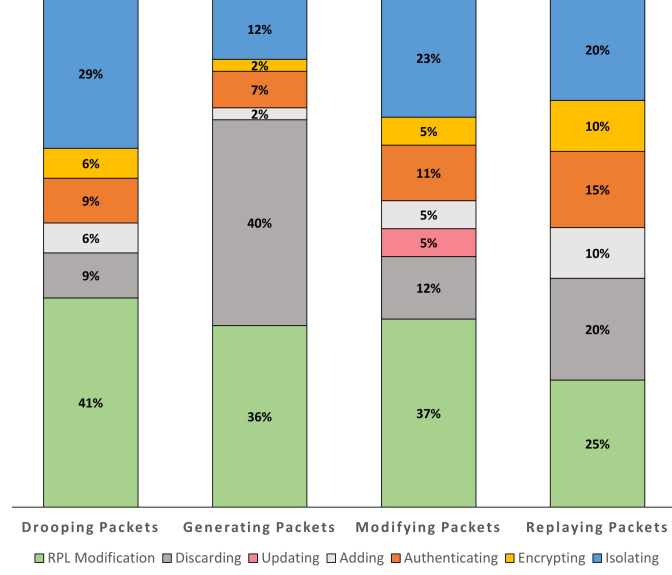


Figure 4.6: *Mitigation solutions percentages for each RPL attack class*

tation requirement for mitigation solutions. For instance, any solution that requires additions or modifications to the RPL protocol, like defining a new objective function, needs to be implemented on the entire network, including the root and all routing nodes. However, other solutions, like discarding malicious DIO packets, can be applied to the nodes that face or might face such an attack without the need to consider all the network nodes or the root. Another observation is that even though encryption and authentication have been studied in multiple solutions, the limitations in memory and computing resources could make them less favorable in front of lightweight solutions like updating the rank value based on neighbor nodes. In addition, many encryption and authentication solutions are built to protect the network against external attackers but overlook that these attacks might be launched by internal nodes as well. Therefore, further analysis has been performed to fully assess the efficiency of mitigation solutions to aid in making the right decision about the chosen mitigation

solution and ensuring the network is secure. We call the mitigation solution efficient when the solution successfully reduces the impact of the attack using the minimum resources considering the contexts and constraints of IoT environments.

We develop our evaluation criteria for each solution to consider the following aspects: The first criterion is the network's node utilizations, which indicates how this solution maintains utilizing the maximum number of active nodes in the routing process. The second one is the node's resource preservation, which represents the complexity of the solution in computation and storage. The following evaluation factor is the modification level, which describes the required amendments to be performed in the network to apply the mitigation solution. The last criterion is the comprehensiveness of the solution, which refers to how the solution considers various attacker forms. i.e., internal and external attackers. To implement these factors, we develop a matrix that represents different levels of satisfaction for each criterion. In addition, each satisfaction level is associated with numerical scores (represented as stars) to be assigned for each mitigation solution. Table 4.7 shows the evaluation criteria of mitigation solutions. Accordingly, each presented mitigation solution will be evaluated using the aforementioned criteria, where higher scores indicate higher satisfaction levels of the solution based on the studied criterion.

For the network's node utilization factor criterion, the scores are either 0 or 1. We assign 0 for all solutions that explicitly exclude nodes of the routing process, either by isolating nodes or overlooking all control packets that regulate routing decisions. Other than that, the solution gets a score of 1 in this criterion. Moving to the node's resources preservation factor, the score varies between 0 and 2. Score 0 has been assigned to

Table 4.7: *Evaluation criteria of mitigation solution Efficiency*

	Criteria			
	Network Nodes Utilizations	Node Resources Preservation	Modification Level	Comprehensiveness
Description	0 -Applying the solution leads to overlooking some nodes in the network	0 -The proposed solution highly consumes node's resources	0 -The solution should be applied on all network nodes, including the root	0 -The solution overlooks some attacker forms i.e., internal and external attackers
and	★ -Applying the solution leads to considering all network nodes	★ - The solution considers operation that requires additional energy, computational, and storage capabilities	★ - The solution is applied on the root and some network nodes	★ - The solution considers various attacker forms, ie, internal and external attackers
Score		★★ - The solution utilizes operations that are frequently used by the protocol	★★ - The solution can be applied on some parts of the network without the need to update the protocol for all nodes	

solutions that include cryptographic algorithms, as these solutions require additional storage and processes to manage the keys, as well as they usually consist of complex algorithms. We give a score of 2 for the solutions that consist of regular operations in RPL, like discarding excessive DIS packets or modifying option headers of RPL packets. However, other solutions are scored 1 as they require additional storage, like saving RPL state in non-volatile memory or mandating storing mode operation. Other solutions score 1 because they perform resource-consuming operations like invoking repairs mechanisms. For the modification level criterion, we consider that the scores vary between 0 and 2. The solutions that can be applied on any node of the network without the need to update the protocol for all devices gets a score of 2. The solution scores 0 when it should be applied on all network nodes, including the root, as it leads to undesirable disconnection in the network for this update. Also, the nodes might be unreachable remotely and require a manual update, which adds an additional security effort to provide this mitigation. When the solution is applied on the root beside some

network nodes, it gets a score of 1, which allows partial implementation on critical nodes and overlooks the rest or allows gradual implementation to maintain the network running under security updates. For the comprehensiveness of the mitigation, the solution scores 1 when it considers various attacker forms, i.e., internal, and external attackers, while it scores 0 when the solution overlooks some attacker forms, like internal attackers.

After defining the evaluation matrix, we evaluate each mitigation solution based on the defined criteria as shown in Table 4.8. The last column in Table 4.8 represents the final score of the mitigation solution, which is obtained by summing all scores of each evaluation criterion. The scores vary between six and one, where the high score indicates the high efficiency of the solution based on the introduced criteria. Table 4.8 reveals that all solutions based on updating RPL packets are the most efficient mitigation solutions, as this update happens at the node level without the need to modify the routing protocols in the root. In addition, updating solutions adjust the routing parameters without considering the source of the bogus packets, i.e., internal or external attacker nodes. Also, most of discarding solutions show the same level of efficiency as updating solution. On the other hand, solutions that are based on authentication and encryption appear to be the least efficient solutions as they require storage and additional computational overhead at the node level. It also requires programming the root and all nodes to implement the selected algorithm. Moreover, this type of solution usually overlooks the routing attacks resulting from internal nodes. Furthermore, we want to give more attention to the most used solutions, which are modifying objective function (RPL1) and rebuilding the DODAG without malicious

node (ISL3). In fact, these solutions are not at the top in terms of efficiency for the following reasons. For (RPL1), adapting a new objective function in the routing requires updating the protocol in the root and all network nodes. For (ISL3), which is rebuilding the DODAG without the suspicious node. This approach can result in the exclusion of certain nodes from the routing process, leading to a loss of valuable network resources. Additionally, this solution requires each node to maintain a block list of banned nodes, which can significantly increase storage requirements. Furthermore, rebuilding the DODAG is a computationally and communication-intensive process that involves all network nodes, making it a costly solution overall.

When evaluating mitigation methods, it is necessary to analyze each attack on an individual basis to determine the most effective solution as illustrated in Table 4.9. For blackhole and selective forwarding attacks, the most efficient solution is to delete inactive nodes that have not participated in routing for a predetermined time window. However, for DDAO, the only proposed solution is DODAG reconstruction. The local repair attack can be mitigated by disabling local repair and resetting the trickle timer. To mitigate packet flooding, neighbor, replay, DODAG inconsistency, and version attacks, an efficient solution that mitigates all of them is to adjust the RPL settings by disabling global repair, local repair, and trickle timer reset. Additionally, the DODAG inconsistency attack can also be mitigated by updating the option header to clear the ‘O’ and ‘R’ flags before forwarding the packets normally, besides disabling the trickle timer reset. Discarding DIO messages is another efficient mitigation solution for packet flooding attacks. To mitigate version attacks, DIO with a false version number can be discarded, or discarding the change in version number. DIS flooding attacks

Table 4.8: *Efficiency scores of mitigation solutions*

Mitigation Code	Network Nodes Utilizations	Node Resources Preservation	Modification Level	Comprehensiveness	Score
RPL1	★	★★	0	★	★★★★
RPL2	★	★★	★★	★	★★★★★
RPL3	★	★★	★★	★	★★★★★
RPL4	★	★★	★★	★	★★★★★
RPL5	★	★	★★	★	★★★★
RPL6	★	★	★★	★	★★★★
RPL7	★	★	★★	★	★★★★
RPL8	★	★★	★★	★	★★★★★
RPL9	★	★★	★	★	★★★★
DIS1	0	★★	★★	★	★★★★
DIS2	★	★★	★★	★	★★★★★
DIS3	★	★★	★★	★	★★★★★
DIS4	★	★★	★★	★	★★★★★
DIS5	0	★★	★★	★	★★★★
DIS6	★	★★	★★	★	★★★★★
DIS7	★	★★	★★	★	★★★★★
UPD1	★	★★	★★	★	★★★★★
UPD2	★	★★	★★	★	★★★★★
ADD1	★	★	0	0	★★
ADD2	★	★	0	0	★★
ADD3	★	★	0	0	★★
AUT1	★	0	0	★	★★
AUT2	★	0	0	0	★
ENC1	★	0	0	0	★
ISL1	★	★★	★★	★	★★★★★
ISL2	0	★★	★★	★	★★★★
ISL3	0	★	★	★	★★★
ISL4	0	★★	★★	★	★★★★

can be efficiently mitigated by discarding excessive DIS messages, disabling the trickle timer reset, or modifying RPL timers by increasing the response delay. DAO insider attacks can also be mitigated by discarding excessive DAO packets. However, the most efficient mitigation technique for DIO flood attacks is to discard all control packets of malicious nodes. The following attacks are efficiently mitigated by discarding DIO packets: increased rank, decreased rank, wormhole, and worst parent attacks. The

decreased rank attack can also be mitigated by updating the rank with the information reported directly by the neighbors. Clone-ID and sybil attacks can be mitigated by discarding previous information and considering the latest information received from each host in the network. Additionally, sybil attacks and DIO suppression attacks can be mitigated by modifying RPL timers and limiting the DIO message replying rate. Copycat attacks are mitigated by discarding all incoming packets from malicious nodes. Finally, the only proposed solution to overcome DAO inconsistency attack is the DODAG reconstruction without the malicious nodes.

Table 4.9: *High efficient mitigation solution per attack*

Attack	Mitigation Solutions																			
	RPL	RPL	RPL	RPL	DIS2	DIS3	DIS4	DIS6	DIS7	UPD	UPD	ISL1	RPL	RPL	RPL	RPL	DIS1	DIS5	ISL2	ISL4
Blackhole												6								
Clone-ID								6												
Copycat																		5		
DAO inconsistency																				3
DAO insider							6													
DDAO																				3
DIO suppression				6																
DIS flooding			6	6	6															
DODAG inconsistency	6	6	6							6										
Hello flood																	5			
Increased rank						6														
Local repair		6	6																	
Neighbor	6	6	6																	
Packet flooding	6	6	6			6														
Replay	6	6	6																	
Selective forwarding												6								
Sinkhole						6					6									
Sybil			6					6												
Version	6	6	6			6			6											
Wormhole						6														
Worst parent						6														

CHAPTER 5

REPLAY ATTACKS IN RPL-BASED INTERNET OF THINGS

This chapter addresses RQ3 concerning the implementation of replay attacks utilizing RPL control messages and their resulting effects. Section 5.1 analyzes and compares the effects of different replay attacks on RPL network topology. In addition, Section 5.2 summarizes different methods and features to detect replay attacks.

5.1 Analysis of RPL Replay Attacks

Multiple RPL replay attacks can be launched based on the type of control message being replayed, the modification performed before replaying the message, and the frequency of replaying the message. Figure 5.1 summarizes the mechanism of these various attacks and answers the first research question regarding the types of control

messages used to launch replay attacks. First, the DIS message, this message is sent by an external node probing the network, and it does not contain information about the sender. Therefore, replaying this message by any other node will not impact the network. The second message is a DIO message that can launch multiple RPL replay attacks. When the attacker node replays the exact received DIO message without adding its IP address or routing information, such as the rank value, it performs a neighbor attack [24], [10], [260]. However, when the attacker frequently sends the received message, the attack becomes a DIO suppression attack [258], [10]. On the other hand, if the attacker changes the IP address of the received message to include its IP address while keeping the routing information of the received node and sending the message frequently, this attack is called a copycat attack [261], [72]. The DAO message is another control packet used to launch a replay attack. In a DAO replay attack, the attacker replays the received DAO message to a neighbor node in the DODAG [3], [101]. Another attack based on replaying DAO messages is a route table falsification attack. Here, the attacker modifies the DAO message by changing the source's IP address and then replays it to its parent [3], [2].

In addition, we determine the impact of replay attacks by first considering used control messages and listing different attacks that might occur. Based on the mentioned consequences of these attacks according to the literature, we analyze and classify the impact level as shown in Table 5.1. The attack has a low impact when it only leads to additional exchanges of control messages without impacting network topology. For moderate impact, the attacker node destructs the optimal structure of DODAG by driving network nodes to use sub-optimized paths or deceiving them

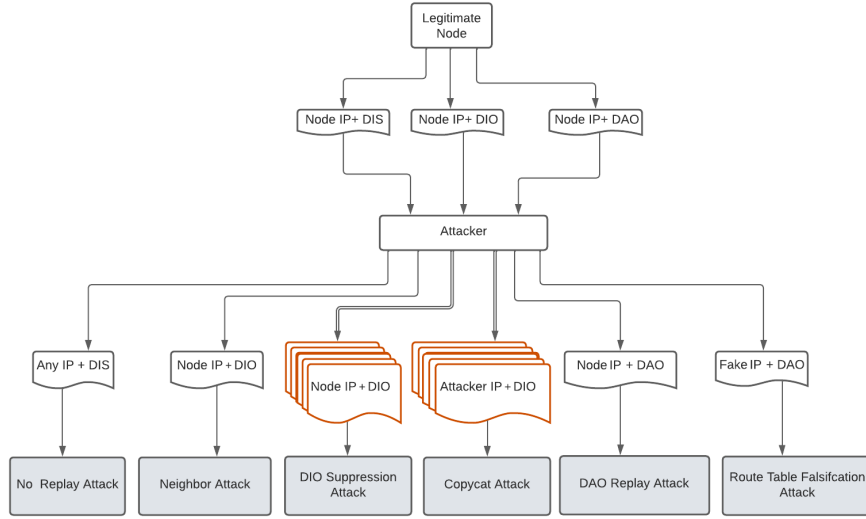


Figure 5.1: *RPL replay attacks mechanism*

into considering an out-of-range node as a neighbor. When the attack has a critical impact, it falsifies the network by either announcing a false or out-of-range node to add non-existing paths or by isolating some nodes from the DODAG.

Table 5.1: *The classification scheme of potential impact on routing*

Impact Magnitude	Impact Description
Low	Performing the attack leads to additional exchanges of control messages without impacting network topology
Moderate	Performing the attack leads to using sub-optimized routing paths or adding an out-of-range node as neighbor
Critical	Performing the attack leads to adding non-existing routing path to DODAG or isolating node(s) from DODAG

5.1.1 Attacks Resulting from Replaying DIO

The DIO control message is the packet that has the most information about the network and the sender. In addition, it is the message responsible for constructing the upward path [41], [40]. Therefore, this control message is exposed to multiple replay attacks. Specifically, three types of replay attacks have been introduced based on replaying DIO messages: neighbor attack [117], [3], DIO suppression attack [258], [10], and copycat attack [261], [72]. We will study these forms of DIO replay attacks in terms of the attackers' mechanisms and their impact on the routing service and the DODAG topology according to Table 5.1.

DIO Replay Attack

When an attacker node replays the received DIO message without modification, this attack is called a neighbor attack. To perform this attack, the attacker node has to replay an exact copy of the received DIO message without adding its information (neither its rank nor the IP address). As a result, this replayed DIO deceives the recipients by considering the original sender of this DIO message in their communication range. The intent of this attack is to compel nodes to pursue sub-optimized or nonexistent paths [5], [117]. The neighbor attack has been studied by many researchers. For instance, Sharma et al. [24] considered the neighbor attack as a type of sub-optimization attack in their classification and comparison of RPL routing attacks. The impact of this attack, according to [24], is the delay in network performance as packets use non-optimal paths. Raoof et al. [3] classified neighbor attacks as a type

of RPL-specific attacks, emphasizing this attack as a result of RPL vulnerabilities. In addition, [3] pointed out that this adds non-significant network delay. However, Mangelkar et al. [117] considered the neighbor attack as a type of confidentiality threat attack that happens as a result of a failure to keep routing information confidential, which leads to adapting sub-optimal or non-existing routes and increasing resource consumption. Furthermore, several studies have demonstrated that this attack increases E2E delay, changes network topology, adds additional control overhead, and loses some data packets [50], [262].

To illustrate the impact of this attack and how it negatively affects the routing service, we need to consider the position of the adversary node, victim nodes, and the node that has been exploited in the DODAG, as the position of the exploited node plays a vital role in amplifying the impact of the neighbor attack [3], [117]. Figure 5.2 illustrates various positions of the attacker node, the exploited node, and the victim node. The critical impact of the neighbor attack happens when the exploited node has a lower rank than the victim’s parent, as mentioned in [3], [5]. This case is shown in Figure 5.2(c). Specifically, the victim receives the replayed DIO message. This message contains information about a new neighbor node from the victim’s perspective. Therefore, the exploited node will be added as a candidate parent in its parent set. Also, it will be added to the victim’s routing table if the DODAG uses a storing mode [41]. In addition, the victim node will calculate its new rank based on the rank of the exploited node and will choose it as a preferred parent [41]. Accordingly, the victim node will also send a DAO message to the exploited node as a new parent. However, as this path does not exist, it leads to high exchange of control messages

and loses some data packets [3]. In Figure 5.2 (b), the exploited node is out of the victim's range. The victim receives the replayed DIO message sent by the attacker. As a result, the victim node will add the exploited node to the parent list as well as the routing table, similar to the previous case. However, the replayed DIO message contains a higher rank than the victim's parent. Therefore, the exploited node will not be considered as a preferred parent, and the upward route will not be impacted. In this case, the impact of the neighbor attack is moderate by adding an out-of-range node in the victim's potential parents and the routing table if the victim is in the storing mode [41]. In addition, this attack leads to a further exchange of control messages that causes some delay.

In Figure 5.2 (a), both the exploited node and victim nodes are within each other's communication range. Therefore, the replayed DIO message, previously received by the victim, will only be a repeated consistent DIO message. The only impact, in this case, is to increase the control message exchanges by triggering the trickle timer of the victim to send its own DIO message. This scenario has a low impact on routing because it does not directly affect network topology nor lead to the consideration of an out-of-range neighbor as a potential parent. It is worth noting that the neighbor attack can be launched by any internal or external node, as the adversary uses a DIO message multicast by each node in the network [24].

Frequent DIO Replay Attacks

The second and third types of DIO replay attacks fall under this category. The second DIO replay attack is a DIO suppression attack, where the attacker replays

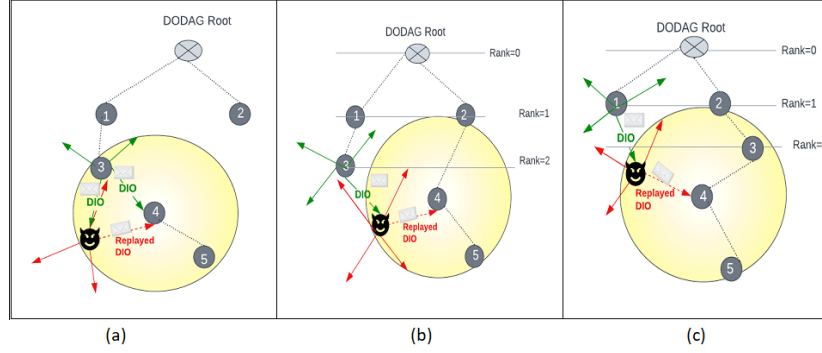


Figure 5.2: *DIO replay attack analysis based on relative location: (a) low impact, (b) moderate impact, and (c) critical impact.*

a received legitimate DIO message many times within a fixed frequency [258]. The DIO suppression attack has the same mechanism as the neighbor attack in terms of replaying the received DIO message without adding any information about the sender node. The difference in this attack appears in replaying the received DIO message within a fixed frequency [258], [82]. This attack aims to hide the existence of some network nodes that could lead to partitioning the network by misusing the trickle timer [3], [82]. According to the trickle algorithm, when a node receives a specific consistent number of DIO messages, the trickling method prevents this node from transmitting its own DIO messages to minimize the congestion by reducing the exchanged control messages in the network [44]. As such, the adversary replays a specific DIO message several times to trigger the suppression feature in the victim node. The suppressed nodes may remain hidden, and some routes may go undiscovered as a result of the ongoing suppression [10], [83].

Several researchers have tackled the consequences of the DIO suppression attack. For instance, Avalia et al. [38] included the DIO suppression attack as a type of attack that

threatens the network in terms of eliminating some control messages and destructing network topology. Other consequences of this attack appeared in the form of degrading the quality of routing services by reducing PDR and network path stretch [258], [263]. In addition, other studies have highlighted the DIO suppression attack as an attack that forces nodes to update their routing tables with outdated and wrong data [258], [3], [227], [5], [85].

To illustrate the impact of DIO suppression attacks, we evaluate only the suppressing feature and its impact. To achieve that, the malicious node is positioned close to the replay source node to focus only on the frequent replay effect. By doing this, every node that receives replayed DIOs can reach the replay source node [258]. However, the attacker node may replay a DIO message from an out-of-range node in frequent replay DIO attacks [258], [261]. The combined impact of suppressing and attempting to reach an out-of-range node is not assessed in this study and is left for future work. Figure 5.3 demonstrates different positions that attacker and victim nodes can take under a DIO suppression attack. Even though Perazzo et al. [258] and Raoof et al. [3] mentioned that the worst impact of this attack leads to network partitioning and isolating some nodes, they have not mentioned the position that causes such impact. To attain this level of harm in the network, we choose the setup shown in Figure 5.3 (c). In this position, the victim node has only the exploited node and the attacker node as potential parents. Besides, the sub-DODAG of the victim node is connected to the DODAG only through the victim node. According to this position, when the victim node receives consistent DIO messages from the exploited and attacker nodes, it suppresses its DIO. As a result, the sub-DODAG of the victim node that relies only

on DIO messages from the victim to have the routing information about the network will not receive any DIO message. Therefore, suppressing the victim node will lead to network partitioning and sub-DODAG isolation from the network. In this case, the DIO suppression attack has a critical impact on the network.

However, Verma et al. [264] emphasized the impact of this attack on the routing performance is small when the network has a small size. Figure 5.3 (a) illustrates this limited effect. In this scenario, the victim node has multiple potential parents. Therefore, when the victim node receives the replayed consistent DIO messages, it suppresses its DIO according to the trickle algorithm. However, the victim node also receives other DIO messages from the other potential parent (node 3). Accordingly, the victim node reacts by transmitting its DIO. This scenario has low impact on network topology because it only leads to additional exchanges of control messages owing to the victim node not suppressing its DIO where the suppression feature is disabled by other parents.

Another setup is introduced in Figure 5.3 (b) to illustrate the impact of creating sub-optimized paths as mentioned by [258] and [5]. In this setup, the victim node has only the exploited and the attacker nodes as potential parents. In addition, the sub-DODAG of the victim node can connect to the DODAG using another node. In this case, the impact of this attack is moderate on the network topology, as even if the attacker can successfully suppress the victim node, its impact will be limited by hiding only the optimal route through the victim node and forcing the sub-DODAG to use a sub-optimal routing path without partitioning the network.

The third DIO replay attack is the copycat attack, which was first studied by

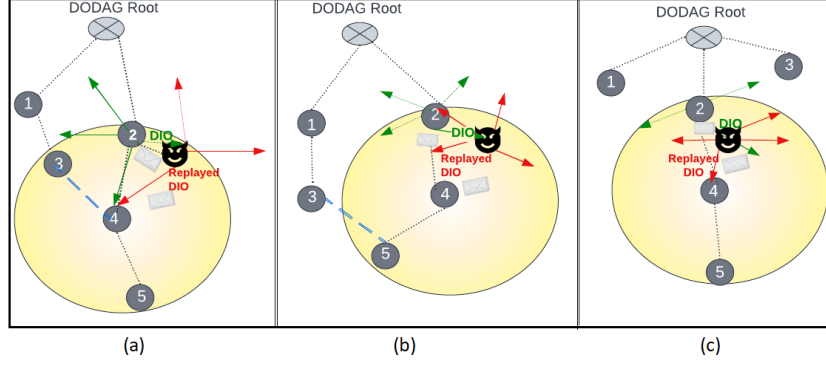


Figure 5.3: Frequent DIO replay attacks analysis based on relative location: (a) low impact, (b) moderate impact, and (c) critical impact.

Verma et al. [261]. In this attack, the attacker receives broadcast DIO messages from a legitimate node, and sends previously received DIO messages many times with a fixed replay interval [71]. There are two types of copycat attacks. The first type is called a spoofed copycat attack, where the attacker replaces the source IP address in the IPv6 packet with the IP address of the original DIO sender. That is, the attacker replays the received DIO message frequently without modification. As a result, the receiver will think the DIO sender is in its communication range. This attack has more impact when victim nodes try to add an attacker that is out of range as a parent, assuming it is a part of the optimal path to the root [261]. In this case, this attack is identical to the DIO suppression attack [258]. The second type is called a non-spoofed copycat attack, where the attacker replays DIO messages with its own IP address. As a result, neighbor nodes will receive false information about the rank of the attacker. This makes these neighbor nodes update their rank values and the selection of their parent using incorrect network information [72]. This attack aims to create sub-optimal routing paths in the attack region [71] and increase the control

message overhead due to the frequency of the replaying packets [72]. It should be noted that the non-spoofed copycat attack will be studied as a copycat attack in this research. Because the mechanism of this attack is similar to the DIO suppression attack (except that it adds the attacker IP), the analysis of the impact of this attack in terms of the attacker and victim nodes' position is identical to the previously explained DIO suppression attack, as shown in Figure 5.3.

Few researchers have tackled the copycat attack. Simoglou et al. [18] classified this attack as a combination of flooding and replay attacks. Pasikhani et al. [17] illustrated copycat attacks are a form of the DIO suppression attack considered to be a topology attack. According to Verma and Ranga [5], the impact of this attack on the network lies in its decreasing PDR and increasing E2E network delay [261], [71], [72].

5.1.2 Attacks Resulting from Replaying DAO

The DAO message plays a vital role in downward traffic. The destination node must send a DAO control message to announce reverse route information along the upward route. Replay attacks launched by a DAO message have not been investigated in the literature. However, Mayzaud et al. and Sharma et al. described attacks that involve replaying the routing information of path sequence in DAO messages by storing legitimate control messages from other nodes and later transmitting them around the network [24], [2]. We study two types of replay attacks based on replay DAO messages. The first type is a DAO replay attack, and the second type is a route table falsification attack. According to Mangelkar et al. [117], all replay attacks are classified as a failure

to secure the integrity of the network. In replay attacks using DAO, a malicious node intends to announce fake routes. The difference appears in the mechanism; in a DAO replay attack, the adversary replays legitimate DAO message not in the victims' sub-DODAG. While in a route table falsification attack, the replayed DAO message is modified to advertise a different source node. It is worth mentioning that launching a replay attack using a DAO is harder than a DIO replay attack because the DAO message is typically unicast from the child node to the preferred parent only [40], [41]. Therefore, the attacker node can be either an internal attacker node or a node that sniffs a DAO packet sent by a child to its parent. We will emphasize the mechanism of each of them, the intent of these attacks as well as their impact on the routing service. We will also highlight the resulting DODAG topology under these attacks based on Table 5.1.

