ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

NEW LIGHTWEIGHT DoS ATTACK MITIGATION TECHNIQUES FOR RPL BASED IoT NETWORKS

Ph.D. THESIS

Ahmet ARIŞ

Department of Computer Engineering

Computer Engineering Programme

APRIL 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

NEW LIGHTWEIGHT DoS ATTACK MITIGATION TECHNIQUES FOR RPL BASED IoT NETWORKS

Ph.D. THESIS

Ahmet ARIŞ
(504122501)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Sema F. OKTUĞ

APRIL 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

RPL TEMELLİ IoT AĞLARI İÇİN DoS SALDIRILARININ ETKİSİNİ AZALTACAK YENİ TEKNİKLER

DOKTORA TEZİ

Ahmet ARIŞ
(504122501)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Sema F. OKTUĞ

NİSAN 2019

Ahmet ARIŞ, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 504122501 successfully defended the thesis entitled "NEW LIGHTWEIGHT DoS ATTACK MITIGATION TECHNIQUES FOR RPL BASED IoT NETWORKS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Sema F. OKTUĞ** İstanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. S. Berna ÖRS YALÇIN** İstanbul Technical University	.....
	Prof. Dr. Fatih ALAGÖZ Boğaziçi University	.....
	Prof. Dr. Bülent ÖRENCİK Beykent University	.....
	Assoc. Prof. Dr. Feza BUZLUCA İstanbul Technical University	.....

Date of Submission : **13 March 2019**
Date of Defense : **30 April 2019**

To the people who give meaning to my life,

FOREWORD

I owe a debt of gratitude to several people who have been supporting me not only during this thesis study, but also my entire life. I always felt very lucky to have met these gorgeous people. However, I might have forgotten to thank to some people here and I do apologize for it beforehand.

First of all, I would like to thank to my advisor Prof. Dr. Sema F. Oktuğ for her endless support. I have always been feeling very lucky to be a PhD student of her. She was always very kind, understanding, friendly and full of energy. She is definitely a role model for me, not only as an advisor, but also as a researcher, instructor, manager, friend and parent. I will always remember our lovely conversations and especially her chocolates that she generously shared with me. I really hope that, I will make her proud of my achievements in the future like her earlier bright students.

I would like to thank to my co-advisor Assoc. Prof. Dr. S. Berna Örs Yalçın for her support since my masters degree. When I met her for the first time, I literally knew nothing and all along those years she shared her knowledge and experience with me. I am grateful to her that she helped me to improve my thesis studies with her comments. She would always tell me "hep dört ayak üstüne düşüyorsun" to express my luck. I hope I will have chance to hear the same saying from her for many more years.

I also would like to thank to my thesis progress committee members, namely Prof. Dr. Bülent Örencik and Assoc. Prof. Dr. Feza Buzluca for their valuable comments and insights, which were definitely helpful for me. It was really important for me to hear their feedback on my studies. Although my thesis finishes now, I hope I will still be their student who keeps on learning from them throughout his career.

During my thesis, I had chance to spend three months in Swedish Institute of Computer Science (SICS) as a visiting researcher. This was made possible thanks to Prof. Dr. Thiemo Voigt and Dr. Shahid Raza who accepted me to conduct research at SICS, my advisor Prof. Oktuğ's support and financial support from COST ACROSS project. I would like to thank to Thiemo and Shahid for accepting me to SICS and to my advisor and COST ACROSS for their support. Spending three months with full of research at SICS was a wonderful experience for me. Thiemo was really friendly and helpful and certainly made my visit better there. I also would like to thank to our neighbor Şerife Yavuz in Kista. She was like our helpful angel starting from our first night in Sweden till the last. I will always be appreciative.

This study was supported by 2211C - Domestic Doctoral Scholarship Program Intended for Priority Areas, No. 1649B031503218 of the Scientific and Technological Research Council of Turkey (TUBITAK) and also by the Research Fund of Istanbul Technical University (Project Number:39182).

I would like to thank to Assoc. Prof. Dr. Berk Canberk. He was a great instructor and a friend for me. I really enjoyed his classes and being his teaching assistant. I have always admired his energy and his research motivations. It is wonderful to work with him right now and I'm learning a lot from him. I'm thankful to him that he introduced

me to Medianova during the last six months of my PhD. I owe thanks to Medianova that they supported me coming to university once a week to complete my PhD.

During my PhD, I started to attend classical guitar lessons at İTÜ Savaş Çekirge Classical Guitar Education and Research Center. Although I could not continue to the lessons, two people's company kept me being a part of the center. These people are Prof. Dr. Nevin Çekirge and Guitarist Haşim Polat. Prof. Çekirge was like a mother for me. She never stopped supporting me. We shared many things, many moments when we were organizing international classical guitar festivals at our university. I learned a lot from her. I'm appreciative for knowing her. I also would like to thank to my guitar teacher Haşim Polat. He was an excellent teacher for me. He was always very friendly and caring. I spent great time whenever we met somewhere and had a night full of guitar. I will always remember spending really lovely times in the garden of Faculty of Mines with Prof. Çekirge and Guitarist Polat.

I would like to thank to the whole members of Faculty of Computer and Informatics Engineering including all workers, research and teaching assistants, professors. I always felt grateful to work in such a nice place with such nice people. I would like to thank my close friends in our faculty, namely Müge Erel-Özçevik, Gözdegül İşgüder-Şahin, Halil Gülaçar, Yusuf Hüseyin Şahin, Elham Dehghan, Kübra Adalı, Tuğçe Bilen, Emir Erdem and Yusuf Özçevik. I shared a lot with them. I am hoping that our friendship will last a lifetime with them. I will always remember the times we spent together. I will especially remember Müge with her unique way of constructing Turkish sentences and pronouncing sounds from the nature, Gözde with meat lunches at our university dining hall, Halil and Yusuf with our Friday studies, Kübra with her after lunch chocolates, Elham with our jokes together, Tuğçe with "ya ne gerek var", Emir with "vice major's daughter" and Yusuf Özçevik with our conversations.

I would like to utter the names of İshak and Hamide Demirbağ, Rahmi Koç, Selim Durak, Mahmut Uçak, Ahmet Kızılca and Burhan Boz whom I met during my stay at the accommodation place at İnce Minareli Cami, Merter. The first three people very extremely helpful, friendly and caring. I will always remember the chicken pie that Hamide abla baked for us. It was great to meet these people and be a friend of them. I'm grateful to each of them.

I also would like to thank to Gönül ve Onur Temizgezek who were like a second family for me in İstanbul. We shared a lot together. Their friendship is very valuable for me that I would always like to keep in touch with them, celebrate Onur's achievements together and see them happy all the time.

Çaçaron Family in Fethiye supported me during my high school, BSc. and MSc. degrees. I have always been appreciative of all the support they have given to me. If I had chance to be a computer engineer, have a masters degree and PhD now, it would not be possible without their help all those years. I would specially like to thank Uğur Çaçaron who has always been like a brother and a friend to me. I always enjoyed his conversations, jokes and his books.

I would also like to thank to my teachers Gürzat Kışlak, Yücel Erdoğan and Ozan Özel and his lovely wife Belda Özel. Gürzat hoca loved me like his son and always supported me. Yücel Erdoğan was the one who lighted the fire in me to start and love reading. Ozan Özel was a wonderful teacher and friend, who gave me the best math private lessons in Fethiye and his wife, Belda Özel was like a sister for me. If I was

going to their house for lessons every week, that was also due to her pastries and meals she would generously offer me.

I would like to thank my close friends, namely Mehmet Soybalı, Adem Zeytuncü and Şaban Tombak. Our friendship with Adem and Şaban started during our childhood and we always enjoyed each other's company whenever we had chance to come together. I'm really happy to have them as my friends. I met Mehmet during masters and we became close friends soon. I really admire his friendship. He is the kind of friend who is honest, who has targets and who with I can share my ideas, beliefs, hopes and fears in any topic. I feel very lucky to have them as my close friends.

Lastly I want to thank my family: my brothers Aziz and Osman, my father Ramazan, my mother Ayşe and my dear love Mairi Jean Wicker. I think whatever I write here to express how thankful I am to each of them would not be enough. I can not imagine a life without them. My brothers have always been with me. My father and mother have always been so understanding, so friendly and caring for me. And my dear love, Mairi has been in my life more than 10 years now. We shared and experienced a lot together. I'm feeling so lucky to be her partner and thinking that my biggest achievement up until now was this. I hope this achievement will last all along my lifetime.

April 2019

Ahmet ARIŞ
(R&D Analyst)

TABLE OF CONTENTS

Page

ABBREVIATIONS

6LoWPAN : IPv6 over Low-Power Wireless Personal Area Networks
BLE : Bluetooth Low Energy
BR : Border Router
CBOR : Concise Binary Object Representation
CCA : Clear Channel Assessment
CoAP : Constrained Application Protocol
CPU : Central Processing Unit
CSMA : Carrier Sense Multiple Access
DAO : Destination Advertisement Object
DIO : DODAG Information Object
DIS : DODAG Information Solicitation
DODAG : Destination Oriented Directed Acyclic Graph
DoS : Denial of Service
DDoS : Distributed Denial of Service
D/DoS : DoS or DDoS
DTLS : Datagram Transport Layer Security
GTS : Guaranteed Time Slot
ICMPv6 : Internet Control Message Procotocol version 6
IDS : Intrusion Detection System
IEEE : The Institute of Electrical and Electronics Engineers
IETF : The Internet Engineering Task Force
IoT : Internet of Things
IPSec : Internet Protocol Security
IPv6 : Internet Protocol version 6
LLN : Low Power and Lossy Network
LoRa : Longe Range
LoRaWAN : Longe Range Wide Area Network
LTE : Long Term Evolution
MAC : Medium Access Control
NB : Narrow Band
OS : Operating System
PHY : Physical Layer
QoS : Quality-of-Service
RAM : Random Access Memory
RF : Radio Frequency
ROM : Read-Only Memory
RPL : IPv6 Routing Protocol for Low Power and Lossy Networks
UDP : User Datagram Protocol
VN : Version Number
VNA : Version Number Attack
WiFi : Wireless Fidelity
WSN : Wireless Sensor Network

SYMBOLS

C (x,y)	**:** Combination
KB	**:** Kilobytes
p	**:** Attacking Probability
FN	**:** False Negative
FP	**:** False Positive
ms	**:** Millisecond
mW	**:** Milliwatt
TN	**:** True Negative
TP	**:** True Positive

LIST OF TABLES

Page

LIST OF FIGURES

Page

NEW LIGHTWEIGHT DoS ATTACK MITIGATION TECHNIQUES FOR RPL BASED IoT NETWORKS

SUMMARY

The information technology is converting every aspect of human life to smart day by day. Currently, the Internet of Things (IoT) is the most promising technology in accordance with Machine Learning, Big Data Analytics, Cloud and Edge Computing to realize such a change. IoT, with its billions of *things*, acting as the primary data source, is and will continue to be playing a crucial role to transform buildings, houses, factories, cities, suburban areas and many places to Internet-connected smart environments.

IoT can be defined as a network of *things* with sensing and/or actuating capabilities that can connect to the Internet and exchange information. The elements of the IoT include but not limited to, sensors, actuators, embedded and wearable devices. These devices form networks with different topologies (e.g., star, mesh) using various communication technologies (e.g., IEEE 802.15.4, Bluetooth, WiFi, LoRa, cellular, etc.).

IoT is a candidate technology in order to realize the future Internet of Services and Industry 4.0 revolution. However, there are serious threats for IoT, which aim to degrade the performance of the network, deplete the batteries of the devices and cause packet losses and delays. These attacks are called as Denial of Service (DoS) attacks, which are already notorious for their effects in existing communication systems. Limited power, processing, storage and radio dictate extremely efficient usage of these resources to achieve high reliability and availability in IoT. However, DoS and Distributed DoS (DDoS) attacks aim to misuse the resources and cause interruptions, delays, losses and degrade the offered services in IoT. For highly reliable and available IoT, such attacks have to be prevented, detected or mitigated autonomously.

The goal of this thesis is to investigate how IoT networks can be secured against D/DoS attacks. For this aim, we focused on a subclass of IoT, namely the Low Power and Lossy Networks (LLNs). In LLNs, elements of the network are resource-constrained (i.e., limited processing, storage, communication and power sources) and the environment is lossy. The IETF and the IEEE proposed several standards to connect such networks to the Internet and allow various applications to be supported efficiently. The set of protocols are called as the *standardized protocol stack*, which includes the IEEE 802.15.4 for PHY and MAC layer, 6LoWPAN adaptation layer between MAC and network layers, IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) for network layer, UDP for transport layer and CoAP for application layer. We considered to study LLNs, because we believe that majority of the devices in IoT will be within the concept of LLNs. Limited resources, lossy environments and possibly mobility make such networks challenging to incorporate robust security solutions. Considering these devices are capable of connecting to the Internet, the task of protecting such networks against attackers gets more challenging.

In this respect, we started our thesis journey with understanding the LLNs with its architectural design, elements and communication protocols. As we study the security of the LLNs, we analyzed the standardization efforts to secure the LLN-based IoT networks. Considering the pros and cons of the security solutions, resource-constraints, QoS requirements, implementation challenges, deployment environments and users, the Internet connectivity and the D/DoS attacks targeting the existing computer networks, we believed that there is a need for the LLNs to be secured against such attacks. Hence, we researched the D/DoS attacks which may target the LLNs. Our research revealed that, the LLNs can be the target of D/DoS attacks which can be either new (due to new protocols and standards) or old (attacks that have been targeting the existing networks). We analyzed the works that investigate the effects of the attacks, cryptography-based security solutions, protocol security studies and intrusion detection and mitigation systems that are specifically designed for the LLNs.

We continued our journey with focusing on the network layer of the LLNs and specifically to the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) that was proposed by the IETF. Among the set of attacks, we targeted the Version Number Attack (VNA). In VNA, an attacker maliciously changes the VN, thus initiates an illegitimate global repair operation in RPL. However, in an ordinary RPL network, global repair operation can be started only by the DODAG root. Our motivation to choose the VNA among the other attacks were manifold. First of all, it was a pretty new DoS attack and it was not studied much in the literature. Since it was new, there was no detection or mitigation study or any in-depth analysis. It was also an interesting attack for us that, it did not need thousands of attackers, but only a single attacker was enough to affect the whole network.

In order to analyze the effect of the VNA, we created a realistic topology considering the IETF's routing requirements documents for the LLNs. Using the Contiki Cooja, we simulated a single attacker at several positions within the RPL DODAG. Our topology had mobile nodes and hence, mobile attackers. We employed a probabilistic attacker model to identify the best setting for an attacker, not only in terms of the position, but also the attacking probability. Our analysis with respect to packet delivery ratio, RPL control message overhead, average network delay and average power consumption metrics showed that, control message overhead and packet delivery ratio results are highly correlated to the position of the attacker. Effect of the attack gets stronger as the attacker gets farther from the DODAG root. In terms of the attacking probability and the effect of the attack, we figured out that, performance of the network gets worse as the probability of attacking increases. Simulation results for mobile attackers show that, mobile attackers have nearly the same detrimental effects on the network as the best attacking positions for a static attacker.

Our analysis with a single attacker revealed that, the VNA is a very serious threat for the RPL-based LLNs. It misuses the global repair operation of the RPL protocol and forces the complete network to be rebuilt over and over again. During this time, an excess amount of RPL control messages are generated, majority of the application data packets are lost, average network delay and power consumption are severely increased. Based on our observations for the strongest attacking positions and the ordinary global repair operation explained in the RPL specification, we came up with a simple mitigation mechanism for VNA, namely *elimination*, that allows the VN updates coming from the direction of the DODAG root and stops the VN updates

coming from the opposite direction (i.e., from the direction of leaf nodes). Elimination technique checks the sender of the incoming DIO (DIO with a greater VN). If the sender has a worse rank than the receiver node, which shows that it is positioned at the lower portions of the DODAG, the DIO is discarded since such an update should not come from that portions of the DODAG. We believed that the elimination technique would be very efficient to mitigate the effect of the VNA consisting of a single attacker. Since it eliminates the VN updates started from the direction of leaf nodes (positions at which VNA is the strongest) towards the DODAG root, the strongest attacking positions could be eliminated by this way. However, if the attacker is not far from the root, in other words, if the attacker is an intermediate node within the DODAG, then it can still affect the lower portions of the DODAG since the elimination technique will allow it. In order to overcome this issue and propose a complete solution, we came up with *shield* mechanism that is based on the elimination technique. In shield, every node works on its own and keeps a list of its neighbors that have a better rank value than itself in a table, which we call as *ShieldList*. If it receives a DIO with a greater VN (which signifies that a VN update is taking place), it firstly checks whether the DIO sender is in its ShieldList or not. If it does not exist within the table, then it discards the DIO. Because, such a DIO can come only from the direction of the DODAG root and possible senders are already listed in the ShieldList table. However, if the DIO sender exists within the table, then comes a trust mechanism, in which the receiving node expects other neighboring nodes (nodes within the ShieldList table) to announce such an update. It waits until majority of the ShieldList entries claim a VN update. Hence, it does not trust a single neighbor claiming a VN update, but trusts more neighbors announcing the same update. Thus, it can provide a solution not only against the attackers at the leaf nodes, but also attackers at the intermediate positions. We analyzed the performance of our new mitigation techniques by means of simulations. Simulation results show that, our novel mitigation techniques can effectively mitigate the effect of the VNA.

In the last part of our thesis study, we focused on multiple VN attackers. We performed two sets of comparisons. In the first set of comparisons, we investigated the effect of increasing the number of attackers. In the second set of comparisons, we analyzed the effect of multiple attackers positions. Simulation results showed that, increasing the number of attackers affects only the packet delivery ratios. In terms power consumption per successfully delivered packets, the lowest values were obtained for the positions that are at the edges of the topology. In terms of distance to the DODAG root, closer positions seem to cause more power consumption per successfully delivered packets values than the farther positions. For packet delivery ratio and average network delay results, attackers at the center of the network caused the highest packet losses and longest delays. These results were interesting for us since our earlier study and a predecessor study had not realized such affects. Therefore, we conducted analysis to find out the underlying reasons. We considered topological differences, logging mechanisms and operating system changes. We concluded that, OS updates are the underlying reason for us to obtain different results for an RPL network under the VNA. Finally, we evaluated the performance of our mitigation techniques against multiple attackers based on packet delivery ratio results. We found out that our mitigation techniques successfully mitigate the effect of multiple attackers.

RPL TEMELLİ IoT AĞLARI İÇİN
DoS SALDIRILARININ ETKİSİNİ AZALTACAK YENİ TEKNİKLER

ÖZET

Bilişim teknolojileri günden güne insan hayatının her alanını akıllı bir hale getiriyor. Bu değişimin gerçekleşmesinde Nesnelerin İnterneti (Internet of Things - IoT) şu anda en umut vadeden teknoloji olarak Makine Öğrenmesi, Büyük Veri Analitiği, Bulut ve Sınırda Hesaplama teknolojileriyle iş birliği içinde bulunuyor. IoT milyarlarca *nesne*siyle bu değişimde en temel veri kaynağı olarak görev alıyor. Binaları, evleri, fabrikaları, şehirleri, kırsal alanları ve daha birçok ortamı internete bağlanan akıllı ortamlara dönüştürmekte hem şu anda, hem de gelecekte aktif bir rol alacak gibi görünüyor.

IoT, algılama ve/veya eyleme yeteneği olan nesnelerin oluşturduğu, internete bağlanabilen ve bilgi alışverişi yapabilen bir ağ olarak tanımlanabilir. Bunlarla sınırlı olmamakla beraber duyargalar, eyleyiciler, gömülü ve giyilebilen cihazlar IoT'nin temel bileşenlerini oluşturmaktadır. Bu cihazlar birçok iletişim teknolojisi (IEEE 802.15.4, Bluetooth, WiFi, LoRa, hücresel ağlar, vb.) kullanarak farklı topolojilerde (yıldız, örgü) IoT ağları oluşturabilirler.

IoT geleceğin Servislerin İnterneti'ni ve Endüstri 4.0 devrimini gerçekleştirmek için aday bir teknolojidir. Ancak IoT'ye yönelik ağ performansını düşürmeyi, cihazların pillerini tüketmeyi, paket kayıplarına ve gecikmelere neden olmayı hedefleyen çok ciddi tehditler bulunmaktadır. Bu tehditler halihazırdaki ağları ve sistemleri etkileyen ve adından sıkça söz ettiren Servis Engelleme Saldırıları'dır (Denial of Service - DoS). IoT cihazlarının büyük bir bölümünde güç, işleme, bellek ve haberleşme kaynaklarının kısıtlı olması bu kaynakların daha güvenilir ve her zaman erişilebilir IoT ağları için mümkün olduğunca verimli kullanılmalarını şart koşmaktadır. Fakat DoS ve Dağıtık DoS (DDoS) saldırıları kaynakları kötücül kullanmayı, hizmet kesintilerine, gecikmelere ve paket kayıplarına neden olmayı ve böylece IoT ağlarının verdikleri hizmetlerin performansını düşürmeyi amaçlamaktadır. Yüksek seviyede güvenilir ve erişilebilir IoT ağları için bu saldırıların otonom bir şekilde engellenmesi, tespit edilmesi ya da etkilerinin azaltılması gerekmektedir.

Bu tezin amacı D/DoS saldırılarına karşı IoT ağlarının nasıl korunabileceğini araştırmaktır. Bu amaçla tezde IoT ağlarının bir alt sınıfı olan Düşük Güçlü ve Kayıplı Ağlar'a (Low Power and Lossy Networks - LLNs) odaklandık. Bu tür ağlarda cihazlar kaynak kısıtlarına (kısıtlı işlem birimi, bellek ve depolama, iletişim arayüzü ve güç kaynakları) sahip olmakta ve iletişimin gerçekleştiği ortamda paket kayıpları yaşanabilmektedir. IETF ve IEEE organizasyonları böyle ağları internete bağlayabilmek ve birçok uygulamanın verimli bir şekilde çalışmasını sağlayabilmek için bir dizi standart önermişlerdir. Bu standartlar kümesine *standartlaşmış protokol yığını* adı verilmektedir. Sırasıyla fiziksel katman ve MAC katmanı için IEEE 802.15.4, MAC ve ağ katmanları arasına IETF 6LoWPAN adaptasyon katmanı,

ağ katmanı için IETF Düşük Güçlü ve Kayıplı Ağlar için IPv6 Yönlendirme Protokolü (IPv6 Routing Protocol for Low Power and Lossy Networks - RPL), iletim katmanı için UDP ve uygulama katmanı için de IETF Kısıtlı Uygulama Protokolü (Constrained Application Protocol - CoAP) standartlaşmış protokol yığınını oluşturmaktadır. Tez çalışmasında LLN'leri çalışmayı düşündük. Çünkü IoT'yi meydana getirecek cihazların büyük bir bölümünün LLN sınıfını oluşturan cihazlar olacağına inanıyoruz. Bu tür ağlarda kaynakların kısıtlı olması, iletişim ortamının kayıplı olması ve cihazların hareketli olabilmesi böyle ortamlarda çalışmayı ve sağlam güvenlik çözümleri sunabilmeyi zorlaştırmaktadır. Bu ağlardaki cihazların internete bağlanabileceği de düşünülürse, LLN'leri saldırganlara karşı korumak çok daha zor bir problem haline gelmektedir.

Bu bağlamda tez çalışmamız LLN'leri mimari yapıları, ağ bileşenleri ve iletişim protokolleri açılarından anlamayla başladı. LLN'lerin güvenliğini çalıştığımız için standardizasyon kurumlarının bu ağlar için önerdikleri güvenlik çözümlerini inceledik. Önerilen güvenlik çözümlerinin artı ve eksilerini, cihazların kaynak kısıtlarını, servis kalitesi (Quality of Service - QoS) gereksinimlerini, güvenlik çözümlerinin doğru gerçeklenmelerinde karşılaşılan sorunları, cihazların çalışma ortamlarını, kullanıcıları, internet bağlantısı yoluyla ağın dışarıya açılabilmesini ve halihazırdaki D/DoS saldırılarını göz önüne aldığımızda LLN'lerin saldırılara karşı korunmalarının bir ihtiyaç olduğuna inandık. Buradan yola çıkarak LLN'leri hedef alabilecek D/DoS saldırılarını araştırdık. Araştırmalarımız bu ağların hem yeni saldırıların (bu ağlar için önerilen yeni standartlar nedeniyle), hem de bilinen saldırıların (halihazırdaki ağlara yapılan D/DoS saldırıları) hedefi olabileceğini ortaya çıkardı. LLN'leri hedef alabilecek saldırıların etkilerini inceleyen çalışmaları, bu saldırılardan korunmak için kriptografi-temelli güvenlik çözümleri öneren çalışmaları, standartlaşmış protokol yığınındaki protokollerin güvenlik açıklarını kapatmayı hedefleyen çalışmaları, LLN'ye özel saldırı tespit ve etki azaltma çalışmalarını analiz ettik.