DAO Replay Attack

To launch the DAO replay attack, the attacker node receives a DAO message from the exploited node. Then, it replays the DAO message and unicasts it to the victim node. As a result, the victim node adds the exploited node to its neighbor table (in the case of storing mode) and forwards this DAO to its ancestors. The ancestors add this fake entry in their routing tables (in the storing mode) in addition to the root's routing table. Figure 5.4 highlights various scenarios of the attacker, exploited, and victim nodes. To illustrate the impact of having longer routing paths that lead to more delays (according to Kamble et al. [101] and Raoof et al. [3]), Figure 5.4 (a) represents the scenario of the exploited and victim nodes, where nodes are completely

reachable. In this case, the attacker node replays the DAO message to a neighbor node. Accordingly, the victim node considers the exploited node as a child, and it forwards this DAO message to all its ancestors. When the root wants to send a message to the exploited node, it sends it through the victim node, and then the victim node delivers it to the exploited node because it is a neighbor node. Therefore, this position in a DAO replay attack has a moderate impact on the routing in terms of using a sub-optimized downward path for the exploited node. Also, in the storing mode, it fills the routing tables of the exploited node and all its ancestors with a false entry. Conversely, Figure 5.4 (b) reveals the case where the DAO replay attack has a critical and harmful impact that includes packet drops and an increase in the number of RPL messages [3] and [101]. In this case, the victim node is out of the exploited node range. When the root tries to reach the exploited node, the traffic will flow toward the victim node. Because the victim node does not reach the exploited node, it drops packets directed toward the exploited node. As a result of failing to deliver the required data to the destination, the route purging process will be invoked to reconstruct the downward paths, which requires additional exchanges of control messages and leads to losing multiple data packets.

DAO Replay Attack with Source Forging

According to the literature, this attack is known as a route table falsification attack, and it requires enabling the storing mode. This attack happens when the attacker modifies received DAO messages to build fake downward routes [2], [3]. We define this attack as follows: the route table falsification attack occurs when a malicious

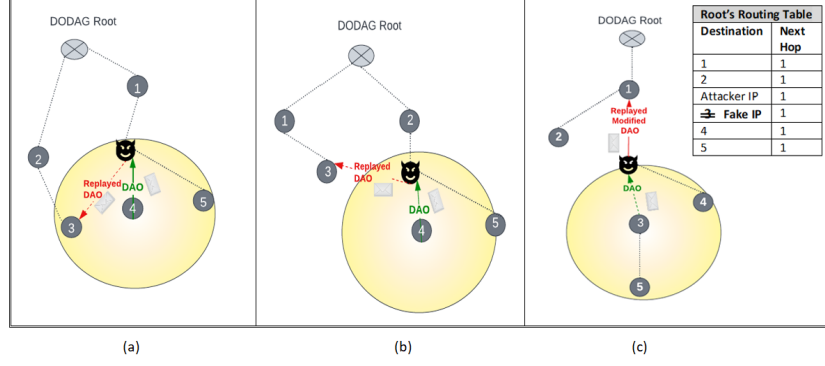


Figure 5.4: DAO replay attacks analysis based on relative location: (a) moderate impact, (b) critical impact, and (c) DAO replay attack with source forging.

node replays the received DAO message after modifying this message by replacing the original source with a fake source. As a result, the replayed DAO message informs other nodes of false routes. The nodes that are advertised in the bogus route might be (a) genuine but not present in the adversary's sub-DODAG or (b) completely fictitious [4]. A thorough examination of how these attacks impact RPL networks has not yet been revealed [3], [4]. Figure 5.4 (c) illustrates the mechanism of this attack that causes false routing entries, as mentioned by Raoof et al. [3] and Kamble et al. [101]. When the attacker node receives a legitimate DAO message from its child, it replays the received DAO message after changing the source IP address to a fake one. The impact of this attack is to hide the downward route of the exploited node because the root cannot reach this node. The effect of this mechanism is critical as the exploited node is isolated from the downward traffic, which leads to lose multiple data packets. This mechanism also announces a false path and adds a fake entry in the routing tables of the root and all victim's ancestors with a storing mode.

Few researchers have studied the route table falsification attack or DAO falsifica-

tion attack. This attack was first explained by Mayzaud et al. [2]. According to [2], a route table falsification attack was considered a topology attack that aimed to impact the optimal paths in DODAG topology. Moreover, Raoof et al. [100] argued that the route table falsification attack is a type of RPL storing mode attack because this attack only happens when RPL runs in storing mode of operation. According to [100], this attack aims to harm routing tables. Verma and Ranga [5] also studied the route table falsification attack and presented this attack as a DAO-related attack, mentioning the impact of forcing legitimate nodes to create non-existing routes. Moreover, [5] discussed a case when a victim node might be filled with wrong routing entries as a result of these fake routes. Table 5.2 summarizes the impact of various replay attacks in RPL based on the attacker’s impact on the DODAG and network topology.

5.2 Detecting Replay Attacks

Researchers utilized different methods and various features to identify and detect replay attacks. For instance, Le et al. [19] presented a monitoring system to detect neighbor attacks, where the monitored values were the number of DIS messages, sequence of DIO messages, stability of the node in the assigned topology, and difference in the rank value between the parent and child. Meanwhile, [260] used location information and received signal strength information in their detecting solution. Moving to another detection system approach proposed in [257], which utilized a trust-based mechanism to detect and isolate neighbor attackers. Another detection method was developed by Farzaneh et al. [140] by considering the total received DIO messages and

Table 5.2: *Evaluation for RPL replay attacks impacts on routing*

Attack	Level of Impact	Positions of Attacker and Exploited Node	Impact
DIO Re-play	Low	Deceived node is neighbor of exploited node	Exchanging additional control messages
	Moderate	Deceived node is out-of-range of exploited node with higher rank than deceived node's parent	+ adding out-of-range neighbor as potential parent
	Critical	Deceived node is out-of-range of exploited node with lower rank than deceived node's parent	++ Attempting to use non-existing upward path
Frequent DIO Replay	Low	Deceived node has multiple potential parents	Exchanging additional control messages
	Moderate	Decedents of deceived node have multiple potential parents	+ Suppressing deceived node; using sub-optimized routing path
	Critical	Deceived node and its sub-DODAG are connected to the DODAG through attacker	++ Partitioning the DODAG
DAO Re-play	Moderate	Deceived node is neighbor of exploited node	Exchanging additional control messages
	Critical	Deceived node is out-of-range of exploited node	+ Adding non-existing downward path; isolating exploited node from downward traffic
DAO Re-play With source forging	Critical	Deceived node forges exploited node's address	Exchanging additional control messages; adding non-existing downward path; isolating exploited node from downward traffic

+ In addition to the impact in the above cell, ++ In addition to the impact in the above two cells

the number of neighbors. While the detection system proposed in [265], an online unsupervised ensemble-based anomaly detection technique, is considered for RPL attack detection, including replay attacks. Another detection scheme was a heuristic-based by Pu and Zhou [85] to detect DIO suppression attacks. This method is built on the threshold value of an acceptable sequence number increment. Similarly, the method introduced by Tahar et al. [266] entails the comparison of the time disparity between two DIO messages with a specified threshold value for each node to identify the node responsible for launching the DIO suppression attacks. Another study proposed by Kim et al. [267], to develop a fuzzy logic-based intrusion detection scheme to detect DIO suppression attacks. Remarkably, few studies were proposed to identify copycat attacks [261], [71], [72]. All these studies suggested using outlier detection to find attacks based on abnormal behavior. It is worth noting that no study has been proposed to detect DAO replay attacks. Figure 5.5 summarizes features commonly used in replay attack detection. The number of RPL packets has been frequently used to detect replay attacks. In contrast, other features like hop count have been rarely utilized for replay attack detection. However, most studies only focused on detecting one type of replay attack. i.e., neighbor or DIO suppression, etc.

To summarize, Table 5.3 shows the proposed detection solutions presented in the literature. It identifies the design approach and the features considered in the design.

Table 5.3: *Summary of proposed security solutions for replay attacks*

References	Designing Approach	Features Used
[97]	Threshold	No. of received DIS, DIO, and DAO packets, sequence of DIS, DIO, and DAO, and parent-ID.
[260]	Location	Received signal strength
[85]	Threshold	Increment in sequence number of messages
[140]	Threshold	No. of received DIO packets, No. of neighbors
[75]	Threshold	No. of received packets from each neighbor
[268]	Threshold	Hop count
[84]	Machine learning	Sender identities in the received messages
[71]	Machine learning	No. of DIO, time interval between successive DIO's
[96]	Machine learning	No. of received and transmitted DIO, No. of received and transmitted DAO, node's rank.
[257]	Trust	No. of successful transmitted packets, No. of packets
[269]	Machine learning	No. of transported DIO and DAO packets, No. of received DIO and DAO packets, times to change parent set, and times to change rank
[247]	Node's Rank	Echelon Metric Based Objective Function
[270]	Deep learning	Convolutional neural network
[267]	Fuzzy logic	DIO counter, Received DIO time, DIO Interval
[266]	Threshold-based	Time between consecutive DIOs
[265]	Machine learning	No. of received and transmitted (DIS, DIO, DAO) packets, No. of received, transmitted, and forwarded UDP packets, average rank, average version forwarding ratio

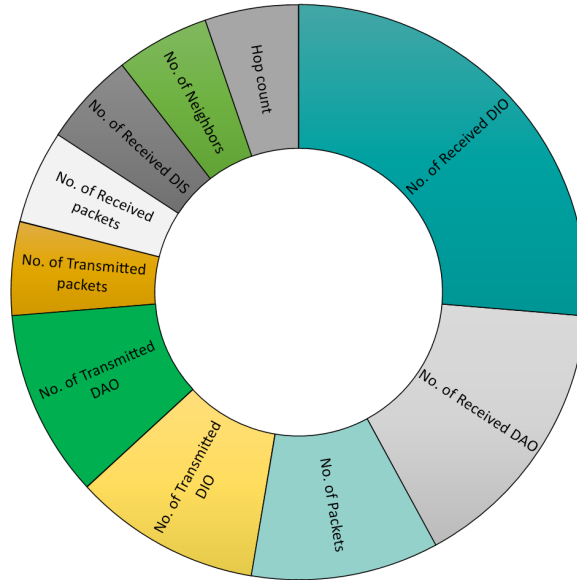


Figure 5.5: *Common used features to detect RPL replay attacks*

CHAPTER 6

DEVS-RPL MODEL

In this chapter, we answer RQ4 about modeling the RPL establishment and maintenance to specify the number and frequency of generated RPL messages. Section 6.1 shows the model objective & scope. Section 6.2 provides the model description and specification. Section 6.3 explains the generation of RPL messages and the sending time.

6.1 Model Objective & Scope

When studying complex systems, such as network protocols, it is essential to analyze their behavior over time. This involves examining the system's structure and the internal time delays that can impact its overall behavior. For instance, in the context of the RPL specification, a node must exchange specific control messages. The goal is to identify the particular control messages that a node is required to exchange and quantify the number of control messages a node generates under normal operating conditions. To build these specifications, we mainly rely on the RPL Standard [41]

and the implementation of RPL in Contiki-NG is an operating system [271].

DEVS introduced by Zeigler in the late 1970s and it provides a hierarchical and modular approach to modeling discrete event systems, where modularity is defined by the ability of a system to interact with its environment through input and output ports [272].

DEVS focuses on processing events rather than studying the system behavior in each individual time step, making it inherently more efficient [22]. This model is crucial for understanding how complex systems evolve over time, particularly when dealing with protocols that involve interactions and delays. In DEVS, each system or model is conceptualized as a “black box”, denoted as S , that receives and sends messages through these ports. The internal behavior of each component within a system is specified using the atomic model [21]. The Atomic DEVS Model is a fundamental building block in DEVS formalism. It describes the behavior of individual components in a system, including how they respond to internal and external events over time[272].

We applied DEVS to model the generation and exchange of control messages in RPL. In our model, we considered the following states to describe the behavior of RPL network nodes:

- Joining: which describes the initial connection attempts of nodes [40].
- Sending: which describes sending a message to another node [39], [40].
- Processing: This represents the state of processing a message by the MC processor, and the node doesn’t generate any RPL message during this state [39], [273].

- Listening: which shows that the node is either receiving or idle [273].

6.2 Model Specification

Formally DEVS is defined as tuples: $(X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a)$. Figure 6.1 provides a visual illustration of our proposed DEVS-RPL model, which can be formally described as follows:

6.2.1 Input, Output, State Set

X is the set of input values that represent external events, and Y is the set of output values. In DEVS-RPL $X = Y = \{DIS, DIO_M^U, DIO_M, DIO_U, DAO\}$.

S is a set of states. $S = \{(phase, \sigma) | phase \in \{“Joining”, “Processing”, “Listening”, “SendingDIS”, “SendingDIO_M”, “SendingDIO_U”, “SendingDAO”, \sigma \in R_{0,\infty}^+\}\}$

6.2.2 Initial State

The starting state in RPL is represented by $\delta_{int}(“s_0 : Joining”, 0)$. This state stays 0-time units and is utilized by all nodes that want to join a DODAG.

6.2.3 Internal and External Transition Functions

Internal transition function is represented as $\delta_{int} : S- > S$ that causes a transition to the next phase when the current state’s time has elapsed. External transition function is defined as $\delta_{ext} : Q \times X^b- > S$ that causes a transition to a new phase as a result of receiving an external event on the input ports. Three parameters are

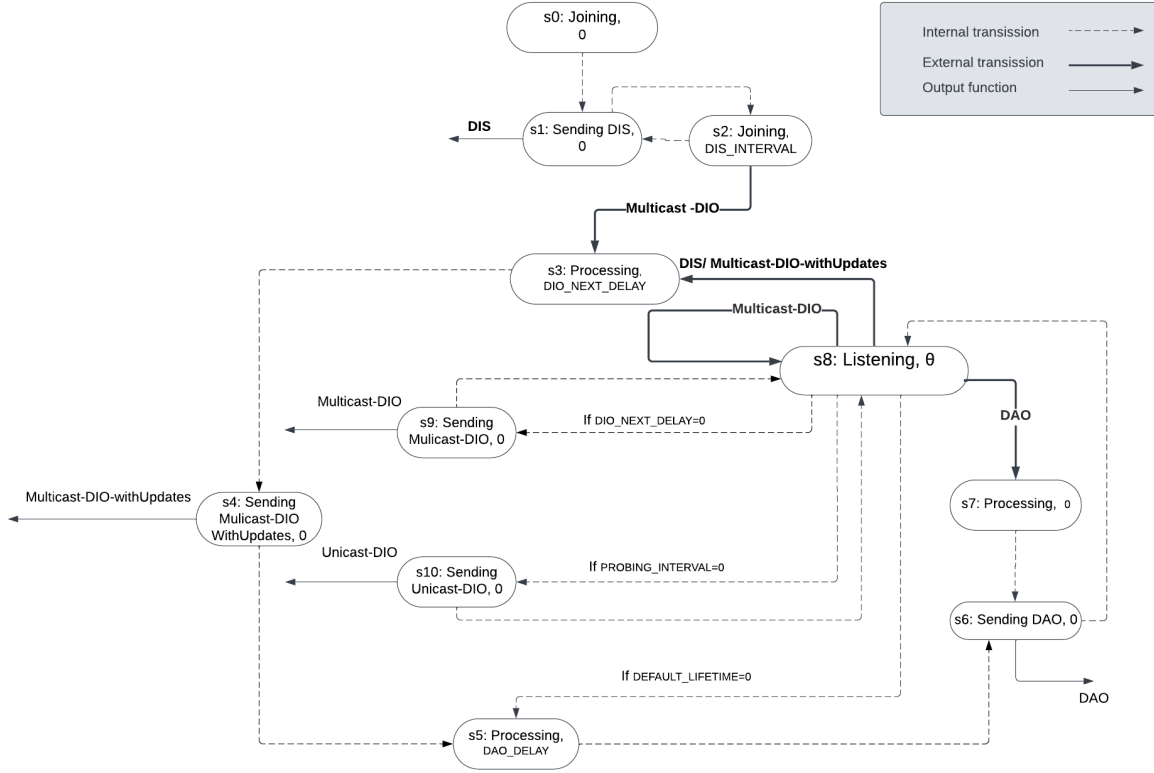


Figure 6.1: *State transitions of DEVS-RPL model*

used to determine the next phase, which are the current state, the port where the input has occurred, and the values of the input. These parameters can be expressed as $Q \in \{(s, e) | s \in S, 0 \leq e \leq t_a(s)\}$ considering that S is total state set and e is the time elapsed since last transmission. In DEVS-RPL, the node changes its state according to the following defined internal and external transition functions:

1. The function $\delta_{int}("s_0 : joining", 0) = ("s_1 : SendingDIS", 0)$ shows that the joining node instantly leaves "joining" state to "Sending DIS" state and it will send its first DIS message. The node lasts in this state 0-time units as it just sends a DIS packet.

2. After sending the first DIS message, the transition function $\delta_{int}("s_1 : SendingDIS", 0) = ("s_2 : Joining", I_{DIS})$ indicates that the node moves back to the joining state, and it lasts for I_{DIS} as specified in RPL [41]. During this interval, the node waits without sending any control message.
3. Accordingly, when I_{DIS} timer expires, and no incoming DIO message is received, the function $\delta_{int}("s_2 : Joining", I_{DIS}) = ("s_3 : SendingDIS", 0)$ tells that the node should move to the "Sending DIS" state. It sends another DIS message [41]. This step will be repeated until the node receives a DIO message.
4. When a node receives a multicast DIO message while in a joining state, the function $\delta_{ext}("s_2 : Joining", \sigma, I_{DIS}, DIO_M) = ("s_3 : Processing", I_{DIO})$ illustrates that the node leaves this state and moves to the "Process" state to process the incoming DIO (to identify its rank and choose its preferred parent) for $I_{DIO} = Min(I_{DIO})$ time unit [41]. Besides, the $I_{Probing}$ is set [271].
5. Once I_{DIO} timer expires, the function $\delta_{int}("s_3 : Processing", I_{DIO}) = ("s_4 : SendingDIO_M^U", 0)$ reveals that the node moves to "Sending Mulicast-DIO-WithUpdates" state and multicast its DIO message [41]. The node spends in this state 0-time units as it just sends DIO packet.
6. After multicasting updating DIO message, the function $\delta_{int}("s_4 : SendingDIO_M^U", 0) = ("s_5 : Processing", D_{DAO})$ shows that the node moves to the "Process" state to prepare a DAO packet for D_{DAO} time unit [41].
7. When D_{DAO} expires, according to the function $\delta_{int}("s_5 : Processing", D_{DAO}) =$

(“ $s_6 : SendingDAO$ ”, 0), the node resets the I_{DAO} and moves to the “Sending DAO” state to unicast DAO message to its parent. The node spends in this state 0-time units as it just sends DAO packet. In addition, Besides, the I_{DAO} is set [271].

8. Following sending DAO message, the function $\delta_{int}(\text{“}s_6 : SendingDAO\text{”}, 0) = (\text{“}s_8 : Listening\text{”}, \theta)$ indicates that the node moves to “Listening” state, where the node can receive control messages. The node stays in this state until the minimum timer lasts (We noted the minimum interval by θ). In this state, the node runs three timers. The first one is $I_{Probing}$, which controls sending unicast DIO packets for probing. The second timer is the I_{DAO} , which controls sending DAO messages to maintain downward routing, and the last one is I_{DIO} , which controls sending multicast DIO messages.

9. The function that reflects the case when a node receives messages with an update in a DODAG including mulicast-DIS, multicast-DIO with a new version number, multicast-DIO with an infinitive rank, or DIO with a new DTSN is $\delta_{ext}(\text{“}s_8 : Listening\text{”}, \sigma, \theta, DIS, DIO_M^U) = (\text{“}s_3 : Processing\text{”}, I_{DIO})$. The node immediately moves to “Processing” to process the incoming DIO (to identify its rank and choose its preferred parent) for $I_{DIO} = Min(I_{DIO})$ time unit [41]. Then, it repeats the steps (5-8).

10. On the other hand, when the $I_{Probing}$ lasts, the function $\delta_{int}(\text{“}s_8 : Listening\text{”}, \theta) = (\text{“}s_{10} : SendingDIO_U\text{”}, 0)$ refers that the node moves to the “Sending Unicast-DIO” state and sends a unicast DIO message. The node

spends in this state 0-time units and returns to the “Listening” state again, and the $I_{Probing}$ also reset [271].

11. After sending a unicast DIO message, the function $\delta_{int}("s_{10} : SendingDIO_U", 0) = ("s_8 : Listening", \theta)$ reveals that the node moves to “Listening” state, where the node can receive control messages. The node stays in this state until the minimum timer θ lasts.

$$\theta = \min(I_{Probing}, I_{DAO} - I_{Probing}, I_{DIO} - I_{Probing})$$

12. When the I_{DIO} lasts, according to this function, $\delta_{int}("s_8 : Listening", \theta) = ("s_9 : SendingDIO_M", 0)$, the node moves to the “Sending Multicast-DIO” state and multicasts DIO message. The node spends in this state 0-time units.

13. After sending a multicast DIO message, the node returns to the “Listening” state again as described in the function $\delta_{int}("s_9 : SendingDIO_M", 0) = ("s_8 : Listening", \theta)$. The node stays in this state until the minimum timer θ lasts.

Accordingly, the θ timers will have the following values:

$$I_{DIO} = \begin{cases} I_{DIO} \times 2 & : \text{if } I_{DIO} \leq \text{Max}(I_{DIO}) \\ I_{DIO} & : \text{Otherwise} \end{cases}$$

$$\theta = \min(I_{Probing} - I_{DIO}, I_{DAO} - I_{DIO}, I_{DIO})$$

14. The function that shows the case when a node receives a multicast DIO message that doesn’t have any DODAG update is $\delta_{ext}("s_8 : Listening", \sigma, \theta, DIO_M) = ("s_8 : Listening", \theta - \sigma)$. Consequently, this message will not impact I_{DIO} timer

[271], and the node stays in its “Listening” state, the remaining interval.

15. In the case that I_{DAO} timer lasts, the function $\delta_{int}("s_8 : Listening", \theta) = ("s_5 : Processing", D_{DAO})$ indicates that the node moves to the “Process” state to prepare a DAO packet for D_{DAO} time unit [271].
16. Consequently, when D_{DAO} expires, the function $\delta_{int}("s_5 : Processing", D_{DAO}) = ("s_6 : SendingDAO", 0)$ specifies that the node resets the I_{DAO} and moves to the “Sending DAO” state to unicast DAO message to its parent [41]. The node spends in this state 0-time units as it just sends DAO packet.
17. After sending a unicast a DAO message, the transition function $\delta_{int}("s_6 : SendingDAO", 0) = ("s_8 : Listening", \theta)$ refers that the node returns to the “Listening” state again where the node can receive control messages. The node stays in this state until the minimum timer θ lasts, where the three timers have the following values: $\theta = \min(I_{Probing} - I_{DAO}, I_{DIO} - I_{DAO}, I_{DAO})$
18. In the case that a node receives a DAO message while in a “Listening” state, the function $\delta_{ext}("s_8 : Listening", \sigma, \theta, DAO) = ("s_7 : Processing", 0)$ reveals that the node leaves this state and moves to the “Process” state to forward the incoming DAO for 0time unit.
19. Accordingly, the function $\delta_{int}("s_7 : Processing", 0) = ("s_6 : SendingDAO", 0)$ points out that the node moves to “Sending DAO” state and unicast DAO message to the preferred parent. The node spends in this state 0-time units as

it just sends DIO packet.

20. After sending a unicast a DAO message, the transition function $\delta_{int}("s_6 : SendingDAO", 0) = ("s_8 : Listening", \theta)$ shows that the node returns to the “Listening” state again, where the node can receive control messages. The node stays in this state until the minimum timer θ lasts.

6.2.4 Confluent Transition Function

This function is executed in cases where the internal and external functions cause a simultaneous transition. In our model, the external transition function proceeds the internal transition function. $\delta_{conf}(phase, \sigma, x) = \delta_{int}(\delta_{ext}(phase, \sigma, \theta, x), \theta - \sigma)$

6.2.5 Output Function

the output function $\lambda : S \rightarrow Y^b$ shows the generated messages to be sent. In DEVS-RPL, according to Figure 6.1, the generated messages are: $\lambda("s_1 : SendingDIS", 0) = DIS$

$$\lambda("s_4 : SendingDIO_M^U", 0) = DIO_M^U$$

$$\lambda("s_9 : SendingDIO_M", 0) = DIO_M$$

$$\lambda("s_{10} : SendingDIO_U", 0) = DIO_U$$

$$\lambda("s_6 : SendingDAO", 0) = DAO$$

6.2.6 Time Advance Function

This function is responsible for providing the time that causes a state transition.

$$t_a(phase, \sigma) = \sigma$$

Table 6.1: *Participant states of each control message*

<i>DIS</i>	<i>DIO_U</i>	<i>DIO_M</i>	<i>DAO</i>
(<i>s</i> ₀ : <i>Joining</i> ",0)	(<i>s</i> ₈ : <i>Listening</i> ", <i>I_{Probing}</i>)	(<i>s</i> ₄ : <i>SendingDIO_M^U</i> ",0)	(<i>s</i> ₆ : <i>SendingDAO</i> ",0)
(<i>s</i> ₁ : <i>SendingDIS</i> ",0)	(<i>s</i> ₁₀ : <i>SendingDIO_U</i> ",0)	(<i>s</i> ₉ : <i>SendingDIO_M</i> ",0)	(<i>s</i> ₈ : <i>Listening</i> ", <i>I_{DAO}</i>)
(<i>s</i> ₂ : <i>Joining</i> ", <i>I_{DIS}</i>)		(<i>s</i> ₃ : <i>Processing</i> ", <i>I_{DIO}</i>)	(<i>s</i> ₇ : <i>Processing</i> ",0)
		(<i>s</i> ₈ : <i>Listening</i> ", <i>I_{DIO}</i>)	(<i>s</i> ₅ : <i>Processing</i> ", <i>D_{DAO}</i>)

6.3 Deriving Messages Generation and Sending Time

According to the presented model. We want to derive the following: First, the generated control messages. Second, the required time for each output message to be generated. We rely on the Output Functions 6.2.1 to consider each output message. In order to determine the required time that each generated message takes, we need to define participant states. let $P \subseteq S$ represent a set of states that lead to generating a specific control message $y \in Y$. The participant state P to generate y control message is: $P(y) = \{p : \lambda(p) = y \text{ or } (\delta_{int}(p) \in P \text{ and } \lambda(p) = \phi)\}$. The required time for any output message to be generated can be extracted from the Time Advance Function (explained in Section 6.2.6), by considering the time of all participant states. We trace all transitions that lead to generating a specific message $y \in Y$ and the consumed time in each of them as: $T(y) = \sum_i t_a(p_i, \sigma) : p_i \in P$. The participant states of each control message are illustrated in Table 6.1.