LLN'leri hedef alabilecek saldırıları ve bu saldırılara karşı koymayı hedefleyen çalışmaları inceledikten sonra standartlaşmış protokol yığınının ağ katmanına odaklandık. Buradaki odak noktamız RPL yönlendirme protokolüydü. RPL'i hedef alan saldırıların arasından Versiyon Numarası Saldırısı'na (Version Number Attack - VNA) yöneldik. VNA'da saldırgan kötücül bir şekilde RPL ağının versiyon numarasını değiştirerek global onarım işlemini başlatır. Ancak normal bir RPL ağında bu işlemi sadece RPL ağını kuran ve Hedef-Odaklı Yönlü Çevrimsiz Çizge'nin (Destination Oriented Directed Acyclic Graph - DODAG) kökü olan düğüm gerçekleştirebilmektedir. RPL'i hedef alan birçok saldırının arasından VNA'yı seçmemizin arkasında birden fazla neden bulunuyordu. Bunlardan ilki, VNA'nın oldukça yeni bir DoS saldırısı olması ve üzerinde çok az çalışılmış olmasıydı. Yeni bir saldırı olması nedeniyle de henüz derinlemesine bir analiz çalışması, bu saldırıya yönelik bir tespit sistemi ya da etkisini azaltmayı hedefleyen bir çalışma bulunmamaktaydı. VNA'nın bize ayrıca ilginç gelen diğer bir yönü de, klasik saldırıların aksine, tek bir saldırgan düğümün bu saldırıyı gerçekleştirmek için yeterli oluşu ve bütün bir RPL ağını tek başına etkileyebilme şansına sahip olmasıydı.

VNA'nın RPL üzerindeki etkisini inceleyebilmek için IETF'in yönlendirme protokol gereksinimleri dökümanlarından yola çıkarak gerçekçi bir benzetim topolojisi oluşturduk. Contiki işletim sisteminin Cooja benzetim ortamını kullanarak topoloji üzerinde tek bir saldırganı farklı pozisyonlara yerleştirerek benzetimler yaptık.

Oluşturduğumuz topolojide hareketli düğümler de yer aldığı için bazı saldırı benzetimlerinde saldırganı hareketli olan düğümlerden de seçtik. Saldırgan modelimizi oluştururken olasılığa göre saldırı gerçekleştiren bir yapı geliştirdik. Böylelikle saldırgan açısından sadece en iyi saldırı pozisyonlarını değil, saldırı olasılıklarını da belirlemeyi hedefledik. Paket başarım oranı, RPL kontrol mesajları sayısı, ortalama ağ gecikmesi ve ortalama güç tüketimi metriklerine göre analizler yaptığımızda kontrol mesajları sayısı ve paket başarım oranının saldırı pozisyonuyla ilişkili olduklarını gördük. Saldırgan DODAG kök düğümünden uzaklaştıkça saldırının etkisinin arttığını anladık. Saldırı olasılığı açısındansa olasılık arttıkça, ağ performansının daha da kötüleştiğini gördük. Saldırının hareketlilikle ilişkisini incelediğimizde ise hareketli saldırganların en etkili hareketsiz saldırganlar kadar ciddi bir şekilde ağ performansını etkilediklerini gözlemledik.

Tek saldırgan ile yaptığımız benzetimlerin sonuçları VNA'nın RPL-temelli LLN'ler için çok ciddi bir tehdit olduğunu ortaya çıkardı. Bu saldırıda global onarım mekanizması kötücül bir şekilde kullanılıyor ve tüm RPL ağı tekrar ve tekrar yeniden oluşturulmaya zorlanıyor. Bu sıradaysa aşırı derecede çok RPL kontrol mesajı üretiliyor, uygulamanın paketlerinin büyük bir bölümü kayboluyor, ortalama ağ gecikmesi ve güç tüketimi değerleri kötü bir şekilde etkileniyor. En etkili saldırı pozisyonları ve RPL dökümanında belirtilen global onarım mekanizmasının beklenen işleyişini göz önüne alarak basit bir saldırı etkisi azaltma yöntemi geliştirdik. *Eleme* (Elimination) adını verdiğimiz bu yöntem sadece DODAG kökü tarafından gelen version numarası güncellemelerine izin verirken (RPL dökümanında belirtilen beklenen global onarım işleyişi), saldırının en etkili olduğu tam tersi yönden (DODAG yaprakları yönü) gelen güncellemelerine ise izin vermiyor. Bu amaçla şu andakinden daha yüksek bir version numarasıyla gelen DIO (RPL kontrol mesajı) mesajlarında gönderenin *sıra*sına (rank) bakıyor ve versiyon numarası güncellemesinin yönünü kolaylıkla tespit edebiliyor. Eğer gönderen düğüm daha kötü bir sıra değerine sahipse, bu onun DODAG üzerinde daha altlarda, yapraklar tarafında, olduğunu belirtiyor. Böyle versiyon güncellemelerine izin verilmemesi gerekiyor, çünkü RPL dökümanına göre böyle güncellemeler sadece DODAG kökü tarafından, yani sıra bilgisi daha iyi olan düğümlerden gelebilir. Eleme tekniğinin saldırının en etkili olduğu noktalardan gelecek saldırıları engellediği için VNA için oldukça etkili bir çözüm olacağına inandık. En etkili saldırı pozisyonlarının etkisi eleme tekniğiyle azaltılsa da saldırganın ağın iç kesimlerinde olduğu durumlarda, ne kadar en etkili saldırı noktaları olmasa da, saldırı hala RPL ağını olumsuz yönde etkileyebilir. Dolayısıyla eleme yöntemi saldırının etkisini azaltsa da tam bir çözüm oluşturamayacağı ortadaydı. VNA için tam bir çözüm bulabilmek için eleme tekniğini baz alan *kalkan* (shield) adını verdiğimiz yeni bir yöntem geliştirdik. Kalkan'da her düğüm diğerlerinden bağımsız bir şekilde hareket ediyor ve çevresinde bulunan ve ondan daha iyi bir sıra değerine sahip olan düğümleri Kalkan Tablosu'nda tutuyor. Bu tablodaki düğümler kendisine göre DODAG köküne daha yakın pozisyonda olan düğümleri belirtiyor. Dolayısıyla versiyon numarası daha yüksek bir DIO'yu sadece bu düğümlerden almayı, aldığında da sadece bir düğümün sözüne inanmayarak tablosundaki düğümlerin çoğunluğunun aynı güncellemeyi iddia etmesini bekliyor. Anlaşılacağı gibi Kalkan yöntemi eleme tekniği ile bir güven yöntemini birleştirerek VNA için tam bir çözüm olmayı amaçlıyor. Eleme tekniği ile saldırının en etkili pozisyonlarını eliyor, güven yöntemiyle de DODAG içinde yer alabilecek saldırıları da etkisizleştirmiş oluyor. Önerdiğimiz iki

yeni yöntemin verimliliğini benzetim ortamında yaptığımız testlerle inceledik ve bu yöntemlerin VNA'nın etkisini büyük bir başarımla azalttığını tespit ettik.

Tez çalışmamızın son bölümünde ise çok saldırganlı VNA'ya odaklandık. Çok sayıda saldırganların ağın performansını nasıl etkilediğini incelemek için iki ayrı analiz yaptık. İlk analizimizde saldırgan sayısının artmasının RPL'in performansını nasıl etkilediğini araştırdık. İkinci incelememizdeyse çok sayıda saldırganları, her bir saldırgan sayısını kendi içinde olmak üzere, pozisyon kombinasyonları açısından test ettik. Benzetimlerimizden elde ettiğimiz sonuçlar saldırgan sayısını arttırmanın sadece paket başarımını etkilediğini gösterdi. Saldırganların pozisyonları için yaptığımız testlerse başarı ile ulaştırılan paket başına harcanan güç açısından ağın köşelerinde (edge) yer alan saldırganların diğer saldırı pozisyonlarına göre daha az güç tüketimine neden olduğunu gösterdi. Ağın merkezinde yer alan saldırganların ise başarıyla ulaştırılan paket başına harcanan güç değerini diğer pozisyonlara göre daha çok arttırdığını gördük. Paket başarım oranı ve ortalama ağ gecikmesi metrikleri içinse sonuçlar ağın merkezinde yer alan düğümlerin daha çok paket kayıplarına ve daha uzun ortalama ağ gecikmelerine neden olduğunu gösterdi. Benzetimlerimizden elde ettiğimiz sonuçlar bizim için oldukça ilginç sonuçlardı. Çünkü tek saldırganlı benzetimlerimizden aynı yönde sonuçlar almamıştık. Bu sorunu çözebilmek adına bir dizi analiz gerçekleştirdik. Bu amaçla iki benzetimde kullanılan DODAG topoloji farklılıklarını inceledik, performans ölçümü için kullandığımız log alma yöntemine ve performans metriği farklılıklarına baktık ve işletim sistemi güncellemelerini analiz ettik. Tüm tetkiklerimiz alınan farklı sonuçların işletim sistemindeki güncellemelerden kaynaklandığını gösterdi. Son olarak bu tez çalışmasında önerdiğimiz Eleme ve Kalkan tekniklerini çok saldırganlı durumlar için paket başarım oranı açısından inceledik. Elde ettiğimiz sonuçlar bize önerdiğimiz tekniklerin çok saldırganlı durumlarda da saldırı etkisini başarıyla azalttığını gösterdi.

1. INTRODUCTION

The information technology is converting every aspect of human life to smart day by day. Currently, the Internet of Things (IoT) is the most promising technology in accordance with Machine Learning, Big Data Analytics, Cloud and Edge Computing to realize such a change. IoT, with its billions of *things*, acting as the primary data source, is and will continue to be playing a crucial role to transform buildings, houses, factories, cities, suburban areas and many places to Internet-connected smart environments [1].

IoT can be defined as a network of *things* with sensing and/or actuating capabilities that can connect to the Internet and exchange information. The elements of the IoT include but not limited to, sensors, actuators, embedded and wearable devices. These devices form networks with different topologies (e.g., star, mesh) using various communication technologies (e.g., IEEE 802.15.4, Bluetooth, WiFi, LoRa, cellular, etc.) [2].

IoT is a candidate technology in order to realize the future Internet of Services and Industry 4.0 revolution. Accommodation of billions of devices with sensing and/or actuation capabilities will introduce crucial problems with management, interoperability, scalability, reliability, availability and security. Autonomous control and reliability of future Internet of Services are directly related to reliability and availability of IoT. However, there are serious threats for IoT, which aim to degrade the performance of the network, deplete the batteries of the devices and cause packet losses and delays. These attacks are called as Denial of Service (DoS) attacks, which are already notorious for their effects in existing communication systems. Limited power, processing, storage and radio dictate extremely efficient usage of these resources to achieve high reliability and availability in IoT. However, DoS and Distributed DoS (DDoS) attacks aim to misuse the resources and cause interruptions, delays, losses and degrade the offered services in IoT. DoS attacks are clearly threats for availability and reliability of IoT. For highly reliable and available IoT, such attacks have to be prevented, detected or mitigated autonomously.

DoS and DDoS attacks can target any communication system and cause devastation. Such attacks make use of the vulnerabilities in the protocols, operating systems, applications and actual physical security of the target system. Readers can easily find several incident news related to D/DoS attacks on the Internet. These attacks are so common that every day it is possible to see them (e.g, please check the digital attack map of Arbor Networks and Google Ideas [3]). It is not hard to predict that IoT will face with D/DoS attacks, either as a target or source of the attacks. In fact, quite recently one of the major Domain Name System infrastructure provider of popular web sites and applications was the target of DDoS attacks where a botnet called as *Mirai* compromised thousands of cameras and digital video recorder players [4]. This incident was the first example of IoT being used as an attack source for DDoS. It clearly showed that, protection of IoT networks from attacks is not sufficient and protection of the Internet from IoT networks is needed as well.

A very interesting report [5] on how security of IoT will be playing an important role in defining the cybersecurity of future was published by UC Berkeley Center for Long-Term Cybersecurity in 2016. A group of people from various disciplines developed five scenarios regarding with what will security be like in the future considering various dimensions including people, governments, organizations, companies, society, culture, technological improvements and of course attackers. Although all of the scenarios are related to the security of IoT, the last two scenarios have direct relations. The fourth scenario puts the emphasis on the ubiquity of IoT in a way that IoT will be everywhere and will be playing a vital role on the management of several applications and systems. This will give attackers more chance to target. In such a world, attackers will able to affect organizations, governments and the daily life of people easier than now. Thus, cybersecurity term will be transformed to just *security* since it will able to affect everything. The last scenario considers the wearable devices and their novel purpose of use. According to the hypothesis, the wearables of future will not only perform basic measurement tasks, but will be used to track emotional states of humans. Advancements in the technologies will allow such a change. Emotional, mental and physical state information which is very important for individuals will be the target of attackers and will be used as a weapon against them. Of course in such a scenario, it will be very crucial for people to manage their

emotional, mental and physical state and this will affect the society in various ways which we can not imagine. This report clearly shows that if we fail to secure the IoT networks, then the ubiquity and proliferation of IoT will not transform the future to smarter but will cause catastrophic effects on human life, environment, culture and society.

Securing IoT networks is not an easy problem since we have to think of device, network and application characteristics, affordable cryptography-based solutions, physical security of the network and devices, compromise scenarios and intrusion detection systems. Designers and administrators will face many trade-offs, where security will be on one side and cost, network lifetime, Quality-of-Service (QoS), reliability and many more will be on the other side.

The goal of this thesis is to investigate how IoT networks can be secured against D/DoS attacks. For this aim, we focused on a subclass of IoT, namely the Low Power and Lossy Networks (LLNs). In LLNs, elements of the network are resource-constrained (i.e., limited processing, storage, communication and power sources) and the environment is lossy [6]. The IETF and the IEEE proposed several standards to connect such networks to the Internet and allow various applications to be supported efficiently. The set of protocols are called as the *standardized protocol stack* [7]. We considered to study LLNs, because we believe that majority of the devices in IoT will be within the concept of LLNs. Limited resources, lossy environments and possibly mobility make such networks challenging to incorporate robust security solutions. Considering these devices are capable of connecting to the Internet, the task of protecting such networks against attackers gets more challenging.

In this respect, we started our thesis journey with understanding the LLNs with its architectural design, elements and communication protocols. Following this, we investigated the D/DoS attacks that may target the LLNs. Such attacks can be either inherited from earlier networking research (e.g., Wireless Sensor Networks (WSN) and Ad-Hoc Networks) or specific to LLNs due to new protocols and the Internet connection. Therefore, it is very crucial to determine the possible attacks. After determining the attacks, we analyzed the standardization efforts and also the literature for security mechanisms. We questioned whether the cryptography-based solutions can provide the ultimate fix to solve the D/DoS problem for LLNs. Our questioning

led us to believe that although there is cryptography, D/DoS attacks can still affect the LLNs. Therefore, we comprehensively analyzed the literature for:

- Studies that analyze the effect of the attacks,
- Studies that propose a cryptography-based security solution to protect the LLNs from the attacks,
- Intrusion Detection Systems (IDS) that aim to detect a set of attacks that target the LLNs,
- Mitigation systems that target to reduce the effect of the attacks.

We continued our journey with focusing on the network layer of the LLNs and specifically to the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [6] that was proposed by the IETF. Among the set of attacks, we targeted the Version Number Attack (VNA) since it was a very interesting DoS attack and was not studied comprehensively in the literature. VNA was interesting because it was specific to the RPL and it could detrimentally degrade the performance of the network easily. We started our analysis with the Cooja [8] simulator of the open source Contiki Operating System (OS) [9]. We set up a realistic scenario in the simulation environment and simulated a single attacker at different parts of the network. When we analyzed the results, we realized that the effect of the attack is correlated to some performance metrics, which was also realized by a predecessor study [10]. Based on our analyses and observations, we came up with two novel mitigation techniques that are very lightweight and that can mitigate the effect of the VNA successfully. Up until this point, our analysis were considering only a single attacker and employing multiple attackers were our future work. So, as the last part of this thesis, we moved to multiple attackers analysis.

The contributions of our thesis can be listed as follows:

- Research and categorization of the D/DoS attacks that may target the LLN-based IoT networks,

- Research and categorization of the studies that target the D/DoS attacks (i.e., analysis studies, cryptography-based security solutions, detection studies, mitigation studies),
- Analysis of the RPL Version Number (VN) Attack in a realistic scenario with respect to performance metrics (e.g., delay, packet delivery ratio, control message overhead and power consumption),
- Proposition of two new lightweight mitigation techniques which provides significant performance improvements to RPL networks under VNA,
- Analysis of multiple attackers in RPL networks.

The rest of the thesis is organized as follows: In Chapter 2, we give preliminary information on IoT networks (i.e., architecture and the standardized protocol stack). In Chapter 3, we review the literature for the security of LLNs. In this chapter, we analyze the cryptography-based security solutions for LLNs, we consider the proposed solutions and questions whether they are enough. Then, we categorize the D/DoS attacks that may target the LLNs. Following this, we analyze the studies examine the D/DoS attacks for LLNs, mitigation systems and protocol security solutions and IDSes for the LLNs. In Chapter 4, we study the effects of the RPL VNA. We propose two novel mitigation techniques for this attack in Chapter 6 and in Chapter 7 we focus on multiple attackers. Finally in Chapter 8, we conclude the thesis.

2. PRELIMINARY INFORMATION ON IoT NETWORKS

2.1 IoT Networks

Internet of Things can be defined in several ways from various angles and there is no standard definition for it. However, from the engineering point of view, IoT is a network of any *thing*s, each supplied with a computing system (i.e., CPU, memory, power source and a communication interface like radio or Ethernet), each is uniquely identifiable and addressable and connected to the Internet. In this section, we will firstly propose a generic architecture for IoT which we think will be helpful to understand IoT environments better. After that, we will summarize the standardized protocol stack [7] we focus on in this study.

2.1.1 Internet of Things architecture

We believe that, exploring the architectural components is a very useful way to see the complete picture and understand IoT environments better. In Figure 2.1, we outline a generic IoT architecture which is based on the general architectures previously proposed in [11–13]. The only difference of our architecture from the reference works is that we separated the IoT Access Network Layer from the IoT-Internet Connection Layer, whereas the reference studies combine them into a single layer called either as *Network Layer* or *Transport Layer*.

In the generic architecture, the lowest layer is the Perception Layer. It consists of sensors, actuators, RFID tags and any other embedded devices. Most of these devices are expected to be small form-factor devices with constrained resources (i.e., power source, processing, storage and communication interface). The majority of IoT devices will use battery as the power source. However, based on the application environment, mains-powered devices or energy-harvesting elements may exist as well. Since power will be a scarce resource, power consumption of the nodes (i.e., devices in the network) has to be minimized. In addition to various techniques to reduce the power

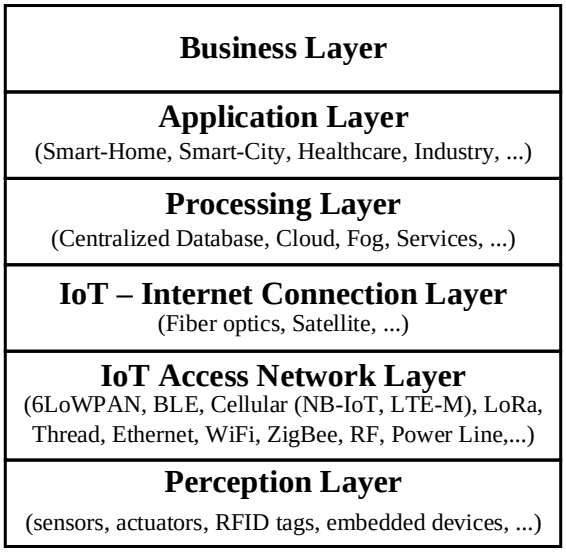

Figure 2.1 : Generic architecture of IoT.

consumption, IoT devices use low-power radios to keep the energy footprint as small as possible and lengthen the network lifetime. Typically low-end microcontrollers with RAM and ROM in the order or KBs constitute the big portion of nodes accommodated in IoT networks. In addition to the resource characteristics, mobility of the devices is important as well. Devices in the Perception Layer can be either static or mobile, but the percentage of mobile devices will be smaller than the static ones.

The IoT Access Network Layer is the second layer in our architecture, in which the nodes in the Perception Layer form a network. In this layer, there are several communication technologies (i.e., 6LoWPAN, Bluetooth Low Energy (BLE), LoRa and LoRaWAN, WiFi, Ethernet, Cellular, ZigBee, RF and Thread) which are candidates for the in-network communication. Most of them are open technologies, whereas some of them are (e.g., ZigBee, LoRa, Cellular) proprietary. These communication technologies provide varying data rates and transmission ranges in return of different power consumptions and costs. Hence, depending on the several design constraints, the nodes in the Perception Layer can form IoT networks with different characteristics. Among these technologies, BLE, WiFi, LoRa and Cellular offer star-based topologies. However, 6LoWPAN, ZigBee and Thread support mesh topologies, where elements of the network can forward others' packets. Some of them are proposed for specific application areas (i.e., Thread was proposed for smart-home

environments). Most of these technologies require a gateway or border router which is used to connect the nodes in IoT network to the Internet.

The third layer in our generic architecture is the IoT - Internet Connection Layer, where a border router or gateway connects the inner IoT network to the Internet via communication technologies, such as fiber optics or satellite communication.

Processing, analysis and storage of the collected data are performed at the Processing Layer. Designers can choose centralized storage and processing systems, or distributed storage and processing systems (e.g., cloud or fog computing environments). Middleware services are provided in this layer based on the processed and analyzed data. This is one of the most important layer in the architecture of IoT, since valuable information is extracted here from the collected data which can be in big volumes, variety and veracity.

The Application Layer is the fifth layer within the generic IoT architecture. In this layer, we see applications in various deployment areas, which make use of the meaningful information obtained from Processing Layer. Applications of IoT can be in home, building, industry, urban or rural environments. Applications of home environments can be health-reporting and monitoring, alarm systems, lighting applications, energy conservation, remote video surveillance [14]. Building environments IoT applications can be Heating Ventilation and Air Conditioning applications, lighting, security and alarm systems, smoke and fire monitoring and elevator applications [15]. Industrial IoT applications can be safety, control and monitoring applications with different emergency classes [16]. In urban environments, there may be broad range of applications. Lighting applications, waste monitoring, intelligent transportation system applications, monitoring and alert reporting are only a few of them. Rural environments may include monitoring applications (e.g., bridges, forests, agriculture, etc.).

The Business Layer is the last layer in the generic architecture, which includes organization and management of IoT networks. Business and profit models are constructed here in addition to charging and management operations [12].

2.1.2 Standardized protocol stack for Low Power and Lossy Networks

Multiple communication technologies exist for the IoT-Access Network Layer as we mentioned in Section 2.1.1. Since Thread, NB-IoT, LTE-M, LoRa/LoRaWAN are very new communication technologies, there are not any studies which focus on the D/DoS attacks that may target such networks. ZigBee is a proprietary technology and it also uses the same physical layer and MAC layer as 6LoWPAN-based networks. Thus PHY and MAC layer attacks for 6LoWPAN-based networks covers the PHY and MAC layer attacks for ZigBee-based networks too. WiFi targets more resource-rich devices than 6LoWPAN. Therefore, it may not be a good candidate for low power and lossy networks-based IoT applications where majority of the devices will be battery-powered devices with small form factors and reasonable costs. BLE technology might be a good option for the IoT-Access Network Layer with very low power consumption, increased data rate and range. However, it suffers from the scalability problem where Bluetooth-based networks face with issues when the number of slaves exceeds seven [17, 18]. Up until Bluetooth 5, park state was supported by Bluetooth which was allowing more than seven slaves to be part of the network in turns. But Bluetooth 5 does not support it any more and instead it brought scatternets, which aims to create multi-hop Bluetooth networks with specific nodes acting as routers between piconets. However, currently no commercial Bluetooth radio supports it and synchronization and routing operations will make the scatternet operation in Bluetooth networks a complex issue to deal with. Hence, considering the aforementioned reasons, we focus on the 6LoWPAN-based IoT networks in this study.