6.3.1 Generating DIS

To formulate the time of sending a DIS message, $T(DIS_i)$ denotes the time of sending an arbitrary DIS_i message. According to the participant states and the time advance

function, with the assumption that the first DIS is sent at time $T(DIS_1)$, the time of sending a DIS can be presented as:

$$T(DIS_i) = T(DIS_1) + I_{DIS} \times (i - 1) \quad (6.1)$$

Therefore, the number of DIS in given window of time W is calculated by:

$$\#DIS = \frac{W - T(DIS_1)}{I_{DIS}} \quad (6.2)$$

It is worth noting that no external event leads to generating DIS messages.

6.3.2 Generating Unicast DIO

To formulate the time of sending DIO_U message, $T(DIO_{U_i})$ denotes the time of sending an arbitrary DIO_{U_i} message. According to participant states and the time advance function, the time of sending DIO_U relies on $I_{Probing}$, which is presented as:

$$T(DIO_{U_i}) = T(DIO_{U_1}) + I_{Probing} \times (i - 1) \quad (6.3)$$

Where $T(DIO_{U_1})$ is the sending time of the first DIO_U message.

As a result, the number of DIO_U in given window of time W is calculated by:

$$\#DIO_U = \frac{W - T(DIO_{U_1})}{I_{Probing}} \quad (6.4)$$

It's crucial to understand that the generation of a DIO_U message is not triggered by any external events, as DIO_U is a probing message that is sent frequently.

6.3.3 Generating Multicast DIO

To formulate the time of sending a DIO_M message, $T(DIO_{M_i})$ denotes the time of sending an arbitrary DIO_{M_i} message. For the time advance functions of the partici-

pant states, we didn't distinguish between the DIO_M whether it contains an update or not, where I_{DIO} is the timer that controls the emission of this message even with receiving external events. By assuming that $T(DIO_{M_1})$ is the sending time of the first DIO_M message. Accordingly, the time of sending a DIO_M can be presented as:

$$T(DIO_{M_i}) = T(DIO_{M_{i-1}}) + I_{DIO} \quad (6.5)$$

where I_{DIO} is first set to $I_{DIO} = \text{Min}(I_{DIO})$, then it is dynamically updated :

$$I_{DIO} = \begin{cases} I_{DIO} \times 2 & : \text{if } I_{DIO} \leq \text{Max}(I_{DIO}) \\ I_{DIO} & : \text{Otherwise} \end{cases}$$

The time between consecutive messages increases exponentially at the beginning when the network is constructed. However, the time between messages is fixed when the network is stable [41]. When a node receives messages that show an update in a DODAG including multicast-DIS, multicast-DIO with a new version number, multicast-DIO with an infinitive rank, or DIO with a new DTSN, the node immediately resets to $I_{DIO} = \text{Min}(I_{DIO})$ [41]. We assume that the difference in sending rate distinguishes between two network stages. The first stage is when the network is under construction, whereas the second one is when it is under maintenance. We define the construction stage as the time before reaching the $\text{Max}(I_{DIO})$.

Table 6.2: *DEVS-RPL specification of received messages*

RPL message	Phase	Minimum	Maximum
DIS	-	0	$\frac{NW}{I_{DIS}}$
DIO_U	-	$\frac{WN}{4I_{probing}}$	$\frac{2NW}{I_{probing}}$
DIO_M	Construction	$\frac{NW}{\text{Max}(I_{DIO})}$	$N \log_2 \frac{W}{\text{Min}(I_{DIO})}$
DIO_M	Maintenance	$\frac{NW}{2\text{Max}(I_{DIO})}$	$\frac{2NW}{\text{Max}(I_{DIO})}$
DAO	Construction	0	$N \log_2 \frac{W}{\text{Min}(I_{DIO}) + D_{DAO}}$
DAO	Maintenance	0	$\frac{2NW}{I_{DAO}}$

6.3.4 Generating DAO

To formulate the time of sending a DAO message, $T(DAO_i)$ denotes the time of sending an arbitrary DAO_i message. According to participant states and the time advance function, the transition that leads to reset I_{DAO} appears in two cases: first is when I_{DAO} expires, and the time is due to send a new DAO. The second case is when the node sends a DIO_M^U that includes an update in the DODAG, this update requires multicast a DIO with update as well as sending a DAO as unicast to the parent node as reflected in Figure 6.1. Denoting the time of sending a DIO_M^U before DAO_i by $T(DIO_{Mj}^U)$, $T(DAO_i)$ can be written as follows:

$$T(DAO_i) = \begin{cases} T(DAO_{i-1}) + (I_{DAO} + D_{DAO}) & : \text{if } T(DAO_{i-1}) \geq T(DIO_M^U) \\ T(DAO_R) + D_{DAO} & : \text{if received DAO} \\ T(DIO_M^U) + D_{DAO} & : \text{Otherwise} \end{cases} \quad (6.6)$$

Therefore, in order to determine the number of DAO messages sent, we counted the DIO messages containing updates. Additionally, we have to calculate the time intervals between each pair of consecutive multicast DIO messages with updates. If this interval is greater than the sum of I_{DAO} and D_{DAO} , we can infer that a DAO message has been sent. Moreover, sending DAO messages is linked to receiving DAO messages, as all received DAO messages are forwarded.

6.3.5 Deriving the Number of Received Messages

After demonstrating the number of generated RPL messages, we want to specify the number of received RPL messages by each node, considering the number of neighbor nodes. Table 6.2 shows the range of each control message the node receives in

normal operations of RPL considering the defined generated messages and the added randomness factor for each defined interval [271]. For the received *DIS*, the minimum number of received *DIS* packets is 0, which indicates that no new node wants to join the DODAG. However, the maximum number of *DIS* packets might reach to the double number of neighbor nodes in each I_{DIS} interval. For the received *DIO_U*, it is essential that at least half of the neighbors receive a minimum of one *DIOU* packet during each round trip of a probing message in every $I_{Probing}$ interval. Each node must actively send *DIO_U* packets to its neighbors to ensure that statistics are consistently updated. Regarding *DIO_M*, we distinguish between the construction time of the network and the maintenance time. During the construction time, the Upper Control Limit (UCL) according to RPL specification is $\log_2 \frac{W}{Min(I_{DIO})}$ by each neighbor. The minimum number of received messages is $\frac{NW}{Max(I_{DIO})}$, which is also the maximum number of the received *DIO_M* during the maintenance time according to RPL specification. During the maintenance time, the minimum number is a message per half of the neighbors in each $Max(I_{DIO})$ interval. Similarly, for DAO, it follows the *DIO_M*; the only difference appears in the timers, which consider D_{DAO} as an additional timer during the construction stage and I_{DAO} as the used timer in the maintenance stage.

CHAPTER 7

PROPOSED DEVS-RPL IDS

This chapter answers RQ5 by showing the design of a lightweight detection system that utilizes the RPL specifications of the number and frequency of exchanged control messages. In details, Section 7.1 shows how to utilize Statistical Process Control (SPC) in anomaly detection. Defining limits in control messages is illustrated in Section 7.2. In addition, Section 7.3 provides the implemented algorithms to detect anomalies.

7.1 Anomaly Detection using Statistical Process Control

The detection system is built on SPC, which utilizes statistical techniques to monitor and manage a process. SPC tools and procedures are valuable for monitoring process behavior and identifying anomalies in internal systems. In the context of RPL, each node monitors the RPL messages it receives and assesses the quantity of received messages. SPC procedures are based on the concept that even if the generation of control messages remains constant, observations of that process will naturally exhibit variability in the statistical sense. For example, random factors can lead to negligi-

ble variations in the number of received messages between nodes, but overall, nodes function normally. If the variability of message generation is understood, control limits can be established so that the majority of new node observations fall within these limits as long as the process remains unchanged (i.e., remains in control). However, when the characteristics of the process change, such as when a node initiates an attack or malfunctions, the features of the observations also change, resulting in different generation or sending behavior that exceeds the control limits [274]. SPC procedures involve two distinct research phases. During the first phase, the nature of the in-control data is captured and specified to establish the in-control distribution. The estimated mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of this phase are then used to confidently determine the upper control limit (*UCL*) and lower control limit Lower Control Limit (*LCL*). Next, in the second phase, the actual monitoring starts, enabling comparisons between incoming data and the specified in-control distribution to determine whether the process is in control. Although there is no study that utilized SPC in attack detection on RPL. Few studies utilized this approach in attack detection in WSN, including [275] and [276]. Typically, in SPC, the monitored value generally is the metric or feature that reflects the behavior of the network. The PDR was monitored by both studies to detect attacks where [275] detected jamming attacks, and [276] focused on distributed denial of service attack detection. The *UCL* and *LCL* are defined to identify deviations from normal behavior. Control Limits for both studies are calculated as follows:

$$UCL = \mu + k \cdot \sigma \quad (7.1)$$

$$LCL = \mu - k \cdot \sigma \quad (7.2)$$

Where μ is the mean value and σ is the standard deviation of the monitored value during normal conditions. k is typically set to 3 for a 99.73% confidence interval.

Anomaly detection in Statistical Process Control (SPC) can be significantly enhanced by utilizing Shewhart, Exponentially Weighted Moving Average (EWMA), and the cumulative sum (CUSUM) charts, each offering unique benefits [277], [278]. Shewhart charts rely on the Central Limit Theorem to assume data is approximately normally distributed, even when population mean and variance are unknown [278]. These charts are ideal for detecting large, abrupt shifts in processes, and they calculate control limits under the assumption of independent subgroups. For processes with mild autocorrelation, creating subgroups with multiple samples can effectively reduce dependency. However, in highly autocorrelated processes, such as slow-moving systems sampled at high rates, control limits may trigger frequent false alarms [278]. On the other hand, the CUSUM chart, with its infinite memory, excels in detecting small, persistent deviations from target values [277]. However, CUSUM is sensitive to small changes, making it less suited for lossy network applications, where packet loss or delays introduce inherent variability. EWMA charts bridge the gap between these two extremes by assigning adjustable weights to historical data, allowing fine-tuning of control limits [278]. This makes EWMA charts the most versatile, particularly for received RPL packets that have high degrees of correlation and dependency. In our study, we rely on EWMA procedure, introduced by Roberts in 1959, which is a statistical method used to detect changes in the mean of time series data. It contin-

uously monitors real-time data, combining past and present information to calculate a weighted sum of the original observations [274]. We selected this method because it effectively combines current data with historical observations through a weighted average, utilizing a weighting factor for both the most current sample observations and historical data [279]. This technique suits the nature of lossy networks where the near history of the observed data is considered to assess the health of the network nodes. EWMA method assigns specific weights to recent observations, allowing it to consider both historical and the changes in the data. At each time i , the exponentially weighted moving average z_i is calculated according to equation (7.3) [280]:

$$z_i = \alpha x_i + (1 - \alpha)z_{i-1} \quad (7.3)$$

x_i denotes the observation at time i , and the initial value z_0 is $\hat{\mu}$ from the first phase. The parameter $0 \leq \alpha \leq 1$ is the weight given to the current observation and determines the rate of considering the previous observations. According to the literature in SPC, it is recommended to use values between 0.05 and 0.25, while lower values of α are employed to detect smaller mean changes [281]. However, the application has the main role of identifying the appropriate value for α . Given α , $\hat{\mu}$, and $\hat{\sigma}$ from the first phase, the upper and lower control limits are computed as:

$$UCL = \hat{\mu} + L\hat{\sigma}\sqrt{\frac{\alpha}{2-\alpha}[1 - (1 - (\alpha))^{2i}]} \quad (7.4)$$

$$LCL = \hat{\mu} - L\hat{\sigma}\sqrt{\frac{\alpha}{2-\alpha}[1 - (1 - (\alpha))^{2i}]} \quad (7.5)$$

The parameter L determines the width of the control limits. And the term $[1 - (1 - (\alpha))^{2i}]$ approaches 1 when i increases. To detect anomalies, the SPC procedure requires

Table 7.1: *The statistical parameters of RPL messages*

RPL mes- sage	Stage	Distribution	$\hat{\mu}$	$\hat{\sigma}$	UCL	LCL
<i>DIS</i>	-	Bernoulli($p = \frac{N-1}{N}$)	$\frac{N-1}{N}$	$\sqrt{\frac{N-1}{N}}$	$2N$	0
<i>DIO_U</i>	-	Poisson($\frac{2W}{I_{probing}}$)	$\frac{2W}{I_{probing}}$	$\sqrt{\frac{2W}{I_{probing}}}$	$\frac{2NW}{I_{probing}}$	$\frac{WN}{4I_{probing}}$
<i>DIO_M</i>	C	Lognormal ($\ln(\frac{NW}{T(DIO_M)}, \sigma^2)$)	$E(X) = \frac{NW}{T(DIO_M)} e^{\frac{\sigma^2}{2}}$	$E(X)\sqrt{e^{\sigma^2} - 1}$	$N \log_2 \frac{W}{Min(I_{DIO})}$	$\frac{NW}{Max(I_{DIO})}$
<i>DIO_M</i>	M	Poisson($\frac{NW}{Max(I_{DIO})}$)	$\frac{NW}{Max(I_{DIO})}$	$\sqrt{\frac{NW}{Max(I_{DIO})}}$	$\frac{2NW}{Max(I_{DIO})}$	$\frac{NW}{2Max(I_{DIO})}$
<i>DAO</i>	C	Lognormal ($\ln(\frac{NW}{T(DIO_M)+D_{DAO}}, \sigma^2)$)	$E(X) = \frac{NW}{T(DIO_M)+D_{DAO}} e^{\frac{\sigma^2}{2}}$	$E(X)\sqrt{e^{\sigma^2} - 1}$	$N \log_2 \frac{W}{Min(I_{DIO})+D_{DAO}}$	0
<i>DAO</i>	M	Poisson($\frac{NW}{I_{DAO}}$)	$\frac{NW}{I_{DAO}}$	$\sqrt{\frac{NW}{I_{DAO}}}$	$\frac{2NW}{I_{DAO}}$	0

specifying the distribution of each RPL message. In the context of DIS messages, nodes are expected to receive these messages during the initial construction phase of the network, specifically within the first interval. Nodes should receive DIS messages from their neighboring nodes, excluding the parent node, since the parent is closer to the root. This proximity allows the parent to begin processing the received DIO and stop sending DIS ahead of its children. Therefore, the likelihood of receiving a DIS message follows a Bernoulli distribution, reflecting the probability of receiving it from non-parent neighbors compared to all neighboring nodes. For DIO and DAO messages, as Table 6.2 indicates, most of the network time, received RPL messages adhere to a Poisson distribution due to their occurrence at a fixed frequency [282]. However, during the construction time, the received messages follow lognormal distribution as the time between message generations increases exponentially. To apply the SPC procedure to detect anomalies, the distribution of each RPL message, the mean, and the standard deviation are illustrated in Table 7.1.

7.2 Defining limits in Control Messages

For each RPL message, our definition for UCL and LCL are based on the minimum and maximum number of received messages illustrated in Table 6.2. We utilized equations(7.4),(7.5) to define specify the associated α and L . It is worth noting that DIS message is only received from new nodes. This message doesn't require applying EWMA, as it should only be sent by a new node and will stop once a node joins the network. Therefore, the UCL of DIS is $2N$ in the first I_{DIS} interval, and the LCL is 0, meaning there are no new nodes.

7.3 DEVS-RPL IDS Monitoring Implementation

We provide the algorithms designed to monitor the network and detect anomalies. Algorithm 1 should be executed in the root node to determine the monitoring window W , the upper and lower limits UCL and LCL for each RPL message according to table 7.1. The weights are also initialized as $0 \leq \alpha \leq 1$. These parameters are transmitted to all nodes in an application packet.

Algorithm 1: Initialize Limits and monitoring interval

Initialize Window: W
Initialize Upper Limits: $U [DIS, DIO_U, DIO_M, DAO]$,
Initialize Lower Limits: $L [DIS, DIO_U, DIO_M, DAO]$,
Initialize Weights: $\alpha [DIS, DIO_U, DIO_M, DAO]$
Send to each node (W, U, L, α)

Each node executes Algorithm 2 to monitor the network and report any detected anomalies in neighbor nodes. According to this algorithm, nodes need to receive the monitoring window (W), UCL and LCL, and initial weights (α) for each RPL message.

Algorithm 2: Monitoring network & detect anomalies

Require: Window: W ,
Upper Limits: $U [DIS, DIO_U, DIO_M, DAO]$,
Lower Limits: $L [DIS, DIO_U, DIO_M, DAO]$,
Weights: $\alpha [DIS, DIO_U, DIO_M, DAO]$
Input: ICMPV6 packet
Initialize $Received [DIS, DIO_U, DIO_M, DAO] = 0$
Initialize $IsAnomaly = False$
Initialize $Mean [DIS, DIO_U, DIO_M, DAO] = 0$
if $ICMPV6.Type=155$ **then**
 if $ICMPV6.code=DIS$ **then**
 $Received [DIS] ++$
 else
 if $ICMPV6.code=DIO$ **then**
 if $ICMPV6.DestinationIPaddress=Multicast\ Address$ **then**
 $Received [DIO_M] ++$
 else
 $Received [DIO_U] ++$
 end
 else
 if $ICMPV6.code=DAO$ **then**
 $Received [DAO] ++$
 end
 end
 end
end
if W expires **then**
 foreach $Message \in [DIS, DIO_U, DIO_M, DAO]$ **do**
 $Mean_{[Message]} =$
 $\alpha_{[Message]} Received_{[Message]} + (1 - \alpha_{[Message]}) Mean_{[Message]}$
 if $Mean_{[Message]} \ll L_{Message}$ or $Mean_{[Message]} \gg U_{Message}$ **then**
 $IsAnomaly = True$
 end
 end
 if $IsAnomaly = True$ **then**
 Send To Root (NodeID, NeighborList)
 end
end

Upon receiving the RPL message, the ICMPV6 packet with type=155 fetches its code to determine whether it is a DIS, DIO, or DAO in order to increase the corresponding counter. When W expires, the mean of received RPL messages is updated according to equation (7.3). If the obtained mean is not within the specified UCL and LCL. Subsequently, the node raises a flag and reports an anomaly in its neighboring nodes.

CHAPTER 8

EXPERIMENTAL EVALUATION

This chapter presents the experiments implemented throughout this study. First, we implemented replay attacks and we evaluated their impacts on the network as demonstrated in Section 8.1. Then, Section 8.2 shows our validation of DEVS-RPL model. Finally, Section 8.3 reveals the implementation and evaluation of our proposed DEVS-RPL IDS detection solution.

8.1 Replay Attacks Impacts

We conducted an empirical evaluation with the objective to examine and compare the local impact of five types of replay attacks, as explained in chapter 5. The evaluated attacks are neighbor, DIO suppression, copycat, DAO replay, and route table falsification, considering the highlighted attackers' positions. The core assumption of the evaluation is that the success of an attacker in degrading the routing service of even a single node demonstrates a potential threat to the entire network, regardless of its size.

Table 8.1: *Simulation parameters of replay attacks evaluation*

Parameter	Value
Sensor nodes type	Zolertia Z1 mote
PHY and MAC layer	IEEE 802.15.4 with CSMA and ContikiMAC
Objective function	MRHOF OF
Mode of operation	Storing
No. of replay attacks	5
Simulation time	5 min, 20 min, 60 min
Node positioning	Illustrated in Figures 5.2, 5.3, and 5.4
Deployment area	100m W $\times$ 100m L
Number of nodes (adversary included)	7

8.1.1 Experimental Setup and Design

The simulation environment uses Contiki-NG because it is an open-source operating system focusing on low-power IoT devices. In addition, Contiki-NG is also lightweight and has flexible resource management modules [271]. RPL protocol is already implemented in Contiki-NG. Cooja is the simulation environment in Contiki-NG that reflects the actual compiled IoT networks.

The experiment has been set to mimic the specific scenarios depicted in Figures 5.2, 5.3, and 5.4 in terms of the number of nodes and their positions. Matching these scenarios allows for clear validation and focused analysis of the attack impacts. It is worth mentioning that small-scale networks with a single attacker are sufficient to capture the behaviors and impact of the RPL attacks under investigation without compromising the validity of the results, as reported in [195], [88]. The simulation parameters are summarized in Table 8.1.

Table 8.2 shows the different replay attack configurations. Under replay attack scenarios, control messages are replayed once for neighbor, DAO replay, and route table falsification attacks [2], [100]. In comparison, DIO suppression and copycat attacks

replay the received DIO messages 10 times within a five-second interval. The attacker node keeps performing the replay attack whenever it receives a control message from the exploited node. In our experiment, the nodes are set to compute their rank based on MRHOF objective function [39].

Table 8.2: *Configuration parameters in different RPL replay attacks*

Replay Attack	Control Message	Modification	No. of Replay	Interval
Neighbor	DIO	-	1	-
DIO suppression	DIO	-	10	5 sec.
Copycat	DIO	Sender's IP	10	5 sec.
DAO replay	DAO	-	1	-
Route table falsification	DAO	Sender's IP	1	-

8.1.2 Evaluation Metrics

To analyze replay attacks' effects on the network, four evaluation metrics are used, as defined below. The evaluation metrics are implemented in the simulation script editor.

- **Packet delivery ratio (PDR):** the packet delivery ratio is calculated by dividing the number of received data packets by the number of sent packets at the destination node.

$$PDR = (PacketsReceived)/(PacketsSent) \quad (8.1)$$

- **End-to-end delay:** E2E delay is measured as the consumed time to transmit packets from one node to another through the network, including both processing and queuing time [250]. Considering n to be the number of received packets,

the delay is calculated as follows:

$$E2E_D = \sum_{1}^n (ReceiveTime - SentTime)/n \quad (8.2)$$

- **Control messages overhead:** this measure calculates the average number of control messages exchanged in the network to maintain the DODAG topology, including DIS, DIO, DAO, and DAO-ACK control messages [252].
- **Power consumption:** the power consumption is computed by recording the total energy needed to transmit all the packets, which is obtained using the Contiki-NG (energest.h) library. The power (in milliwatts) is calculated by multiplying the current of each state by the period of time the sensor node spends operating in each of the four states, CPU, low power mode (LPM), transition (Rx), and receive (Tx) together by the voltage [19]. The voltage and current values are taken from the Z1 sensor datasheet [283], as shown in Table 8.3. In general, the energest time is the number of ticks spent in a specific state in each interval, which needs to be divided by Contiki's RTIMER_ARCH_SECOND = 32768 [132] to obtain the time in seconds. The power consumption for all states is computed using Equation 8.3 [19]. To compute the average power consumption of impacted nodes in the network, the power consumed by all nodes, excluding the attackers, is summed. This value is averaged by the number of neighbors to find the local impact of average power consumption.

$$PowerConsumption = \frac{EnergestValue \times Current \times Voltage}{RTIMER_ARCH_SECOND \times Period} \quad (8.3)$$

Table 8.3: *Z1 mote power specification*

Attribute	Value
Voltage	3 V
Current transmit (Tx)	17.4 mA
Current receive (Rx)	18.8 mA
Current CPU	10 mA
Current idle	0.023 mA

8.2 DEVS-RPL Model Validation

8.2.1 RPL Network Dataset

To verify the correctness of the proposed DEVS-RPL model in terms of the derived sent messages, and the sending time, we conducted extensive simulations using different RPL network topologies and extended simulation times. The simulation environment utilizes Cooja simulator. We used multiple network layouts: random, binary, and grid layouts. Each selected layout has the following characteristics: The random layout consists of different nodes with diverse distances and numbers of neighbors that vary between 2 and 5 neighbors, as illustrated in Fig.8.2(a). The binary layout allows each node to have a minimum of one and a maximum of three neighbors, as shown in Fig. 8.2(b), which creates an efficient representation for a sparse network. Conversely, we choose a grid layout to represent a dense network. This topology guarantees maximum redundant paths, with each node surrounded by a minimum of three (for nodes at the edge of the grid) and a maximum of eight neighbors, as demonstrated in Fig.8.2(c). To obtain a detailed view of the network, the simulation was run under normal conditions for a duration of 7 hours, providing a comprehensive understanding of the network's behavior.

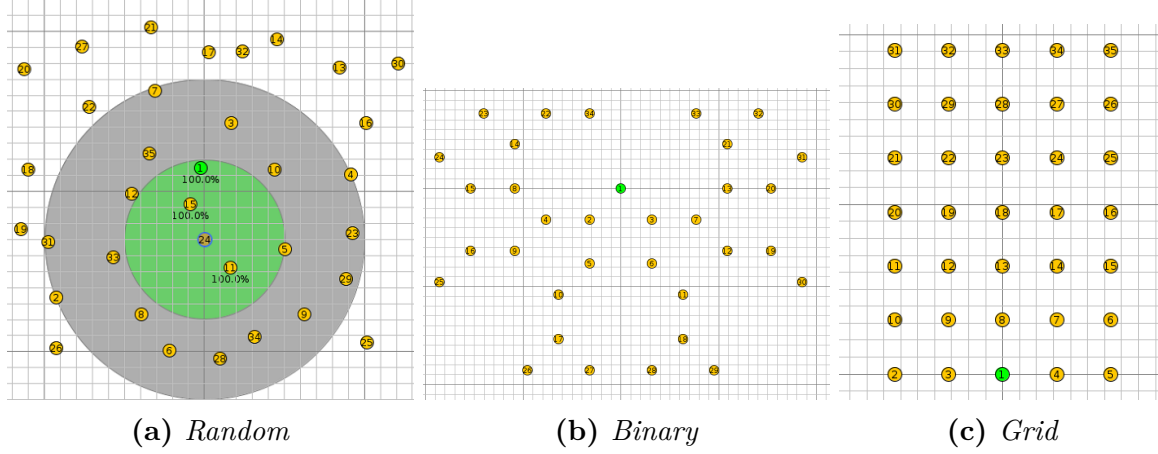


Figure 8.1: *Network layouts in used dataset*

The specifications of the configuration are conveniently illustrated in Table 8.4. It is worth noting that the root node is distinguished by a pink color.

Table 8.4: *RPL simulation settings*

Simulation Settings	Random Layout	Binary Layout	Grid Layout
Simulation area	200 x 200	450 x 350	250 x 300
Number of nodes	34 + 1 Sink		
Mote type	Zolertia Z1		
Simulation time	7 Hours		
Mode of operation	Storing		
Objective function	MRHOF		

In addition, we provide all parameter values as summarized in Table 8.5. These parameters and the added randomness factor are extracted from `rpl-conf.h` and `rpl-timers.c`. We used these parameters to feed DEVS-RPL to validate the model by comparing the generated control messages based on this model to the number we got from the different network layouts.

Table 8.5: RPL parameters

Parameter	Value	Randomness factor
I_{DIS}	60 seconds	$I_{DIS}/2$
$I_{Probing}$	60 seconds	$I_{Probing}/2$
$Min(I_{DIO})$	4 seconds	-
$Max(I_{DIO})$	1048 seconds	-
I_{DIO}	-	$I_{DIO}/2$
D_{DAO}	4 seconds	-
I_{DAO}	30 minutes	$I_{DAO}/2$

Table 8.6: DEVS-RPL parameters

Stage	Control message	Time between messages	# Messages
-	DIS	$Uni(0.5*I_{DIS}, I_{DIS})$	1
-	DIO_U	$Uni(0.5*I_{Probing}, I_{Probing})$	1
Construction	DIO_M	$Exp(T(DIO_{M_i}))$	1
Maintenance	DIO_M	$Uni(0.5*Max(I_{DIO}), Max(I_{DIO}))$	1
Construction	DAO	$Exp(T(DIO_{M_i}) + D_{DAO})$	Average DODAG Degree
Maintenance	DAO	$Uni(0.75*I_{DAO}, I_{DAO}) + Max(I_{DIO}) + D_{DAO}$	Average DODAG Degree

8.2.2 DEVS-RPL Model Dataset

DEVS-RPL model was used to generate control messages using SimQuick. SimQuick is a powerful and user-friendly tool that allows modeling and simulating various processes using Microsoft Excel [284]. We modeled the generated control messages and the time between each message according to equations (6.1) - (6.6), besides the added/-subtracted randomness factor in Table 8.5.