The IEEE and the IETF proposed several standards and protocols in order to connect resource-constrained nodes to the Internet within the concept of IoT. Palattela et al. [7] proposed a protocol stack for low power and lossy IoT networks which makes use of the protocols/standards proposed by IEEE and IETF. The standardized protocol stack is shown in Figure 2.2 .

The standardized protocol stack includes IEEE 802.15.4 [19] for physical layer and MAC layers. This standard promises energy-efficient PHY and MAC operations for low power and lossy networks and is also used by Thread and ZigBee technologies.

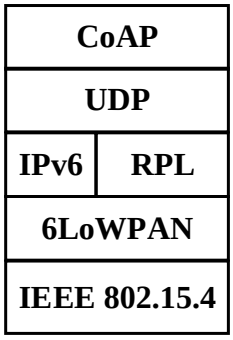

Figure 2.2 : Standardized protocol stack.

The expected cardinality of the IoT networks (e.g., of the order of billions) and already exhausted IPv4 address space force IoT to use IPv6 addresses. However, when IPv6 was proposed, low power and lossy networks were not considered, which resulted in the incompatible packet size issue. The maximum transmission unit of IEEE 802.15.4-based networks is far too small compared to IPv6 packet sizes. In order to solve this problem, IETF proposed an adaptation layer, IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) [20]. 6LoWPAN makes use of header compressions to permit transmission of IEEE 802.15.4 fragments carrying IPv6 packets.

RPL [6] was proposed by the IETF as IPv6 routing protocol for low power and lossy networks. We will explain RPL in more detail in Section 4.

The standardized protocol stack for low power and lossy networks employs the Constrained Application Protocol (CoAP) [21] for the application layer. CoAP is built on top of UDP and supports Representational State Transfer architecture. By means of CoAP, even resource-constrained nodes can be part of the World Wide Web. In order to optimize the data carried by CoAP messages, the IETF proposed another standard for the binary representation of the structured data called Concise Binary Object Representation (CBOR) [22] on top of CoAP.

3. LITERATURE REVIEW ON THE SECURITY OF LOW POWER AND LOSSY NETWORKS

In Section 2.1.2, we briefly summarized the standardized protocol stack which consists of standards and protocols proposed by IEEE and IETF for low power and lossy networks. In this chapter, we focus on the security of IoT networks which accommodate the standardized protocol stack.

Securing a communication network is not an easy task and requires a comprehensive approach. In such a study, we have to determine assets, think of threats and consider compromise scenarios and possible vulnerabilities. Following these, we have to find the suitable solutions which will help us to ensure a *secure* system.

Considering the solutions, the first thing that probably comes into our minds is the cryptography. Cryptography promises to provide *confidentiality* and *integrity* of the messages, *authentication* of the users and systems and *non-repudiation* of the transactions. Confidentiality means that the content of the message is kept secret from eavesdroppers. Integrity ensures that the content of the message is not changed and is still the same as the first time it was produced. Authentication allows the end points of the communicating parties to identify each other and determine the correct target of the communication. Non-repudiation prevents one end of the communication to deny its actions that it performs and protects the other end.

In this chapter, we firstly outline the cryptography-based security solutions for the low power and lossy networks which employ the standardized protocol stack. After that, we analyze the protocols and point out the advantages and disadvantages. Then we will inquire whether cryptography is enough for us or not.

3.1 Cryptography-Based Security Solutions for Low Power and Lossy Networks

A number of cryptography-based solutions exist so as to secure the low power and lossy networks that employ the standardized protocol stack. These solutions are shown in Figure 3.1.

CoAP → CoAPs	
UDP → DTLS	
RPL & IPv6 →	**IPSec, Secure RPL Control Messages**
6LoWPAN →	
IEEE 802.15.4 →	**IEEE 802.15.4 PHY & Link Layer Security**

Figure 3.1 : Cryptography-based security solutions for Low Power and Lossy Networks.

IEEE 802.15.4 PHY and Link Layer Security [19] provides security for the communication between two neighbors in IEEE 802.15.4-based networks. This hop-by-hop security solution promises confidentiality, authenticity and integrity against insider attackers.

The Internet Protocol Security (IPSec) [23] aims to provide end-to-end security. It consists of a set of protocols, which are Authentication Headers, Encapsulating Security Payloads and Security Associations. Authentication Headers provides authentication and integrity, whereas Encapsulating Security Payloads promises confidentiality in addition to authentication and integrity. Designers can select either of them but regardless of the selection, Security Associations has to run initially to setup the security parameters. IPSec provides security for IP-based protocols and it is independent from the protocols above the network layer.

In addition to IPSec, RPL provides secure versions of the control messages. Although it is optional, confidentiality, integrity and authentication of the control messages are assured.

Datagram Transport Layer Security (DTLS) [24] aims to secure UDP-based applications. Similar to the other solutions, it ensures the confidentiality, integrity and authenticity of datagrams.

CoAP provides security bindings for DTLS in CoAPs scheme. It lets designers to choose to run DTLS with preshared keys, public keys and/or certificates in order to secure CoAP traffic. Although Figure 3.1 does not show any other security mechanisms working at the application layer and above, the IETF has draft documents (i.e., Object Security of CoAP (OSCoAP), CBOR Object Signing and Encryption

(COSE) and Ephemeral Diffie-Hellman over COSE (EDHOC)) which aim to provide security at the application layer and above.

3.2 Which Security Solution to Use?

As we can see, there are a number of security solutions to protect low power and lossy networks and it is hard to determine which solution to use.

IEEE 802.15.4 PHY and Link Layer Security is independent from the network layer protocols and most of the radios support it. Independence from the upper layer protocols means that we do not have to change anything with them. However, since IEEE 802.15.4 PHY and Link Layer Security provides the security between two neighbors, trustworthiness of every node on the routing path becomes a very crucial issue. If the routing path has a malicious node, then the security of the messages routed through this path cannot be guaranteed. In addition to this, IEEE 802.15.4 PHY and Link Layer Security works only in the IoT - Access Network Layer in our generic architecture and when messages leave this layer and enter the Internet, they are no more protected [25].

IPSec provides end-to-end security and is independent from the upper layer protocols. End-to-end security guarantees security between two hosts which can be in different networks. Designers do not have to worry about the trustworthiness of the other nodes, devices or networks on the path. However, it brings burden to 6LoWPAN layer, where packets with IPSec require header compression [25]. In addition to this, Security Associations is connection-oriented and simplex, which means if two hosts want to send packets secured with IPSec to each other, then each of them individually need to establish SAs [26]. Furthermore, firewalls may limit the packets with IPSec and Internet Service Providers tend to welcome packets with IPSec as business-class packets and prefer to charge them more. So, if IoT data will be secured with IPSec, there are a number of issues we have to consider before using it.

DTLS serves as the security solution between two UDP-based applications running on different end-points. Although it aims to protect the application layer data, it does not promise security for anything else. This means, if we employ DTLS as the only security solution, then we cannot protect IP headers when packets are passing through

the IoT - Access Network Layer and through the Internet. So, security of the routing becomes susceptible to the attacks, such as DoS and DDoS attacks. This is why the primary security concern of DTLS is on D/DoS attacks.

3.3 We Have Cryptography-Based Solutions, Are We All Set?

In Section 3.1 we outlined the cryptography-based security solutions very briefly. As we explained in Section 3.2, each solution comes with its advantages and disadvantages. It is not an easy task to select the appropriate solution. However, there are a number of other issues which we have to consider when protecting IoT networks.

First of all, cryptography is generally thought to be heavy weight, and is full of resource consuming operations implemented in software and/or hardware. When we consider the resource limitations of the devices in low power and lossy networks, affordability of such solutions becomes questionable. Designers have to face the trade off between security and very crucial parameters such as cost, network life time and performance.

Secondly, although cryptography-based solutions are proved to be secure, proper implementation of the protocols and algorithms is extremely important. However, most of the implementations of these solutions have vulnerabilities as reported by researchers [27]. In addition to this, in order to shorten the development time, engineers tend to use the code examples shared on forums. These code examples working properly does not mean that they are vulnerability-free [28].

Physical security of the networks and devices are as important as our other concerns. It is directly related to the applicable type of attacks. If the physical security of the deployment area is weak, which is the case for most of the deployments, and if devices do not have protection mechanisms against tampers which is due to reduce the cost, then it is possible for attackers to insert a malicious device or grab a device and extract the security parameters and leave a malicious device back.

In addition to the cost of cryptography, issues with correct implementation and physical security, we have to consider users as well. We know that most of the people are not security-aware and usability of security mechanisms have problems [29]. Therefore, compromise scenarios have to think of users and external people involving with the IoT network, applications and deployment areas.

Table 3.1 : D/DoS Attacks which may target IoT Networks.

Physical Layer	MAC Layer	6LoWPAN Layer	Network Layer	Transport and Application Layer
Node Capture	Jamming	Fragment Duplication	Rank	Flooding
Jamming	GTS	Buffer Reservation	Version Number	Desynchronization
Spamming	Backoff Manip.		Local Repair	SYN Flood
	CCA Manip.		DODAG Inconsistency	Protocol Parsing
	Same Nonce		DIS	Processing URI
	Node Spec. Flooding		Neighbor	Proxying and Caching
	Replay Protection		Sybil	Risk of Amplification
	ACK Attack		Sinkhole	Cross-Protocol
	Man-in-the-Middle		Selective Forw.	IP Address Spoofing
	Ping-Pong Effect		Wormhole	
	Bootsrapping		CloneID	
	Stenography			
	PANID Conflict			

Although we have cryptography, our networks and systems are still susceptible to some type of attacks, called Denial of Service attacks [30]. In the next section, we will examine the DoS and DDoS attacks which may target low power and lossy networks employing the standardized protocol stack.

3.4 Denial of Service Attacks Targeting Internet of Things Networks

Denial of Service attacks aim to misuse the available resources in a communication network and degrade or stop the services offered to ordinary users. Since IoT will be one of the main building block of Internet of Services, detection, mitigation and prevention of such attacks are very crucial.

In this section, we present and explain the D/DoS attacks which may target IoT networks. Table 3.1 categorizes such attacks with respect to the layers of the standardized protocol stack. This categorization is an extended version of our previous study [31].

3.4.1 D/DoS attacks to the physical layer

Physical Layer D/DoS attacks are *node capture*, *jamming* and *spamming*.

As its name implies, in *node capture* attacks, attackers capture the physical nodes within the network. The aim of the attackers may be creating routing holes or tampering the device and extracting security parameters. After that, they may place the node back with the compromised software or place the node with a replica of it. By this way, they can apply various attacks (e.g., other attacks categorized as higher layer attacks).

Physical Layer *jamming* attacks comprise of malicious devices creating interference to the signals transmitted in the physical layer [32]. Attackers can constantly, randomly or selectively (i.e., jamming signals carrying specific packets, such as routing or data packets) apply jamming.

In *spamming* attack, attackers place malicious QR codes to the deployment areas which cause users to be forwarded to malicious targets on the Internet [33].

3.4.2 D/DoS attacks to the MAC layer

MAC Layer D/DoS attacks are *link layer jamming*, *GTS*, *backoff manipulation*, *CCA manipulation*, *same nonce attack*, *node specific flooding*, *replay protection attack*, *ACK attack*, *man-in-the-middle*, *ping-pong effect*, *bootstrapping attack*, *PANID conflict* and *stenography*.

Link layer jamming is a type of jamming where frames are jammed instead of signals as in the physical layer [32].

IEEE 802.15.4 standard has an optional feature called as Guaranteed Time Slot (GTS) which works in beacon-enabled operational mode. GTS is intended for timely critical applications that require strict timing with channel access and transmissions. Nodes have to request and allocate time slots in order to use this feature. However, if attackers cause interference during this process (e.g., by jamming), then ordinary nodes cannot register themselves for the guaranteed time slots and thus QoS of the application gets affected. This attack is called *GTS* attack [34].

ACK attack consists of attackers creating interference to Acknowledgment (ACK) frames and thus causing a node to believe that its fragment was not successfully received by the receiving node [32]. By this way targeted nodes are forced to retransmit the same fragment and consume more power. QoS of the running application would be affected by it too. Moreover, it may cause the sender node believe that its next hop neighbor is filtering the messages.

Clear Channel Assessment (CCA) mechanism is used by nodes to sense the channel and find out if any other node is currently using the channel or not. This approach is commonly used to prevent collisions. However, attackers can skip CCA and access the channel, which causes collisions. By this way, delays, retransmissions and unnecessary energy usage occurs. This attack is called *CCA manipulation* [32].

Backoff manipulation attack compromises the backoff periods of Carrier Sense Multiple Access (CSMA)-based medium access with attackers choosing shorter backoff times instead of longer [32]. By this way, they get the chance to use the channel as much as possible and limit the other users' channel accesses.

Sequence numbers are used in the IEEE 802.15.4 standard in order to prevent malicious devices sending the previously sent fragments over and over. However, in *replay protection attacks* [32], attackers can still misuse it by sending frames with bigger sequence number than the targeted ordinary node. This would cause the receiving node drop the fragments coming from the ordinary node since it now looks like it is sending old fragments.

As we mentioned in Section 3.1, IEEE 802.15.4 PHY and Link Layer Security is a candidate security mechanism for IoT security, which promises to protect the communication between two neighbor nodes. If nodes share the same key and nonce values in the implementation of IEEE 802.15.4 PHY and Link Layer Security, then attackers may extract the keys by eavesdropping the messages which happens in the *same nonce* attack [32].

The *PANID Conflict* attack misuses the conflict resolution procedure of IEEE 802.15.4 which functions when two coordinators are placed close to each other in a deployment area and holding the same Personal Area Network ID. Malicious nodes may transmit a

conflict notification message when there is actually no conflict to force the coordinator to initiate the conflict resolution process [32].

Another MAC Layer D/DoS attack is the *ping-pong effect*, where malicious nodes intentionally switch between different PANs [32]. If attackers choose to do it frequently, then they may cause packet losses, delays and extra overhead to the already limited resources.

In the *bootstrapping attack*, attackers aim to obtain useful information about a new node joining the network. In order to do so, firstly a targeted node is forced to leave the network by the attackers. Then when it tries to join the network again, attackers obtain the bootstrapping information which they may use to associate a malicious node to the network [32].

Node specific flooding attacks are a type of flooding attack which is applied at the MAC layer [32]. In this attack, malicious nodes send unnecessary fragments to the target node which aims to consume its resources and thus is no longer able to serve for its ordinary purpose.

The *Stenography* attack abuses the unused fields in the frame format of IEEE 802.15.4. Unused bits can be used by the attackers to carry hidden information [32].

3.4.3 D/DoS attacks to the 6LoWPAN layer

Hummen et al. [35] proposed two attacks, namely *fragment duplication* and *buffer reservation*, which may target the 6LoWPAN Adaptation Layer.

In *fragment duplication* attack, attackers duplicate a single fragment of a packet and thus force the receiving node to drop the fragments of the corresponding packet. In this attack, attackers abuse the approach of 6LoWPAN standard which deals with the duplicate fragments. The standard advises to drop the fragments of a packet in case of duplicates so as to get rid of the overhead of dealing with duplicates and save resources. However, malicious nodes can turn this naive mechanism into a DoS attack very easily.

In *Buffer reservation* attacks, attackers reserve the buffer space of the targeted node with incomplete packets and keep it occupied as long as possible. Since resources are limited, nodes cannot afford to spare extra buffer space for the incomplete packets of other nodes. Thus, during the time the attacker holds the buffer space, ordinary nodes'

fragments cannot be accepted. Readers should note that, this behavior of 6LoWPAN is possible when 6LoWPAN is configured to forward the fragments according to the route-over approach, where all fragments of a packet are reassembled by the receiving node before being forwarded.

3.4.4 D/DoS attacks to the network layer

D/DoS attacks which may target the IoT Network Layer can be divided into two categories: RPL-specific and non-RPL-specific attacks. RPL-specific attacks are *rank*, *version number*, *local repair*, *DODAG inconsistency* and *DIS* attacks. Non-RPL-specific attacks are the ones which are already known from the WSNs, and other communication networks research. Although they look old-fashioned, they are still applicable in RPL-based networks. Non-RPL-specific attacks are *sybil*, *sinkhole*, *selective forwarding*, *wormhole*, *cloneID* and *neighbor* attacks.

3.4.4.1 RPL-specific attacks

D/DoS attacks which may target RPL networks abuse the vulnerabilities of the RPL protocol design. RPL, designed by the IETF for the routing of IPv6 packets on low power and lossy networks, has vulnerabilities with the control plane security and attackers can easily misuse it. DoS attacks which misuse the operation of the protocol and are specific to RPL are *rank*, *version number*, *local repair*, *DODAG inconsistency* and *DIS* attacks. These attacks will be explained in detail in the next chapter.

3.4.4.2 Attacks not specific to RPL

Attacks which are not specific but still applicable to RPL are *neighbor*, *sybil*, *sinkhole*, *selective forwarding*, *wormhole* and *cloneID*.

A malicious node can apply the *neighbor* attack by retransmitting the routing control messages it hears [36]. This behavior causes neighbor nodes to think that the source of the control message is close to them and take actions accordingly. Actions could be sending control messages back, trying to select it as a preferred parent, etc. If the attacker uses a high power radio, then it may affect a large portion of the network by this way.

In *sybil* attacks, a malicious node seems to act as multiple nodes, introducing itself with multiple logical identities [30]. If there is a voting mechanism running in the IoT network (e.g., voting based security mechanisms, cluster head selection), attackers can apply sybil to change the results and thus take control of the complete network or a portion of the network.

The *CloneID* attack is similar to the sybil attack but works in a different dimension. The attacker in this case places the clones of a malicious node or normal node to the multiple positions at the network [30, 37]. This attack has similar aims as sybil and it may also be called *node replication* attack.

Sinkhole attacks are another type of attacks where malicious nodes advertise good routing parameters to show themselves as candidate parents. In RPL, attackers can advertise good ranks, which causes the neighbor nodes to select it as the preferred parent [30, 37, 38]. When a malicious node is selected as the preferred parent by neighbor nodes, then it can apply other attacks, such as selective forwarding.

In *selective forwarding* attacks, a malicious node inspects the incoming packets, drops the ones it is interested in and forwards the rest [30, 37]. For example, it may forward only the routing messages, whereas it may drop the data packets. Or, malicious node may filter specific packets sourced from or destined to specific addresses.

The last attack we explore in this category is the *wormhole* attack [30,39]. In wormhole attacks, at least a couple of malicious nodes create a hidden communication channel by means of multiple radios and transfer the overheard messages transmitted at one end point to another. This may work bidirectional as well. By this way, two sets of nodes around each attacker believe that they are in the communication range of each other, which causes several issues.

3.4.5 D/DoS attacks to the transport and application layer

D/DoS attacks which may target Transport and Application Layers are *flooding*, *desynchronization*, *SYN flood*, *protocol parsing*, *processing URI*, *proxying and caching*, *risk of amplification*, *cross-procotol* and *IP address spoofing* attacks [21, 40].

The majority of the attacks mentioned here were not studied in the literature and the IETF considers them as possible threats for CoAP.

3.5 Studies that Analyze D/DoS Attacks for Internet of Things

The previous section was about the possible D/DoS attacks which can target the IoT networks. Starting from this section, we will analyze the studies for the aforementioned attacks. In this section, we will explore the works which investigate the effects of the attacks.

Sokullu et al. [34] proposed GTS attacks to IEEE 802.15.4 in 2008. In their work, they also analyzed the effects of the attack in the bandwidth utilization of Contention Free Period (CFP). They considered single and multiple attackers where attackers can either attack randomly or intelligently. They found out significant decrease in the bandwidth utilization of CFP periods due to GTS attacks.

Le et al. analyzed the rank attack in [41] in RPL networks in 2013. In this work, they applied the rank attacks with different cases where the attacker constantly applies the attack or switches between legitimate and malicious behaviors frequently. Analysis with respect to combinations of attacking cases show that if the rank attack is applied in a dense part of the network, then its effect is more detrimental. They also realized that, the number of affected nodes, number of generated DIO messages, average end-to-end delay and delivery ratio can be the indicator of such attacks.

Mayzaud et al. studied RPL VNA in [10] in 2014. They investigation with a single attacker in a grid topology at varying positions showed that the location of the attacker is correlated to the effects of the attack. If the attacker is located far from the DODAG root within the grid, then its effect is larger than when attacker is closer to the root.

The Version number attack is analyzed by another work [42] proposed by Aris et al. in 2016. In this study, the authors considered a factory environment consisting of varying topologies (i.e., grid and random) with different node mobilities (e.g., static and mobile nodes). A probabilistic attacker model is incorporated here. Based on the simulations, in addition to the location-effect correlation found in Mayzaud's work [10], the authors found out that the mobile attackers' effect can be as detrimental as the farthest attacking

Table 3.2 : Categorization of the studies that analyze the D/DoS Attacks for IoT.

Proposal	Target Attack	Finding
Sokullu [34]	GTS (MAC Layer)	Significant bandwidth utilization decrease in CFP
Le [41]	Rank (Routing)	Dense networks are more vulnerable
Mayzaud [10]	Version Number (Routing)	*Attacking position-effect of the attack* correlation
Aris [42]	Version Number (Routing)	Mobile attackers are more detrimental and attack triples the power consumption of the network

position in the network. They also showed that, version number attacks can increase the power consumption of the nodes by more than a factor of two.

Table 3.2 categorizes the studies which analyze the effects of the D/DoS attacks for IoT. When we review the studies in this section, we realize that researchers focused on the IoT-specific attacks rather than the attacks which we are already familiar with from the Wireless Sensor Networks research (i.e., selective forwarding, wormhole, sinkhole, etc.). In addition to this, three of the studies found out correlations with the success of the attack and the attack settings. Such findings can be extremely useful in defending IoT networks against the attackers and designing better detection and mitigation systems which consider these findings. In Table3.1, we had provided a categorization of the D/DoS attacks for IoT and it is clear that many attacks have not been implemented and analyzed in a similar manner.

3.6 Mitigation Systems and Protocol Security Solutions for Internet of Things

Mitigation systems are proposed by researchers in order to minimize the effects of the attacks. Such systems are far from being a complete security solution but still can increase the strength of the system against attackers. In this context, existing protocols are enriched with additional features by the designers which can mitigate the detrimental effects of the D/DoS attacks. On the other hand, protocol security solutions referred here consist of mechanisms which aim to secure a communication protocol or a specific part of it. In this section, readers can find the studies which either propose a security solution or mitigate the effect of the attacks.

Dvir et al. proposed VeRA [43], a security solution for the crucial version number and rank parameters carried in DIO messages in 2011. Their solution makes use of hash chains and message authentication codes in order to securely exchange these RPL parameters in DODAG Information Object (DIO) messages.

Weekly et al. [38] evaluated the defense techniques for sinkhole attacks in RPL in 2012. They compared a reduced implementation of VeRA to their novel technique called as Parent Failover. Parent Failover uses Unheard Node Set which includes the IDs of the nodes that the BR did not hear from. Each node blacklists its parent if it sees itself in the list in this technique.

Wallgren et al. [30] proposed implementations of routing attacks (i.e., selective forwarding, sinkhole, hello flood, wormhole, sybil) which are not specific to RPL. They did not analyze the effects of the attacks. However, they made comments on possible mitigation/detection mechanisms against such attacks. Their mitigation ideas include usage of geographical location information, incorporation of cryptography schemes, using multiple routes and/or RPL instances and keeping track of the number of nodes within the network. Although the authors suggest to use such mechanisms against the corresponding attacks, they did not implement the mitigation mechanisms and analyze the performance of them.

Hummen et al. [35] proposed two novel attacks to 6LoWPAN adaptation layer, which are fragment duplication attack and buffer reservation attack. They also proposed two novel mitigation mechanisms against these attacks. For fragment duplication attacks, the authors proposed hash chain structures which create a binding for fragments of a packet to the first fragment of the corresponding packet. In order to mitigate the effects of buffer reservation attacks, they suggested to split the reassembly buffer into fragment-sized slots and let multiple fragments belonging to different packets use it. They merged split buffer approach with a fragment discard mechanism in case of overloaded buffer conditions.

In 2014, Seghal et al. [44] proposed a mitigation study which targets DODAG inconsistency attacks. According to the authors, RPL uses a threshold to mitigate the effects of such an attack. In RPL, a node receiving a data packet with flags indicating an inconsistency drops the packet and resets its trickle timer. A node can do this

Table 3.3 : Categorization of the mitigation systems and protocol security solutions.