8.2.3 Dataset Distribution Validation

To assess the distribution of the received message, we begin by visualizing the obtained dataset with a histogram. Next, we fit the distributions and conduct a goodness-of-fit test (such as the Kolmogorov-Smirnov test) to determine which distribution best aligns with the empirical data [282]. For this evaluation, we used the Accord.NET

Framework, an open-source .NET machine learning framework that provides tools for pattern recognition, statistical analysis, and predictive modeling [285].

- **KS Test Statistic:** The KS test statistic measures the maximum difference between the empirical distribution function (EDF) of a sample and the cumulative distribution function (CDF) of a reference distribution [282]. A lower value of the KS statistic signifies a closer alignment between the sample and the reference distribution.
- **P-Value and ‘Best Fits’:** The p-value indicates the likelihood of observing a Kolmogorov-Smirnov (KS) statistic as extreme as the one calculated, assuming the null hypothesis is true (that the data follows the specified distribution). A high p-value (greater than 0.05) suggests that we fail to reject the null hypothesis, implying that the data likely adheres to the specified distribution. In contrast, a low p-value (less than or equal to 0.05) leads us to reject the null hypothesis, indicating that the data does not fit the distribution [282].

8.3 Evaluation of DEVS-RPL IDS Detection

8.3.1 Evaluation Setup and Assumptions

To evaluate our proposed DEVS-RPL IDS, we conducted a comparison of performance between our proposed solution and other specification-based IDS including [19], [195], [247], [74]. The proposed IDS tested against replay attacks: neighbor, suppression, DAO replay, copycat, and route table falsification attacks (RTF). Besides, We also

tested DEVS-RPL IDS against other types of RPL attacks, including blackhole attacks (dropping packet attacks), DIS flooding (packet-generating attacks), and local repair (modifying packets attacks). Our experiments also use Cooja as the simulation environment. We selected Zolertia Z1 sensors that run the proposed DEVS-RPL IDS. We choose Z1 as it is more efficient in energy consumption and memory usage than other sensor types like Sky mote [286], [287]. To evaluate the performance of DEVS-RPL IDS to detect attacks considering different network densities, We used multiple network layouts: random, binary, and grid layouts as shown in Figure 8.2. The light green color identifies the root node, while the pink color highlights the attacker node. The setup specifications are illustrated in Table 8.7. In addition, Table 8.7 illustrates the necessary values we used in Algorithms 1, 2 to calculate the parameters. These interval values are extracted from `rpl-conf.h` and `rpl-timers.c`.

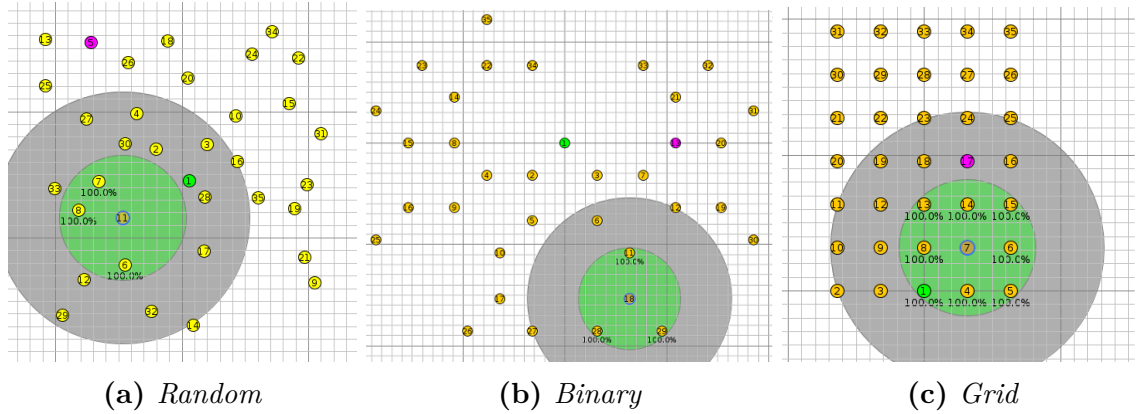


Figure 8.2: *Network layouts used for the evaluation*

Table 8.7: *Simulation parameters of DEVS-RPL IDS evaluation*

Layout	Binary, grid, and random
Number of nodes (adversary included)	34 + 1 Sink
Mote type	Zolertia Z1
Simulation time	60 Minutes
Mode of operation	Storing
Objective function	MRHOF
I_{DIS}	60 seconds
$I_{Probing}$	60 seconds
$Min(I_{DIO})$	4 seconds
$Max(I_{DIO})$	1048 seconds
D_{DAO}	4 seconds

8.3.2 Evaluation Metrics

We used several metrics to measure the accuracy and effectiveness of the proposed IDS in classifying malicious and normal packets. In addition, we measure the network performance with the implementation of DEVS-RPL IDS. These metrics are the following:

- **Detection Accuracy:** The accuracy of attack detection is the ratio between the correct prediction to all predictions in the system, and it is calculated as stated in equation (8.4) [288]:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (8.4)$$

- **False Positive Rate (FPR):** Another evaluation metric is the false positive or false alarm rate. This metric represents the percentage when IDS raises the alarm where nodes function normally and no intrusion happens. The False alarm rate is calculated by dividing the number of raised alarms where there was no attack by the total number of alarms as illustrated in equation (8.5) [288]:

$$FPR = \frac{FP}{TN + FP} \quad (8.5)$$

- Sensitivity or Recall (True Positive Rate (TPR)): This metric is also called the sensitivity rate. The true positive rate is calculated by dividing the number of raised alarms where an attack occurs by the total number of attacks as shown in equation (8.6) [19]:

$$TPR = \frac{TP}{TP + FN} \quad (8.6)$$

- Precision (Positive Predictive Value): Precision quantifies the ratio of correctly predicted malicious behavior (TP) to the total predicted malicious behavior (TP + FP) [19]:

$$Precision = \frac{TP}{TP + FP} \quad (8.7)$$

- Specificity (True Negative Rate): Specificity measures the proportion of correctly identified normal behavior (TN) to the total actual normal behavior (TN + FP) [289].

$$Specificity = \frac{TN}{TN + FP} \quad (8.8)$$

- F1 Score: The F1 Score is the harmonic mean of Precision and Recall [23]:

$$F1 \text{ Score} = 2 \frac{Precision \cdot Recall}{Precision + Recall} \quad (8.9)$$

- Matthews Correlation Coefficient (MCC): MCC is a balanced measure that accounts for all four categories in the confusion matrix (TP, TN, FP, FN) [163].

It is defined as follows:

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (8.10)$$

- Detection Latency: the detection latency is measured as the amount of time taken to detect the existence of attacks.
- Power Consumption: the power consumption is computed as previously demonstrated in Subsection 8.1.2.
- Memory Consumption: This metric is defined as the overhead of implementing the detection solution in terms of memory requirement, including both static memory (RAM) and flash memory (ROM) to be dedicated for overall code and data [195], [247].

CHAPTER 9

RESULTS AND DISCUSSION

This chapter outlines the outcomes of the executed experiments. Section 9.1 presents the results obtained from implementing replay attacks and their impact on the network. While Section 9.2 encompasses the validation results of DEVS-RPL and the correctness evaluation. Furthermore, Section 9.3 addresses RQ5 and presents the evaluation metrics of the introduced IDS, including accuracy, false positive rate, true positive rate, as well as memory and power consumption.

9.1 Replay Attacks Evaluation

9.1.1 Packet Delivery Ratio (PDR)

To evaluate PDR, Figure 9.1 shows the local effect of various forms of replay attacks on PDR on nodes that are neighbors of the attacker. Figure 9.1 reveals neighbor attacks in all its positions have no impact on data delivery. Despite advertising for an out-of-range node, the victim node finds a legitimate path to transmit data packets successfully. Nevertheless, DAO replay and route table falsification attacks lead to

losing 15% of the data packets when the attacker has a critical impact. We found the lost packets in the DAO replay attack are the data sent by the root to the node that has been exploited where no data packets can be delivered to them. In this case, when the root sends a data packet to the exploited node, the data is sent to the node that has received the replayed DAO packet. When this node is unable to reach the exploited node, the data packet is dropped. For route table falsification attacks, even though the data is sent through its legitimate route toward the destination, the data packet is directed to a forged IP. Therefore, it does not reach the target node.

For DIO suppression and copycat, both attacks have identical impacts on whether the DIO packet is modified before replaying or not. The results reveal the attacker's position has the largest impact on data delivery. When the attacker node locates in the critical position, the PDR is dropped to 50%. Also, other positions of the attacker lead to losing around 20% of the data packets. As such, the most harmful replay attacks on the PDR are DIO suppression and copycat attacks. However, the attack with the lowest impact on PDR is neighbor attack.

9.1.2 End-to-End (E2E) Delay

Figure 9.2 shows how each replay attack impacts E2E delay on the attacker's neighbors. The most impact in terms of delay is from neighbor attacks, particularly when an attacker is placed in a low-impact position and replays DIO messages of a node that is a neighbor to the deceived node. In this case, the delay reaches up to 50% higher than the no-attack scenario. We found the delay occurs when sending and receiving data packets belonging to the exploited node whose DIO message is used

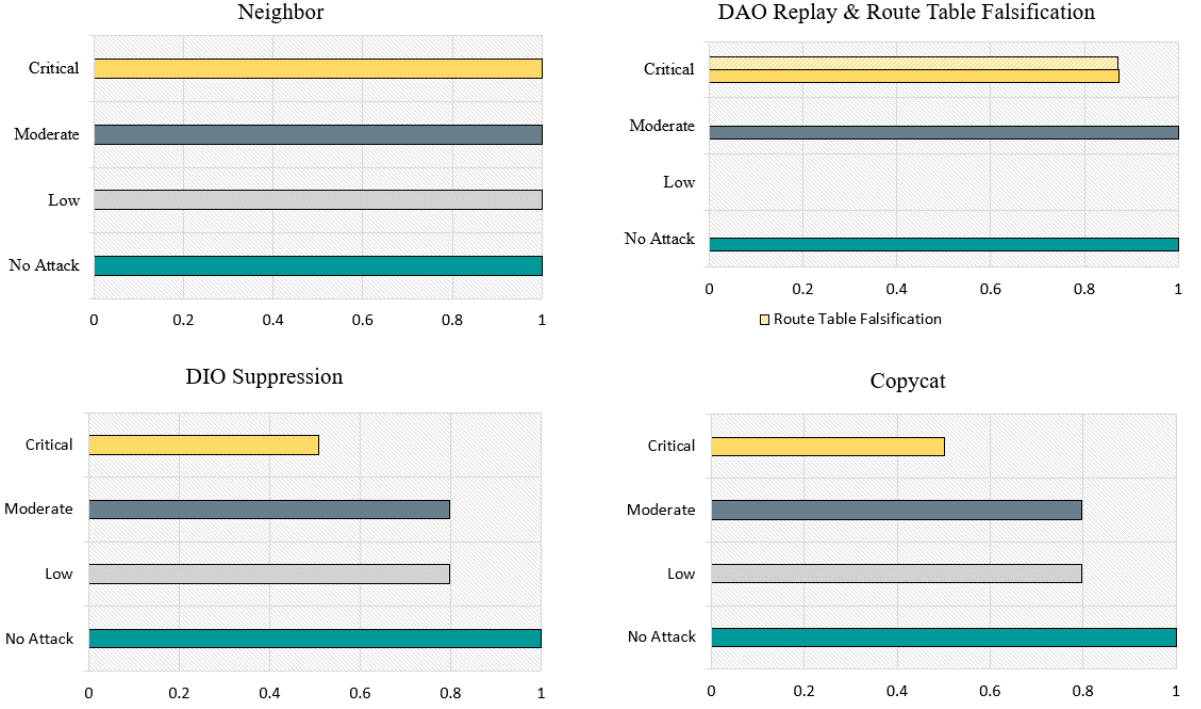


Figure 9.1: *Local impact of replay attacks on PDR*

to launch replay attacks. For DAO replay and route table falsification attacks, the delay increases by around 23% in the case of critical impact. Nevertheless, when the attacker performs a moderate-impact DAO replay attack, the latency is negligible. DIO suppression and copycat attacks have similar behavior in terms of end-to-end delay. The most affected case in the latency is the moderate impact scenario when the victim's child can join DODAG through another parent. In particular, the delay occurs at the child of the deceived node, where this node frequently cannot access its parent and needs to consider a sub-optimized routing path. It is worth mentioning that when the attacker node has a critical impact, the data packets that have been successfully delivered cause less delay than when a network is not under attack. The reason behind this behavior is that data packets of the exploited node and the de-

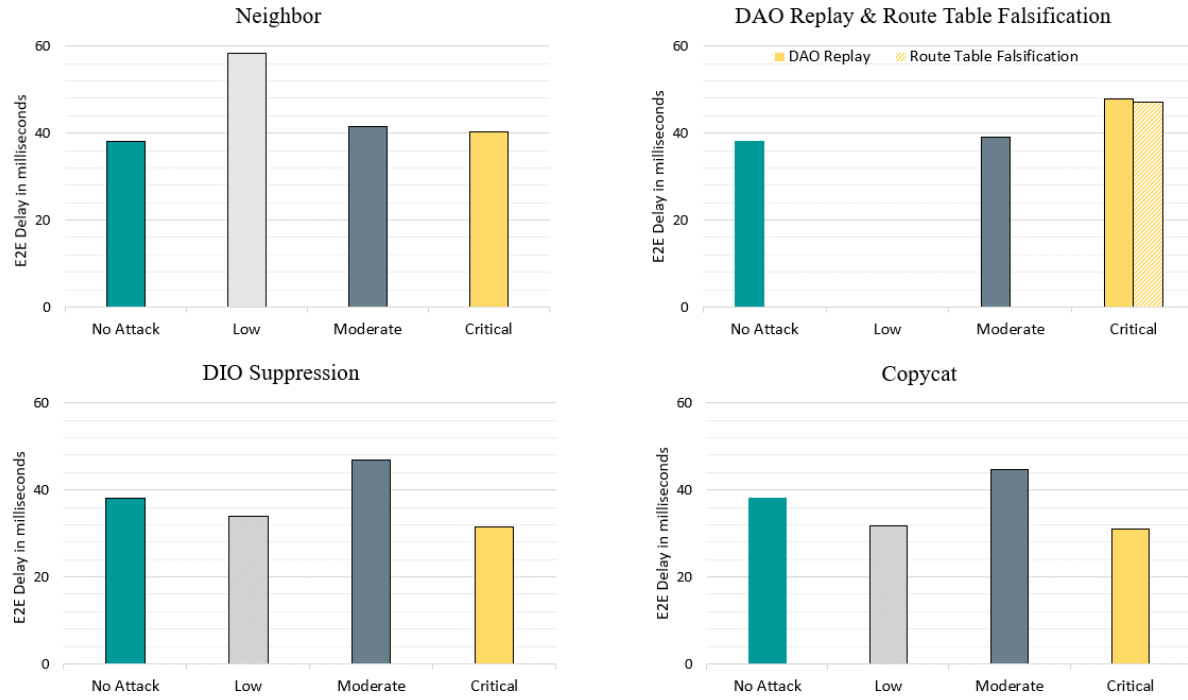


Figure 9.2: *Local impact of replay attacks on E2E delay*

ceived node and its child are not delivered when the attacker is launching the attack by frequently replaying DIO packets. Most of the delivered packets belong to nodes close to the root and out of the attacker's range. In summary, replay attacks that cause high delay are neighbor and route table falsification attacks, whereas the attacks with the lowest impact on latency are DIO suppression and copycat attacks.

9.1.3 Control Message Overhead

Figure 9.3 illustrates the average exchanged control messages. In general, attacks resulting in replaying DIO lead to more exchanges in RPL messages, as DIO messages are usually broadcast to all neighbors. Also, according to RPL, all receiver nodes must process each incoming DIO to track the changes in DODAG and transmit their

own DIO message. The attacks that cause the highest exchange of control messages are copycat and DIO suppression attacks, where the increase in control messages is between 500% and 800%. Specifically, the copycat attack, when an attacker has a low impact, shows the highest control overhead where the increase percentage reaches about 800%. This increase is due to the exploited node in this position having more neighbors than in other cases where the exploited node reacts to every received DIO packet by generating its DIO message according to the trickle timer specification that leads to the replay of more DIO messages by the attacker. Even though copycat and DIO suppression attacks have a similar setup, in the copycat attack, the attacker node adds its IP to the replayed messages, making receiver nodes consider these messages different.

Meanwhile, DAO replay and route table falsification attacks lead to a slight increase in the exchange of control messages. This result is justified by the nature of DAO messages, which are unicast only to the parent node. This finding also highlights the difficulty of detecting this type of replay attack.

9.1.4 Power Consumption

When the network is under replay attacks, the control messages overhead emphasize the wasted energy extinguished in exchanging futile control messages. Because the additional overhead consumes nodes' batteries, these attacks impact network availability. Figure 9.18 shows the impact of the replay attacks on average power consumption for neighbor nodes of the attacker. In general, replay attacks have a noticeable effect on the nodes' power due to the additional exchange of control messages. Figure 9.18

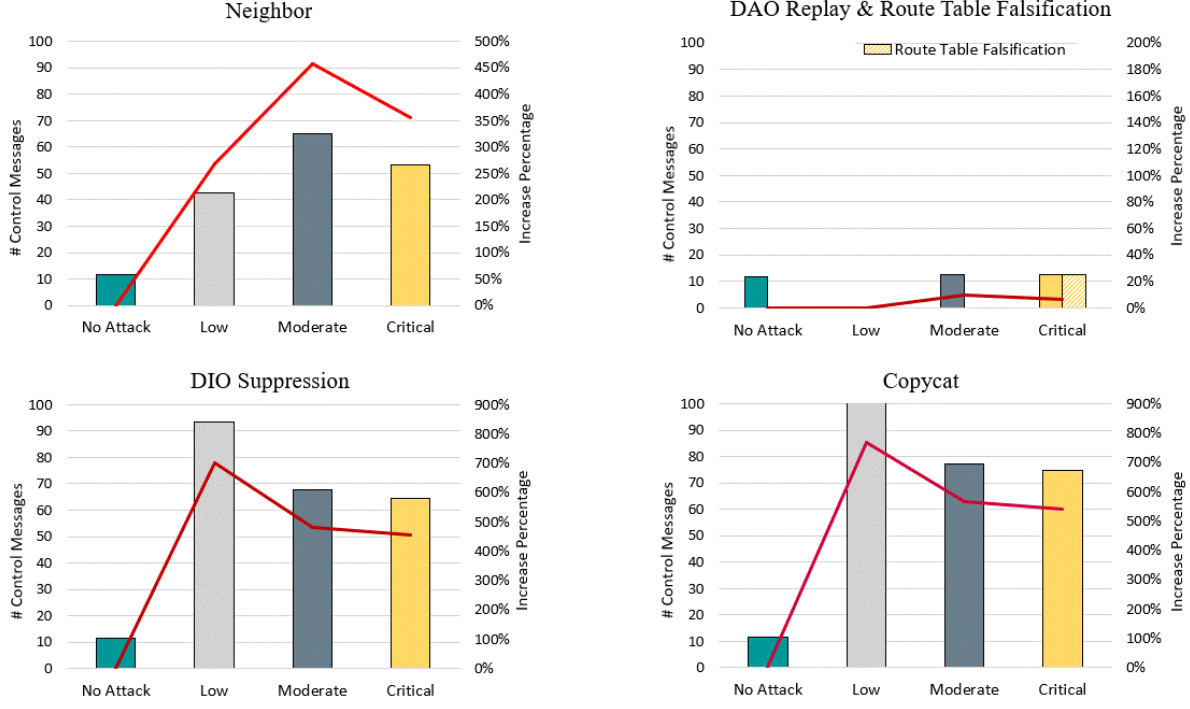


Figure 9.3: *Local impact of replay attacks on average control message overhead*

confirms that power consumption is correlated with messages overhead. Accordingly, DIO suppression and copycat attacks affect the average power consumption of the network the most. In comparison, DAO replay and route table falsification attacks affect it the least.

9.2 DEVS-RPL Validation Results

The RPL networks dataset consists of 57238 control messages. Table 9.1 illustrates the number of generated messages for each network layout and indicates the type of each control message.

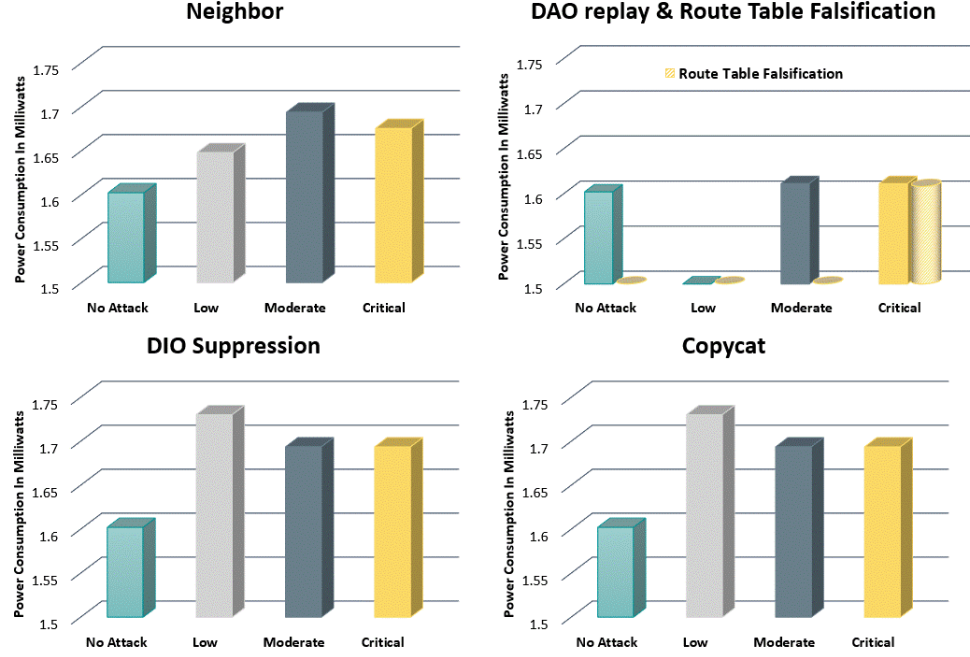


Figure 9.4: *Replay attacks impact on average power consumption*

Table 9.1: *Control messages distribution in RPL network*

# Messages	Random	Binary	Grid
DIS	34	34	34
DIO_U	14652	14679	14681
DIO_M	768	782	710
DAO	3496	4381	3119
Total	18818	19876	18544

9.2.1 Generated DIS

Regarding the DIS control message, based on Table 9.1, DIS appears to be the lowest generated and transmitted packets. Nodes only send DIS once when they decide to join the network. Figure 9.5, presents the sending time of network nodes in various configurations as well as the sending time in DEVS-RPL model. As seen in Figure 9.5, irrespective of the layout, nodes transmit DIS messages within the first minute of booting time, with an average delay of approximately 10 seconds. DEVS-RPL shows

a slightly higher delay within an average of 13 seconds. Consequently, we can conclude that in the RPL network, DIS messages are sent within the first minute of booting time and only once if the sensor has neighboring nodes within its communication range. For the DIS messages received, Figure 9.6a illustrates that the number of received messages closely resembles a Bernoulli distribution with a high p-value of 0.456, indicating that the observed differences between the dataset and the Bernoulli distribution are not statistically significant. Additionally, the KS test result in 0.131 suggests a small difference, indicating that the Bernoulli distribution is an appropriate model for the data.

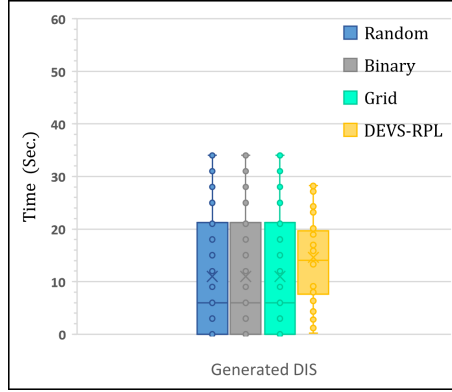
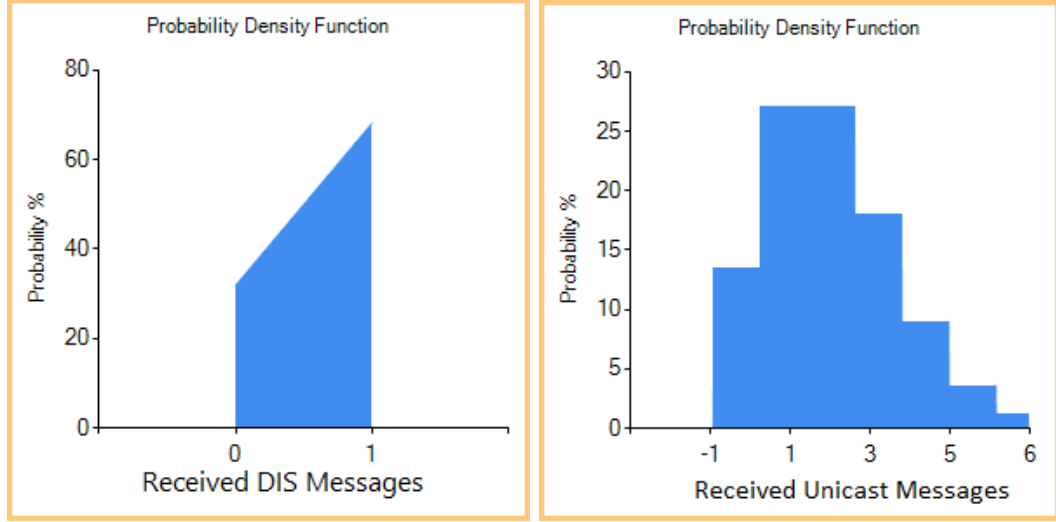


Figure 9.5: *Sending time of DIS messages*

9.2.2 Generated Unicast DIO

Moving to generated DIO_U messages, Figure 9.7 demonstrates the average unicast DIO messages each node sends throughout the simulation in front of the DEVS-RPL model. Although there was an initial discrepancy during the first minute of the experiment, with the DEVS-RPL model generating an average of 1.3 DIO_U message while



(a) $DIS \sim \text{Bernoulli}(p = 0.68, q = 0.32)$, (b) $DIO_U \sim \text{Poisson}(\lambda = 2)$, $K - S$ Test : 0.131, $P - \text{Value}$: 0.456 S Test : 0.677, $P - \text{Value}$: 0.647

Figure 9.6: *Distribution of received DIS and DIO_U*

the other network layouts generated 2.5-3 messages, all network layouts generated an average of 1.2 messages per minute for the rest of the simulation, consistent with the DEVS-RPL model. Based on these results, we can assert that the DEVS-RPL model precisely describes the generation of DIO_U messages. For the first minute, the additional DIO_U messages are related to discovering new neighbors, which leads to an additional DIO_U exchange to get the neighbor's ETX value [290].