Proposal	Target Attack	Mitigation/Security Mechanism
VeRA [43]	Rank and Version Number (Routing)	Hash chains and Message Authentication Codes
Weekly [38]	Sinkhole (Routing)	Reduced VeRA and Unheard Node Set
Wallgren [30]	Routing attacks not specific to RPL	Geographical Location Info., Cryptography, Multiple Paths and Instances, Cardinality of the Network
Hummen [35]	Fragment Duplication and Buffer Reservation (MAC)	Content Chaining Using Hash Chains, Split Buffer with Fragment Discard
Seghal [44]	DODAG Inconsistency (Routing)	Adaptive Threshold for Inconsistency Situations
Mayzaud [45]	DODAG Inconsistency (Routing)	Adaptive Threshold for Inconsistency Situations
Ramani [46]	CloneID (Routing), General DoS	Distributed Firewall

until reaching a threshold. After this threshold it does not reset the trickle timer any more. This proposal changes the constant threshold of RPL to an adaptive threshold to mitigate the effects of the attack better.

Another mitigation technique for DODAG inconsistency attacks was proposed by Mayzaud et al. [45] in 2015. It is an improved version of the mitigation technique proposed in Seghal's work [44]. In the former study, packets with 'R' flags set were counted, whereas in this study, the number of trickle timer resets are counted. Based on this, a node either drops the packets and resets trickle timer, or forwards the packets with modifying the R and O flags to the normal state.

In 2016, Ramani proposed a two-way firewall [46] for low power and lossy networks. The two-way firewall analyzes the traffic destined to the 6LoWPAN network and traffic leaving from the network. The proposed firewall was tested against the CloneID and simple DoS attacks. The main module of the firewall works on the BR and becomes active when packets destined to the CoAP and DTLS ports are captured. Packets are parsed into incoming and outgoing packets and their IP addresses and ports are verified. After this check, information related to the packet is saved and checked against the protocol rules. Erroneous packets are dropped here. Also the nodes in the 6LoWPAN network are equipped with mini-firewall modules which inform the main firewall about their behavior.

Table 3.3 categorizes the Mitigation Systems and Protocol Security Solutions for IoT. Mitigation mechanisms against routing attacks constitute the majority of the studies in this section. Researchers targeted both RPL-specific attacks and other routing attacks which can be applied to RPL as well. Considering the resource limitations in IoT networks, we can see that three proposals use hash functions as lightweight solutions.

3.7 Intrusion Detection Systems for Internet of Things

In this section, we will survey the literature for intrusion detection systems proposed against D/DoS attacks for IoT networks. This section is organized as follows: Firstly, we will briefly give some background information about IDSes. After that, we will analyze the IDSes proposed for IoT.

3.7.1 Intrusion Detection Systems

Intrusion Detection Systems serve as a strong line of defense for computer networks against the attackers. Without IDS, the puzzle of a *secure* network is incomplete. As explained in Section 3.3, despite having cryptography-based solutions, attacks are still possible and IDS comes into the picture here, where it monitors and analyzes the traffic, data, behavior or resources and tries to protect the network from attackers.

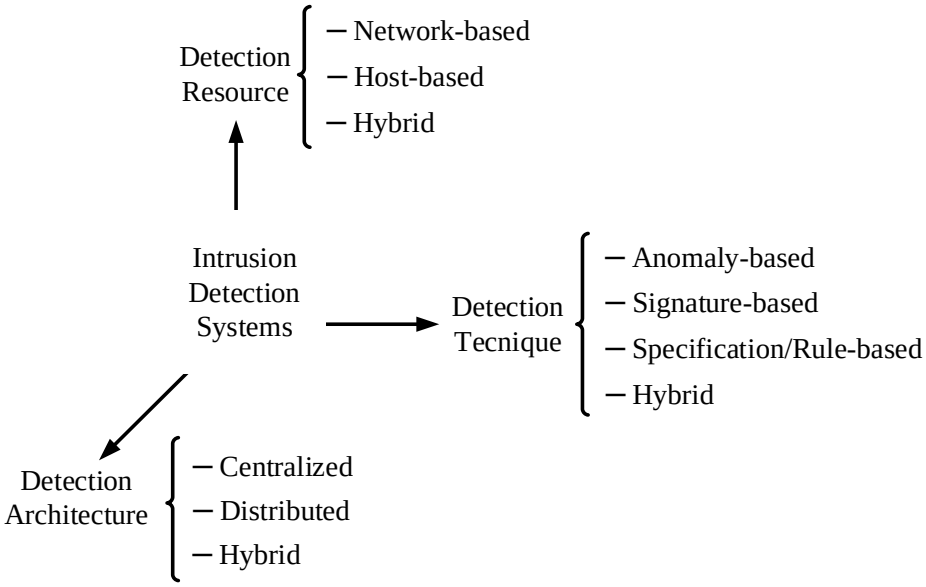

Figure 3.2 : Intrusion Detection Systems.

Intrusion Detection Systems can be explored from various points of view. Figure 3.2 shows a 3D Cartesian Plane of IDSes, where axes depict important categories which may be helpful to classify the IDSes. Although not shown, there may be other dimensions in this figure, such as operation frequency and targeted attacks.

3.7.1.1 Intrusion Detection Systems: detection techniques

IDSes can be divided into four classes based on the detection technique. These are *anomaly-based*, *signature-based*, *specification/rule-based* and *hybrid* systems where the former two are the most popular ones.

Anomaly-based systems learn the behavior of the system when there is no attack and create a *profile*. Deviations from the profile show possible anomalies. Anomaly-based IDSes can detect the new attacks since attacks are expected to cause deviations from the ordinary behavior. However, they can create false alarms and incorrectly classify legitimate connections as intrusion attempts. In addition to this, anomaly-based techniques are generally believed to be more complex and to use more resources than the other detection techniques.

Signature-based systems aim to detect intrusions by making use of attack signatures/patterns. Typically signatures are stored in a database and IDS tries to match them when analyzing the connections, packets or resources. If the database does not have a signature for an attack, which happens in case of new attacks, such systems cannot detect it. Otherwise they promise high detection rates for the known attacks and they do not suffer from false alarms. If we use signature-based techniques, we have to consider how to deal with new attacks since our IDS will probably skip them. Also we have to think about the storage cost of the signatures.

Specification/rule based systems require specifications of the protocols/systems and create rules based on the specifications. These rules separate legitimate connections from the malicious ones. In such systems creation of the specification and coverage of the created rules are important issues which affect the performance of the IDS.

Hybrid intrusion detection systems consider advantages and disadvantages of the previous three detection techniques and aim to benefit from multiple of them at the same time. Of course such a decision may be costly in terms of the available resources.

3.7.1.2 Intrusion Detection Systems: detection resources

Intrusion Detection Systems can be divided into three categories in terms of the resources they use for detection. These are *network-based*, *host-based* and *hybrid* detection systems.

Network-based IDSes use the incoming and outgoing monitoring traffic to/from the network in addition to the internal traffic to detect the intrusions. Network-based IDSes can have a global view of the network and use it to boost the detection performance. However, such systems lack the information about the individual resource consumptions and logs of the nodes within the network which may be crucial for the detection of specific attacks.

Host-based intrusion detection systems consider the traffic only coming to and leaving from the host. Such systems monitor the resources and logs of the hosts as well which may provide hints about attacks. Since they work locally, they cannot have a global knowledge about the state of other nodes or the network which can be very useful to increase the performance of the IDS.

Hybrid IDSes combine the strengths of network-based and host-based systems that benefit from both network and node resources.

3.7.1.3 Intrusion Detection Systems: detection architecture

Architecture of Intrusion Detection Systems can be *centralized*, *distributed* or *hybrid*.

Centralized IDSes place the intrusion detection to a central location and all of the monitoring information has to be collected here. One of the main reasons to select a central point for intrusion detection can be the available resources. As mentioned previously, anomaly detection techniques can be resource-hungry and it may not be feasible to place them on every node due to resource-constraints. Therefore, a resource-rich node, such as border router, can accommodate the intrusion detection system. However, centralized systems come along with communication overhead since monitoring data has to be carried all the way to the central location. If malicious nodes prevent monitoring data from reaching to the centralized IDS, then they may achieve to mislead the IDS.

In *Distributed* IDSes, intrusion detection runs locally at every node in the network. In order to afford an IDS at every node, designers have to tailor the detection technique or algorithm according to the available resources. This approach clearly does not have any communication overhead, however the IDS has only local information to analyze in order to detect the intrusions.

Hybrid IDSes again harmonize both of the detection architectures and try to benefit from each of them as much as possible. In such systems, IDS is divided into modules and these modules are distributed along the network. Modules at every node can apply intrusion detection to a certain extend, may share less information (in comparison with centralized IDSes) with the centralized module and thus both reduce the communication overhead and enjoy the rich resources of the centralized module.

3.7.2 Intrusion Detection Systems proposed for Internet of Things

Cho et al. [47] proposed a botnet detection mechanism for 6LoWPAN-based networks in 2009. They assumed that the nodes in the IoT network use TCP transport layer protocol. Nearly seven years before the Mirai botnet, this study considered how IoT networks can be used as a botnet for DDoS attacks. The authors thought that, if there exists a malicious node on the forwarding path, then it can forge the packets and direct them to the target victim address based on the command of the bot master. A detection mechanism is placed at the 6LoWPAN gateway node which analyzes the TCP control fields, average packet lengths and number of connections to detect the botnets. The idea is based on hypothesis that the ordinary IoT traffic should be very homogeneous and botnet would cause significant deviations on the traffic.

Le et al. proposed a specification-based IDS [48] for IoT in 2011. It targets rank and local repair attacks. Their work assumes that a monitoring network is set up at the start of the network with minimum number of trustful monitoring nodes which has full coverage of the RPL network and has capability to do additional monitoring jobs. In this context, it has a distributed architecture and it is a network-based IDS. Each Monitoring Node stores the IDs, ranks and preferred parents of neighboring nodes. Monitoring Nodes accommodate a Finite State Machine of RPL with normal and anomaly states to detect the attacks. If a Monitoring Node cannot decide whether a node is an attacker or not, then it can ask the other Monitoring Nodes. This IDS

was not implemented and the authors did not specify the format of the communication between the monitoring nodes.

Misra et al. [49] proposed a learning automata based IDS for DDoS attacks in IoT. When there is an attack taking place, packets belonging to the malicious entities need to be sampled and dropped. This study aims to optimize this sampling rate by means of Learning Automata. Firstly DDoS attacks are detected at each IoT node in the network based on the serving capacity thresholds. When the source of the attack is identified, all of the nodes are informed about it. In the next step, each node samples the attack packets and drops them. This is when the Learning Automata solution comes to the scene. Sampling rate of the attack packets are optimized by means of the Learning Automata.

SVELTE IDS [37] was proposed by Raza et al. in 2013. It targets sinkhole and selective forwarding attacks. It has a hybrid architecture where it places lightweight IDS modules (i.e., 6LoWPAN mapper client and mini firewall module) at the resource-constrained devices and the main IDS (i.e., 6LoWPAN mapper, intrusion detection module and distributed mini firewall) at the resource-rich Border Router (BR). 6LoWPAN Mapper at the BR periodically sends requests to the mapper clients at nodes. Mapper clients reply with their ID, rank, their parent ID, IDs of neighbors and their ranks. Based on the collected information about the RPL DODAG, SVELTE compares ranks to find out inconsistencies. It also compares the elements of the white-list and elements of current RPL DODAG and uses nodes' message transmission times to find out the filtered nodes. The mini firewall module is used to filter outsider attackers. In addition to these, nodes change their parents with respect to packet losses encountered. In terms of the detection resources used, we can classify SVELTE into network-based IDSes. We can also put it into the category of specification/rule-based IDSes.

In the same year with SVELTE, Kasinathan et al. proposed a centralized and network-based IDS [40] and its demo [50] for 6LoWPAN-based IoT networks. The motivation of this study is the drawback of centralized IDSes which suffer from internal attackers. As mentioned in Section 3.7.1.3, internal attackers may prevent monitoring data from reaching the centralized IDS. This is due to the fact that monitoring data is sent through the shared wireless medium, which can be interfered by attackers. In

this IDS architecture, the authors place monitoring probes to the IoT network which have wired connections to the centralized module. Evaluation of their architecture was done via a very simple scenario where they used an open source signature-based IDS. A monitoring probe sends monitoring data under the UDP flood attack.

Amaral et al. [51] proposed a network-based IDS for IPv6 enabled WSNs. In the proposed scheme, watchdogs which employ network-based IDS are deployed in specific positions within the network. These nodes listen their neighbors and perform monitoring of exchanged packets. IDS modules at each watchdogs use rules to detect the intrusion attempts. These rules are transmitted to watchdogs through a dedicated channel. In order to dynamically configure the watchdogs, the authors used policy programming approach.

In 2015, Pongle et al. proposed an IDS [39] for wormhole attacks. Their IDS has a hybrid architecture similar to SVELTE. Main IDS is located at the BR and lightweight modules are located at the nodes. This study assumes that the nodes are static and the location of each node is known by the BR at the beginning. The main IDS collects neighbor information from nodes and uses it to find out the suspected nodes whose distance is found to be more than the transmission range of a node. The probable attacker is detected by the IDS based on the collected Received Signal Strength Indicator measurements related to the suspected nodes. In terms of the detection resource, we can consider this study as a network-based IDS. And from the detection technique point of view, it can be counted as a specification/rule-based IDS.

Another IDS proposed in 2015 was INTI [52] which targets sinkhole attacks. INTI consists of four modules. The first module is responsible for the cluster formation, which converts the RPL network to a cluster-based network. The second module monitors the routing operations. The third module is the attacker detection module, where reputation and trust parameters are determined by means of Beta distribution. Each node sends its status information to its leader node, which in turn determines the trust and reputation values. Threshold values on these parameters define whether a node is an attacker or not. The fourth module isolates the attacker by broadcasting its information. INTI can be classified as an anomaly-based IDS with distributed IDS architecture. In terms of the detection resources, we can put it into the category of hybrid IDSes.

Sedjelmaci et al. proposed an anomaly-detection technique [53] for low power and lossy networks in 2016. Unlike the other IDSes targeting specific attacks and aiming to detect them, this study focuses on the optimization of running times of detection systems. The motivation of the study is derived from the fact that anomaly-based systems require more resources compared to signature-based systems. If our system can afford to be a hybrid system, having both of the detection systems, then we have to optimize the running time of the anomaly-detection module in order to lengthen the lifetime of the network. The authors choose game-theory in this study for the optimization of the running time of the anomaly detection system. They claim that, thanks to the game-theory, anomaly detection runs only when a new attack is expected to occur. Anomaly detection runs only during such time intervals and create attack signatures. The signature-based system in turn puts this signature to its database and runs more often than the anomaly-detection system.

Mayzaud et al. proposed a detection system [54] for version number attacks in 2016. Their system uses the monitoring architecture which was proposed in the authors' earlier work [55]. Their monitoring system makes use of the multiple instance support of RPL protocol. It consists of special monitoring nodes with long range communication radios. These nodes are assumed to be covering the whole network and can send the monitoring information to the DODAG root using the second RPL instance that was setup as the monitoring network. In the proposed IDS, monitoring nodes eavesdrop the communication around them and send the addresses of their neighbors and addresses of the nodes who sends DIO messages with incremented version numbers to the root. The root detects the malicious nodes by means of the collected monitoring information. However, the proposed technique suffers from high false positives. This IDS can be counted as a network-based IDS with centralized detection architecture. We can also categorize it as a specification/rule-based IDS.

Another IDS proposed in 2016 was Saeed et al.'s work [56]. This study focuses on the attacks targeting a smart building/home environment where readings of sensors are sent to the server via a base station. In this study, the focus is on the attacks that target the base station. These attacks include software-based attacks and other attacks (i.e., performance degradation attacks, attacks to the integrity of the data). The anomaly-based with a centralized architecture is located at the base station. It consists

of two layers. The first layer is responsible for analyzing the behavior of the system and detecting anomalies. It uses Random Neural Networks to create the profile and detect the anomalies. The second part is responsible from the software-based attacks. It comes up with a tagging mechanism to pointer variables. Accesses with the pointer are aimed to be limited with respect to the tag boundaries.

Le et al. proposed a specification-based IDS [36] which is based on their previous work [48]. Their IDS targets rank, sinkhole, local repair, neighbor and DIS attacks. Firstly the proposal obtains an RPL specification via analysis of the trace files of extensive simulations of RPL networks without any attacker. After the analysis of the traces for each node, states, transitions and statistics of each state are obtained. These are merged to obtain a final Finite State Machine of RPL which helps them to find out instability states and required statistics. This study organizes the network in a clustered fashion. The IDS is placed at each Cluster Head which sends requests to cluster members periodically. Members reply with neighbor lists, rank and parent information. For each member, Cluster Head stores RPL related information. It runs five mechanisms within the concept of IDS. These mechanisms are understanding the illegitimate DIS messages and checks for fake DIO messages, rank inconsistencies and rules, and instability of the network. Cluster Head makes use of three thresholds to find out the attackers. These are number of DIS state and instability state visits, and number of faults. This IDS can be counted as network-based IDS with a distributed architecture.

Aris and Oktug proposed a novel IDS design [57] in 2017 which is an anomaly-based IDS with hybrid architecture. In this study lightweight monitoring modules are placed at each IoT nodes and the main IDS is placed at the BR. Monitoring modules send RPL-related information and resource information of the node. The main IDS module periodically collects the monitoring information and also works as a firewall, where it can analyze the incoming and outgoing traffic from and to the Internet. In this study, each IDS module working on different RPL networks can share suspicious events information with each other. Each IDS works autonomously and detects anomalies using the monitoring information of 6LoWPAN network, firewall information and suspicious events information. When anomalies are detected, nodes within the network are informed via white-lists, whereas other IDSes are informed via the exported

Table 3.4 : Categorization of Intrusion Detection Systems for IoT.

Proposal	Det. Arch.	Det. Technique	Det. Resource
Cho [47]	Centralized	Anomaly-based	Network-based
Le [48]	Distributed	Specification/Rule-based	Network-based
Misra [49]	Distributed	Specification/Rule-based	Network-based
SVELTE [37]	Hybrid	Specification/Rule-based	Network-based
Kasinathan [40]	Centralized	Signature-based	Network-based
Amaral [51]	Distributed	Specification/Rule-based	Network-based
Pongle [39]	Hybrid	Specification/Rule-based	Network-based
INTI [52]	Distributed	Anomaly-based	Hybrid
Sedjelmaci [53]	Distributed	Hybrid	Host-based
Mayzaud [54]	Centralized	Specification/rule-based	Network-based
Saeed [56]	Centralized	Anomaly-based	Network-based
Le [36]	Distributed	Specification/rule-based	Network-based
Aris [57]	Hybrid	Anomaly-based	Hybrid

suspicious events information. This anomaly-based IDS is a hybrid IDS in terms of architecture and the detection resources used.

Table 3.4 categorizes the IDSes for IoT. This table shows that majority of the systems are specification/rule-based. It clearly shows that, researchers focused on the protocols (i.e., RPL) rather than a common approach of creating a profile of normal behavior. This observation is also related to signature-based systems being rarely proposed for IoT. The reason may be due to the hardness of creating the signatures for the aforementioned attacks in IoT environments. In terms of the detection architecture, we can see that researchers consider every possible architecture and there is no outperforming option here. When we analyze the detection resources used, most of the studies are network-based IDSes. This shows that, node resources and logs are not yet used frequently by IoT security researchers. This may be due to the already limited resources of the nodes which may already be used 100% (e.g., RAM) or no space to store logs. But it is interesting to see that no proposal considers to use the deviation of the power consumption as an intrusion attempt.

Table 3.5 compares the studies in terms of target attacks and implementation environments. The majority of the attacks targeted by IDSes for IoT are routing attacks as shown in the table. A big portion of the studies focus only on a single attack, whereas only a few studies consider multiple attacks. When we consider these attacks, nearly all of them are insider attacks. This means, IoT security researchers in this concept are not

Table 3.5 : Target attacks & implementation environments of Intrusion Detection Systems for IoT.

Proposal	Target Attacks	Implem. Env.
Cho [47]	Botnets	Custom Simulation
Le [48]	Rank, Local Repair	Not-implemented
Misra [49]	General DoS	Custom Simulation
SVELTE [37]	Sinkhole, Selective-forwarding	Contiki Cooja
Kasinathan [40]	UDP flooding	Project Testbed
Amaral [51]	General DoS	Project Testbed
Pongle [39]	Wormhole	Contiki Cooja
INTI [52]	Sinkhole	Contiki Cooja
Sedjelmaci [53]	General DoS	TinyOS TOSSIM
Mayzaud [54]	Version Number	Contiki Cooja
Saeed [56]	Software-based attacks, Integrity attacks, Flooding and other	Prototype impl.
Le [36]	Rank, Sinkhole, Local Repair, Neighbor, DIS	Contiki Cooja

thinking of the threats sourced from the Internet yet. In addition to this, only one study targets software-based attacks. However, we know that embedded system developers choose programming in C language, which may open software-based vulnerabilities to the attackers targeting IoT. In terms of the implementation environment, Contiki Cooja is the environment selected by most of the researchers.

4. RPL AND VERSION NUMBER ATTACK

4.1 RPL

RPL [6] was proposed by the IETF as IPv6 routing protocol for low power and lossy networks. Formation of IEEE 802.15.4-based mesh networks was made possible by the RPL routing protocol, which constructs Destination Oriented Directed Acyclic Graphs (DODAG). A DODAG root creates a new RPL instance and lets other nodes to join the network by means of control messages. Nodes exchange control messages to be a part of the network and to set up upward and downward routes, which are routes towards to the root and routes towards to the leaf nodes respectively. Once the routes are set, nodes can start sending data messages according to the organization of their application program.

The RPL specification defines four types of control messages which are carried in the Internet Control Message Procotocol version 6 (ICPMv6) packets in IoT networks. These messages are DODAG Information Solicitation (DIS), DODAG Information Object (DIO), Destination Advertisement Object (DAO) and DAO-Acknowledgment (DAO-ACK). DIS messages are broadcasted by new nodes to obtain the information about the RPL instance in order to join the network. When a node sends a DIS message, the neighbor nodes reply with DIO messages which carry important information about the RPL network (i.e., DODAG ID, instance ID, rank and related information, version number, mode of operation, etc.). When a node receives DIO messages from its neighbors, it calculates/updates its rank and informs neighbors about its rank and the RPL information with a new DIO message. In addition to this, based on the rank of its neighbors, it selects the one with the lowest rank value as the preferred parent and informs that node with a DAO message. DAO messages are thus used to set up parent-child relationships based on the information obtained via DIO messages. When a node sends a DAO message, the receiving node can optionally reply back with a DAO acknowledgment message and thus a parent-child relationship is set up. An example

RPL network is shown in Figure 4.1. Note that, when a node changes its preferred parent, it sends a special form of DAO message, called as no-path DAO to inform its old preferred parent about such a change. In order to minimize the overhead of control messages, RPL uses Trickle Timer [58] to reduce the number of control messages created as network gets more stable. Nodes are expected to follow the rules of the RPL specification in order to create loop-free and efficient RPL DODAGs.

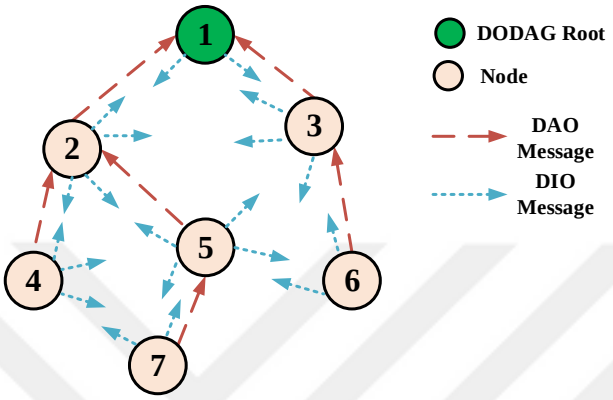

Figure 4.1 : An example RPL network.

Among the set of parameters carried in DIO messages, rank, version number and mode of operation are the most crucial ones.

In the RPL, position of a node which is the relative distance of a node from the DODAG root is named as *rank*. Rank is carried in DIO messages and it is calculated by each node based on the Objective Function and the rank of neighbor nodes. Objective Function types, include, but are not limited to, hop count, expected transmission count, remaining energy. RPL lets network administrators to select a suitable Objective Function based on the QoS requirements. Rank is used to create an efficient DODAG according to the application needs and to set up the child-parent relationship. For optimized and loop-free DODAGs, nodes have to follow the rules dictated in the specification.

In low power and lossy networks, faults and problems tend to occur. RPL has two repair mechanisms in order to recover from such issues and keep the DODAG healthy. These mechanisms are namely global repair and local repair.

The global repair mechanism sets up the complete DODAG from scratch and affects the whole LLN. The indicator of global repair operation is the *VN* parameter carried in DIO messages. DODAG root initiates the global repair operation by

incrementing/updating the VN. When it is updated, such an update is propagated within the DODAG via DIO messages. During this process, nodes re-calculate their ranks, re-create parent-child relationships and exchange several control messages to re-build the DODAG. The RPL specification [6] indicates that, only the root has right to change the VN and initiate the global repair operation. Moreover, the specification does not dictate any rule on when or in which condition to apply such an operation and leaves the decision to the implementers. Thus, we see different global repair implementations in different operating systems (e.g., Contiki, Riot, TinyOS) for IoT.

The local repair is an alternative repair solution of RPL which aims to solve the local inconsistencies. This mechanism cost less than the global repair mechanism since it involves a smaller portion of the network. If nodes find out inconsistencies (e.g., loops, packets with wrong direction indicators), then they start the local repair mechanism which consists of exchanging control messages and re-creating the parent-child relationships and getting appropriate ranks again.

In RPL, upward routes (i.e., the routes towards the DODAG root) are created by means of DIO messages, whereas downward routes (i.e., the routes from root towards leaf nodes) are created by DAO messages. RPL has a data path validation mechanism, in which headers of the IPv6 data packets carry RPL flags that indicate the direction of the packet (i.e., upward or downward) and possible inconsistencies with the rank of the previous sender/forwarder. When a node receives a packet with those flags indicating an inconsistency, it drops the packet and starts the local repair mechanism.

The Mode of Operation field carried in DIO messages signifies how RPL organizes routes in a DODAG. There are two types of Mode of Operations, namely storing and non-storing. In storing mode of operation, nodes store routing tables which contain addresses of their child nodes. Contiki 3.0 and earlier versions work in storing Mode of Operation by default. In non-storing Mode of Operation, which is also known as source routing, only the DODAG root stores the routes and it sends the route information at every packet, which increases the header size of the packets.

4.2 RPL Security

The RPL specification proposes the secure versions of the control messages which aim to provide confidentiality, integrity and authentication of the routing protocol. In terms of non-cryptography-based solutions for the RPL, usage of the geographical information, routing messages over multiple paths, dynamically selecting the next hop, redundant traffic generation, introducing quota, etc. were suggested by another RFC [59].

In terms of cryptography-based solutions, IPSec [23] may be considered for end-to-end security and IEEE 802.15.4 PHY and Link Layer Security [60] may be considered for hop-by-hop security. While ensuring confidentiality, authentication and integrity, both of the schemes have their own advantages and disadvantages. Apart from the standards-based solutions, other lightweight cryptography-based solutions proposed for WSNs and IoT can be considered for RPL-based networks too [61].

However, there are issues to consider about the cryptography-based solutions, especially in the context of LLNs [62] as we explained in Section 3.3. Firstly, it can be costly for resource constrained devices to employ cryptography. QoS of the applications and network lifetime may be affected by long-taking and costly cryptography operations. Secondly, since billions of nodes are expected to be deployed, it is highly possible that majority of the nodes will not be tamper-proof to reduce the cost, which makes it easy for attackers to grab the nodes, extract the security parameters, reprogram and apply the attacks. Moreover, lossy nature of the LLNs and resulting retransmissions, fragmentation of large packets for MAC layer, scalability of the solutions to hundreds/thousands of IoT nodes, number of message exchanges required and production, transfer and update of keys can highly affect the performance, success and acceptance of cryptography-based solutions for RPL-based IoT networks. Lastly, although algorithms are proven to be strong, correct implementation is not always ensured and many implementations suffer from vulnerabilities caused by improper implementations [27].

Finally, even though cryptography is used, networks and systems are still vulnerable to D/DoS attacks and this is applicable not only to RPL but to all networks. Therefore,

it is very important to accommodate lightweight and additional security solutions to protect LLNs from the attackers.

4.3 Denial of Service Attacks Which May Target RPL-Based Networks

We had categorized the D/DoS attacks which may target IoT networks based on the protocol stack in Table 3.1 and explained the attacks in Section 3.4. As it can be seen from the Table, the RPL can be the target of several DoS attacks which are either specific to the RPL or inherited from WSNs, MANETs, etc. [30, 31, 62].

D/DoS attacks which may target RPL-based IoT networks can be divided into two categories: RPL-specific and non-RPL-specific attacks. RPL-specific attacks are *rank*, *version number*, *local repair*, *DODAG inconsistency* and *DIS* attacks. Non-RPL-specific attacks are the ones which are already known from WSNs, and other communication networks research. Although they look old-fashioned, they are still applicable in RPL-based networks. These attacks are *sybil*, *sinkhole*, *selective forwarding*, *wormhole*, *cloneID* and *neighbor* attacks. We had explained the attacks which are not specific to RPL in Section 3.4.4.2. Whereas in this section, we will explain RPL-specific attacks.

The first attack we explain is the *DIS* attack where DIS messages which are broadcasted by new nodes to join the RPL network are maliciously used. Attackers can send unnecessary DIS messages in DIS attack [36], which causes the neighboring nodes to reset their DIO timers and send DIO messages frequently. Thus, the attacker forces nodes to generate redundant control messages and consume more power.

The rank is a very crucial parameter of the RPL protocol which represents a node's position within the DODAG as we mentioned previously. For optimized and loop-free DODAGs, nodes have to follow the rank rules (i.e., minimum hop rank increase, etc.). However malicious nodes may use rank in various ways to apply D/DoS attacks. In [36] and [41], an attacker node selects the neighbor with worst rank as a preferred parent instead of choosing the one with the best rank. Thus an inefficient DODAG is created which causes delays and excess number of control messages. In [48], the attacker intentionally skips applying the rank check which breaks the rank rule constructing the loop-free parent-child relationships.

The data path validation mechanism of RPL can be the target of DoS attacks too. In the ordinary operation, when a node receives a packet with those flags indicating an inconsistency, it drops the packet and starts the local repair mechanism. In *DODAG inconsistency* attack [44], attackers can set the corresponding flags of a data packet before they forward it and force the receiver node to drop the packet and start local repair.

In DODAG inconsistency attack, local repair operation is started by the nodes who are affected from the attack as a result of the attack. However malicious nodes can start local repair when there is no need so as to misuse the resources. This type of attack is called *local repair* attack [36, 48].

The last RPL-specific attack is the *Version Number Attack* that we will explain in detail in the next section.

4.4 RPL Version Number Attack

RPL has two repair mechanisms to keep the DODAG healthy. One of them is the global repair operation where the entire DODAG is re-created. According to the RPL specification, only the root can initiate the global repair mechanism by incrementing the VN parameter. As we mentioned earlier, every DODAG has a corresponding VN which is carried in DIO messages. Nodes in a RPL network find out the global repair operation by checking the VN in the incoming DIO messages. When a node realizes that the incoming DIO has a newer VN, it starts global repair operation assuming that the DODAG root started the network-wide global repair operation. As a result of global repair, nodes break all parent-child relationships, re-calculate their ranks, reset DIO timers (i.e., trickle timer) and exchange several control messages to set up the new DODAG.

There is no mechanism in RPL which guarantees that only the DODAG root can change the VN field. As we explained earlier on, RPL networks can make use of security mechanisms to protect both from insider and outsider threats. However, even though cryptography is used, malicious insider nodes can still change the VN and force the entire network to set up the DODAG from scratch [10, 42]. This attack is called as the *Version Number Attack*.

5. DETRIMENTAL EFFECTS OF THE VERSION NUMBER ATTACK ON RPL NETWORKS

In the literature, the VNA has been studied in only a few works since it is a pretty new type of a DoS attack for RPL-based LLNs. The studies either analyze the effect of the attack or secure the VN parameter, or propose a detection system. In this chapter, we study the RPL VNA in-depth and analyze the attack from various points of views. The unique aspects of our work can be seen in our analysis of a realistic network topology that has both static and mobile nodes with different cardinalities for which our inspiration came from the IETF routing requirement documents. We also analyze how the VNA affects the power consumption of the nodes. We incorporated a probabilistic attacking model where the attacker attacks with a probability of p (e.g., 0, 0.3, 0.5, 0.7, 1). We also provide the performance results with respect to various values of p. The rest of the chapter is organized as follows: In Section 5.1, we briefly explain the former study on the effects of the VNA. In Section 5.2, we propose our in-depth analysis, simulation environment and simulation results.

5.1 Literature Review on the Effects of the RPL Version Number Attack

In terms of studies that aim to analyze the effects of the VNA, there exists only one study proposed by Mayzaud et al. [10]. They studied the effects of a single attacker in a grid topology at varying positions. They realized that a correlation exists between the position of the attacker and success of the attack in terms of RPL control message overhead and packet delivery ratio. According to the study, if the attack is applied from positions which are relatively far from the DODAG root, effect of the attack is stronger in comparison with the attacks applied from closer positions.

5.2 Effects of the RPL Version Number Attack

5.2.1 Realistic topology and attacker model

In this study, we analyzed the VNA in a realistic scenario according to various points of view. In order to simulate a realistic scenario and understand how LLNs are expected to take place in application areas, we reviewed the IETF routing requirements documents for urban [63], industrial [16], building [15] and home [14] environments. According to the common characteristics shared by the application areas, we created a topology as shown in Figure 5.1. The topology consists of 44 nodes, where 4 nodes are mobile (nodes $41-44$) and rest are static (nodes $2-40$). The node with ID 1 is the border router and is the root of the DODAG. The initial form of the DODAG before mobile nodes start to move is also given in the figure. Twenty four nodes are placed on a grid topology with 50m distance between. Fifteen nodes which are located at the right top corner of the topology are randomly placed with different node densities. Nodes in the grid topology represent the nodes placed in a factory building or work place. The nodes that are placed at top right corner represent sensor nodes placed on a production line or a machine. The mobile nodes represent the vehicles or people moving in the area with wearable devices. Both static and mobile nodes measure temperature data and send it to the root periodically.

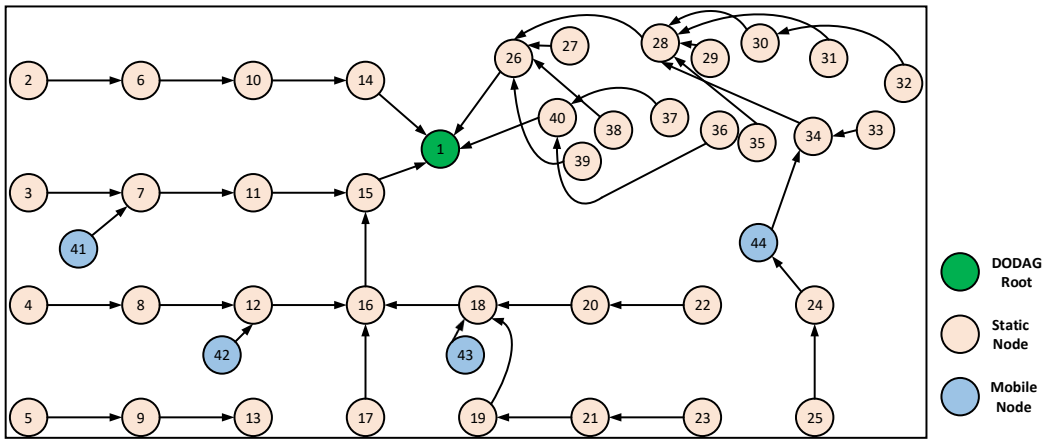

Figure 5.1 : Network topology.

The attacker node in our simulations is exactly the same as the other nodes except for sending DIO messages with changed VN with a probability of p (0, 0.3, 0.5, 0.7,

1). We believe that an intelligent attacker would want to go undetected easily but still would like to harm to the network as much as possible. With that reason, it might decide to attack with different probabilities.

5.2.2 Simulation environment and parameters

In our simulations, we used the Cooja simulator [8] of the Contiki Operating System version 2.7. [9]. In Cooja, we used emulated Tmote Sky motes [64]. Each node sends temperature readings to the root periodically (i.e., every minute). We record the traffic and other logs for 50 simulation minutes. The mobile nodes start to move at 3rd minute according to random way point mobility model. The simulation environment has distance-based loss model with no external interference.

In order to analyze the effect of the attack in our realistic simulation scenario, we firstly ran simulations without any attack so as to get the baseline performance results. Then we analyzed the effect of the attack with *p=1* with respect to the location of the attacker by changing its position in the network. To do that, we randomly selected some nodes (i.e., nodes $2, 5, 12, 14, 19, 25, 32, 36, 40, 41$ and 44). While choosing the nodes we paid attention to choose nodes at different positions with respect to the root, nodes with different mobility patterns(i.e., static, mobile) and with different cardinalities (i.e., nodes in the grid or dense segment of the network). After simulations with *p=1*, we applied the same procedure for *p=0.7*, *p=0.5* and *p=0.3*. For each attacking position, we ran 10 simulations and calculated performance metrics with 95% confidence interval.

5.2.3 Performance metrics and simulation results

Our performance metrics are average packet delivery ratio, average delay (i.e., average network delay), control traffic overhead and average power consumption. We used the total received and total lost packets that carry the temperature information to calculate the average packet delivery ratio. In order to calculate the average network delay, we divided total network delay by total packets received at the root. We do not consider the lost packets. The control traffic overhead constitutes the total DIS, DIO, DAO and DAO-ACK messages sent by the nodes. We take the average of every nodes' power consumption according to Powertrace logs and nominal operating condition values

of Tmote Sky as in the work of Raza et al. [37]. We give the simulation results in Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5 for the packet delivery ratio, average delay, RPL control packets overhead and average power respectively.

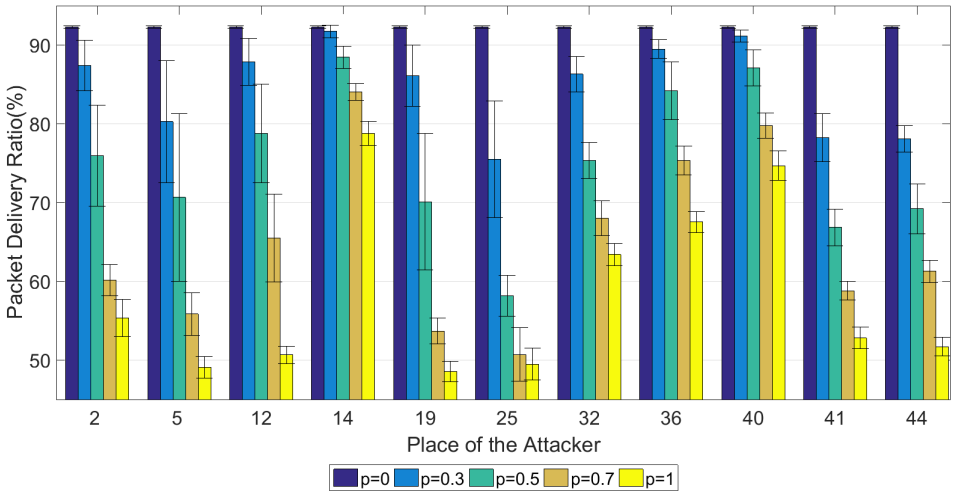

Figure 5.2 : Packet delivery ratio results.

In terms of the packet delivery ratio, shown in Figure 5.2, the baseline delivery ratio is 92% when there is no attack (i.e., *p=0*). As the probability of the attacking increases, the delivery ratio drops down clearly. For the nodes 14 and 40, the ratio is better than the rest of the attacking places and probabilities. The worst delivery ratios were obtained when the attacker is positioned at 5, 19 and 25. Our results validate the work of Mayzaud et al. [10], in the way that the attack is more effective when the attacker is located far from the root. The underlying reason is that, when the attacker is far from the root, it takes longer time for the root to realize that the version number has been changed. Until the time that root finds out the change, rest of the nodes get affected from the attack. Whereas, when the attacker is closer to the root, it takes less time for the root to realize the change. When we analyze the effect of the attack in dense segments of the network, our results do not validate Mayzaud et al. in terms of the attack and the neighbor count correlation. In our simulations, among the attacking positions, node 36 is the one with the highest number of neighbors (e.g.,11). But it has the third best delivery ratio after nodes 14 and 40. The same observation applies to the node 40 as well. It has nine neighbors and in comparison with the node 14, which is close to the root and has only three neighbors, node 40 does not have significantly lower delivery ratios. For the topology given in Figure 5.2, nodes 5, 19 and 25 are the nodes with the worst delivery ratios. When we consider the mobile nodes (i.e., nodes

41 and 44), we see that the mobile attackers' performance results are very similar to these nodes' results. Therefore, we observe that the version number attack performed by the nodes far from the root and by the nodes that are mobile have nearly the same detrimental effects on packet-delivery ratio.

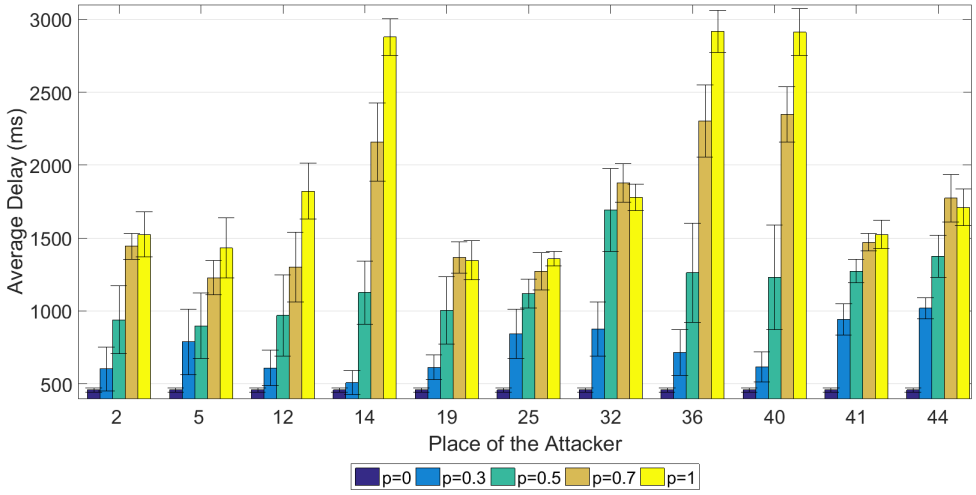

Figure 5.3 : Average network delay results.

The average delay (i.e., average network delay) results show a similar pattern to the packet delivery ratio, the higher the values of p, the longer the average delay gets. According to the simulation results shown in Figure 5.3, this observation applies to the majority of the attacking cases. However, when the attacker is located at nodes 19, 32 and 44, attacking with p=*0.7* causes longer average delays than other p values. The version number attack causes the average delay to be nearly 6 times longer in the worst case. As in the article of Mayzaud et al., the average delay values do not relate to the attacker's position due to channel conditions, routing decisions according to changing expected transmission count values.

The control packet overhead results shown in Figure 5.4 are very similar to the packet delivery ratio results. The overhead increases by 7500% when the attacker attacks from the position of the node 25 with the probability of 0.7. Similar to the packet delivery ratio, attacking from the nodes 5, 19 and 25 affects the performance the worst. The mobile attacker results are almost as bad as the results for the leaf nodes.

The power consumption of the nodes is another important performance metric when we target LLNs where nodes have battery constraints. As shown in the Figure 5.5, the version number attack can effect the power consumption of the nodes significantly.

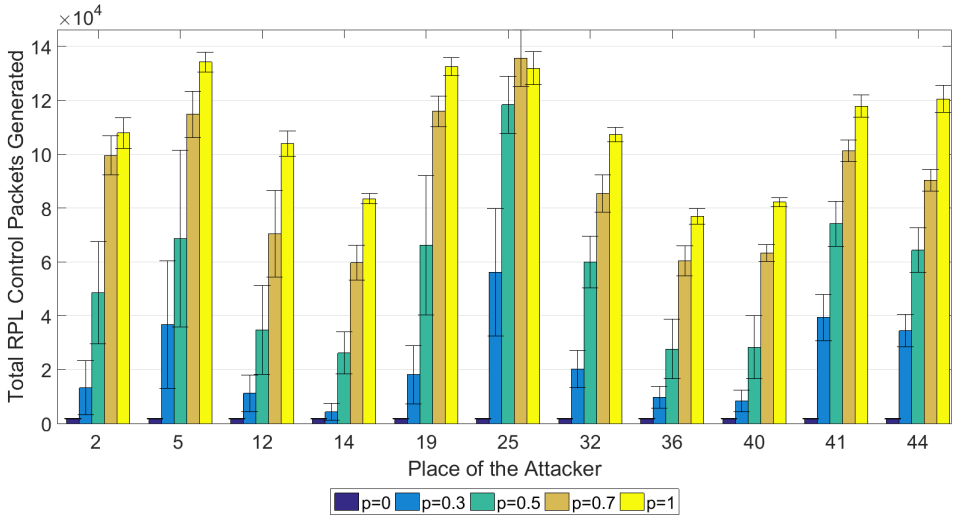

Figure 5.4 : Control packet overhead results.

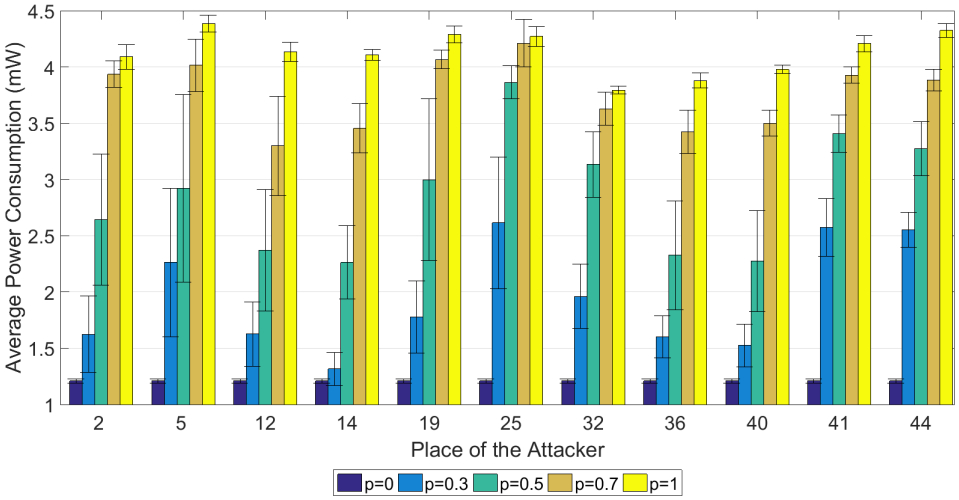

Figure 5.5 : Average power consumption results.

In the worst case, the average power consumption of the nodes increases by 265% and almost reduces the average lifetime of the nodes to 14 of the usual. According to the results, we can not observe any correlation between the power consumption and placement of the attacker as for the average delay results. The mobile attackers, as for the previous performance metrics, are very effective in terms of power consumption and approximately have the worst results.

5.2.4 Conclusion of the section

The simulations show that the packet delivery ratio and the control packet overhead results are highly correlated to the location of the attacker. In terms of the power consumption and the average network delay, the location of the attacker is not the major factor in the success of the attack.

The attacking probability, p, analysis show that, for the higher values of p, performance of the network degrades the worse. With this reason, the attacker has to make a decision whether to affect the performance clearly and be detected easily by an existing mechanism or affect the performance slightly with small values of p and increase the chance of going undetected.

In terms of the mobility, simulation results show that the mobile attackers have nearly the same detrimental effects on the network as the nodes far from the root. According to these results, securing the mobile entities in a LLN is very crucial. If an adversary achieves to place a malicious node onto a mobile network entity, then it can easily drain the resources and reduce the network lifetime.

6. NEW LIGHTWEIGHT MITIGATION TECHNIQUES FOR RPL VERSION NUMBER ATTACKS

In this chapter, we present two lightweight mitigation techniques for RPL VNA which affect the performance of IPv6-connected LLNs detrimentally as we analyzed in the previous chapter. Simple yet effective mitigation mechanisms proposed in this chapter promise significant performance improvements in an RPL network under attack. By means of the proposed techniques, the delay caused by the attacker can be shortened up to 87%, the average power consumption can be reduced up to 63%, the control message overhead can be lowered up to 71% and the data packets delivery ratio can be increased up to 86%. The proposed techniques, while allowing the ordinary RPL operation, trade off the mitigation performance against the resource overheads and thus allow network administrators to choose the right scheme for their RPL network.

The rest of the chapter is organized as follows. In Section 6.1, we review the studies on security, detection and mitigation mechanisms for RPL VNA. In Section 6.2, we analyze the severity of the attack on LLNs. Section 6.3 presents the novel lightweight mitigation techniques, whereas Section 6.4 evaluates the performance of the proposed mechanisms. We give additional analyzes in Section 6.5 and finally in Section 6.6, we conclude the chapter.