For the unicast messages received, Figure 9.6b illustrates that the number of received messages closely resembles a Poisson distribution with a high p-value of 0.647, suggesting that the data following a Poisson distribution cannot be rejected. However, the KS test result 0.677 is insignificant, which infers some deviations from the expected fit in certain areas. Overall, though, this distribution serves as a reasonable approximation for the data.

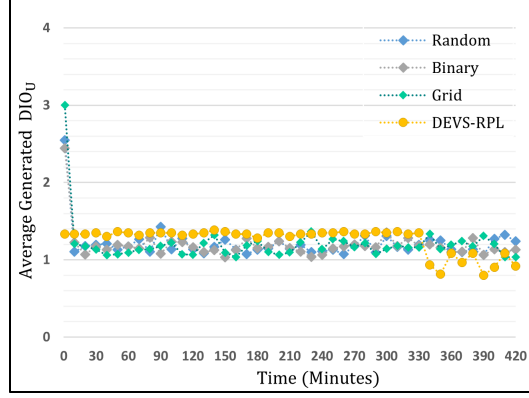


Figure 9.7: Average generated DIO_U during the simulation time

9.2.3 Generated Multicast DIO

To quantify the generated DIO_M , we need to identify the construction and maintenance stages. We choose an interval of 30 minutes for our empirical evaluation. In the construction stage, which is the first 30-minute interval, the average exchanged DIO_M is 10. In the maintenance stage, it is slightly less than 2 (around 1.87), which shows a clear distinction between the construction stage and the maintenance stage in terms of the number of the produced DIO_M in the network.

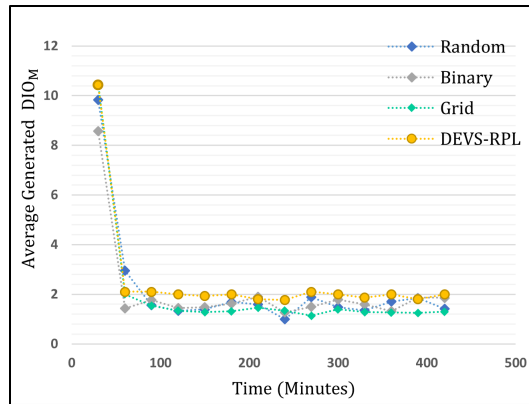
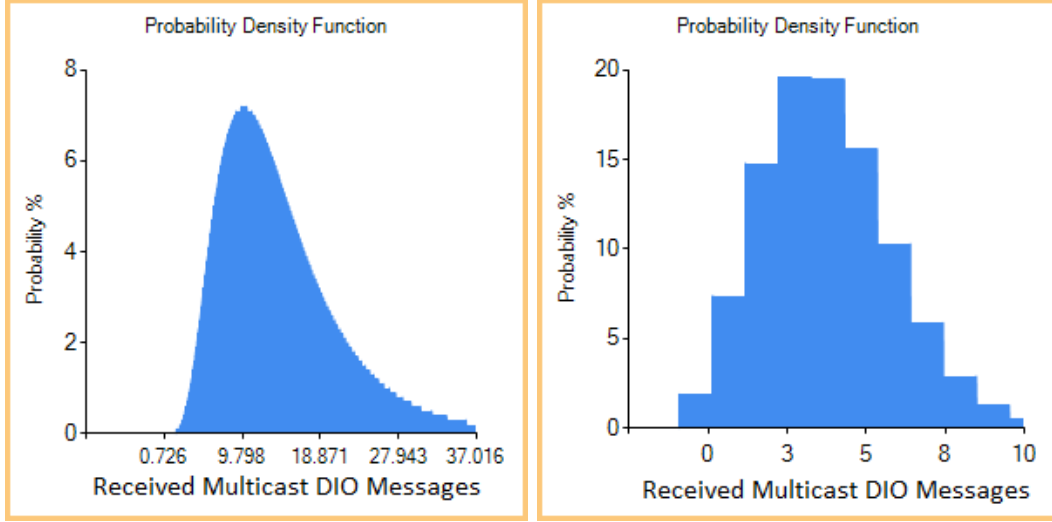


Figure 9.8: Average generated DIO_M during the simulation time

In Figure 9.8, during the construction stage, the density of the network affects the

number of exchanged DIO_M messages. The grid layout generates the highest average of 10.5 messages per node, while the binary layout generates the minimum with an average of 8.5 messages per node. The average number of DIO_M messages generated by DEVS-RPL is 10.4, which aligns with the generation rate in a random layout. During the maintenance stage, DEVS-RPL generates an average of 2 DIO_Ms in each interval. For the random and binary layouts, generation rates are between 1.2 and 1.9, while the grid layout varies between 1.3 and 1.5 DIO_M messages per node. The suppression feature is overlooked in this model, causing DEVS-RPL to estimate a slightly higher generation rate than actual values, mostly noticeable in the grid layout. We also analyzed the received DIO_M messages. Figure 9.9a indicates that the messages are well-represented by a Log-normal distribution during the construction period. This is supported visually by the histogram, which shows a skewed peak and a long right tail. Furthermore, the K-S test statistic is 0.07, which is quite low, while the p-value is 0.906. These results suggest a strong match between the received DIO_M messages and the theoretical Log-Normal distribution. Conversely, Figure 9.9b illustrates that the histogram resembles a Poisson distribution. It displays a peak near the mean (approximately 3-4 messages) and a tapering tail. The KS statistic for the Poisson distribution is 0.142, indicating a moderate deviation between the received messages and the theoretical cumulative distribution functions. Although the p-value is 0.154, which is above 0.05, suggesting that the data could fit a Poisson distribution, the fit is not particularly strong due to some asymmetry or variation observed in the data compared to the theoretical Poisson distribution.



(a) DIO_M in construction phase $\sim \text{Log-normal}$ ($\mu = 2.54$, $\sigma^2 = 0.49$), $K - \text{Poisson}(\lambda = 3.983)$, $K - S \text{ Test} : 0.142$, $S \text{ Test} : 0.07$, $P - \text{Value} : 0.906$
(b) DIO_M in maintenance phase $\sim \text{Poisson}$ ($\lambda = 3.983$), $K - S \text{ Test} : 0.142$, $P \text{ Value} : 0.154$

Figure 9.9: Distribution of received DIO_M messages

9.2.4 Generated DAO

In Figure 9.10, the DAO messages generated over the simulation time are shown. During the construction stage, all network layouts produced around 22 DAO messages per node, with DEVS-RPL generating an equal number. In the maintenance stage, the grid layout had the lowest generation rate, while the binary layout had the highest with 5 and 7 DAO messages per node, respectively. To assess the received DAO messages, Figure 9.11a illustrates the distribution of incoming DAOs during the construction phase. The chart and the associated statistics indicate that the dataset likely follows a Log-normal distribution, as evidenced by its skewness and distinctive tail characteristics. The KS test statistic is 0.105, and the p-value is 0.583, both suggesting a reasonable fit for the data to a Log-normal distribution. In contrast, Figure 9.11b depicts the distribution of received DAOs during the maintenance phase. The

histogram reveals a bell-shaped distribution, which aligns with the expected shape of a Poisson distribution at higher rates. Furthermore, the symmetry and spread of the data support a close approximation to the theoretical Poisson curve. The KS test statistic of 0.583 and a p-value of 0.834 confirm that the data fits well with a Poisson distribution.

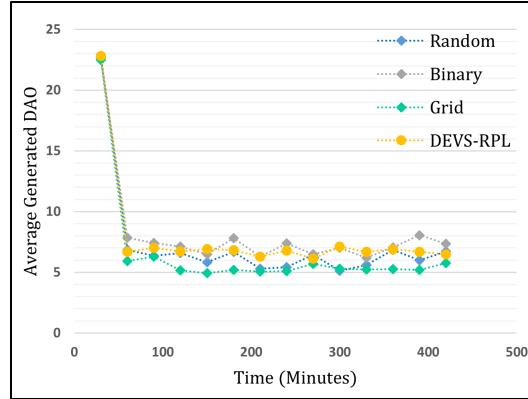
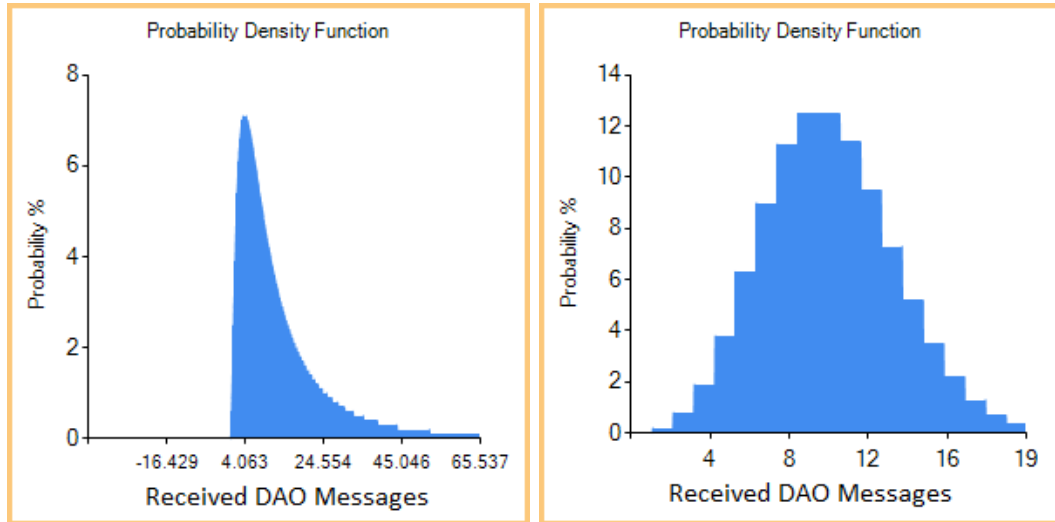


Figure 9.10: Average generated DAO during the simulation time



(a) DAO in construction phase $\sim$ Log - normal ($\mu = 2.23$, $\sigma^2 = 0.94$), $K - S$ Test : 0.583, S Test : 0.105, $P - Value$: 0.583
(b) DAO in maintenance phase $\sim$ Poisson($\lambda = 10$), $K - S$ Test : 0.583, S Test : 0.105, $P - Value$: 0.834

Figure 9.11: Distribution of received DAO messages

Table 9.2: *Statistical metrics for error and deviation analysis*

	DIS	DIO_U	DIO_M	DAO
S.D.	0	0.2	2.184	4.158
MAE	0	0.221	0.455	0.649
MAPE (%)	0.00	4.09	8.21	9.15
MAD	0	0.0167	0.1	0.165
TMAE($\alpha = 10\%$)	0	0.212	0.412	0.513
RMdSE	0	0.292	0.499	0.797

9.2.5 Error Evaluation

To assess the precision of DEVS-RPL, we calculate the percentage error. The error evaluation is based on the difference between the messages generated in each layout and those generated according to the DEVS-RPL model. More precisely, in each 30-minute interval, we calculate the absolute difference between the number of generated messages. Multiple statistical metrics for error and deviation analysis have been calculated: The standard deviation (S.D.), the mean absolute error (MAE), the mean absolute percentage error (MAPE), Median Absolute Deviation (MAD), Trimmed Mean Absolute Error (TMAE), and Root Median Squared Error (RMdSE). Table 9.2 shows the calculated values.

In Table 9.2, both MAE and MAPE values have been presented to quantify the average deviation between the generated messages in the Cooja simulator and the values generated by DEV-RPL. Notably, all generated messages recorded less than 10% MAPE, which assertively indicates that DEVS-RPL remained less than 10% away from the actual values on average over the whole time period. The DIS message is completely error-free, as it is generated only once. The DIO_U , which relies on a single timer $I_{Probing}$ with its randomness factor, has the lowest error value with a MAPE 4%

error percentage. On the other hand, the DAO message, which relies on several events such as lifetime updates, DIO_M^U , and received DAO, recorded a 9% error percentage, making it more susceptible to errors. The median absolute deviation(MAD) is a robust measure of how spread out a set of data is. The variance and standard deviation are also measures of spread, but they are more affected by extremely high or extremely low values and non-normality. Table 9.2 shows the standard Deviation and MAD for each RPL message. The table indicates that MAD values are lower than their S.D., which aligns with the expectation that MAD is typically less influenced by outliers. In the case of the DIS message, DSVS-RPL demonstrates no variability, implying that all values are constant. In addition, DEVS-RPL data shows reduced variability and small deviations for DIO_U messages comparable to DIO_M messages. Regarding DAO messages, its MAD is considerably smaller, suggesting a higher occurrence of outliers within the DEVS-RPL data. TMAE is a variation of MAE that trims a certain percentage of extreme values (outliers) before computing the mean. This provides a more robust measure of error by reducing the impact of outliers. With respect to TMAE at α set to 10%, Table 9.2 demonstrates that DIS and DIO_U messages display the smallest errors and the highest prediction accuracy. In contrast, DIO_M and DAO messages are less accurate, with DAO being the least reliable in terms of prediction performance. Furthermore, Table 9.2 shows that the trimming at $\alpha = 10\%$ effectively removes extreme outliers, allowing for a greater focus on significant core errors. The range of TMAE underscores the variability in prediction accuracy across different features. In terms of RMdSE, which measures the average difference between a statistical model's predicted values and the actual observed values. It quantifies

the dispersion of residuals, illustrating how closely the observed data aligns with the predicted values. According to Table 9.2, an RMdSE of 0 signifies no error for DIS messages, which aligns with the model’s assumption that DIS generation in the first minute is both constant and accurately predicted. For DIO_U messages in DEVS-RPL, the RMdSE reflects a low prediction error and correlates well with the findings from Contiki-NG. In contrast, the RMdSE for DIO_M messages indicates a moderate prediction error. While DIO_M does not perform as effectively as DIO_U , it still makes a meaningful contribution to the model, albeit with reduced precision. The RMdSE is highest for DAO messages, suggesting that they contribute the least to the model’s overall accuracy.

9.3 DEVS-RPL IDS Performance Results

To evaluate DEVS-RPL IDS performance, Figure 9.12 shows EWMA $\bar{X}$ chart for RPL messages and Figure 9.13 shows EWMA standard deviation charts over the simulation time. The illustrated control charts of nodes are taken for normal and under various RPL attack situations. In normal conditions where no attack was introduced, the EWMA $\bar{X}$ scores correctly remain within the control limits. Additionally, The EWMA standard deviation being near zero as Figure 9.13 reveals, which reflects a stable, predictable process with no significant shifts or anomalies. In contrast, under RPL attacks, nodes reveal a noticeable increase in $\bar{X}$. The most significant increase in $\bar{X}$ is attributable to the DIS flooding attack, during which nodes are inundated with a substantial influx of DIS messages. This inundation results in the generation

and exchange of an excessive number of DIO messages. In contrast, DAO replay attack has the least attack impact on $\bar{X}$ as the nature of the attack depends on RPL message, which is limited in less frequency than DIO message. In addition, Figure 9.13 reveals that blackhole and flooding attacks exhibit the highest variability, indicating severe and persistent process instability. At the same time, other attacks like RTF has moderate variability, showing ongoing but less extreme fluctuations. It is also noticeable that suppression attack shows smaller but noticeable spikes, indicating periodic disturbances.

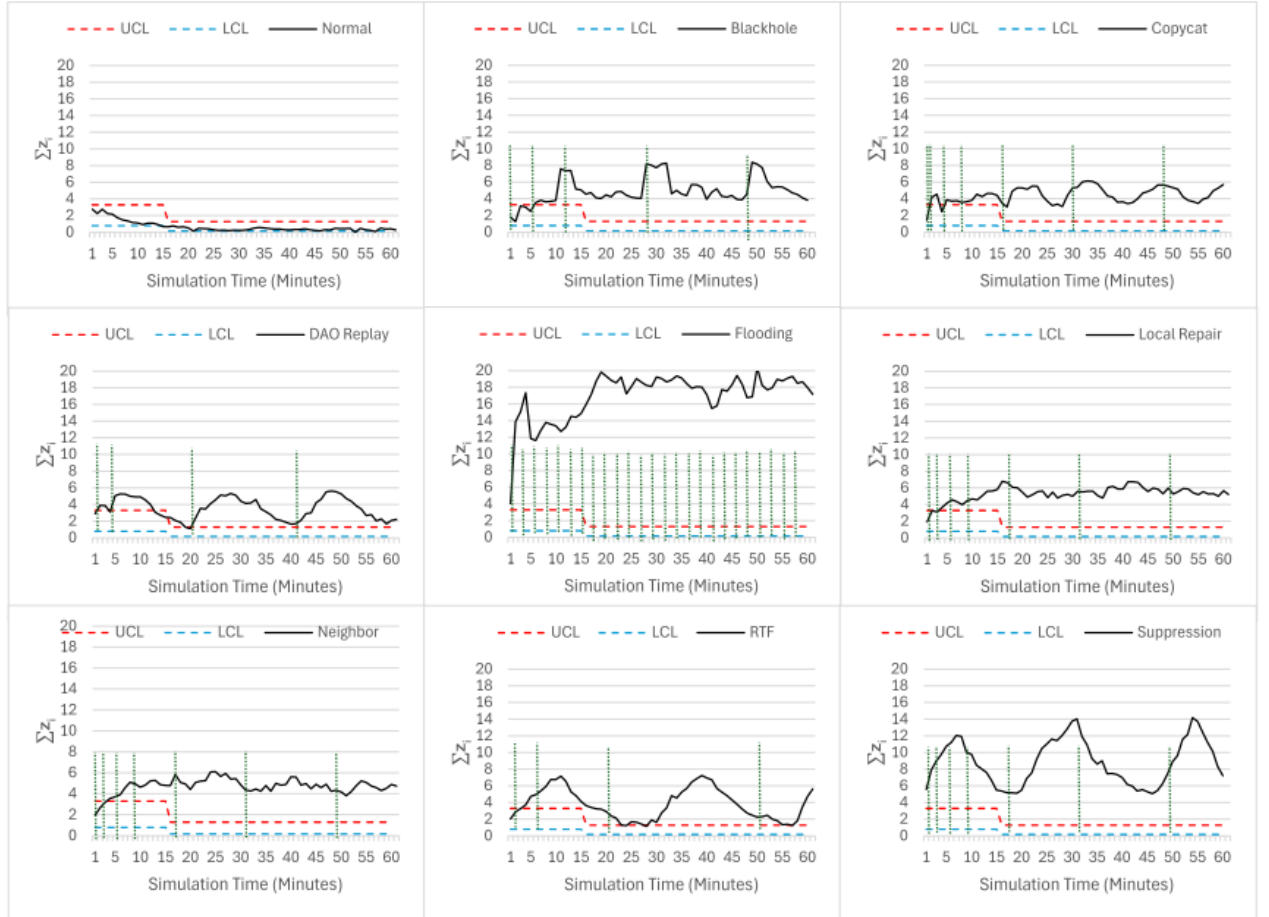


Figure 9.12: *EWMA control charts ($\bar{X}$ chart), with vertical lines indicating the moments of attacks*

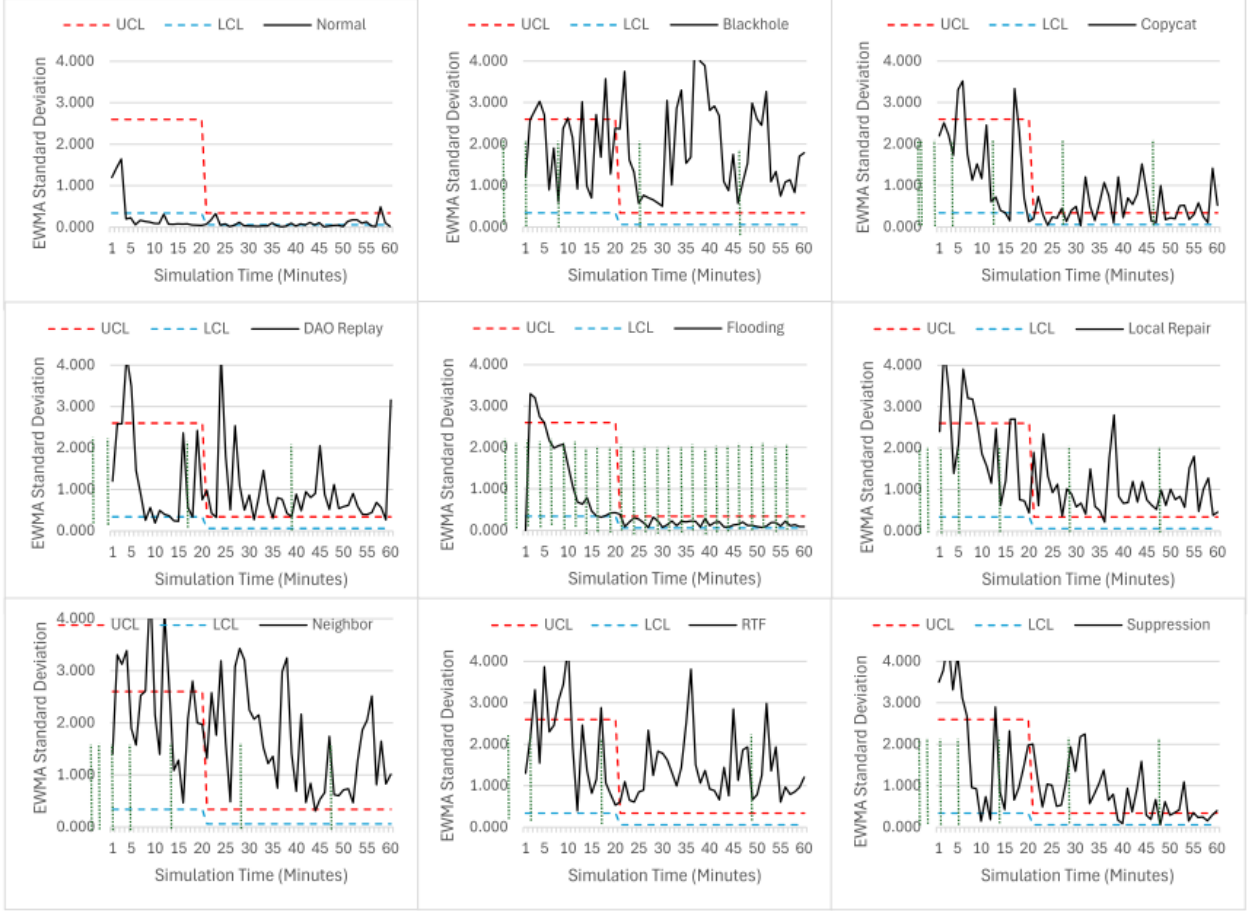


Figure 9.13: *EWMA control charts (S chart) with vertical lines indicating the moments of attacks*

9.3.1 Detection Efficiency

To evaluate the accuracy, sensitivity, and false alarm rate, we use the confusion matrix to measure the performance of the detection method in terms of the accuracy in detecting malicious and normal nodes and producing a low false alarm rate [17]. The confusion matrix is an outline of the prediction output in the detection method where the number of correct and false predictions is calculated based on malicious and normal

behavior. The confusion matrix used is illustrated in Table 9.3.

Table 9.3: *Confusion matrix to evaluate DEVS-RPL IDS*

Predicted Behavior	Actual Behavior	
	Malicious (P=8340)	Normal (N=4320)
Malicious (P=7990)	TP =7776	FP=214
Normal (N=4670)	FN=360	TN=4310

The presented confusion matrix in Table 9.3 is analyzed to calculate several performance metrics commonly used in classification problems, including accuracy, precision, recall, F1-score, and more. Table 9.4 shows the performance of the proposed DEVS-RPL IDS according to the classification evaluation metrics. The demonstrated scores indicate that the detection system operates effectively, achieving an accuracy of 96.23% in identifying both malicious and normal behaviors. Precision shows that most predictions of malicious behavior are accurate, while recall reflects the system’s ability to identify the majority of such behaviors. Specificity highlights its strong performance in classifying normal behavior, and the F1 Score balances precision and recall to affirm overall reliability. However, MCC although lower, offers a comprehensive view of performance by addressing classification imbalances. Overall, this analysis highlights the system’s strengths and areas for improvement. The high accu-

Table 9.4: *Detection evaluation metrics*

Metric	Value
Accuracy	96.23%
F1 Score	96.21%
MCC	27.64%
Precision	97.21%
Recall (Sensitivity)	95.62%
Specificity	95.27%

racy suggests a robust detection system for most cases, while the slightly lower MCC indicates opportunities to better balance predictions and minimize false positives and negatives.

The graph depicted in Figure 9.14 illustrates the average accuracy, FPR, and TPR overtime during the simulation. During the first three minutes, both the accuracy and TPR were relatively low, around 65%, due to the detection system’s struggle to identify attacks during the construction of the DODAG. Meanwhile, the FPR was high, at around 45%. However, after this initial period, the accuracy and TPR of the DEVS-RPL IDS began to improve, reaching 93% accuracy at minute 8. Subsequently, the accuracy and TPR of the detector reached nearly 100%, while the FPR dropped to 0%.

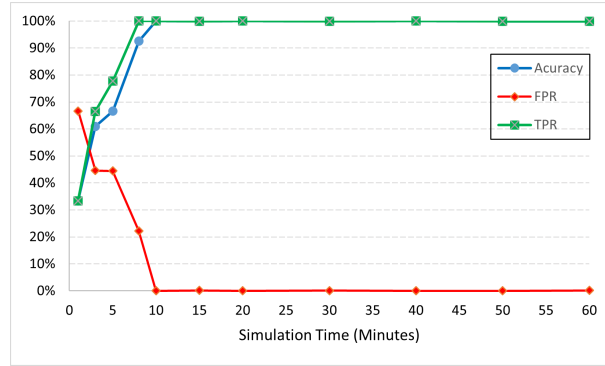


Figure 9.14: *Average accuracy, FPR, and TPR over simulation time*

Figure 9.15 presents a comparison of the detection accuracy of DEVS-RPL IDS with other specification-based detectors, namely EMBOF-RPL [247], Altered SVELTE [187], LEADER [173], and RAIDER [74]. EMBOF-RPL exhibited the highest accuracy in attack detection, averaging 98%. DEVS-RPL IDS achieved a detection accuracy of around 95%. It is pertinent to note that while EMBOF-RPL was tested

against three attacks, DEVS-RPL IDS underwent testing against eight distinct attacks.

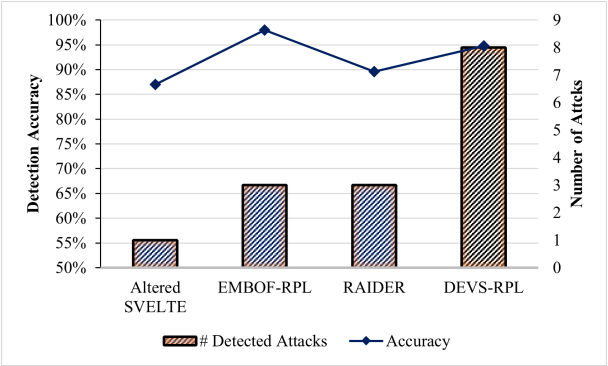


Figure 9.15: Comparisons of detection accuracy between DEVS-RPL IDS and specification-based detectors

In addition, Figure 9.16 shows TPR and FPR between DEVS-RPL IDS and specification-based detectors Ioulianou’s IDS [150], EFSM IDS [19], and Altered SVELTE [187]. According to the illustrated percentages, DEVS-RPL IDS outperforms the other solutions in terms of achieving the highest TPR with around 96% and the lowest FPR with a rate of 5%.