6.1 Literature Review on Security, Detection and Mitigation Mechanisms for RPL Version Number Attack

In the literature, there exist a myriad of studies on different security aspects of IoT and RPL specifically. RPL can be the target of various DoS attacks and many studies including [30, 31, 62, 65, 66] proposed overview and taxonomy of attacks against RPL. In addition to this, several IDSes were proposed for RPL-based IoT networks as well as for WSNs [67, 68]. A review of IDSes for IoT can be found in the study of Zarpelão et al. [69] and in our previous study [62]. Cryptography-based security solutions for WSN and IoT were reviewed and compared in the work of Nguyen et al. [61].

Security of RPL started to take the attention of more and more researchers and recently proposals addressing different attacks were published including [70–72]. However, we cannot see many studies targeting VNA. In the literature, the VNA has been studied in only a few works since it is a pretty new type of a DoS attack for RPL-based LLNs. The studies either analyze the effect of the attack or secure the VN parameter, or propose a detection system. Since we are proposing novel mitigation techniques for VNA, we analyze only the corresponding proposals in this section in detail. The categorization of the studies that target RPL VNA is shown in Table 6.1.

Table 6.1 : Categorization of the works related to RPL Version Number Attack.

Work	Type*	Finding / Security Scheme / Detection Technique	Implementation / Simulation Environment	Legitimate VN Updates Considered?
Mayzaud et al. [10]	AS	Attacker location - success of the attack correlation	Contiki Cooja	No
Arış et al. [42]	AS	Attacker location - success of the attack correlation, detrimental effect of the mobile attackers	Contiki Cooja	No
Dvir et al. [43]	SS	Hash chains and Message Authentication Codes to secure VN	Unimplemented	Inherently
Perrey et al. [73]	SS	Digital signatures are used to secure VN	RIOT and testbed	Inherently
Nikravan et al. [74]	SS	Identity-based Offline-Online Signature Based scheme to secure VN	Unimplemented	Inherently
Firoz et al. [75]	DS	Cooperative verification-based detection system	Contiki Cooja	No
Mayzaud et al. [54]	DS	Distributed monitoring-based detection system	Contiki Cooja	No

* (AS: Analysis Study, SS: Security Study, DS: Detection Study.)

Mayzaud et al. [10] and Aris et al. [42] analyzed the effects of the VNA. The former study found correlation between the position of the attacker and success of the attack in terms of RPL control message overhead and packet delivery ratio. According to the study, if the attack is applied from positions which are relatively far from the DODAG

root, effect of the attack is stronger in comparison with the attacks applied from closer positions. Aris et al. verified the findings of Mayzaud et al. and also showed that mobile attackers can be as much disastrous as the farthest attacking positions in an RPL network.

As a security study, there are three studies which consider securing the VN field in DIO messages by means of cryptographic mechanisms. Dvir et al. [43] proposed the VeRA which makes use of hash chains and Message Authentication Codes to secure the rank and VN fields carried in RPL DIO control messages. Although the VeRA sounds promising for the security of the two important fields, implementation of the scheme is missing and evaluation has not been made in terms of resource overhead and network performance. It was also considered to be vulnerable to a number of rank attacks by [73] and [74]. Perrey et al. [73] proposed TRAIL for VN and rank attacks. In terms of the VN attack, TRAIL uses digital signatures to secure the VN, however the authors do not give any details on the accommodated signature scheme. In addition to this, TRAIL seems to have scalability issues as the number of nodes in the LLN increases. The other security study was presented by Nikravan et al. [74] which makes use of Identity-based Offline-Online signature based scheme. The idea is to secure the VN and rank fields of DIO messages by means of signatures. In their approach, the DODAG root generates the private keys for the nodes in the LLN. The nodes use private keys to generate their offline signatures, which are used to sign the aforementioned fields in the DIO messages in the online phase. Nodes sign and unsign the fields in DIO messages and verify the senders by means of operations like point multiplications, exponentiations and pairings. Although this study sounds very promising, several issues to consider exist. First of all, performance of the proposed scheme was analyzed only quantitatively without any implementation or realistic simulation. Therefore, RPL operation with the proposed technique is questionable in terms of the long computation times it requires, the static network topology it assumes, the secure communication channel it needs within the LLN to deliver the private key and the extra space overhead it will cost to store signatures on the max. 127B transmission unit of IEEE 802.15.4 protocol. In addition to these, the proposed technique may cause high false positives and negatives which were not analyzed.

Firoz et al. [75] proposed a cooperative verification mechanism to detect the VNA. In their scheme, nodes exchange verification messages with their two-hop neighbors and identify the attacker by collecting the VN information from them. Although cooperative verification idea sounds promising, several issues exist for this study. Firstly, no evaluation exists for the extra communication overhead, RAM and ROM requirements of the proposed mechanism. Secondly their scheme requires every node to know the addresses of their two hop neighbors. Additionally, the proposed mechanism itself can be source of a DoS attack where a malicious node redundantly sends verification request messages misusing the resources of neighbors. Mayzaud et al. [54] proposed a VNA detection mechanism which uses a distributed monitoring architecture proposed by the authors in one of their earlier study [55]. It is assumed that the monitoring architecture covers the whole network and consists of nodes with longer range radios. In the detection mechanism, if nodes send DIOs with greater VN, then the monitoring nodes report them to the centralized detection module. Based on the reports, the nodes which were reported by multiple monitoring nodes are put to the malicious nodes list. This mechanism suffers from high false positives and the authors try to decrease it by node placement strategies.

As shown in the Table 6.1, analysis studies and detection studies targeting VNA use Contiki and its Cooja simulator for implementation and test purposes. Among the cryptography-based security schemes, only the work of Perrey et al. [73] was implemented and tested. However, rest of the security schemes were not implemented and they provided only the analytical performance results. As the security schemes aim to secure the VN field in the DIO messages by means of cryptography, they inherently consider the legitimate VN updates. On the other hand, in terms of the detection studies, we would like to emphasize that none of the detection mechanisms considers legitimate VN updates initiated by the root, which may affect the performance of the aforementioned studies significantly.

6.2 On the Severity of the RPL Version Number Attack

In the previous sections, we had explained the global repair operation of the RPL, relation of Version Number parameter carried in DIO messages with the global repair. We had also mentioned how RPL specification points out the ordinary operation global

repair and how it does not dictate any rule on when or in which condition to start global repair and it leaves the decision to the implementers. We name the action of a malicious node changing the VN intentionally and initiating an illegitimate global repair, thus forcing the rest of the nodes to rebuild the DODAG as VNA.

RPL VNA is a very serious DoS attack which can threaten RPL-based LLNs. This statement can be validated by the findings of the studies such as our study [42] (we explained it in the previous chapter) and [10], which show that, VNA can have catastrophic effects on node resources, network and application performances. VNA can cause IoT applications to suffer from huge packet loss (up to 30% [10], 50% [42]), increased delays (almost 2x [10], 6x% [42]). Malicious global repair operations initiated by the attacker can cause excess amount of RPL control messages to be generated (almost 18x [10], 75x% [42]) and therefore increased power consumption results (almost tripled [42]). Hence, RPL VNA clearly threatens RPL-based IoT applications, wide-range acceptance of RPL and investments.

VNA stems from the fact that there is no mechanism in RPL which guarantees that VN can be incremented only by the DODAG root. At first glance cryptography-based security solutions come to the scene to provide such a guarantee. However, considering the issues mentioned earlier on this thesis on cryptography-based solutions and existing studies, we can not claim that this problem is over. Intrusion Detection Systems may be another option. However, such systems need to be lightweight, provide high accuracy and do not inherently cause new DoS attacks to be mounted. Lightweight and efficient mechanisms which will work with RPL in harmony and which will efficiently protect from or detect or mitigate VNA are required to secure the RPL networks against such attacks.

6.3 Novel Mitigation Techniques for RPL Version Number Attack

In this section, we propose two novel mitigation techniques against RPL VNA with varying mitigation performances and resource requirements. The mitigation techniques were inspired from our earlier work [42], where we analyzed the effect of the VNA with a probabilistic attacker model in a factory environment scenario. In that study, after exhaustive simulations, we realized that when the attack was applied from positions far from the root, the effect of the attack for two performance metrics (i.e.,

packet delivery ratio and RPL control message overhead) was clearly higher. We would like to remind the readers that, Mayzaud et al. [10] had proposed the same findings, which signifies that the strongest positions for an attacker to affect the performance of the network are positions that are at the side of leaf nodes. These findings led us to a conclusion that, if one wants to mitigate the effect of the VNA, it would be crucial to filter the VN updates coming from the leaf nodes' direction. This observation is also supported by the RPL specification in a way that only the root has right to change the VN, which means such an update has to come from the opposite direction of the leaf nodes. Thus our first mitigation technique is formed, where each node eliminates the VN updates coming from the direction of the leaf nodes. However, even though the strongest attacking positions are eliminated by this way, this technique cannot mitigate the effect of the VNA applied from the rest of the attacking positions. Therefore, based on the first mitigation technique we employed a trust mechanism that relies on changing the VN based on the observation of majority of the neighboring nodes with better rank values as the second mitigation technique.

6.3.1 Mitigation technique I: elimination

In the first mitigation technique, a node expects VN updates coming from the direction from DODAG root towards leaf nodes. Therefore, when it receives a DIO message with a greater VN, it checks the sender and if it is not coming from the nodes closer to the root than itself, it eliminates such updates by discarding the DIO as shown in Figure 6.1. Here, it uses the rank information which signifies the relative position of a node to the root in a RPL network. It eliminates the VN update if sender's rank is better (i.e., has a lower value) than its rank.

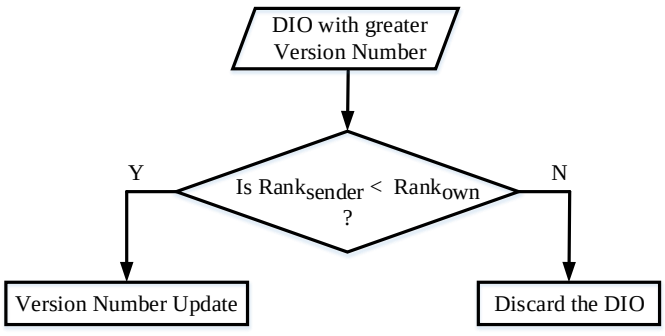

Figure 6.1 : Elimination technique.

The Elimination technique is very simple and can be implemented by a single additional comparison operation, which makes it very suitable to implement for resource-constrained devices. Since VNA affects the performance of the network more when applied from the positions close to the leaf nodes, this technique eliminates such malicious updates disseminating to the whole RPL network. However, if an attacker attacks from other positions, it can still misuse the resources of the network and the nodes. Therefore, although the Elimination technique can be hypothesized to mitigate the effects of the VNA to a certain extent, a better solution is needed which addresses every attacking position.

6.3.2 Mitigation technique II: shield

The second mitigation scheme, which we call as Shield, is an improved version of the Elimination technique. The idea behind the Shield is not to trust any single VN update information coming from the nodes with a better rank. Instead, expect majority of such nodes to apply such an update if this is a legitimate VN update. For this purpose, every node in an RPL DODAG keeps a list of neighbor nodes which have better rank than itself in a table called as ShieldList. For each such node, it stores the neighbor identity information (i.e., IP address) and the observed VN fields as shown in the Figure 6.2. The figure shows a sample DODAG employing the Shield Mitigation Technique. As shown in the Figure 6.2, Node 6 has three neighbors, which are Node 3, Node 5 and Node 7 (assuming only these nodes are in the transmission range of Node 6). Assuming that only Node 3 and Node 5 have better rank values than Node 6, it stores the identity and VN information of only these nodes and not Node 7.

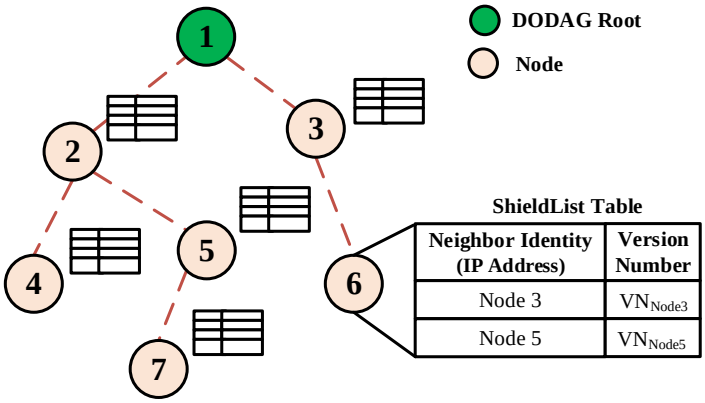

Figure 6.2 : ShieldList table structure in a RPL DODAG.

Every node in a RPL DODAG has to maintain its ShieldList by means of analyzing the incoming DIO messages. The rank information of nodes may change depending on the mobility pattern or changes in the topology (e.g., errors, routing holes, rank calculated according to remaining energy based Objective Function, etc.) and therefore a node may perform addition or removal of nodes on the table according to the up to date rank information.

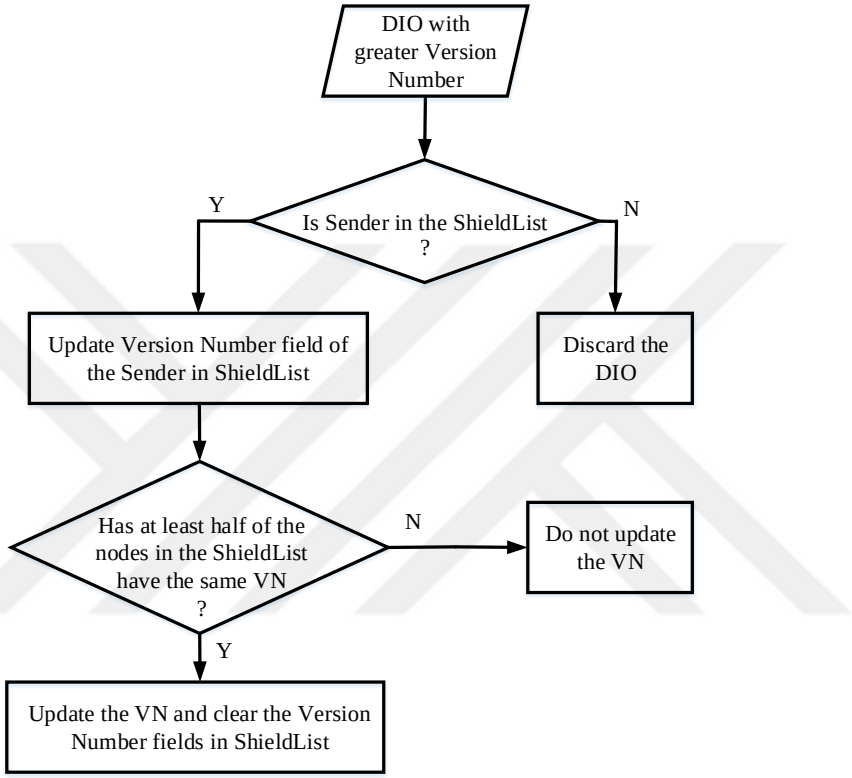

Figure 6.3 : Shield technique.

The operation of the Shield mechanism is outlined in the flow chart in Figure. 6.3. If a node receives a DIO with a greater VN than the current VN, it checks whether the sender exists in its ShieldList table, since the neighbor nodes which are positioned at a higher hierarchy on the DODAG and which can disseminate the legitimate VN update information are already determined and stored in the ShieldList. If it does not exist in the table, then the DIO message carrying this new VN is discarded. On the other hand, if sender of the DIO message exists in the table, then the receiving node updates the Version Number field of the sender in its table and checks whether majority of the nodes in the table share the same VN information. If at least half of the table entries share the same VN update, it updates its VN. After updating VN and being part of the

global repair operation, the corresponding node clears the Version Number fields in the ShieldList table.

In comparison with the Elimination technique, Shield mechanism requires each element of a DODAG to keep a table that stores identity and VN information of the neighboring nodes with better rank values. ShieldList tables of every node is independent from each other and do not require any communication overhead to exchange tables or fill the table entries. Every node on the DODAG autonomously maintains its table and makes decisions whether to update its VN or not. However, maintaining the table and processing the flow chart requires additional processing power in the Shield Technique but in return, it promises a more robust mitigation performance.

At a first glance both of the novel mitigation techniques we propose in this study may look very simple. However, since we target LLNs which have resource constraints, proposed solutions to secure such networks have to be simple, not complicated and resource-friendly.

6.4 Performance Evaluations

In order to evaluate the performance of the Elimination and Shield techniques, we considered the attacker model, the legitimate VN updates, the simulation topologies and settings and also the performance metrics.

6.4.1 Attacker model

In this study we considered a single attacker which joins to the ordinary operation of the RPL but only initiates malicious VN updates. The attacker in our study starts to attack a certain amount of time after the DODAG gets stable. The attacker can apply the attack from any position in the topology. Whenever it sends a DIO message with respect to RPL Trickle Timer mechanism, it increments the VN in the DIO message. In this respect, we can refer such an attacking behavior as an almost constant attacking model. Such an attacker model may sound very simple. However, we would like to inform the readers that, we had analyzed a probabilistic attacker model (i.e., attacking with respect to probability of p where $p \in \{0, 0.3, 0.5, 0.7, 1\}$) in our prior work [42]. Based on the analysis, we realized that the higher the probability of attacking, the

stronger the effect of it. Since we are not proposing a detection system in this study, it is very reasonable to choose the strongest attacking settings in order to evaluate the mitigation performance of our novel mitigation techniques. In this study, we assume that the DODAG root is secure and not compromised.

6.4.2 Legitimate version number updates

As we explained previously, the RPL specification does not dictate any rule on when or in which condition to apply a global repair operation and leaves the decision to the implementers. With this reason, we checked the RPL implementations of TinyOS [76], RIOT [77] and Contiki [78] operating systems for global repair operations. Although all of them change the VN when a node receives a DIO with a greater VN, only TinyOS has a periodic global repair operation with 60-minute periods. The remaining OSes do not have such an autonomous mechanism for the DODAG root. We also analyzed the CISCO RPL Configuration Guide [79] which specifies the default time interval for global repair operations as 30 minutes. Since the legitimate VN updates is an important issue to consider when targeting the VNA, we decided to employ global repairs initiated by the root with intervals according to the Negative Exponential Distribution. Hence, we simulate events happening according to the Poisson hits and causing the root to start global repair.

6.4.3 Simulation topologies

In this work, we considered grid and random node placement strategies to analyze how the mitigation performance changes with the topology characteristics. Based on the combinations of these node placement strategies, we formed four different topologies, namely Grid, Grid Corner, Grid Random and Random which are shown in Figure 6.4. All of the topologies consist of 36 static nodes and a root placed in a $150x150m$ area. The root is placed in the center in the Grid Topology and at the corner in the Grid Corner Topology. The Grid Random Topology is organized as a mixture of grid and random node placement strategies where half of the nodes are placed on the grid randomly and the remaining nodes are scattered randomly to the area. In the Random topology, random placement is applied to all nodes.

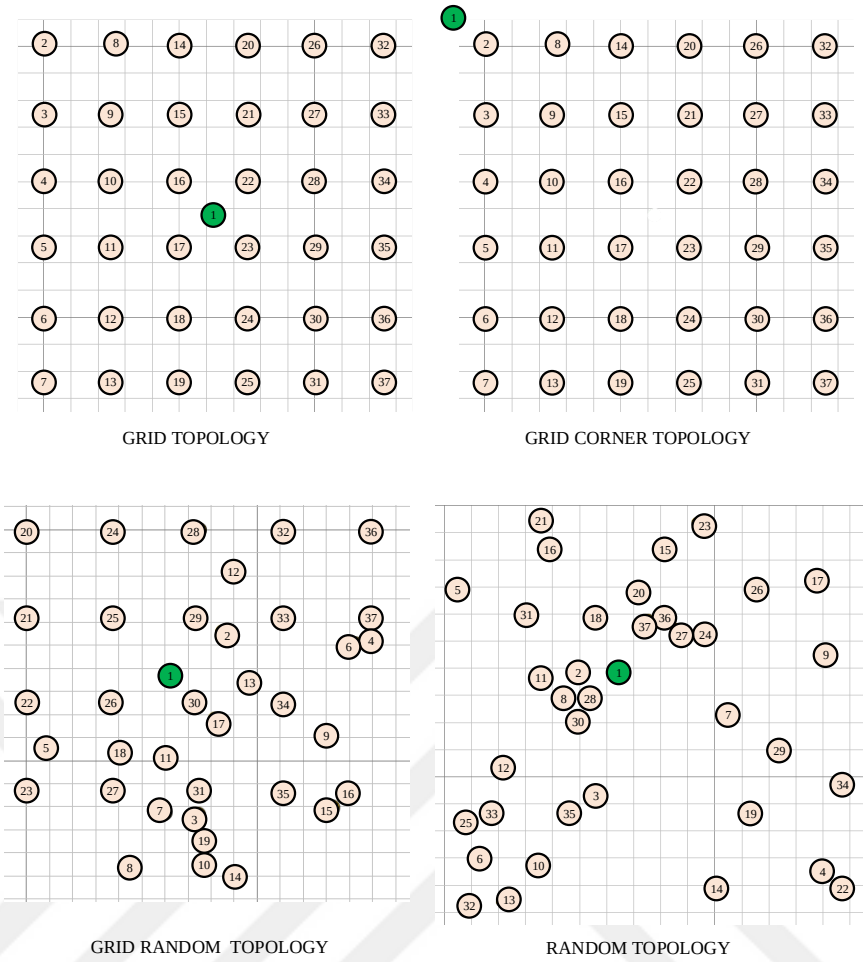

Figure 6.4 : Topologies for the simulation.

6.4.4 Performance metrics

The performance metrics employed can be presented in three classes:

- **Network Performance Metrics:** Average network delay, RPL control message overhead and data packets delivery ratio,
- **Resource Requirement Statistics:** Average power consumption, RAM and ROM requirements,
- **Accuracy Metrics:** True Positives, False Negatives, True Negatives, False Positives.

In our simulations, nodes send temperature readings periodically to the root by UDP packets. The average network delay is calculated using the delay of successfully received UDP packets at the root. RPL VNA misuses the global repair operations.

Since RPL uses control messages (i.e., DIS, DIO, DAO) to maintain the DODAG, VNA affects the number of generated control messages, which we count in this study. We calculate data packet delivery ratio by means of the ratio of successfully received data packets at the root.

The proposed techniques must mitigate the effects of the VNA by filtering malicious VN updates while ensuring the ordinary operation of RPL. This means that the root should be able to increment the VN to start global repair operation while the malicious nodes should not be allowed to. Therefore, measuring the accuracy of filtering the malicious and legitimate updates is crucial. If a node changes its VN based on the legitimate VN update, this behavior is tagged as True Positive (TP). If it considers the legitimate update as a malicious update and does not change its VN, it is tagged as False Negative (FN). On the other hand, if the incoming VN update is a malicious one and if our node does not change its VN, then we tag it as True Negative (TN). If it changes its VN by misinterpreting the malicious update, then it is tagged as False Positive (FP).

In resource overhead metrics, average power consumption is determined based on the average of the total power consumption of nodes (except for the root and the attacker). RAM and ROM requirements show how much RAM and ROM spaces are required to implement the proposed mitigation techniques besides actual RPL application.

6.4.5 Simulation environment and results

In order to evaluate the performance of the Elimination and Shield techniques, we placed the attacker in every possible position in each of the four aforementioned topologies and ran ten simulations for each attack position (i.e., in total 3x4x36x10 = 4320 simulations for attack cases and 3x4x10 = 120 simulations for attack-free cases) each lasting 50 minutes. We believe that the simulation duration is long enough to evaluate the performance of the network clearly, since the DODAG gets stable within the first three minutes. We measured the performance metrics and provided the results with 95% confidence interval.

The simulation settings are shown in Table 6.2. We employed Contiki Cooja [8] for our simulations. Contiki [9] is an open source operating system for Low Power and Lossy Networks-based IoT systems. We chose Contiki and its simulator Cooja

Table 6.2 : Simulation parameters.

Num. of nodes	36
Node Type	Tmote Sky
Operating System	Contiki 3.0
Simulator	Contiki Cooja
Radio Medium	Unit Disk Graph Medium: Distance Loss
Topology Dimensions	$150x150m$
Transmission Range	$50m$
Interference Range	$100m$
Simulation duration	50 minutes
Number of simulations	10 per attack position
Confidence interval	95%
Legitimate VN updates	5 updates per simulation
Per node traffic rate	1 packet per minute

because it provides the implementations of the standardized protocols (i.e., IEEE 802.15.4 PHY and MAC layers, 6LoWPAN adaptation layer, RPL routing protocol, IPv6, UDP and CoAP) which were proposed be the IEEE and the IETF for LLNs. The Cooja simulator provides emulations of well known WSN motes (e.g., Tmote Sky, MicaZ, MSP430-based motes) as well as several plugins for various purposes such as power profiling. Contiki and Cooja lets researchers to quickly create a network of resource-constrained motes running the mentioned protocols and modify the source codes so as to create and test new techniques and analyze the performance.

The emulated Tmote Sky [64] nodes in the topologies with $150mx150m$ dimensions run Contiki OS [9] versioned 3.0 with RPL-UDP application where each node sends a temperature reading to the root per minute. The transmission and interference ranges as well as the radio medium are the default settings of Contiki Cooja. Each node employs the standardized protocol stack [7] implementation of Contiki 3.0 where IEEE 802.15.4 radio, ContikiMAC and IEEE 802.15.4 Framer, CSMA, 6LoWPAN, RPL and IPv6, and UDP implementations constitute the protocol stack. As we explained previously, the default RPL implementation of Contiki 3.0 does not possess any global repair mechanism for the DODAG root. Therefore, in order to consider the legitimate VN updates we changed the RPL implementation of the DODAG root, so that in each simulation the root initiates five global repairs on average with the intervals determined based on the Negative Exponential Distribution.

6.4.5.1 Network performance results

The simulation results for different network performance metrics with respect to topologies are shown in Figure 6.5, Figure 6.6 and Figure 6.7 where attack-free RPL, RPL under attack, RPL with Elimination under attack and RPL with Shield under attack cases are compared. No attack scenario is also studied for each topology. For attack scenarios, we placed the attacker to every position one by one, ran ten simulations and measured the performance. We did this for each topology and for three different Contiki RPL UDP techniques. These are plain RPL, RPL with Elimination (*RPL+ELIMINATION*) and RPL with Shield (*RPL+SHIELD*) mitigation techniques. The results shown in the figures for RPL, RPL+ELIMINATION and RPL+SHIELD are the averages of the obtained results for each attacker position.

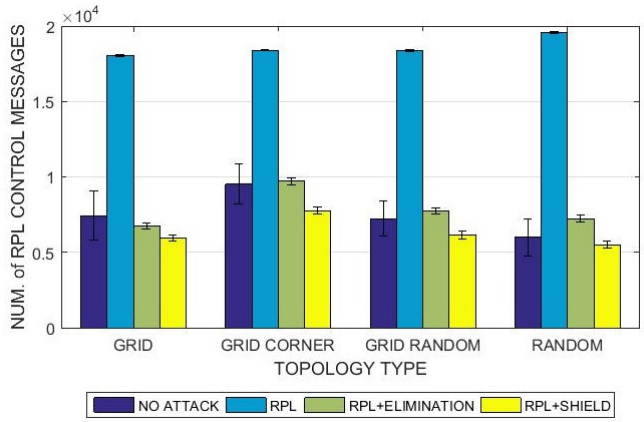

Figure 6.5 : RPL control message overhead results with respect to topologies.

The RPL control message overhead results are shown in Figure 6.5. We can see that, VNA causes excess amount of control messages to be generated in comparison with the attack-free case for all topology types. The number of control messages is almost doubled for Grid Corner topology, tripled for Random topology and in between for Grid and Grid Random topologies. The reason is the degree of node densities where random placement can form topologies with dissimilar node densities, whereas the node densities in a grid topology are more uniform. This influences the number of parent nodes and children nodes, where DODAG can be formed with less parent nodes serving to more children nodes as the degree of randomness in the node placement increases from grid to random. Hence more children nodes can hear the control messages of their corresponding parent nodes and thus DODAG can be created

and maintained by means of lesser number of control messages. However, when a malicious node changes the VN, such a change can be disseminated within the network quicker than the topologies with grid-based placements. In Figure 6.5, we can see the efficiency of our novel mitigation techniques in reducing the excess number of generated control messages. Both of the mitigation techniques mitigate the effect of the attack successfully but the best solutions were obtained by the Shield technique where the reduction is 57% for Grid Corner, 66% for Grid Random, 67% for Grid and 71% for Random topologies. The interesting thing is that, for all topologies RPL with Shield under attack generates even less control messages than the attack free RPL and this is true for the Elimination technique too but except for the Grid topology. The underlying reason for such an interesting behavior is the filtering of the control messages during the ordinary VN updates. During the legitimate VN updates, nodes change their VN only when DIOs with greater VN come from a specific direction (in Elimination) or when half of the ShieldList entries share the same updated VNs (in Shield). Hence some or a portion of legitimate DIO messages are filtered and this reduces the number of generated control messages. Although less control messages are generated and some legitimate messages are filtered in the attack-free case, nodes can still update their VNs according to legitimate updates and RPL keeps on operating as it will be shown in the Accuracy Results Table in Section 6.4.

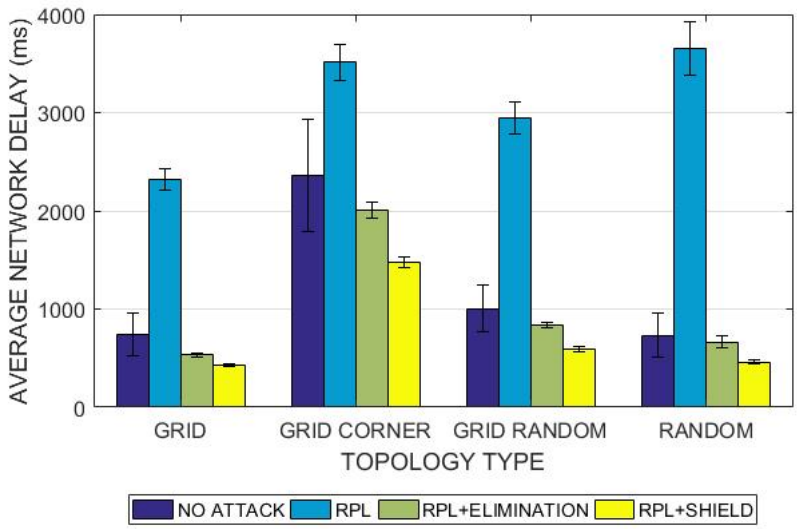

Figure 6.6 : Average network delay results with respect to topologies.

The average network delay results with respect to topologies are shown in the Figure 6.6. As shown in the figure, the effect of the VNA on the average network

delay gets greater proportional to the degree of randomness in the topology. The effect is greater for the Random topology in comparison with the Grid Random topology and that is greater than the effect for the rest of the grid topologies. The underlying reason is similar to the case for RPL control message overhead and related to the degree of node densities among topologies. Uniformity of node densities influences the number of neighbors and also the channel availabilities, which in turn affect the delay. Since we see longer paths in the Grid Corner topology, the delay is longer than all. As shown in the figure, up to fourfold longer delays are introduced due to VNA. Our mitigation techniques efficiently mitigate the effect of the attack on the delay. We can see that, even the simplest mitigation technique (i.e.Elimination) can mitigate the effects of the attack, whereas Shield technique again provides the best outcomes. Shield reduces the excessive amount of delay caused by the attacker by 58% for Grid Corner, 79% for Grid Random, 81% for Grid and 87% for Random topologies. The average network delay results also present an interesting outcome similar to the case in RPL control message overhead that, the proposed techniques provide delays even shorter than all of the scenarios without attack. The reason is again related to the filtering effect of the proposed techniques, which cause nodes to generate less control messages and keep the channel less busy. By this way, average network delays even shorter than the attack-free cases can be obtained.

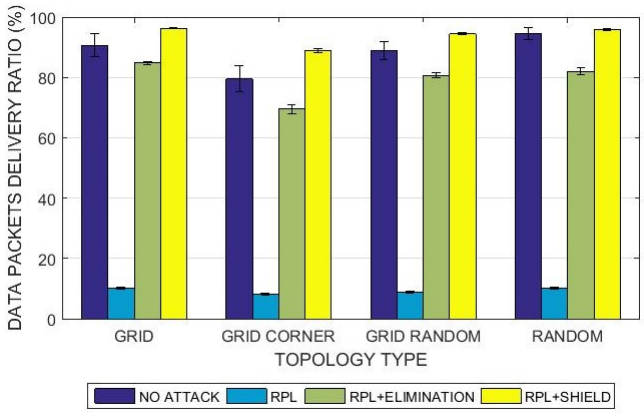

Figure 6.7 : Data packet delivery results with respect to topologies.

The data packet delivery ratio results are shown in Figure 6.7. We can see that the VNA causes detrimental effects on the packet delivery ratio, which drops down to 8% for Grid Corner and Grid Random, and to 10% for Grid and Random topologies. These results show that, RPL-based IoT applications can suffer from VNA as the attack

causes around 90% of application packets cannot be delivered successfully. In terms of the attack-free case, majority of the packet loss stems from MAC layer issues related to reaching the max. retransmissions due to collisions or without collisions, not being able to allocate space because of either a full packet queue or a full neighbor queue. However, when we analyzed the lost packets in attack cases, we found out that majority of packets are dropped at the routing layer due to routing errors introduced by VNA. Illegitimate global repairs forced by the attacker causes nodes to break and to try to re-establish routes and during this process packets are dropped due to non-existing routes. The proposed mitigation techniques however, mitigate the effect of the attack effectively. The packet delivery ratio under the VNA can be increased by 61% to 74% with the Elimination technique and 80% to 86% with the Shield technique. Once again, Shield provides a delivery ratio even better than the attack-free case since it reduces the exchanged control messages as shown in Figure 6.5 and provides better channel conditions.

6.4.5.2 Resource requirement statistics

In this section, we analyze the resource requirement statistics of our proposed techniques. The resources we take into account are the power and memory resources which are very crucial for LLNs that consist of resource-constrained devices.

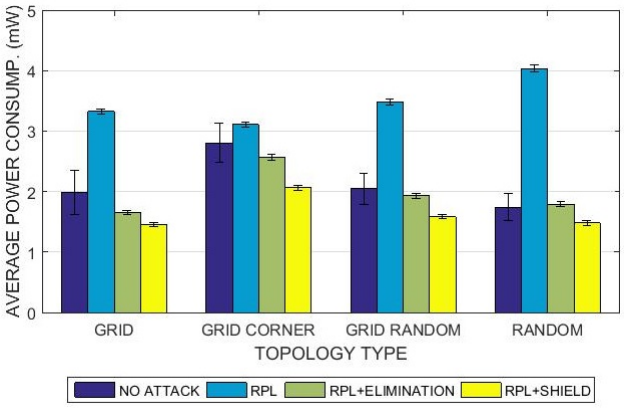

Figure 6.8 : Average power consumption results with respect to topologies.

The average power consumption results with respect to different topologies are shown in Figure 6.8. Among all the topologies we employed, the highest average power consumption results for the attack-free case was observed for Grid Corner topology. This outcome is reasonable since longer paths exist for Grid Corner topology than others. We can clearly see that VNA increases the average power consumption of

the nodes for all topologies except for the Grid Corner topology. For Grid Corner, it looks like there is not much difference between the attack-free case and RPL under attack case. We analyzed this strange outcome and figured out a very interesting reason. The channel conditions with several retransmission attempts in attack-free case and malicious global repairs with resulting routing errors in RPL under attack case somehow balance the power consumption between these two cases. Grid Corner topology lost nearly 20% of the data packets as shown in the Figure 6.7. When we checked the MAC layer, we saw many drops caused from reaching the max. retransmissions due to collisions or without collisions, not being able to allocate space because of either a full packet queue or a full neighbor queue. Most of the packets were dropped at the MAC layer and a small portion of them were dropped at the network layer due to routing errors. This observation explains the high power consumption when there is no attack since nodes waste energy due to several retransmission attempts at the MAC layer. When we checked the case with RPL under attack, we realized that 91% of the data packets are lost. We observed that most of the packets were dropped at the routing layer due to route errors where either the route does not exist or the next hop cannot be found. This shows that when Grid Corner is under attack, since global repair operation aims to create the DODAG from scratch, VNA causes route errors and the ordinary nodes drop majority of their packets at the routing layer, which does not consume as much energy as several MAC layer retransmissions in the attack-free case. When we check the efficiency of our novel mitigation techniques, we can see that, it is possible to reduce the power consumption overhead of the VNA up to 63%. For almost all scenarios, the proposed techniques provide even lower power consumption than attack-free RPL which is similar to the results for network performance metrics due to aforementioned reasons.

Table 6.3 : RAM and ROM statistics.

Image	Total RAM (Byte)	Total ROM (Byte)
Contiki RPL UDP	7530	47363
Contiki RPL UDP + Elimination	7530	47375 (↑ 12)
Contiki RPL UDP + Shield	7962 (↑ 432)	48109 (↑ 746)

The nodes in our simulations use the emulated Tmote-Sky motes [64] which has an MSP430 microcontroller with 10KB of RAM and 48KB of flash. Each node runs

Contiki RPL UDP application where a temperature reading is sent to the DODAG root. The RAM and ROM requirement statistics of the proposed mitigation techniques and the ordinary RPL are shown in the Table 6.3. Since the Elimination technique is very simple to implement, all it requires is 12 bytes of extra ROM usage. However, the Shield technique requires additional code blocks to maintain the ShieldList table and also to process the flow chart shown in the Figure 6.3. Therefore an additional 746 bytes of ROM space is required. The RAM overhead of the Shield technique is 432 bytes. Just to recall that, the ShieldList table stores the IPv6 address and Version Number fields for each neighboring node with better ranks. We set the table size to 20 considering the Contiki RPL max. number of neighbors settings. Each entry in the table occupies 20 bytes where 16 bytes of it is used for the IPv6 address. Thus the 400 bytes of the RAM overhead is used by the ShieldList table. Note that reducing the overhead of the table is possible by storing the IDs of the nodes instead of IP addresses.

6.4.5.3 Accuracy results

In this section, we investigate how accurately legitimate VN updates are discriminated from the malicious ones with respect to varying topology characteristics for our mitigation techniques. Although it was skipped by the previous VN detection studies, we believe that this is a very crucial point to consider since it can directly affect the performance of proposed mechanisms against VNA. Before analyzing the results, we would like to notice the readers that the DODAG root initiates VN updates with random intervals with respect to Negative Exponential Distribution in our scenarios. We did not consider employing periodic VN updates in order to evaluate the accuracy of the proposed techniques comprehensively.

Table 6.4 shows the accuracy results of the proposed techniques. We can see that the mitigation techniques have high True Negatives (more than 98%) which shows that they are very effective at filtering the malicious VN updates for all scenarios. True Negative results clearly show that our novel mechanisms can successfully block the malicious global repair attempts originated from the attacker.

In terms of allowing the legitimate VN updates initiated by the DODAG root (i.e., True Positive results), we can see that the proposed mitigation techniques allow the legitimate updates with varying rates.

Table 6.4 : Accuracy results.

Topology	Accuracy	RPL (%)	RPL+Elimination (%)	RPL+Shield (%)
Grid	TP	100	49,6	9,38
	FN	0	50,4	90,62
	TN	0	99,38	99,41
	FP	100	0,62	0,59
Grid Corner	TP	100	29,97	4,97
	FN	0	70,03	95,03
	TN	0	98,6	99,32
	FP	100	1,4	0,68
Grid Random	TP	100	40,32	5,63
	FN	0	59,68	94,37
	TN	0	99,2	99,49
	FP	100	0,8	0,51
Random	TP	100	46,44	4,93
	FN	0	53,56	95,07
	TN	0	99,5	99,64
	FP	100	0,5	0,36

Better results for True Positives were obtained thanks to Elimination technique since only the direction of the DIO messages with greater VN are taken into account by a node whether to change the VN or not. We would like to remind the readers that only a single DIO message possessing the mentioned property is enough for a node to update its VN for the Elimination technique. Since legitimate update messages come from the direction from root towards leaf nodes, the Elimination technique allows RPL to continue its ordinary operation. As we explained in Section 6.4.5, the nodes in our topologies employ ContikiMAC radio duty cycling protocol where nodes sleep and wake up periodically (i.e., 8 wake-ups per second). Each node sleeps and wakes up independent from the rest of the nodes, which is the default setting of ContikiMAC. Therefore, nodes which have the same rank values in the DODAG (i.e., which are at the same level in the DODAG) may not hear the DIO messages coming from their parents at the same time. Hence, the dissemination of the global repair operation may show different patterns on different sub-DODAGs of the RPL, a sub-DODAG may obtain the newer VN earlier than another sub-DODAG. A node may receive a DIO with an updated VN from a child node with this reason. Because the VN update information has reached the child node earlier than its parent thanks to its neighbors on another

sub-DODAG. This is the actual reason of the filtered out DIO messages carrying the legitimate VN update information for the Elimination technique.

In the Shield technique a node waits until majority of the neighboring nodes with better rank values change their VN before updating its VN. Therefore the legitimate updates are filtered by a node until majority of its neighbors in its ShieldList table claim the same VN update, which explains more legitimate updates were filtered in comparison with the Elimination technique.

As shown in the Table 6.4, although our techniques seem to have high False Negatives, True Positives are never zero for both of the mitigation techniques, which guarantees that the ordinary operation of RPL is not disrupted since all of the nodes in the DODAG can still update their VNs. High False Negative results experienced by employing the proposed techniques are acceptable because the proposed techniques significantly shorten the delay caused by the attacker, reduce the average power, lower the control message overhead and increase the data packets delivery ratio as well as let nodes to update their VNs.

6.5 Discussion and additional analysis

The simulation results with respect to various topologies show that it is possible to mitigate the detrimental effects of the RPL VNA by employing the proposed mitigation mechanisms. The Elimination technique with its simplicity provides a mitigation performance to a certain extent and requires only a tiny amount of ROM space. On the other hand, the Shield mechanism provides the best mitigation performance with more resource requirements than the former technique. Thus a trade-off exists where we pay more attention to the resource constraints but get lower mitigation performance or we accept to assign more resources for the sake of the best mitigation performance.

In this paper, the proposed mitigation techniques are based on the idea that the legitimate VN updates can not come from the nodes which are in the direction of the leaf nodes, in other words which have worse ranks than ours. Because, the rank information represents the relative distance of a node from the root and the nodes closer to the root have better rank than the ones farther from the root. In our simulations, rank of a node is calculated based on the parent's rank and the expected transmission

count. However, it is possible that some nodes in the network may fail and routing holes can occur. If a routing hole appears, then the legitimate VN updates may have to follow different paths which may look like breaking our basis idea on the direction of VN update dissemination. In order to evaluate such a situation, we ran a separate simulation on the Random topology, where such a routing hole can be created by removing the node at the position 7 (i.e., Node 7). The left hand side of the Figure 6.9 shows the sub-DODAG and the corresponding parent-child relationships before the routing hole. Our mitigation schemes expect the legitimate updates to be disseminated as follows: firstly, Node 7, then the nodes 9, 29 and 34 and then the rest of the nodes in the sub-DODAG. We removed Node 7 and created a routing hole just before a legitimate VN update initiated by the DODAG root. We observed that this segment of the network starts to rebuild the sub-DODAG and create new parent-child relationships as shown in the right hand side of the figure. Until the sub-DODAG is rebuilt, nodes do not change their VN. But when the sub-DODAG is rebuilt, the nodes change their VN one by one starting from new parents to new children according to our mitigation schemes.

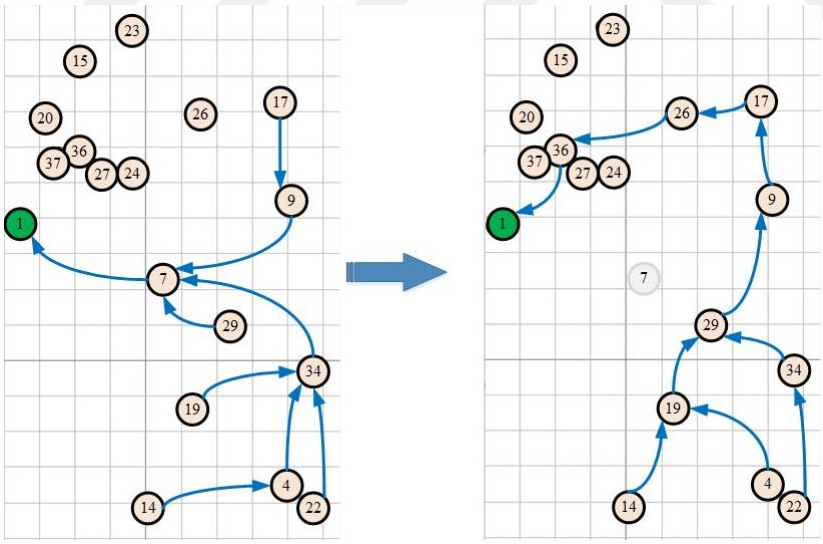

Figure 6.9 : A segment of the Random topology before the routing hole and after the repair.

6.6 Conclusion of the section

In this paper, we propose two mitigation techniques for the VNA with different performance outcomes and resource requirements. The first mitigation technique eliminates the VN updates coming from the direction of the leaf nodes. By this way, it makes sure that the strongest positions for the VNA are mitigated. However, for the rest of the attacking positions, it cannot mitigate the effects of the attack. Therefore, we propose a second mitigation technique in order to mitigate the effect of the attack regardless of the attacking positions. The second technique allows a node to change its VN only when majority of its neighbors with better ranks claim a VN update. The mitigation effectiveness of the proposed schemes is evaluated on four different topologies. The results show that, it is possible to reduce the detrimental effects of the attack by employing the proposed techniques while allowing the legitimate VN updates.

Moving from this study, many things can be done as a future work. One of them is the analysis of the multiple VN attackers situation which has not been considered in the literature yet. Another issue would be to analyze the performance of a hybrid mitigation scenario. If our network has a heterogeneous structure with varying node resources, then we may want to run the Elimination technique in the nodes with the lowest resources and for the rest of the nodes, we can consider the Shield technique. Or, based on the remaining battery level, a node may decide to move from Shield to Elimination. Another issue to consider can be the incorporation of the mobility which can change the rank of the nodes and the topology dynamically and may affect the performance of the attack mitigation.

7. RPL VERSION NUMBER ATTACK WITH MULTIPLE ATTACKERS

In this chapter, we present our analyses on multiple VN attackers. In order to analyze and understand the effects of multiple attackers, we firstly created an initial topology and performed measurements. Following this, we conducted our analyses with a single attacker, two attackers and three attackers w.r.t. various positioning combinations on a grid topology.

7.1 Initial Analyses

In order to analyze and understand the effects of multiple VN attackers in RPL networks, we ran a set of simulations. We wanted to gain insights on how multiple independent attackers affect RPL operation as well as how it shows itself in network performance.

7.1.1 Initial topology and the attacker model

We set up an initial topology as shown in Figure 7.1. The initial topology consists of randomly positioned 13 nodes, amongst 3 of them (i.e., nodes with IDs 7, 12 and 13) are the attackers. The DODAG root (i.e., the node with ID 1) initiates global repair operation in the RPL network w.r.t. Negative Exponential Distribution as in Chapter 6. We would like to remind the readers that, we had modified the default RPL implementation in order to simulate events. Our idea was that events happen and DODAG root updates the VN as a result of such events. In the default RPL implementation (i.e., in Contiki 3.0) as we explained previously, there is no defined way for a DODAG root to update the VN and DODAG root updates the VN whenever it receives a DIO message with a greater VN.

The attacker nodes in the topology are exactly the same as the other nodes except for sending DIO messages with an incremented VN. There are three attacker nodes and each attacker maliciously updates the VN without any cooperation with other attackers.

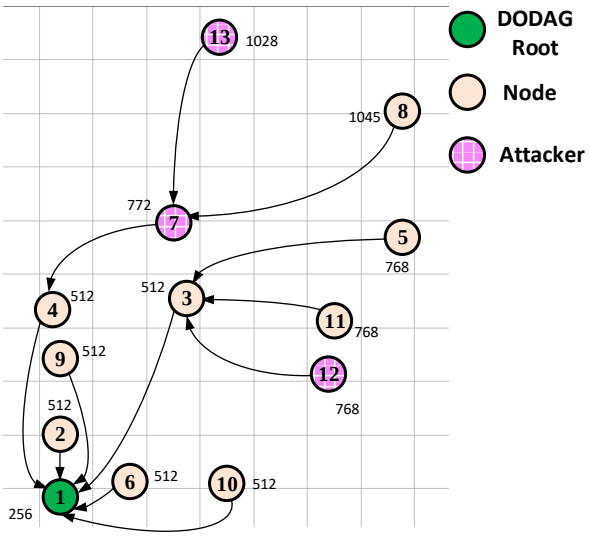

Figure 7.1 : Initial topology with node IDs, ranks and parent-child relationships.