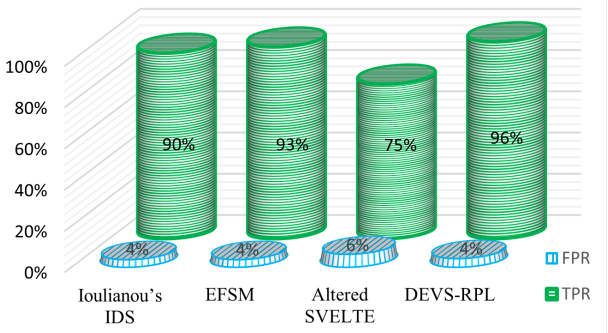


Figure 9.16: Comparisons of TPR and FPR between DEVS-RPL IDS and specification-based detectors

9.3.2 Detection Delay

Furthermore, Figure 9.17 illustrates the time taken by the DEVS-RPL IDS to detect various attacks. Instant detection occurred for DIS flooding and DIO suppression attacks. The DIS flooding attack involved the frequent sending of DIS messages, a clear violation of the RPL specification, making it easily recognizable. The DIO suppression attack, which replays a DIO without including the sender node's ID within a short interval, was also promptly identified by the IDS. Subsequently, DAO replay and copycat attacks were detected after one minute. These attacks violated RPL specifications regarding message destination and frequency while maintaining the true sender node ID. Local repair, neighbor, and RTF attacks were identified approximately four minutes after initiation. Local repair involves modifying packets to trigger the local repair process, a common occurrence in lossy networks, but frequent triggers signaled intrusion or node dysfunction to the IDS. Neighbor and RTF attacks, replay attacks with source forging, took longer for the IDS to identify. Finally, the detection of blackhole attacks took around six minutes due to the nature of the lossy network, which occasionally experiences packet loss.

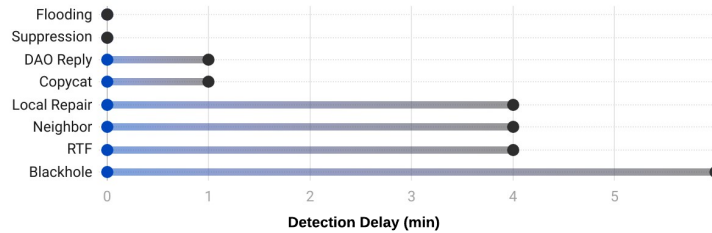


Figure 9.17: *The detection latency of DEVS-RPL IDS in detecting attacks*

9.3.3 Power Consumption

Due to the limited resources in IoT for network nodes, the efficiency of the proposed solution, in terms of network lifetime, must be evaluated using the average power consumption. Figure 9.18 shows the average power consumption after implementing DEVS-RPL IDS and other specification-based IDS, which are EFSM [19], SVELTE [195], and EMBOF-RPL [247]. Upon adding the detection solution, EMBOF-RPL added the lowest amount of power consumption around 2%. DEVS-RPL IDS added an approximately 3% increase in power consumption (0.04 Milliwatts), making it a resource-friendly solution. It is worth mentioning that the variance in power consumption in naive RPL implementation is attributed to using different sensor types. Tmote sky nodes were utilized in the other studies, while this study employs Zolertia Z1 sensors.

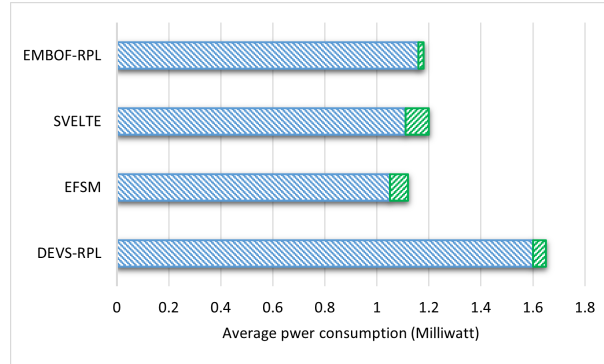


Figure 9.18: Comparisons of average power consumption between DEVS-RPL IDS and specification-based detectors

9.3.4 Memory Consumption

The RAM and ROM requirements of the proposed detection and the ordinary RPL are shown in Table 9.5, which reveals a small addition in the memory consumption around (1%). The increase in ROM consumption (148 bytes) is due to the added code that counts and evaluates the limits of the RPL message. Additionally, the RAM overhead (68 bytes) is caused by the added variables that count each RPL message, besides the variables that save the limits and the weights.

Table 9.5: *Memory consumption*

Image File	Total ROM (Bytes)	Total RAM (Bytes)
Contiki RPL (udp-client)	47485	6458
Contiki RPL+ IDS (udp-client)	47633 (+148 ↑)	6526 (+68 ↑)

As a result, adopting a fully distributed approach of this IDS requires a specification on the nodes to be at least C1 or better where data size (e.g., RAM) is around 10 KB and code size (e.g., Flash) is around 100 KB as shown in Table 9.6.

Table 9.6: *Classes of constrained devices [1]*

Name	data size (e.g., RAM)	code size (e.g., Flash)	Examples
Class 0, C0	« 10 KiB	« 100 KiB	ATtiny
Class 1, C1	10 KiB	100 KiB	STM32F103CB
Class 2, C2	50 KiB	250 KiB	STM32F103RC
Class 3, C3	100 KiB	500..1000 KiB	STM32F103RG
Class 4, C4	300..1000 KiB	1000..2000 KiB	"Luxury"

CHAPTER 10

CONCLUSIONS AND FUTURE DIRECTIONS

10.1 Overview

In this study, we have created an intrusion detection system to identify replay attacks of RPL control messages on IoT devices. To achieve this goal, we have conducted a systematic literature review on RPL routing attacks and mitigation solutions. We thoroughly examined RPL attacks and classified them based on their attack vector. We also established the first taxonomy that categorizes all proposed RPL mitigation solutions based on their techniques. The primary objective of the proposed taxonomies is to categorize attacks and their mitigation solutions based on commonalities. Both taxonomies comprise general patterns that abstractly describe the methodology or technique employed. Following, we investigated the problem of RPL replay attacks in IoT environments where attackers record and send control messages to disrupt the routing topology. The evaluation reveals multiple RPL replay attacks based on the

type of replayed control packet, the modification performed before replaying, and the frequency of replaying the packet. Specifically, we study and analyze five forms of replay attacks. Three types have been introduced based on replaying DIO messages, namely, neighbor, DIO suppression, and copycat attacks. We also analyze the impact on the DODAG topology under these attacks, whether by constructing sub-optimized routing paths or isolating some nodes from the DODAG. In addition, we introduce the first study of two replay attacks launched by DAO messages. The first is the DAO replay attack, and the second is the route table falsification attack. We investigated the DODAG topology under these attacks in terms of blocking the downward traffic and dropping packet delivery. Furthermore, we highlight the existing countermeasures to detect RPL replay attacks, including threshold, trust, location, and machine learning. To design our specification-based IDS, we first developed a DEVS model called DEVS-RPL to describe the generation of RPL messages in nodes according to RPL. Specifically, DEVS-RPL estimates the mean and frequency of each generated RPL message by defining the participant states and utilizing the time advance function. As a proof of concept for DEVS-RPL, we conducted extensive simulations using various RPL network topologies, including random, binary, and grid layouts, to provide comprehensive network behavior evaluation. The results show that the simulated networks produce RPL messages with the same generation rate as the DEVS-RPL model with $\text{MAPE} \leq 9$. Consequently, we propose DEVS-RP IDS, a specification-based IDS that utilizes RPL specifications and short-term observations of exchanged control messages to detect intrusions and dysfunction in RPL networks. The DEVE-RPL IDS, built on an SPC, to monitor the received RPL messages and assess the number

of received messages according to the defined upper and lower limits. The EWMA was specifically used to monitor the network and detect outliers and anomalies in node behavior. We evaluate the performance of the proposed solution to detect replay attacks: neighbor, suppression, DAO replay, copycat, and route table falsification attacks. Besides, we test the detection of DEVS-RPL IDS on other types of attacks, including blackhole, DIS flooding, and local repair attacks. Our evaluation experiment shows that the proposed solution successfully detects these attacks with accuracy reaching 95% with only a 4% false alarm rate. Also, DEVS-RPL IDS added a limited increase, around 3% in power and 1% in memory consumption, making it efficient to deploy on constrained nodes.

10.2 Limitations

While the proposed DEVS-RPL IDS demonstrates high accuracy, low false alarm rates, and efficient resource usage, it does have several limitations that must be addressed for broader applicability and robustness.

- This solution is limited to certain RPL Attacks. The proposed IDS is primarily designed to detect attacks that result in deviations in the amount of exchanged RPL packets. Experimental results indicate that the tested attacks significantly affect the number of RPL messages exchanged, allowing for successful detection. However, certain RPL attacks, such as selective forwarding and increased/decreased rank attacks, can disrupt routing services without changing the volume of RPL packets. According to the literature, these attacks are more subtle and

do not trigger noticeable anomalies in control packet behavior. As a result, the current detection system cannot effectively identify such attacks.

- The proposed DEVS-RPL IDS is dependent on Contiki OS Implementation. This solution is specifically developed based on the RPL protocol specification and its implementation within the Contiki OS environment. While Contiki OS is widely used for RPL-based IoT networks, other operating systems, such as RIOT OS, TinyOS, and Zephyr OS, also support RPL. This study did not consider variations or differences in RPL implementation across these platforms. Additionally, other versions and variations of RPL, such as RPL Lite, are not included in the current model. This reliance limits the portability and generalizability of the detection system to other platforms.
- Scalability limitations hinder the assessment of DEVS-RPL IDS on a larger scale. The evaluation of this detection system was conducted using Z1 sensor motes, which are limited by their hardware specifications. By default, Z1 sensors support a maximum of 8 routing table entries due to their constrained RAM (8 KB). This limitation directly impacts the ability to assess the system's scalability in larger network setups with more neighbors. To accommodate a higher number of neighbors, different sensor hardware with larger routing table capacities would be necessary, along with a redesigned experimental setup. This remains an area for future exploration.

To sum up, the limitations of this detection system relate to its scope of attack detection, dependence on a specific operating system and protocol version, and hardware

constraints affecting scalability. Addressing these limitations through further research and development will enhance the system’s robustness, adaptability, and scalability.

10.3 Concluding Remarks

- The current research status of RPL attacks and the mitigation mechanisms shows that the most studied attack is decreased rank attacks and then blackhole attacks. Conversely, there has been minimal exploration of DAO insider, DAO inconsistency, and DDAO attacks.
- It is intriguing to note that modifying RPL remains the most used technique for thwarting attacks, making up approximately 33% of the applied techniques. Specifically, around 16% of the research studies have concentrated on adjusting the objective function based on node behaviors. The second used approach to mitigate RPL attacks is isolating the attacker node, which has been utilized in 27% of the reviewed studies. On the other hand, authentication, encryption, and network updates have only been implemented in a small percentage of studies.
- We conducted a thorough analysis to determine the efficiency of mitigation solutions and help select the appropriate one to ensure network security. To evaluate each solution, we developed criteria, including network node utilization, resource preservation, modification level, and comprehensiveness. We then analyzed each attack individually to determine the most efficient mitigation solution.
- The experiments that measure network performance under replay attacks reveal

that these attacks can severely impact a network. Specifically, DIO suppression and copycat attacks have the greatest potential effect on a network owing to their impact on data delivery (PDR has dropped to 50%) and the network resources (the increase in control messages is between 500% and 800%). In addition, we found that DAO replay and route table falsification attacks isolate the downward traffic toward the exploited node. Our results also confirm that neighbor attacks lead to considerable delays that reach up to 50% higher than the no-attack scenario.

- Developing DEVS-RPL to describe the generation of RPL messages by each node establishes a reliable baseline for identifying the normal behavior by specifying the mean and frequency of each generated RPL message. In addition, we determine the upper control and lower control limits of each received RPL message using SPC to introduce a lightweight detection system that detects intrusions and anomalies in RPL using the exponentially weighted moving average (EWMA) procedure.
- Introducing a lightweight solution with the added overhead of 3% in power and 1% in memory consumption enhances the feasibility of using this system online without the need for pre-training or historical data to detect anomalies.

10.4 Directions for Future Work

Future directions of this work include the following:

- Implementing a signature-based IDS based on the proposed RPL taxonomy to detect and classify network traffic patterns based on attack signatures. According to the presented taxonomy, the detector can identify the attack according to which part of a packet of ICMPv6 network packets is misused.
- Assessing the effectiveness of the DEVS-RPL IDS in detecting additional RPL attacks. This evaluation will involve measuring its capability to identify and report other types of attacks.
- Designing an updating-based mitigation solution to mitigate replay attacks as this type of solution has been found to be lightweight and resource-friendly. There is great potential in exploring updating-based solutions to counter replay and other types of attacks, specifically those that involve modifying packets, like version, increased rank, and DAO inconsistency attacks.

REFERENCES

- [1] C. Bormann, M. Ersue, and A. Keranen, “Rfc 7228: Terminology for constrained-node networks,” 2014.
- [2] A. Mayzaud, R. Badonnel, and I. Chrisment, “A Taxonomy of Attacks in RPL-based Internet of Things,” *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016. [Online]. Available: <https://hal.inria.fr/hal-01207859>
- [3] A. Raoof, A. Matrawy, and C. H. Lung, “Routing Attacks and Mitigation Methods for RPL-Based Internet of Things,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1582–1606, 2019.
- [4] A. O. Bang, U. P. Rao, P. Kaliyar, and M. Conti, “Assessment of Routing Attacks and Mitigation Techniques with RPL Control Messages: A Survey,” *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–36, Mar. 2023. [Online]. Available: <https://dl.acm.org/doi/10.1145/3494524>
- [5] A. Verma and V. Ranga, “Security of RPL Based 6LoWPAN Networks in the Internet of Things: A Review,” *IEEE Sensors Journal*, vol. 20, no. 11, pp. 5666–5690, 2020.

- [6] S. O’Dea, “Wide-area and short-range IoT device installed base worldwide 2014-2027,” <https://www.statista.com/statistics/1016276/wide-area-and-short-range-iot-device-installed-base-worldwide/>, 2022.
- [7] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” *Future Generation Computer Systems*, vol. 82, pp. 395–411, May 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167739X17315765>
- [8] J. Tripathi, J. de Oliveira, and JP. Vasseur, “Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC Editor, Tech. Rep. RFC6687, Oct. 2012. [Online]. Available: <https://www.rfc-editor.org/info/rfc6687>
- [9] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, “Cyber Threats to Industrial IoT: A Survey on Attacks and Countermeasures,” *IoT*, vol. 2, pp. 163–186, 3 2021.
- [10] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, “A Comprehensive Review on Secure Routing in Internet of Things: Mitigation Methods and Trust-based Approaches,” *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–1, 2020.
- [11] M. E. Whitman and H. J. Mattord, *Principles of information security*. Cengage learning, 2011.

- [12] M. Ciampa, *Security+, Guide to Network Security Fundamentals Second Edition Update*, 2008.
- [13] F. Azzedin and H. Albinali, “Security in internet of things: Rpl attacks taxonomy,” in *Proceedings of the 5th International Conference on Future Networks and Distributed Systems*, 2021, pp. 820–825.
- [14] I. Butun, P. Österberg, and H. Song, “Security of the internet of things: Vulnerabilities, attacks and countermeasures,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 10 2019. [Online]. Available: <http://arxiv.org/abs/1910.13312><http://dx.doi.org/10.1109/COMST.2019.2953364>
- [15] A. Mayzaud, “Monitoring and Security for the RPL-based Internet of Things,” Ph.D. dissertation, Université de Lorraine, 2017.
- [16] A. Seyfollahi and A. Ghaffari, “A Review of Intrusion Detection Systems in RPL Routing Protocol Based on Machine Learning for Internet of Things Applications,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–32, Aug. 2021.
- [17] “Intrusion Detection Systems in RPL-Based 6LoWPAN: A Systematic Literature Review,” *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12 940–12 968, 2021.
- [18] G. Simoglou, G. Violettas, S. Petridou, and L. Mamatas, “Intrusion detection systems for rpl security: A comparative analysis,” *Computers and Security*, vol. 104, 2021.

- [19] 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.
- [20] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, “Intrusion detection systems in the Internet of things: A comprehensive investigation,” *Computer Networks*, vol. 160, pp. 165–191, 2019. [Online]. Available: <https://doi.org/10.1016/j.comnet.2019.05.014>
- [21] B. P. Zeigler and H. S. Sarjoughian, “Introduction to devs modeling and simulation with java: Developing component-based simulation models,” *Technical Document, University of Arizona*, 2003.
- [22] N. Akhtar, M. Niazi, F. Mustafa, and A. Hussain, “A discrete event system specification (DEVS)-based model of consanguinity,” *Journal of Theoretical Biology*, vol. 285, no. 1, pp. 103–112, Sep. 2011. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0022519311003031>
- [23] T. A. Al-Amiedy, M. Anbar, B. Belaton, A. A. Bahashwan, I. H. Hasbullah, M. A. Aladaileh, and G. A. Mukhaini, “A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things,” *Internet of Things*, vol. 22, p. 100741, Jul. 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2542660523000641>
- [24] D. Sharma Sr Assistant Professor, I. Mishra Sr Assistant Professor, and S. Jain, “Impact factor: 4.295 A Detailed Classification of Routing Attacks against

- RPL in Internet of Things,” *International Journal of Advance Research*, vol. 3, pp. 692–703, 2017. [Online]. Available: www.ijariit.com
- [25] S. Keele *et al.*, *Guidelines for performing systematic literature reviews in software engineering*. Technical report, ver. 2.3 ebse technical report. ebse, 2007.
- [26] H. Albinali and F. Azzedin, “Replay attacks in rpl-based internet of things: Comparative and empirical study,” *Computer Networks*, vol. 257, p. 110996, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128624008284>
- [27] —, “Towards RPL attacks and mitigation taxonomy: Systematic literature review approach,” *IEEE Transactions on Network and Service Management*, 2024.
- [28] R. Alghofaili, H. Albinali, and F. Azzedin, “Build–launch–consolidate framework and toolkit for impact analysis on wireless sensor networks,” *Journal of Sensor and Actuator Networks*, vol. 13, no. 1, p. 17, 2024.
- [29] H. Albinali, F. Azzedin, and M. Riaz, “DEVS-RPL IDS: A Specification-Based Intrusion Detection on RPL-Based Network Using Discrete Event Model & Statistical Process Control,” 2024.
- [30] H. Albinali, R. Alghofaili, and F. Azzedin, “Analytical study of local repair attack on rpl-based 6lowpan networks in internet of things,” in *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, 2023, pp. 641–646.

- [31] J. Neeli and S. Patil, “Insight to security paradigm , research trend & statistics in internet of things(IoT),” *Global Transitions Proceedings*, vol. 2, no. 1, pp. 84–90, 2021. [Online]. Available: <https://doi.org/10.1016/j.gltp.2021.01.012>
- [32] N. Sultana and M. Tamanna, “Exploring the benefits and challenges of Internet of Things (IoT) during Covid-19: a case study of Bangladesh,” *Discover Internet of Things*, vol. 1, no. 1, 2021. [Online]. Available: <https://doi.org/10.1007/s43926-021-00020-9>
- [33] M. Ghaleb and F. Azzedin, “Towards scalable and efficient architecture for modeling trust in IoT environments,” *Sensors*, vol. 21, no. 9, 2021.
- [34] L. Kakkar, D. Gupta, S. Saxena, and S. Tanwar, “Iot architectures and its security: a review,” in *Proceedings of the Second International Conference on Information Management and Machine Intelligence: ICIMMI 2020*. Springer, 2021, pp. 87–94.
- [35] I. Alaoui Ismaili, A. Azyat, N. Raissouni, N. Ben Achhab, A. Chahboun, and M. Lahraoua, “Comparative Study of ZigBee and 6LoWPAN Protocols: Review,” in *Proceedings of the Third International Conference on Computing and Wireless Communication Systems, ICCWCS 2019, April 24-25, 2019, Faculty of Sciences, Ibn Tofail University -Kénitra- Morocco*. Kenitra, Morocco: EAI, 2019. [Online]. Available: <http://eudl.eu/doi/10.4108/eai.24-4-2019.2284215>
- [36] H. Lin and N. W. Bergmann, “Iot privacy and security challenges for smart home environments,” *Information*, vol. 7, no. 3, p. 44, 2016.

- [37] N. Accettura and G. Piro, “Optimal and secure protocols in the IETF 6TiSCH communication stack,” in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. Istanbul, Turkey: IEEE, Jun. 2014, pp. 1469–1474. [Online]. Available: <http://ieeexplore.ieee.org/document/6864831/>
- [38] K. Avila, D. Jabba, and J. Gomez, “Security Aspects for RPL-Based Protocols: A Systematic Review in IoT,” *Applied Sciences*, vol. 10, no. 18, p. 6472, 2020.
- [39] B. Mohamed and F. Mohamed, “QoS Routing RPL for Low Power and Lossy Networks,” *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.
- [40] O. Gaddour and A. Koubâa, “RPL in a nutshell: A survey,” *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2012.06.016>
- [41] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” no. RFC6550, p. RFC6550, Mar. 2012. [Online]. Available: <https://www.rfc-editor.org/info/rfc6550>
- [42] O. Iova, P. Picco, T. Istomin, and C. Kiraly, “RPL: The Routing Standard for the Internet of Things... or Is It?” *IEEE Communications Magazine*, vol. 54, no. 11, pp. 16–22, 2016.

- [43] A. Parasuram, D. Culler, and R. Katz, “An Analysis of the RPL Routing Standard for Low Power and Lossy Networks,” *Technical Report No. UCB/EECS-2016-106*, p. 98, 2016.
- [44] P. Levis; T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The Trickle Algorithm,” *Internet Engineering Task Force (IETF)*, 2011. [Online]. Available: <http://www.rfc-editor.org/info/rfc6206>.
- [45] C. Pu and K. K. R. Choo, “Lightweight Sybil Attack Detection in IoT based on Bloom Filter and Physical Unclonable Function,” *Computers and Security*, vol. 113, 2 2022.
- [46] R. Masadeh, B. AlSaaidah, E. Masadeh, M. R. Al-Hadidi, and O. Almomani, “Elastic Hop Count Trickle Timer Algorithm in Internet of Things,” *Sustainability*, vol. 14, no. 19, p. 12417, Sep. 2022. [Online]. Available: <https://www.mdpi.com/2071-1050/14/19/12417>
- [47] P. Thubert, “Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC Editor, Tech. Rep. RFC6552, Mar. 2012. [Online]. Available: <https://www.rfc-editor.org/info/rfc6552>
- [48] O. Gnawali and P. Levis, “Rfc 6719: The minimum rank with hysteresis objective function,” *Internet Engineering Task Force (IETF) Request For Comments*, 2012.

- [49] M. B. Yassien, O. Alzoubi, M. Shatnawi, and A. A. Rawashdeh, “Performance analysis of RPL objective functions,” *ACM International Conference Proceeding Series*, no. OCTOBER, 2019.
- [50] A. Raoof, A. Matrawy, and C.-H. Lung, “Enhancing routing security in iot: Performance evaluation of rpl secure mode under attacks,” *arXiv preprint arXiv:2004.07815*, 2020.
- [51] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, and G. Dini, “An implementation and evaluation of the security features of RPL,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10517 LNCS, no. November, pp. 63–76, 2017.
- [52] A. Raoof, A. Matrawy, and C.-H. Lung, “Secure routing in iot: Evaluation of rpl’s secure mode under attacks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.
- [53] C. L. Duenas Santos, A. M. Mezher, J. P. Astudillo León, J. Cardenas Barrera, E. Castillo Guerra, and J. Meng, “Q-rpl: Q-learning-based routing protocol for advanced metering infrastructure in smart grids,” *Sensors*, vol. 24, no. 15, p. 4818, 2024.
- [54] Y. Wang, Y. Li, J. Lei, and F. Shang, “Robust and energy-efficient rpl optimization algorithm with scalable deep reinforcement learning for iiot,” *Computer Networks*, vol. 255, p. 110894, 2024.

- [55] M. A. B. Brito and D. J. Molinares, “Rpl routing metrics for 5g networks: Systematic review in iiot,” in *2023 IEEE Colombian Caribbean Conference (C3)*. IEEE, 2023, pp. 1–6.
- [56] K. A. Darabkh, H. H. AlAdwan, M. Al-Akhras, F. Jubair, and S. Rahamneh, “A revolutionary rpl-based iot routing protocol for monitoring building structural health in smart city domain utilizing equilibrium optimizer algorithm,” *Soft Computing*, pp. 1–40, 2024.
- [57] A. Bang and U. P. Rao, “Performance evaluation of rpl protocol under decreased and increased rank attacks: A focus on smart home use-case,” *SN Computer Science*, vol. 4, no. 4, p. 329, 2023.
- [58] Z. Zhang, H. Hu, and X. Hu, “Routing protocol for healthcare applications data over the 6lowpan-based wireless sensor networks,” *Procedia Computer Science*, vol. 225, pp. 2153–2162, 2023.
- [59] A. Giri *et al.*, “Advancing healthcare networks: Optimizing multi-objective rpl for diverse traffic in low-power, lossy environments,” *Journal of Electrical Systems*, vol. 20, no. 7s, pp. 482–494, 2024.
- [60] S. Khandelwal, S. Shinde, S. Srivastava, S. Choudhary, and U. Kshirsagar, “Evolution of protocols used in environmental monitoring using wsn,” 2023.
- [61] A. Wakili and S. Bakkali, “Aof: An adaptive algorithm for enhancing rpl objective function in smart agricultural iot networks,” *International Journal of Intelligent Networks*, vol. 5, pp. 325–339, 2024.

- [62] R. Sahay, G. Geethakumari, B. Mitra, and V. Thejas, “Exponential Smoothing based Approach for Detection of Blackhole Attacks in IoT,” in *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. Indore, India: IEEE, Dec. 2018, pp. 1–6.
- [63] D. Airehrour, J. Gutierrez, and S. K. Ray, “Securing RPL routing protocol from blackhole attacks using a trust-based mechanism,” *26th International Telecommunication Networks and Applications Conference, ITNAC 2016*, pp. 115–120, 2017.
- [64] F. Ahmed and Y. B. Ko, “Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks,” *Security and Communication Networks*, vol. 9, no. 18, pp. 5143–5154, 2016.
- [65] D. Airehrour, J. A. Gutierrez, and S. K. Ray, “A Trust-Aware RPL Routing Protocol to Detect Blackhole and Selective Forwarding Attacks,” *Australian Journal of Telecommunications and the Digital Economy*, vol. 5, no. 1, p. 50, 2017.
- [66] G. Soni and R. Sudhakar, “A L-IDS against Dropping Attack to Secure and Improve RPL Performance in WSN Aided IoT,” in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*. Noida, India: IEEE, Feb. 2020, pp. 377–383.
- [67] E. G. Ribera, B. M. Alvarez, C. Samuel, P. P. Ioulianos, and V. G. Vassilakis, “Heartbeat-based detection of blackhole and greyhole attacks in RPL networks,”

- in *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. IEEE, 2020, pp. 1–6.
- [68] S. M. H. Mirshahjafari and B. S. Ghahfarokhi, “Sinkhole+CloneID: A hybrid attack on RPL performance and detection method,” *Information Security Journal: A Global Perspective*, vol. 28, no. 4-5, pp. 107–119, Sep. 2019.
- [69] S. R. Taghanaki, S. B. Arzandeh, and A. Bohlooli, “A decentralized method for detecting clone id attacks on the internet of things,” in *5th International Conference on Internet of Things and Applications, IoT 2021*. Institute of Electrical and Electronics Engineers Inc., 5 2021.
- [70] F. Gara, L. B. Saad, and R. B. Ayed, “An efficient intrusion detection system for selective forwarding and clone attackers in IPv6-based wireless sensor networks under mobility,” *International Journal on Semantic Web and Information Systems*, vol. 13, no. 3, pp. 22–47, 2017.
- [71] A. Verma and V. Ranga, “Addressing copycat attacks in IPv6-based low power and lossy networks,” in *Science and Information Conference*. Springer, 2020, pp. 415–426.
- [72] —, “CoSec-RPL: detection of copycat attacks in RPL based 6LoWPANs using outlier analysis,” *Telecommunication Systems*, vol. 75, no. 1, pp. 43–61, 2020.
- [73] M. Sheibani, B. Barekatain, and E. Arvan, “A lightweight distributed detection algorithm for ddao attack on rpl routing protocol in internet of things,” *Pervasive and Mobile Computing*, vol. 80, 2 2022.

- [74] D. B. Gothawal and S. Nagaraj, “An intelligent and lightweight intrusion detection mechanism for RPL routing attacks by applying automata model,” *Information Security Journal: A Global Perspective*, vol. 32, no. 1, pp. 1–20, Jan. 2023.
- [75] C. Pu and B. Groves, “Energy depletion attack in low power and lossy networks: Analysis and defenses,” in *2nd International Conference on Data Intelligence and Security, ICDIS 2019*. Institute of Electrical and Electronics Engineers Inc., 6 2019, pp. 14–21.
- [76] A. A. Anitha and L. Arockiam, “Ada-IDS: AdaBoost Intrusion Detection System for ICMPv6 based Attacks in Internet of Things,” *IJACSA*, vol. 12, no. 11, 2021.
- [77] B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem, I. Romdhani, and L. Mackenzie, “Addressing the DAO Insider Attack in RPL’s Internet of Things Networks,” *IEEE Communications Letters*, vol. 23, no. 1, pp. 68–71, 2019.
- [78] I. Wadhaj, B. Ghaleb, C. Thomson, A. Al-Dubai, and W. J. Buchanan, “Mitigation mechanisms against the dao attack on the routing protocol for low power and lossy networks (rpl),” *IEEE Access*, vol. 8, pp. 43 665–43 675, 2020.
- [79] M. Alzubaidi, M. Anbar, S. Al-Saleem, S. Al-Sarawi, and K. Alieyan, “Review on mechanisms for detecting sinkhole attacks on RPLs,” *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, no. July, pp. 369–374, 2017.

- [80] M. Zaminkar and R. Fotohi, “SoS-RPL: Securing Internet of Things Against Sinkhole,” *Wireless Personal Communications*, vol. 114, pp. 1287—1312, 2020.
- [81] M. Alzubaidi, M. Anbar, Y.-W. Chong, and S. Al-Sarawi, “Hybrid Monitoring Technique for Detecting Abnormal Behaviour in RPL-Based Network,” *jcm*, pp. 198–208, 2018.
- [82] R. Sahay, G. Geethakumari, and B. Mitra, “A novel network partitioning attack against routing protocol in internet of things,” *Ad Hoc Networks*, vol. 121, 10 2021.
- [83] B. Groves and C. Pu, “A gini index-based countermeasure against sybil attack in the internet of things,” in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.
- [84] C. Pu, “Sybil attack in RPL-based Internet of Things: Analysis and defenses,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4937–4949, 2020.
- [85] C. Pu and S. Hajjar, “Mitigating forwarding misbehaviors in rpl-based low power and lossy networks,” in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, vol. 2018-January. Institute of Electrical and Electronics Engineers Inc., 3 2018, pp. 1–6.
- [86] A. Verma and V. Ranga, “Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, pp. 1–25, 2020.

- [87] 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*. Institute of Electrical and Electronics Engineers Inc., oct 2014.
- [88] 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*, no. June, pp. 17–31, 2015.
- [89] E. Canbalaban and S. Sen, “A Cross-Layer Intrusion Detection System for RPL-Based Internet of Things,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12338 LNCS, pp. 214–227, 2020.
- [90] E. Aydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsstrom, and M. Gidlund, “A Central Intrusion Detection System for RPL-Based Industrial Internet of Things,” *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, vol. 2019-May, 2019.
- [91] E. Kfoury, J. Saab, P. Younes, and R. Achkar, “A self organizing map intrusion detection system for RPL protocol attacks,” *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 11, no. 1, pp. 30–43, 2019.

- [92] A. Rehman and M. A. Lodhi, “Rank Attack using Objective Function in RPL for Low Power and Lossy Networks,” 2016.
- [93] G. Violettas, G. Simoglou, S. Petridou, and L. Mamatas, “A softwarized intrusion detection system for the rpl-based internet of things networks,” *Future Generation Computer Systems*, vol. 125, pp. 698–714, 12 2021.
- [94] S. M. Muzammal, R. K. Murugesan, N. Jhanjhi, M. S. Hossain, and A. Yassine, “Trust and Mobility-Based Protocol for Secure Routing in Internet of Things,” *Sensors*, vol. 22, no. 16, p. 6215, Aug. 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/16/6215>
- [95] B. Farzaneh, M. Koosha, E. Boochanpour, and E. Alizadeh, “A new method for intrusion detection on RPL routing protocol using fuzzy logic,” in *2020 6th International Conference on Web Research (ICWR)*. IEEE, 2020, pp. 245–250.
- [96] A. Agiollo, M. Conti, P. Kaliyar, T. N. Lin, and L. Pajola, “DETONAR: Detection of Routing Attacks in RPL-Based IoT,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1178–1190, 2021.
- [97] A. Le, J. Loo, Y. Luo, and A. Lasebae, “Specification-based IDS for securing RPL from topology attacks,” *IFIP Wireless Days*, vol. 1, no. 1, pp. 4–6, 2011.
- [98] R. Challa and K. S. Rao, “Resource Based Attacks Security Using RPL Protocol in Internet of Things,” *ISI*, vol. 27, no. 1, pp. 165–170, Feb. 2022.
- [99] A. Thomas, T. Gireesh Kumar, and A. K. Mohan, “Neighbor Attack Detection in Internet of Things,” in *Advanced Computational and Communication*

- Paradigms*, S. Bhattacharyya, N. Chaki, D. Konar, U. K. Chakraborty, and C. T. Singh, Eds. Singapore: Springer Singapore, 2018, vol. 706, pp. 187–196, series Title: Advances in Intelligent Systems and Computing.
- [100] A. Raoof, C.-H. Lung, and A. Matrawy, “Securing rpl using network coding: The chained secure mode (csm),” *IEEE Internet of Things Journal*, 2021.
 - [101] A. Kamble, V. S. Malemath, and D. Patil, “Security attacks and secure routing protocols in RPL-based Internet of Things: Survey,” *2017 International Conference on Emerging Trends and Innovation in ICT, ICEI 2017*, pp. 33–39, 2017.
 - [102] J.-D. Kim, M. Ko, and J.-M. Chung, “Physical Identification Based Trust Path Routing Against Sybil Attacks on RPL in IoT Networks,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 1102–1106, May 2022.
 - [103] D. Arshad, M. Asim, N. Tariq, T. Baker, H. Tawfik, and D. Al-Jumeily OBE, “THC-RPL: A lightweight Trust-enabled routing in RPL-based IoT networks against Sybil attack,” *PLoS ONE*, vol. 17, no. 7, p. e0271277, Jul. 2022.
 - [104] L. Wallgren, S. Raza, and T. Voigt, “Routing attacks and countermeasures in the rpl-based internet of things,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, p. 794326, 2013.
 - [105] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, “A study of RPL DODAG version attacks,” *Lecture Notes in Computer Science*

(including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*), vol. 8508 LNCS, pp. 92–104, 2014.

- [106] 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.
- [107] R. Sahay, G. Geethakumari, B. Mitra, and I. Sahoo, “Efficient Framework for Detection of Version Number Attack in Internet of Things,” in *Intelligent Systems Design and Applications*, A. Abraham, A. K. Cherukuri, P. Melin, and N. Gandhi, Eds. Cham: Springer International Publishing, 2020, vol. 941, pp. 480–492, series Title: Advances in Intelligent Systems and Computing.
- [108] A. Arı̇, S. B. Örs Yalçın, and S. F. Oktuğ, “New lightweight mitigation techniques for rpl version number attacks,” *Ad Hoc Networks*, vol. 85, pp. 81–91, 3 2019.
- [109] F. I. Khan, T. Shon, T. Lee, and K. Kim, “Wormhole attack prevention mechanism for rpl based llm network,” in *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2013, pp. 149–154.
- [110] P. Kaliyar, W. B. Jaballah, M. Conti, and C. Lal, “LiDL: Localization with early detection of sybil and wormhole attacks in IoT Networks,” *Computers & Security*, vol. 94, p. 101849, 2020.
- [111] M. S. Ahsan, M. N. M. Bhutta, and M. Maqsood, “Wormhole attack detection in routing protocol for low power lossy networks,” in *2017*

- International Conference on Information and Communication Technologies (ICICT)*, IEEE. IEEE, 12 2017, pp. 58–67. [Online]. Available: <http://ieeexplore.ieee.org/document/8320165/>
- [112] S. Deshmukh-Bhosale and S. S. Sonavane, “A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things,” *Procedia Manufacturing*, vol. 32, pp. 840–847, 2019. [Online]. Available: <https://doi.org/10.1016/j.promfg.2019.02.292>
- [113] C.-M. Chen, S.-C. Hsu, and G.-H. Lai, “Defense Denial-of Service Attacks on IPv6 Wireless Sensor Networks,” in *Genetic and Evolutionary Computing*, T. T. Zin, J. C.-W. Lin, J.-S. Pan, P. Tin, and M. Yokota, Eds. Cham: Springer International Publishing, 2016, vol. 387, pp. 319–326, series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-319-23204-1_32
- [114] A. Seyfollahi, M. Moodi, and A. Ghaffari, “Mfo-rpl: A secure rpl-based routing protocol utilizing moth-flame optimizer for the iot applications,” *Computer Standards and Interfaces*, vol. 82, 8 2022.
- [115] U. Kiran, “IDS To Detect Worst Parent Selection Attack In RPL-Based IoT Network,” in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. Bangalore, India: IEEE, Jan. 2022, pp. 769–773.
- [116] A. Sehgal, A. Mayzaud, R. Badonnel, I. Chrisment, and J. Schönwälder, “Addressing DODAG inconsistency attacks in RPL networks,” *2014 Global Infor-*

- mation Infrastructure and Networking Symposium, GIIS 2014*, no. September, 2014.
- [117] S. Mangelkar, S. N. Dhage, and A. V. Nimkar, “A comparative study on RPL attacks and security solutions,” *Proceedings of 2017 International Conference on Intelligent Computing and Control, I2C2 2017*, vol. 2018-Janua, pp. 1–6, 2018.
 - [118] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, “A security threat analysis for the routing protocol for low-power and lossy networks (rpls),” *RFC 7416*, p. 131, 2015.
 - [119] A. Mathur, T. Newe, and M. Rao, “Defence against black hole and selective forwarding attacks for medical wsns in the iot,” *Sensors*, vol. 16, no. 1, p. 118, 2016.
 - [120] P. S. Nandhini and B. M. Mehtre, “Intrusion detection system based rpl attack detection techniques and countermeasures in iot: A comparison,” *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, pp. 666–672, 2019.
 - [121] A. M. Said, A. Yahyaoui, F. Yaakoubi, and T. Abdellatif, “Machine Learning Based Rank Attack Detection for Smart Hospital Infrastructure,” in *The Impact of Digital Technologies on Public Health in Developed and Developing Countries*, M. Jmaiel, M. Mokhtari, B. Abdulrazak, H. Aloulou, and S. Kallel, Eds. Cham: Springer International Publishing, 2020, vol. 12157, pp. 28–40, series Title: Lecture Notes in Computer Science.

- [122] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, and I. Ahmedy, “Compression header analyzer intrusion detection system (CHA-IDS) for 6LoWPAN communication protocol,” *IEEE Access*, vol. 6, pp. 16 623–16 638, 2018.
- [123] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, “Using the RPL protocol for supporting passive monitoring in the Internet of Things,” *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 366–374, 2016.
- [124] N. M. Müller, P. Debus, D. Kowatsch, and K. Böttinger, “Distributed anomaly detection of single mote attacks in rpl networks.” in *ICETE (2)*, 2019, pp. 378–385.
- [125] A. Verma and V. Ranga, “Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things,” in *2019 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU)*. IEEE, 2019, pp. 1–6.
- [126] A. Nikam and D. Ambawade, “Opinion metric based intrusion detection mechanism for RPL protocol in IoT,” in *2018 3rd International Conference for Convergence in Technology (I2CT)*. IEEE, 2018, pp. 1–6.
- [127] G. Thamilarasu and S. Chawla, “Towards deep-learning-driven intrusion detection for the internet of things,” *Sensors*, vol. 19, no. 9, p. 1977, 2019.
- [128] S. Budania and M. V. Shenoy, “RPL*: An Explainable AI-based routing protocol for Internet of Mobile Things,” *Internet of Things*, vol. 27, p. 101283,

Oct. 2024. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2542660524002245>

- [129] R. Sahay, G. Geethakumari, B. Mitra, and N. Goyal, "Investigating Packet Dropping Attacks in RPL-DODAG in IoT," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. Bombay, India: IEEE, Mar. 2019, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/9033926/>
- [130] M. M. Iqbal, A. Ahmed, and U. Khadam, "Sinkhole Attack in Multi-sink Paradigm: Detection and Performance Evaluation in RPL based IoT," in *2020 International Conference on Computing and Information Technology (ICCIIT-1441)*. Tabuk, Saudi Arabia: IEEE, Sep. 2020, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/9213797/>
- [131] A. Mayzaud, R. Badonnel, and I. Chrisment, "Detecting version number attacks in RPL-based networks using a distributed monitoring architecture," in *2016 12th International Conference on Network and Service Management (CNSM)*. Montreal, QC, Canada: IEEE, Oct. 2016, pp. 127–135.
- [132] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach," *Computer Communications*, vol. 98, pp. 52–71, 2017.
- [133] G.-H. Lai, "Detection of wormhole attacks on IPv6 mobility-based wireless sensor network," *J Wireless Com Network*, vol. 2016, no. 1, p. 274, Dec. 2016.

- [134] V. Kumar, G. Oikonomou, and T. Tryfonas, “Traffic forensics for IPv6-based Wireless Sensor Networks and the Internet of Things,” in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. Reston, VA, USA: IEEE, Dec. 2016, pp. 633–638.
- [135] M. Khardioui, A. Bamou, M. D. El Ouadghiri, and B. Aghoutane, “Implementation and evaluation of an intrusion detection system for iot: against routing attacks,” in *Advanced Intelligent Systems for Sustainable Development (AI2SD’2019) Volume 6-Advanced Intelligent Systems for Networks and Systems*. Springer, 2020, pp. 155–166.
- [136] D. Shreenivas, S. Raza, and T. Voigt, “Intrusion detection in the RPL-connected 6LoWPAN networks,” in *Proceedings of the 3rd ACM international workshop on IoT privacy, trust, and security*, 2017, pp. 31–38.
- [137] M. Alzubaidi, M. Anbar, and S. M. Hanshi, “Neighbor-Passive Monitoring Technique for Detecting Sinkhole Attacks in RPL Networks,” in *Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence*. Jakarta Indonesia: ACM, Dec. 2017, pp. 173–182.
- [138] U. Shafique, A. Khan, A. Rehman, F. Bashir, and M. Alam, “Detection of rank attack in routing protocol for Low Power and Lossy Networks,” *Annales des Telecommunications/Annals of Telecommunications*, vol. 73, no. 7-8, pp. 429–438, 2018.

- [139] R. Sahay, G. Geethakumari, K. Modugu, and B. Mitra, "Traffic Convergence Detection in IoT LLNs: A Multilayer Perceptron based Mechanism," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. Bangalore, India: IEEE, Nov. 2018, pp. 1715–1722.
- [140] M. A. M. Behnam Farzaneh and S. Jamali, "An Anomaly-Based IDS for Detecting Attacks in RPL-Based Internet of Things," in *2019 5th International Conference on Web Research (ICWR)*. IEEE, 2019, pp. 61—66.
- [141] S. R. Taghanaki, K. Jamshidi, and A. Bohlooli, "DEEM: A Decentralized and Energy Efficient Method for detecting sinkhole attacks on the internet of things," in *2019 9th International Conference on Computer and Knowledge Engineering (ICCCKE)*. Mashhad, Iran: IEEE, Oct. 2019, pp. 325–330.
- [142] V. Neerugatti and A. Rama Mohan Reddy, "Detection and Prevention of Black Hole Attack in RPL Protocol Based on the Threshold Value of Nodes in the Internet of Things Networks," *IJITEE*, vol. 8, no. 9S3, pp. 325–329, Aug. 2019.
- [143] S. Shin, K. Kim, and T. Kwon, "Detection of malicious packet dropping attacks in RPL-based internet of things," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 31, no. 2, p. 9, 2019.
- [144] Fatima-tuz-Zahra, N. Jhanjhi, S. N. Brohi, and N. A. Malik, "Proposing a Rank and Wormhole Attack Detection Framework using Machine Learning," in *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. Karachi, Pakistan: IEEE, Dec. 2019, pp. 1–9.

- [145] S. Murali and A. Jamalipour, “A Lightweight Intrusion Detection for Sybil Attack Under Mobile RPL in the Internet of Things,” *IEEE Internet Things J.*, vol. 7, no. 1, pp. 379–388, Jan. 2020.
- [146] K. N. Qureshi, S. S. Rana, A. Ahmed, and G. Jeon, “A novel and secure attacks detection framework for smart cities industrial internet of things,” *Sustainable Cities and Society*, vol. 61, p. 102343, Oct. 2020.
- [147] M. Al Qurashi, C. M. Angelopoulos, and V. Katos, “An architecture for resilient intrusion detection in ad-hoc networks,” *Journal of Information Security and Applications*, vol. 53, p. 102530, 2020.
- [148] D. B. Gothawal and S. V. Nagaraj, “Anomaly-Based Intrusion Detection System in RPL by Applying Stochastic and Evolutionary Game Models over IoT Environment,” pp. 1323–1344, 2020.
- [149] V. Neerugatti and A. Rama Mohan Reddy, “Artificial intelligence-based technique for detection of selective forwarding attack in rpl-based internet of things networks,” in *Emerging Research in Data Engineering Systems and Computer Communications: Proceedings of CCODE 2019*. Springer, 2020, pp. 67–77.
- [150] P. P. Ioulianou and V. G. Vassilakis, “Denial-of-service attacks and countermeasures in the RPL-based internet of things,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11980 LNCS, pp. 374–390, 2020.

- [151] S. Deshmukh-Bhosale and S. S. Sonavane, “Design of Intrusion Detection System for Wormhole Attack Detection in Internet of Things,” in *Advanced Computing and Intelligent Engineering*, B. Pati, C. R. Panigrahi, R. Buyya, and K.-C. Li, Eds. Singapore: Springer Singapore, 2020, vol. 1082, pp. 513–523, series Title: Advances in Intelligent Systems and Computing.
- [152] A. Althubaity, R. Ammar, and S. Han, “Detecting Rules-related Attacks in RPL-based Resource-Constrained Wireless Networks,” in *2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. Louisville, KY, USA: IEEE, Dec. 2020, pp. 1–8.
- [153] A. D. Seth, S. Biswas, and A. K. Dhar, “Detection and Verification of Decreased Rank Attack using Round-Trip Times in RPL-Based 6LoWPAN Networks,” in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. New Delhi, India: IEEE, Dec. 2020, pp. 1–6.
- [154] J. Foley, N. Moradpoor, and H. Ochenyi, “Employing a Machine Learning Approach to Detect Combined Internet of Things Attacks against Two Objective Functions Using a Novel Dataset,” *Security and Communication Networks*, vol. 2020, pp. 1–17, Feb. 2020.
- [155] S. O. M. Kamel and S. A. Elhamayed, “Mitigating the Impact of IoT Routing Attacks on Power Consumption in IoT Healthcare Environment using Convolutional Neural Network,” *IJCNIS*, vol. 12, no. 4, pp. 11–29, Aug. 2020.

- [156] C. Samuel, B. M. Alvarez, E. Garcia Ribera, P. P. Ioulianou, and V. G. Vassilakis, "Performance Evaluation of a Wormhole Detection Method using Round-Trip Times and Hop Counts in RPL-Based 6LoWPAN Networks," in *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. Porto, Portugal: IEEE, Jul. 2020, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9249612/>
- [157] Fatima-tuz-Zahra, N. Jhanjhi, S. N. Brohi, N. A. Malik, and M. Humayun, "Proposing a Hybrid RPL Protocol for Rank and Wormhole Attack Mitigation using Machine Learning," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*. Sakaka, Saudi Arabia: IEEE, Oct. 2020, pp. 1–6.
- [158] W. Choukri, H. Lamaazi, and N. Benamar, "RPL rank attack detection using Deep Learning," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. Sakheer, Bahrain: IEEE, Dec. 2020, pp. 1–6.
- [159] C. D. Morales-Molina, A. Hernandez-Suarez, G. Sanchez-Perez, L. K. Toscano-Medina, H. Perez-Meana, J. Olivares-Mercado, J. Portillo-Portillo, V. Sanchez, and L. J. Garcia-Villalba, "A dense neural network approach for detecting clone id attacks on the RPL protocol of the IoT," *Sensors*, vol. 21, no. 9, p. 3173, 2021.
- [160] M. Chowdhury, B. Ray, S. Chowdhury, and S. Rajasegarar, "A Novel Insider

- Attack and Machine Learning Based Detection for the Internet of Things,” *ACM Trans. Internet Things*, vol. 2, no. 4, pp. 1–23, Nov. 2021.
- [161] A. Patel and D. Jinwala, “A reputation-based RPL protocol to detect selective forwarding attack in Internet of Things,” *International Journal of Communication Systems*, vol. 35, no. 1, p. e5007, 2022.
- [162] S. Yilmaz, E. Aydogan, and S. Sen, “A Transfer Learning Approach for Securing Resource-Constrained IoT Devices,” *IEEE Trans. Inform. Forensic Secur.*, vol. 16, pp. 4405–4418, 2021.
- [163] S. Sharma and V. K. Verma, “AIEMLA: artificial intelligence enabled machine learning approach for routing attacks on internet of things,” *The Journal of Supercomputing*, pp. 1–31, 2021.
- [164] M. Mohapatro and I. Snigdh, “An Experimental Study of Distributed Denial of Service and Sink Hole Attacks on IoT based Healthcare Applications,” *Wireless Pers Commun*, vol. 121, no. 1, pp. 707–724, Nov. 2021.
- [165] M. U. Kiraz and A. Yilmaz, “Comparison of ML algorithms to detect vulnerabilities of RPL-based IoT devices in intelligent and fuzzy systems,” in *Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation: Proceedings of the INFUS 2021 Conference, held August 24-26, 2021. Volume 2*. Springer, 2022, pp. 254–262.
- [166] J. Karande and S. Joshi, “DEDA: An algorithm for early detection of topology attacks in the internet of things,” *IJECE*, vol. 11, no. 2, p. 1761, Apr. 2021.

- [167] S. Nayak, N. Ahmed, and S. Misra, “Deep Learning-Based Reliable Routing Attack Detection Mechanism for Industrial Internet of Things,” *Ad Hoc Networks*, vol. 123, p. 102661, Dec. 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1570870521001748>
- [168] A. M. Said, A. Yahyaoui, and T. Abdellatif, “Efficient Anomaly Detection for Smart Hospital IoT Systems,” *Sensors*, vol. 21, no. 4, p. 1026, Feb. 2021.
- [169] F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, “Fault-tolerant AI-driven Intrusion Detection System for the Internet of Things,” *International Journal of Critical Infrastructure Protection*, vol. 34, p. 100436, Sep. 2021.
- [170] D. S. Nair and S. K. Bj, “Identifying Rank Attacks and Alert Application in WSN,” in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*. Coimbatre, India: IEEE, Jul. 2021, pp. 798–802.
- [171] S. S. Ambarkar and N. Shekokar, “Impact Analysis of RPL Attacks on 6Lo WPAN based Internet of Things network,” in *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. Bangalore, India: IEEE, Jul. 2021, pp. 1–5.
- [172] S. Sinha and S. B, “Impact of DoS attack in IoT system and identifying the attacker location for interference attacks,” in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*. Coimbatre, India: IEEE, Jul. 2021, pp. 657–662.

- [173] S. Karmakar, J. Sengupta, and S. D. Bit, "LEADER: Low Overhead Rank Attack Detection for Securing RPL based IoT," in *2021 International Conference on COMMunication Systems & NETworkS (COMSNETS)*. Bangalore, India: IEEE, Jan. 2021, pp. 429–437.
- [174] A. Mehbodniya, J. L. Webber, M. Shabaz, H. Mohafez, and K. Yadav, "Machine learning technique to detect sybil attack on IoT based sensor network," *IETE Journal of Research*, pp. 1–9, 2021.
- [175] M. D. Momand, M. Khan Mohsin, and Ihsanulhaq, "Machine Learning-based Multiple Attack Detection in RPL over IoT," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*. Coimbatore, India: IEEE, Jan. 2021, pp. 1–8.
- [176] H. B. Patel and D. C. Jinwala, "6MID:Mircochain based Intrusion Detection for 6LoWPAN based IoT networks," *Procedia Computer Science*, vol. 184, pp. 929–934, 2021.
- [177] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, "ML-LGBM: A Machine Learning Model Based on Light Gradient Boosting Machine for the Detection of Version Number Attacks in RPL-Based Networks," *IEEE Access*, vol. 9, pp. 83 654–83 665, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9448047/>
- [178] S. Sinha and K. G, "Network layer DoS Attack on IoT System and location identification of the attacker," in *2021 Third International Conference on In-*

- ventive Research in Computing Applications (ICIRCA)*. Coimbatore, India: IEEE, Sep. 2021, pp. 22–27.
- [179] M. Sharma, H. Elmiligi, and F. Gebali, “A Novel Intrusion Detection System for RPL-Based Cyber-Physical Systems,” *IEEE Can. J. Electr. Comput. Eng.*, vol. 44, no. 2, pp. 246–252, 2021.
- [180] A. W. Burange and V. Deshmukh, “Secured Routing System for Low Energy Networks,” in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 3*. Springer, 2021, pp. 165–173.
- [181] T. A. Alamiedy, M. F. Anbar, B. Belaton, A. H. Kabla, and B. H. Khudayer, “Ensemble feature selection approach for detecting denial of service attacks in RPL networks,” in *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25*. Springer, 2021, pp. 340–360.
- [182] A. Jamil, M. Q. Ali, and M. E. A. Alkhalec, “Sinkhole Attack Detection and Avoidance Mechanism for RPL in Wireless Sensor Networks,” *AETiC*, vol. 5, no. 5, pp. 94–101, Mar. 2021.
- [183] C. Prajisha and A. Vasudevan, “An Intrusion Detection System for Blackhole Attack Detection and Isolation in RPL Based IoT Using ANN,” in *Advanced Computing: 11th International Conference, IACC 2021, Msida, Malta, December 18–19, 2021, Revised Selected Papers*. Springer, 2022, pp. 332–347.

- [184] E. Garcia Ribera, B. Martinez Alvarez, C. Samuel, P. P. Ioulianou, and V. G. Vassilakis, “An Intrusion Detection System for RPL-Based IoT Networks,” *Electronics*, vol. 11, no. 23, p. 4041, Dec. 2022.
- [185] N. Abosata, S. Al-Rubaye, and G. Inalhan, “Customised Intrusion Detection for an Industrial IoT Heterogeneous Network Based on Machine Learning Algorithms Called FTL-CID,” *Sensors*, vol. 23, no. 1, p. 321, Dec. 2022.
- [186] T. Thiyagu, S. Krishnaveni, and R. Arthi, “Deep learning approach for RPL wormhole attack,” in *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021*. Springer, 2022, pp. 321–330.
- [187] A. Mujtaba, S. Amjad, A. Mubarik, and M. U. Younus, “Enhancement of accuracy of the Rank Inconsistency Detection Algorithm,” *Pakistan Journal of Engineering and Applied Sciences*, vol. 30, p. 14, 2022.
- [188] D. K. Sharma, S. K. Dhurandher, S. Kumaram, K. Datta Gupta, and P. K. Sharma, “Mitigation of black hole attacks in 6LoWPAN RPL-based Wireless sensor network for cyber physical systems,” *Computer Communications*, vol. 189, pp. 182–192, May 2022.
- [189] M. Pishdar, Y. Seifi, M. Nasiri, and M. Bag-Mohammadi, “PCC-RPL: An efficient trust-based security extension for RPL,” *Information Security Journal: A Global Perspective*, vol. 31, no. 2, pp. 168–178, Mar. 2022.

- [190] F. Zahra, N. Z. Jhanjhi, N. A. Khan, S. N. Brohi, M. Masud, and S. Aljahdali, "Protocol-Specific and Sensor Network-Inherited Attack Detection in IoT Using Machine Learning," *Applied Sciences*, vol. 12, no. 22, p. 11598, Nov. 2022.
- [191] F. Zahra, N. Jhanjhi, S. N. Brohi, N. A. Khan, M. Masud, and M. A. AlZain, "Rank and Wormhole Attack Detection Model for RPL-Based Internet of Things Using Machine Learning," *Sensors*, vol. 22, no. 18, p. 6765, Sep. 2022.
- [192] R. Alghamdi and M. Bellaiche, "A cascaded federated deep learning based framework for detecting wormhole attacks in IoT networks," *Computers & Security*, vol. 125, p. 103014, Feb. 2023.
- [193] R. Alkanhel, D. Sami Khafaga, E.-S. M. El-kenawy, A. A. Abdelhamid, A. Ibrahim, R. Amin, M. Abotaleb, and B. M. El-den, "Hybrid Grey Wolf and Dipper Throated Optimization in Network Intrusion Detection Systems," *Computers, Materials & Continua*, vol. 74, no. 2, pp. 2695–2709, 2023.
- [194] E. Arvan, M. Koohi Habibi Dehkordi, and S. Jalili, "Secured location-aware mobility-enabled RPL," *Journal of Network and Computer Applications*, vol. 209, p. 103516, Jan. 2023.
- [195] 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.
- [196] U. Farooq, M. Asim, N. Tariq, T. Baker, and A. I. Awad, "Multi-mobile agent trust framework for mitigating internal attacks and augmenting RPL security," *Sensors*, vol. 22, p. 4539, 6 2022.

- [197] C. Pu, “Mitigating DAO inconsistency attack in RPL-based low power and lossy networks,” *2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC 2018*, vol. 2018-Janua, pp. 570–574, 2018.
- [198] J. Hui and J. Vasseur, “The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams.” [Online]. Available: https://www.hjp.at/doc/rfc/rfc6553.html{#}sec{__}5.1
- [199] A. Sehgal, A. Mayzaud, R. Badonnel, I. Chrisment, and J. Schonwalder, “Addressing DODAG inconsistency attacks in RPL networks,” in *2014 Global Information Infrastructure and Networking Symposium (GIIS)*. Montreal, QC: IEEE, Sep. 2014, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/6934253/>
- [200] Nan Jiang, Jianfei Liu, Wei Xiao, and Hongzhou Shi, “Routing attacks prevention mechanism for RPL based on micropayment scheme,” in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Chennai, India: IEEE, Mar. 2016, pp. 835–841. [Online]. Available: <http://ieeexplore.ieee.org/document/7566250/>
- [201] M. Surendar and A. Umamakeswari, “Indres: An intrusion detection and response system for internet of things with 6lowpan,” in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2016, pp. 1903–1908.

- [202] A. Althubaity, H. Ji, T. Gong, M. Nixon, R. Ammar, and S. Han, “ARM: A hybrid specification-based intrusion detection system for rank attacks in 6TiSCH networks,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Limassol: IEEE, Sep. 2017, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/8247593/>
- [203] F. Gara, L. Ben Saad, and R. Ben Ayed, “An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. Valencia, Spain: IEEE, Jun. 2017, pp. 276–281. [Online]. Available: <http://ieeexplore.ieee.org/document/7986299/>
- [204] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, “A Trust-Based Intrusion Detection System for Mobile RPL Based Networks,” in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. Exeter: IEEE, Jun. 2017, pp. 735–742. [Online]. Available: <http://ieeexplore.ieee.org/document/8276832/>
- [205] N. K. Thanigaivelan, E. Nigussie, S. Virtanen, and J. Isoaho, “Hybrid internal anomaly detection system for iot: Reactive nodes with cross-layer operation,” *Security and Communication Networks*, vol. 2018, 2018.
- [206] J. Jiang, Y. Liu, and B. Dezfouli, “A Root-based Defense Mechanism Against

- RPL Blackhole Attacks in Internet of Things Networks,” in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Honolulu, HI, USA: IEEE, Nov. 2018, pp. 1194–1199.
[Online]. Available: <https://ieeexplore.ieee.org/document/8659504/>
- [207] I. Tomić, P.-Y. Chen, M. J. Breza, and J. A. McCann, “Antilizer: Run Time Self-Healing Security for Wireless Sensor Networks,” in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. New York NY USA: ACM, Nov. 2018, pp. 107–116.
[Online]. Available: <https://dl.acm.org/doi/10.1145/3286978.3287029>
- [208] B. A. Alabsi, M. Anbar, S. Manickam, and O. E. Elejla, “Ddos attack aware environment with secure clustering and routing based on rpl protocol operation,” *IET Circuits, Devices and Systems*, vol. 13, pp. 748–755, 9 2019.
- [209] H. B. Patel and D. C. Jinwala, “Blackhole detection in 6lowpan based internet of things : An anomaly based approach,” in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 12 2019, pp. 947–954.
- [210] S. Choudhary and N. Kesswani, “Cluster-based intrusion detection method for internet of things,” in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 11 2019, pp. 1–8.
[Online]. Available: <https://ieeexplore.ieee.org/document/9035319/>
- [211] Deshmukh-Bhosale, Snehal and Sonavane, S.S, “Design Of Intrusion Detection System For Dos Attack In 6lowpan And RPL Based IoT Network,” *IJITEE*,

- vol. 8, no. 11, pp. 3840–3844, Sep. 2019.
- [212] D. B. Gothawal and S. Nagaraj, “Intrusion detection for enhancing rpl security,” *Procedia Computer Science*, vol. 165, pp. 565–572, 2019.
- [213] S. Y. Hashemi and F. S. Aliee, “Dynamic and comprehensive trust model for iot and its integration into rpl,” *The Journal of Supercomputing*, vol. 75, no. 7, pp. 3555–3584, 2019.
- [214] C. Pu, “Energy depletion attack against routing protocol in the internet of things,” in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1 2019, pp. 1–4. [Online]. Available: <https://ieeexplore.ieee.org/document/8651771/>
- [215] A. Tandon and P. Srivastava, “Trust-based enhanced secure routing against rank and sybil attacks in iot,” in *Twelfth International Conference on Contemporary Computing (IC3)*. IEEE, 8 2019, pp. 1–7.
- [216] A. Verma and V. Ranga, “Addressing Flooding Attacks in IPv6-based Low Power and Lossy Networks,” *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2019-Octob, no. December, pp. 552–557, 2019.
- [217] M. C. Belavagi and B. Muniyal, “Multiple intrusion detection in rpl based networks.” *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 10, no. 1, 2020.

- [218] S. Cakir, S. Toklu, and N. Yalcin, “Rpl attack detection and prevention in the internet of things networks using a gru based deep learning,” *IEEE Access*, vol. 8, pp. 183 678–183 689, 2020.
- [219] M. Conti, P. Kaliyar, and C. Lal, “A robust multicast communication protocol for Low power and Lossy networks,” *Journal of Network and Computer Applications*, vol. 164, p. 102675, Aug. 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1084804520301491>
- [220] Z. A. Almusaylim, A. Alhumam, W. Mansoor, P. Chatterjee, and N. Z. Jhanjhi, “Detection and mitigation of rpl rank and version number attacks in smart internet of things,” 2020. [Online]. Available: www.preprints.org
- [221] A. Gajbhiye, D. Sen, A. Bhatt, and G. Soni, “Dplpln: Detection and prevention from flooding attack in iot,” in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 9 2020, pp. 704–709. [Online]. Available: <https://ieeexplore.ieee.org/document/9215381/>
- [222] A. Althubaity, T. Gong, K. K. Raymond, M. Nixon, R. Ammar, and S. Han, “Specification-based distributed detection of rank-related attacks in rpl-based resource-constrained real-time wireless networks,” in *2020 IEEE Conference on Industrial Cyberphysical Systems, ICPS 2020*. Institute of Electrical and Electronics Engineers Inc., 6 2020, pp. 168–175.
- [223] S. Luangoudom, D. Tran, T. Nguyen, B. Khoa, C. Centre, H. A. Tran, G. Nguyen, and Q. T. Ha, “svblock: mitigating black hole attack in low-power

- and lossy networks,” *Int. J. Sensor Networks*, vol. 32, pp. 77–86, 2020.
- [224] S. S. Ambarkar and N. Shekokar, “An efficient authentication technique to protect iot networks from impact of rpl attacks,” *International Journal of Engineering Trends and Technology*, vol. 69, pp. 137–145, 10 2021.
- [225] A. Bang and U. P. Rao, “A novel decentralized security architecture against sybil attack in rpl-based iot networks: a focus on smart home use case,” *The Journal of Supercomputing*, pp. 1–36, 2021.
- [226] A. Jamalipour and S. Murali, “A taxonomy of machine learning based intrusion detection systems for the internet of things: A survey,” *IEEE Internet of Things Journal*, 2021.
- [227] M. A. Boudouaia, A. Ali-Pacha, A. Abouaissa, and P. Lorenz, “Security against rank attack in rpl protocol,” *IEEE Network*, vol. 34, pp. 133–139, 7 2020.
- [228] M. A. Boudouaia, A. Abouaissa, A. Ali-Pacha, A. Benayache, and P. Lorenz, “RPL rank based-attack mitigation scheme in IoT environment,” *Int J Commun Syst*, vol. 34, no. 13, Sep. 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/dac.4917>
- [229] R. Fotohi and H. Pakdel, “A lightweight and scalable physical layer attack detection mechanism for the internet of things (iot) using hybrid security schema,” *Wireless Personal Communications*, vol. 119, pp. 3089–3106, 8 2021.

- [230] U. Kulau, S. Müller, S. Schildt, F. Büsching, and L. Wolf, “Investigation & mitigation of the energy efficiency impact of node resets in rpl,” *Ad Hoc Networks*, vol. 114, 4 2021.
- [231] V. Neerugatti and A. R. M. Reddy, “Machine learning based technique for detection of rank attack in RPL based internet of things networks,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9 Special Issue 3, pp. 244–248, 2019.
- [232] F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, “Multicast DIS attack mitigation in RPL-based IoT-LLNs,” *Journal of Information Security and Applications*, vol. 61, p. 102939, Sep. 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2214212621001563>
- [233] P. S. Nandhini, P. Srinath, P. Veeramanikandan, and S. Malliga, “Version attack detection using claim algorithm in rpl based iot networks: Effects and performance parameters evaluation,” in *2nd International Conference on Smart Electronics and Communication, ICOSEC 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 209–215.
- [234] B. Patel and P. Shah, “Direct Neighbour Sink Reputed Trust Based Intrusion Detection System to Mitigate Sinkhole Attack in RPL for IoT Networks,” *JESTR*, vol. 14, no. 1, pp. 38–35, Feb. 2021. [Online]. Available: <http://www.jestr.org/downloads/Volume14Issue1/fulltext31412021.pdf>

- [235] A. Patel and D. Jinwala, “A trust-integrated rpl protocol to detect blackhole attack in internet of things,” *International Journal of Information Security and Privacy (IJISP)*, vol. 15, no. 4, pp. 1–17, 2021.
- [236] K. Prathapchandran and T. Janani, “A trust aware security mechanism to detect sinkhole attack in rpl-based iot environment using random forest – rfrtrust,” *Computer Networks*, vol. 198, 10 2021.
- [237] B. Rakesh and P. S. H, “Novel authentication and secure trust based rpl routing in mobile sink supported internet of things,” *Cyber-Physical Systems*, 2021.
- [238] K. Sanders and S. S. Yau, “An effective approach to protecting low-power and lossy iot networks against blackhole attacks,” in *2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCoM) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, Dec 2021, pp. 65–72.
- [239] D. Airehrour, J. A. Gutierrez, and S. K. Ray, “SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things,” *Future Generation Computer Systems*, vol. 93, pp. 860–876, 2019.
- [240] A. Raoof, C.-H. Lung, and A. Matrawy, “Securing RPL using Network Coding: The Chained Secure Mode (CSM),” *IEEE Internet of Things Journal*, 2021.

- [241] A. R. Zarzoor, “Securing rpl routing path for iot against rank attack via utilizing layering technique,” *International Journal on Electrical Engineering and Informatics*, vol. 13, pp. 789–800, 12 2021.
- [242] A. D. Seth, S. Biswas, and A. K. Dhar, “Mitigation technique against network isolation attack on rpl in 6lowpan network,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2021-December. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 68–73.
- [243] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, and L. T. Jung, “Smtrust: Proposing trust-based secure routing protocol for rpl attacks for iot applications,” in *2020 International Conference on Computational Intelligence (ICCI)*. IEEE, 10 2020, pp. 305–310. [Online]. Available: <https://ieeexplore.ieee.org/document/9247818/>
- [244] M. Zaminkar, F. Sarkohaki, and R. Fotohi, “A method based on encryption and node rating for securing the rpl protocol communications in the iot ecosystem,” *International Journal of Communication Systems*, vol. 34, 2 2021.
- [245] A. Alsirhani, M. A. Khan*, A. Alomari, S. Maryam, A. Younas, M. Iqbal, M. H. Siqqidi, and A. Ali, “Securing Low-Power Blockchain-Enabled IoT Devices Against Energy Depletion Attack,” *ACM Trans. Internet Technol.*, p. 3511903, Mar. 2022. [Online]. Available: <https://dl.acm.org/doi/10.1145/3511903>
- [246] I. S. Alsukayti and A. Singh, “A Lightweight Scheme for Mitigating RPL Version Number Attacks in IoT Networks,” *IEEE Access*, vol. 10, pp. 111 115–111 133,

2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9923901/>
- [247] A. O. Bang and U. P. Rao, “EMBOF-RPL: Improved RPL for early detection and isolation of rank attack in RPL-based internet of things,” *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 642–665, 2022. [Online]. Available: <https://doi.org/10.1007/s12083-021-01275-3>
- [248] V. R. J. Manne and S. Sreekanth, “Detection and Mitigation of RPL Routing Attacks in Internet of Things,” in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. New Delhi, India: IEEE, Mar. 2022, pp. 481–485. [Online]. Available: <https://ieeexplore.ieee.org/document/9763140/>
- [249] C. Miranda, G. Kaddoum, A. Boukhtouta, T. Madi, and H. A. Alameddine, “Intrusion Prevention Scheme Against Rank Attacks for Software-Defined Low Power IoT Networks,” *IEEE Access*, vol. 10, pp. 129 970–129 984, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9978298/>
- [250] P. Nandhini, S. Kuppuswami, S. Malliga, and R. DeviPriya, “A Lightweight Energy-Efficient Algorithm for mitigation and isolation of Internal Rank Attackers in RPL based Internet of Things,” *Computer Networks*, vol. 218, p. 109391, Dec. 2022. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S138912862200425X>
- [251] M. M. Sithik and B. M. Kumar, “Intelligent agent based virtual clustering and multi-context aware routing for congestion mitigation in secure RPL-IoT

- environment,” *Ad Hoc Networks*, vol. 137, p. 102972, Dec. 2022. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1570870522001470>
- [252] S. Ankam and D. S. Reddy, “A mechanism to detecting flooding attacks in quantum enabled cloud-based lowpower and lossy networks,” *Theoretical Computer Science*, vol. 941, pp. 29–38, Jan. 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0304397522004923>
- [253] Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for Internet of Things,” *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.jnca.2014.01.014>
- [254] H. Sedjelmaci, S. M. Senouci, and T. Taleb, “An accurate security game for low-resource iot devices,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9381–9393, 2017.
- [255] M. Surendar and A. Umamakeswari, “InDReS: An Intrusion Detection and response system for Internet of Things with 6LoWPAN,” *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, pp. 1903–1908, 2016.
- [256] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu, “AVOIDIT: A Cyber Attack Taxonomy,” 2009.

- [257] P. P. Ioulianou and V. G. Vassilakis, “A Trust-Based Intrusion Detection System for RPL Networks : Detecting a Combination of Rank and Blackhole Attacks,” pp. 124–153, 2022.
- [258] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, “DIO suppression attack against routing in the internet of things,” *IEEE Communications Letters*, vol. 21, no. 11, pp. 2524–2527, 2017.
- [259] S. Choudhary and N. Kesswani, “Detection and prevention of routing attacks in internet of things,” in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/Big-DataSE)*. IEEE, 2018, pp. 1537–1540.
- [260] A. Thomas, T. Gireesh Kumar, and A. K. Mohan, “Neighbor attack detection in internet of things,” in *Advanced Computational and Communication Paradigms*. Springer, 2018, pp. 187–196.
- [261] A. Verma and V. Ranga, “The impact of copycat attack on RPL based 6LoWPAN networks in Internet of Things,” *Computing*, 2020. [Online]. Available: <https://doi.org/10.1007/s00607-020-00862-1>
- [262] P. Pongle and G. Chavan, “Real time intrusion and wormhole attack detection in internet of things,” *International Journal of Computer Applications*, vol. 121, no. 9, 2015.

- [263] C. Dogan, S. Yilmaz, and S. Sen, “Analysis of RPL Objective Functions with Security Perspective,” no. February, pp. 71–80, 2022.
- [264] A. Verma and V. Ranga, “Comment on “DIO Suppression Attack Against Routing in the Internet of Things”,” *TechRxiv*, 2019.
- [265] S. Budania and M. V. Shenoy, “Oead: An online ensemble-based anomaly detection technique for rpl network,” in *Proceedings of the 25th International Conference on Distributed Computing and Networking*, 2024, pp. 316–321.
- [266] G. Tahar, B. A. Z. Eddine, and B. Djalila, “Lightweight mitigation against dio suppression attack in rpl-based iot networks,” in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 2024, pp. 1–6.
- [267] C. Kim, C. So-In, Y. Kongsorot, and P. Aimtongkham, “Flsec-rpl: a fuzzy logic-based intrusion detection scheme for securing rpl-based iot networks against dio neighbor suppression attacks,” *Cybersecurity*, vol. 7, no. 1, p. 27, 2024.
- [268] N. Yadav and A. Bhatt, “A Secure IoT Communication against Reactive Jamming Attack in CRN Subtitle,” vol. 5, pp. 1543–1549, 2019.
- [269] M. Wei, C. Rong, E. Liang, and Y. Zhuang, “An intrusion detection mechanism for IPv6-based wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 18, no. 3, p. 155013292210779, 2022. [Online]. Available: <https://doi.org/10.1177/15501329221077922>

- [270] K.-H. Le, M.-H. Nguyen, T.-D. Tran, and N.-D. Tran, “IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT,” *Electronics*, vol. 11, no. 4, p. 524, 2022.
- [271] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, “The Contiki-NG open source operating system for next generation IoT devices,” *SoftwareX*, vol. 18, p. 101089, 2022.
- [272] T. Antoine-Santoni, J. Santucci, E. De Gentili, and B. Costa, “Discrete Event Modeling and Simulation of Wireless Sensor Network Performance,” *SIMULATION*, vol. 84, no. 2-3, pp. 103–121, Feb. 2008. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0037549708091641>
- [273] R. Estepa, A. Estepa, G. Madinabeitia, and E. Garcia, “RPL Cross-Layer Scheme for IEEE 802.15.4 IoT Devices With Adjustable Transmit Power,” *IEEE Access*, vol. 9, pp. 120 689–120 703, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9523554/>
- [274] A. C. Smit, E. Schat, and E. Ceulemans, “The Exponentially Weighted Moving Average Procedure for Detecting Changes in Intensive Longitudinal Data in Psychological Research in Real-Time: A Tutorial Showcasing Potential Applications,” *Assessment*, vol. 30, no. 5, pp. 1354–1368, Jul. 2023. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/10731911221086985>
- [275] M.-A. E. Houssaini, A. Aaroud, A. E. Hore, and J. Ben-Othman, “Detection of Jamming Attacks in Mobile Ad Hoc Networks Using Statistical Process

- Control,” *Procedia Computer Science*, vol. 83, pp. 26–33, 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1877050916301181>
- [276] H. Sounni, E. K. Najib, and L. Fatima, “DISTRIBUTED DENIAL OF SERVICE ATTACKS DETECTION USING STATISTICAL PROCESS CONTROL IN CENTRALIZED WIRELESS NETWORKS,” vol. 17, 2022.
- [277] J. S. Oakland, *Statistical process control*, 5th ed. Oxford ; Boston: Butterworth-Heinemann, 2003, oCLC: ocm50302293.
- [278] K. Dunn, *Process Improvement Using Data*, 2023, accessed: December 18, 2024. [Online]. Available: <https://learnche.org/pid/PID.pdf?10d109>
- [279] P. K. Carson and A. B. Yeh, “Exponentially weighted moving average (ewma) control charts for monitoring an analytical process,” *Industrial & engineering chemistry research*, vol. 47, no. 2, pp. 405–411, 2008.
- [280] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, “Exponentially weighted moving average charts for detecting concept drift,” *Pattern recognition letters*, vol. 33, no. 2, pp. 191–198, 2012.
- [281] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2007.
- [282] J. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol, *Discrete-event system simulation*, fifth edition, pearson new international edition ed., ser. Always learning. Harlow: Pearson, 2014.

- [283] “Z1 datasheet,” <https://zolertia.sourceforge.net/wiki/index.php/Z1>, 2013.
- [284] D. Hartvigsen, *Simquick, Process Simulation With Excel*. Prentice-Hall, Inc., 2004.
- [285] C. R. Souza, “Accord.NET Machine Learning Framework,” Dec. 2014. [Online]. Available: <http://accord-framework.net/>
- [286] N. Dalwadi and M. Padole, “Performance analysis of wireless motes in iot,” in *ICT Analysis and Applications: Proceedings of ICT4SD 2022*. Springer, 2022, pp. 383–397.
- [287] H. S. Zenalabdin, A. Buhari, and T. E. Nyamasvisva, “Performance Analysis of IoT Protocol Stack over Dense and Sparse Mote Network Using COOJA Simulator,” *J. Phys.: Conf. Ser.*, vol. 1529, no. 5, p. 052007, May 2020. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1529/5/052007>
- [288] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, “Iot intrusion detection taxonomy, reference architecture, and analyses,” *Sensors*, vol. 21, no. 19, 2021.
- [289] D. Ray, P. Bhale, S. Biswas, P. Mitra, and S. Nandi, “A Novel Energy-Efficient Scheme for RPL Attacker Identification in IoT Networks Using Discrete Event Modeling,” *IEEE Access*, vol. 11, pp. 77 267–77 291, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10185920/>

- [290] Z. Shah, A. Levula, K. Khurshid, J. Ahmed, I. Ullah, and S. Singh, “Routing Protocols for Mobile Internet of Things (IoT): A Survey on Challenges and Solutions,” *Electronics*, vol. 10, no. 19, p. 2320, Sep. 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/19/2320>

VITAE

- Name: Hussah Ali Hussain Albinali
- Nationality: Saudi
- Date of Birth: June, 9th, 1985
- Email: *h.a.albinali@gmail.com*
- Permenant Address: 5139, Safwa Street, Shualah Dist., Dammam, 34361 Saudi Arabia
- **Academic background:**
 - **2019-2024:** PhD (Computer Science), King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.
 - **2014-2016:** MS (Computer Science), Georgia State University, Atlanta, United States.
 - **2003-2006:** B.Sc. (Computer Science), King Faisal University, Dammam, Saudi Arabia.

- **Teaching experience:**

- **2017-2020:** Lecturer, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
- **2008-2014:** Teaching Assistant, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

- **Journal papers:**

- Albinali, H., & Azzedin, F. (2024). Replay attacks in RPL-based Internet of Things: Comparative and empirical study. *Computer Networks*, 110996.
- Albinali, H., & Azzedin, F. (2024). Towards RPL Attacks and Mitigation Taxonomy: Systematic Literature Review Approach. *IEEE Transactions on Network and Service Management*.
- Alghofaili, R., Albinali, H., & Azzedin, F. (2024). Build–Launch–Consolidate Framework and Toolkit for Impact Analysis on Wireless Sensor Networks. *Journal of Sensor and Actuator Networks*, 13(1), 17.

- **Patents:** Azzedin, F., & Albinali, H. (2023). System And Method to Detect and Countermeasure RPL Attacks In IoT Network (Patent No. 547185US). OBLON, Mclelland, MAIER & Neustadt, L.L.P. Status: “Green Light”.

- **Conferences papers:**

- Albinali, H., Azzedin, F., & Riaz, M. (2024). DEVS-RPL: Design Formal Discrete Event Model of Routing Protocol over LLN. *Proceedings of the 2024 IEEE Cybermatics Congress*.

- Albinali, H., Alghofaili, R., & Azzedin, F. (2023, December). Analytical Study of Local Repair Attack on RPL-Based 6LoWPAN Networks in Internet of Things. In Proceedings of the 7th International Conference on Future Networks and Distributed Systems.
- Azzedin, F., & Albinali, H. (2021, December). Security in Internet of Things: RPL attacks taxonomy. In Proceedings of the 5th International Conference on Future Networks and Distributed Systems.
- Albinali, H., & Alzahrani, F. A. (2021, March). Faster R-CNN for detecting regions in human-annotated micrograph images. In 2021 International Conference of Women in Data Science at Taif University (WiDSTaif) (pp. 1-6). IEEE.
- Albinali, H., Alharbi, M., Alharbi, R., & Aljabri, M. (2022, December). Synchronization Techniques for Multi-threaded Web Server: A Comparative Study. In 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 373-379). IEEE.
- Albinali, H., Han, M., Wang, J., Gao, H., & Li, Y. (2016, December). The roles of social network mavens. In 2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN) (pp. 1-8). IEEE.