Figure 7.1 shows the form of the DODAG with parent-child relationships and rank values before the attack starts and when the network is stable.

7.1.2 Simulation environment and parameters

In our initial simulations, we again used the Cooja simulator [8] of the Contiki OS version 3.0 [9]. In Cooja, we used emulated Tmote Sky motes [64]. As in our previous simulations, the nodes run RPL UDP application where the DODAG root runs the UDP server and rest of the nodes - including the attackers - run UDP client application. UDP clients send a temperature reading every minute to the UDP server. We record the traffic and other logs during the simulation time. The simulation environment has distance-based loss model with no external interference.

7.1.3 Performance metrics

In our previous studies, we had *packet delivery ratio*, *control message overhead*, *average network delay* and *average power consumption* metrics. In this study, we considered analyzing further parameters. Hence, in addition to the previous metrics, we monitored *routing loops*, *absence of a route*, *packet drops due to errors signaled in RPL header options of IPv6 packets*, *MAC layer retransmissions* and *change in rank*.

7.1.4 Initial simulation results

7.1.4.1 Results

We performed simulations on the initial topology. During the analysis of the simulation logs we realized an interesting behavior when the RPL network is under the attack of multiple VN attackers. We found out that, rank values of all of the nodes start increasing and after a while they reach the infinite rank value. Figure 7.2 shows the changes in the topology. Before getting to the details, we would like to firstly explain some useful information which may help readers to understand the figure and the sub-figures.

Figure 7.2 shows the DODAG root, ordinary nodes and the attacker nodes. Ranks of the nodes are shown beside the nodes in the figure. In the RPL networks, typically the lowest rank value is 256 and it is owned only by the DODAG root. Rest of the rank values usually change between 512 and the infinite rank value. In Contiki 2.7 and 3.0, the infinite rank value is 65535. If a node has an infinite rank in RPL, it means that, it does not have a preferred parent and it can not be a preferred parent of another node. It is simply disconnected from the RPL network. The figure also shows the parent-child relationships. Parent-child relationship is indicated by an arrow starting from the child node and ending at the preferred parent node. The dashes signify that a child node has sent a No-path DAO message to its old preferred parent. Just to remind the readers that, a node sends a No-path DAO to its old preferred parent when it chooses a new preferred parent.

The figure shows the topological changes starting from the initial form of the DODAG till the stage where all of the non-root nodes have infinite rank values. Here, we provided only five snapshots of the RPL network showing that change. However, such a change does not happen only in five steps and there could be more distinct snapshots which we could not show due to space limitations. When rank values of every non-root node reaches infinity, they stay infinite until the root applies the VN update. When root updates the VN (w.r.t. Negative Exponential Distribution), then starting from root till leaf nodes, nodes start to obtain their *usual* or *ordinary* rank values again. This

observation was quite interesting for us. We had not observed such a behavior in our earlier studies, since our simulation log analyses had coarser granularity.

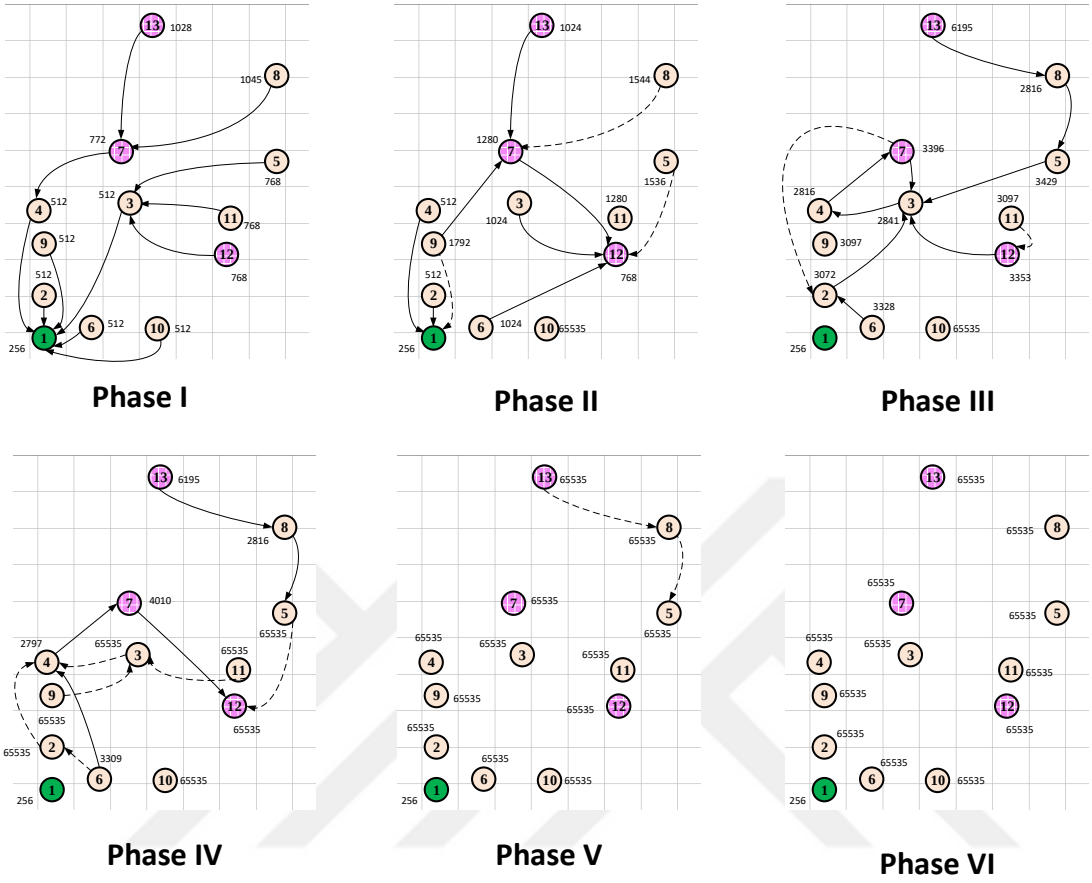

Figure 7.2 : Change in the ranks and parent-child relationships on the initial topology.

7.1.4.2 Underlying reason

We focused on this issue for a while and analyzed the simulation logs, ran more simulations and found out the underlying reason. In Contiki's RPL implementation, when a node receives a DIO with a newer VN (which happens in both legitimate and malicious VN updates), it firstly removes all the nodes in its parents table. Then it adds the DIO sender to its table and tries to select it as the preferred parent. Regardless of the rank of the DIO sender, the receiver does this operation. Therefore, if DIO sender had a worse rank value before the reception of the DIO (i.e., DIO with newer VN), then the receiver node selects it as the preferred parent and calculates its own rank based on its new preferred parent's rank. Which means after the VN update, it gets a worse rank value before the update. This explains the increase in rank values in time when VN attack takes place.

However, this is not the complete answer. Because when we were checking the rank values for a node throughout the simulation time, we were not seeing a linear increase from 512 to 65535 in time. A node can select the sender of the DIO with a newer VN as a parent if it has an acceptable rank. If sender's rank is unacceptable (i.e., infinite rank) or selecting it as a parent will cause too much increase in rank, then the receiver node does not select it as a preferred parent. In such a case, it sets its rank as infinite.

Moving from this on, we realized that, when a node changes its preferred parent, it sends a no-path DAO message to its old parent soon. However, it waits for four seconds to send a DAO message to its new parent. This is how Contiki RPL works. We believe, Contiki developers might have thought that a node may receive many DIO messages in a short while and it is wise to consider all of them and send only one DAO message to the DIO sender with the best rank among all senders. Anyway, during the attack, nodes start sending many control messages and every minute they also try to send a UDP packet. Which means, channel conditions get busier and collisions take place. Nodes have chance to successfully transmit their messages if they do not drop the packets in their queues. During this time, packets (both data packets and control packets) wait in the queues and they can not reach their destinations on time. These channel condition related issues and nodes delaying sending DAO messages for four seconds cause loops in the DODAG. When a node detects a loop, it sets the rank of the other node who caused the loop as infinite. If it tries to select preferred parent and realizes that, no node in its list has an acceptable rank, which means no suitable parent is found, it starts local repair mechanism, where it sets its own rank to infinite. In this way, one by one, nodes start to obtain infinite rank values and after a while, no parent-child relationship exists within the DODAG. This continues until the DODAG root sends a DIO message with an updated VN. When the root does this, the nodes close to the root start to obtain 'normal' rank values and create parent-child relationships again.

The outcomes of our initial analysis are as follows:

- VN attack is an interesting attack in a way that only a single DIO with a malicious VN update can easily cause all of the nodes to be part of the attack by disseminating the malicious update through the network,

- Not every attack affects the network efficiently. Sometimes neighboring nodes can not hear the malicious updates due to various reasons (i.e., sleeping while duty cycling, receiver buffer being full at the MAC layer, collisions, etc.),
- Since VN attack affects the channel conditions, it causes delays for both data and control traffic, which makes the system more chaotic,
- It is important to observe the behavior of rank change and loops when considering the VN attacks.

7.2 RPL Version Number Attack with Multiple Attackers

Up until this point in our thesis, our studies were based on a single VN attacker in RPL networks. After our initial analysis in the previous section, we wanted to investigate multiple attackers. The motivation of this study was to reply the following questions:

- Is the number of attackers directly proportional to the effect of the attack?
- What is the best positioning for multiple attackers?
  - Attackers close to each other or far from each other,
  - Position combinations (e.g., all at the center of the DODAG, at the edges of the DODAG or combinations),
- Which performance metrics are affected by multiple attackers?
- What is the best setting from the attacker's points of view?

7.2.1 Topology and the attacker model

We created a grid topology with 16 nodes for our multiple VN attackers analysis. Figure 7.3 shows the topology for the analyses of multiple attackers and the form of the DODAG when network gets stable. All of the nodes in the topology are static nodes.

In terms of the attacker model, the first issue to decide was the number of attackers. For this aim, we considered to limit the the number of attackers to 3, which is nearly 20% of the nodes (i.e., in an RPL DODAG with 16 nodes) that can be malicious. The next thing

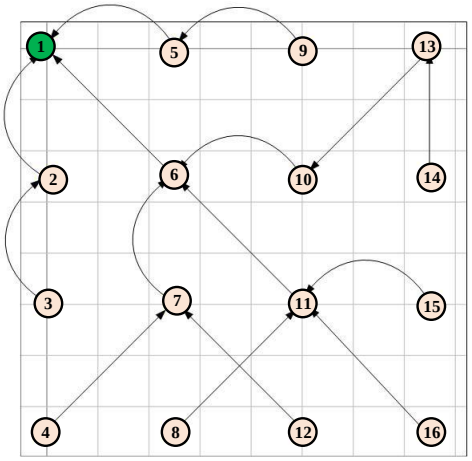

Figure 7.3 : The grid topology and the form of the stable DODAG.

to consider was the positioning of the attackers. Our earlier studies in this thesis had revealed that, the location of the attacker is important for the VN attack. Therefore we had to consider the attacking positions for multiple attackers. However where to place the attackers was our concern now. The first option can be placing a single attacker, 2 attackers and 3 attackers to every possible position combinations within the DODAG. That means, 15 different positions for single attacker, 105 (i.e., $C(15,2)$) different position combinations for 2 attackers and 455 (i.e., $C(15,3)$) position combinations for 3 attackers for our current topology. We assume that every non-root node can be an attacker in these calculations. Since we run at least 10 simulations for each attack case, it makes thousands of simulations in total. So as to reduce the number of simulations, we came up with a leveling approach, which divides the DODAG into breadth-first search like levels. If we redraw the DODAG based on the levels of nodes as in trees, we can get four levels based on the parent-child relationships as shown in Figure 7.4.

Instead of running simulations for every possible position combination for multiple attackers, we decided to form position combinations based on the levels in Figure 7.4. This approach significantly reduced the number of simulations. For 2 attackers, we selected 48 position combinations and for 3 attackers, we selected 54 position combinations considering the location of the attackers in the network (e.g., center, edge), distance between nodes (e.g., close, far) on the topology and combinations of the levels. The attacker combinations are outlined in Table 7.1, Table 7.2 and Table 7.3 for single, two and three attackers respectively.

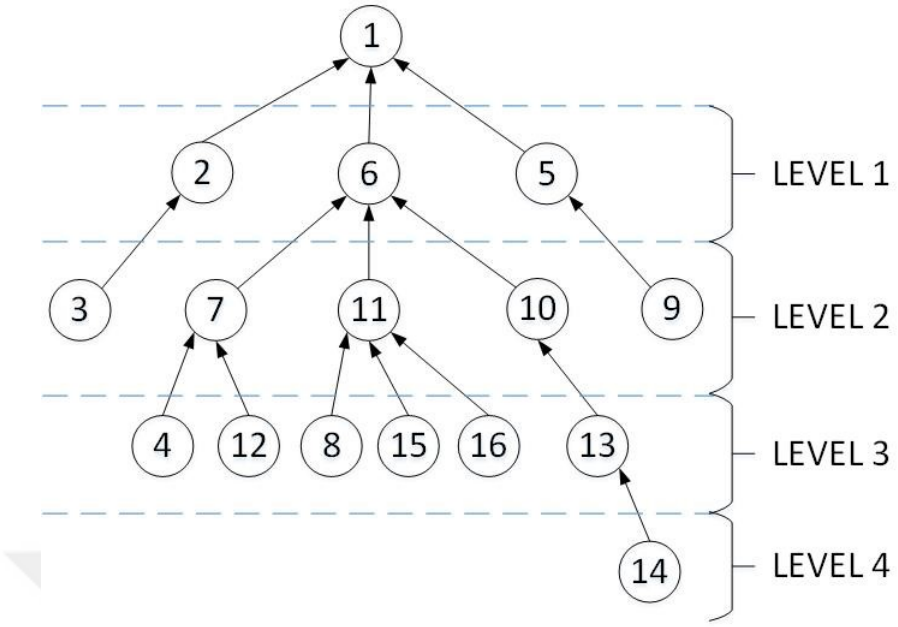

Figure 7.4 : Layered DODAG of the topology.

Table 7.1 : Single attacker positions.

Level	Attacker Position
Level-1	2, 5, 6
Level-2	3, 7, 9, 10, 11
Level-3	4, 8, 12, 13, 15, 16
Level-4	14

Table 7.2 : Positions of two attackers.

Level	Attackers' Positions
Level-1	2-5, 2-6
Level-2	3-7, 3-9, 3-10, 3-11, 7-10
Level-3	4-8, 4-12, 4-13, 4-15, 4-16, 12-16, 13-16
Levels-(1-2)	2-3, 2-7, 2-10, 2-11, 3-6
Levels-(1-3)	2-12, 2-13, 2-15, 2-16, 6-12, 6-13, 6-16
Levels-(1-4)	2-14, 5-14, 6-14
Levels-(2-3)	3-4, 3-8, 3-12, 3-13, 3-15, 3-16, 10-16, 11-16
Levels-(2-4)	3-14, 7-14, 9-14, 10-14, 11-14
Levels-(3-4)	4-14, 8-14, 12-14, 13-14, 15-14, 16-14

Table 7.3 : Positions of three attackers.

Level	Attackers' Positions
Level–1	2-5-6
Level–2	3-7-11, 3-10-11, 7-10-11
Level–3	4-8-12, 4-12-13, 4-13-15, 4-13-16, 8-12-15, 8-12-16, 12-13-15, 12-15-16
Levels–(1–2)	2-3-5, 2-3-6, 2-7-10, 2-10-11, 3-5-11, 6-7-11
Levels–(1–3)	2-4-12, 2-5-12, 4-6-13, 6-8-15, 6-12-15
Levels–(1–4)	2-5-14, 2-6-14
Levels–(2–3)	3-4-7, 3-7-16, 3-12-13, 3-12-15, 7-10-16, 7-11-12, 7-11-15, 7-11-16
Levels–(2–4)	3-7-14, 7-10-14, 7-11-14
Levels–(3–4)	4-13-14, 8-12-14, 12-14-15
Levels–(1–2–3)	2-8-9, 2-9-15, 5-9-13, 6-11-12
Levels–(1–2–4)	6-10-14, 6-11-14
Levels–(1–3–4)	2-3-14, 5-13-14, 5-14-15, 6-14-15
Levels–(2–3–4)	3-12-14, 3-14-16, 11-14-15

7.2.2 Simulation environment and parameters

In our simulations, we used the Cooja simulator of the Contiki OS version 3.0. In Cooja, we used emulated Tmote Sky motes. As in our previous simulations, the nodes run RPL UDP application where the DODAG root runs the UDP server and rest of the nodes - including the attackers - run UDP client application. UDP clients send a temperature reading every minute to the UDP server. We record the traffic and other logs during the simulation time. The simulation environment has distance-based loss model with no external interference.

In terms of the RPL global repair mechanism, we kept the default RPL implementation in which the DODAG root only the updates the VN whenever it receives a DIO with a greater VN.

7.2.3 Performance metrics

In our initial anaylsis, we had several performance metrics which we could analyze. These were *packet delivery ratio*, *control message overhead*, *average network delay*, *average power consumption*, *routing loops*, *absence of a route*, *packet drops due to errors signaled in RPL header options of IPv6 packets*, *MAC layer retransmissions* and *change in rank*. Instead of using all of these metrics, we wanted to consider the

attackers' point of view in order to evaluate the performance. Hence, an attacker may target three goals:

- Causing more data packets to be lost,
- Causing longer delays,
- Causing the waste of more energy and thus depleting the batteries.

Therefore, we focused on the packet delivery ratio, average network delay and power consumption results in terms of number of attackers and attacking position combinations. We also measured rest of the metrics and considered them as supplementary performance metrics in order to understand the results regarding with these three performance metrics.

7.2.4 Simulation results

We ran simulations for attack-free case, for single attacker, for two attackers and for three attackers cases. We ran 10 simulations for each case, which makes in total $(1 + 15 + 48 + 54) \times 10 = 1180$ simulations, where 1×10 simulations are for attack-free case, 15×10 simulations are for single attacker case, 48×10 for two attackers case and 54×10 simulations for three attackers cases. We obtained the results with 95% confidence interval.

In terms of the simulation results, we considered to provide results from two perspectives:

- Comparison of number of attackers with respect to each performance metric,
- Comparison of position of the attackers with respect to each performance metric and different number of attackers.

In the first class of comparisons, we obtained the results for all single attacking positions, then for all two attackers positions and three attackers positions. In Section 7.2.4.1, we show how each performance metric changes when we increase the number of attackers.

Figure 7.5 : Coloring scheme for single attacker case.

Figure 7.6 : Coloring scheme for two attackers case.

Figure 7.7 : Coloring scheme for three attackers case.

In the second class of comparisons, we thought of using a coloring scheme where colors starting from red to light green depict the closure of attacker(s) to the DODAG root. The red color signifies the attacker(s) is/are close to the DODAG root and the light green signifies that the attacker(s) is/are far from the DODAG root. We provided the coloring schemes for single, two and three attackers in Figure 7.5, Figure 7.6 and Figure 7.7 respectively. As we have used the leveling approach (see Figure 7.4) and combinations of levels (see Table 7.2 and Table 7.3), we assigned the colors with respect to the sum of the levels of each attacker in multi-attacker cases. For example, for single attacker case, if the attacker is in Level 1, then it is assigned to red and if it is in Level 4, it is assigned to light green. For two attackers cases, if level sum of the attackers is 2, then this combination is assigned to red and if level sum of the attackers is 7, then that combination is assigned to light green. For three attackers cases, colors are assigned with respect to level sums that can be between 3 and 10. In all simulation results, the blue color signifies the case where no attacker exists (i.e., shown as *N/A* in the figures).

7.2.4.1 Comparison of number of attackers

We compared the three performance metrics with respect to number of attackers. The simulation results are shown in Figure 7.8 for packet delivery ratio, Figure 7.9 for average network delay and Figure 7.10 for average power consumption. We would like to inform the readers that coloring scheme is not used in these figures. It is used in the following subsection.

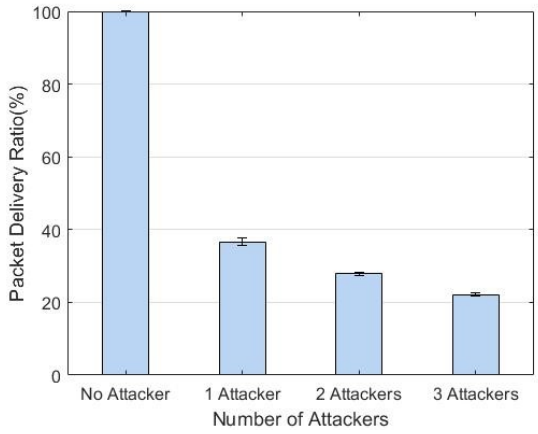

Figure 7.8 : Comparison of number of attackers in terms of packet delivery ratio.

Packet delivery ratio results show that higher the number of attackers, lower the delivery ratio. When there is no attacker, we do not see any packet loss since the topology is simple and there is no external interference. However, as the number of attackers increases, packet delivery ratio drops down to 36% for single attacker, 27% for two attackers and finally 22% for three attackers. In terms of supplementary performance metrics for dropped packets, we measure packet drops due to route errors (i.e., state when no route exists), drops due to IPv6 packets' header processing errors and drops at the MAC layer (i.e., reaching the max. retransmission count, buffer allocation errors, full neighbor queues). When we have a look at the supplementary performance metrics results, we see that majority of packet drop messages are route error messages. It is followed by the drops at the MAC layer. Packet drops due to header processing errors are less than both of the other metrics.

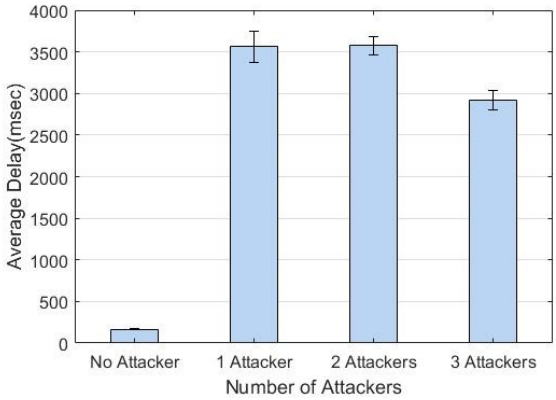

Figure 7.9 : Comparison of number of attackers in terms of average network delay.

Average network delay results show that single attacker and two attackers cause longer delays than three attackers. Results show that VN attack causes between sixteenfold and twentyfold increase in delays comparison to the no attack case. We can not explain the reason of three attackers case having shorter average network delay results than single attacker and two attackers. In the calculation of average network delay results, we consider only the packets which successfully arrive to the DODAG root. Considering the packet delivery ratios in Figure 7.8, we see that number of successfully delivered packets w.r.t. number of attackers changes like $single\,attacker > two\,attackers > three\,attackers$. However, we can not see a similar or opposite relation between the number of attackers for average network delay.

Average power consumption results show that effect of single, two and three attackers are quite close to each other. We can not really claim that, increasing the number of attackers seems to increase the average power consumption. When we think of power consumption of a node, we can relate it with parameters that reside at different layers of the protocol stack. At the application layer, it is related to the number of data packets generated, which is in this case same for all attacking cases. At the transport layer, we can relate it with the transport protocol, which is UDP for all attacking cases. At the network layer, we can relate power consumption with number of generated RPL control messages, number of route errors which cause a packet to be dropped without going down to MAC layer and header processing errors of IPv6 packets which again cause a packet to be dropped. When we check the average number of RPL control messages generated and header processing error counts (i.e., we measure these parameters as supplementary performance metrics), we see that average of all of the results for each attacker cardinalities are close to each other. In terms of route errors, with slight differences, results have $single\,attacker < two\,attackers < three\,attackers$ relationship. Hence, network layer results do not imply a significant power consumption difference between different number of attackers when we consider the average of all attacking positions for each cardinality of attackers. At the MAC layer, power consumption is related to transmission attempts of IEEE 802.15.4 frames. In this layer, we do not have the average or total number of transmission attempts. But, we have MAC layer error counts. Number of fragments dropped at MAC layer for different number of attackers

has $single attacker > two attackers > three attackers$ relation. The results are again not too distant from each other. So, all of the supplementary metrics we measure imply average power consumption results to have a behavior as shown in the Figure 7.10.

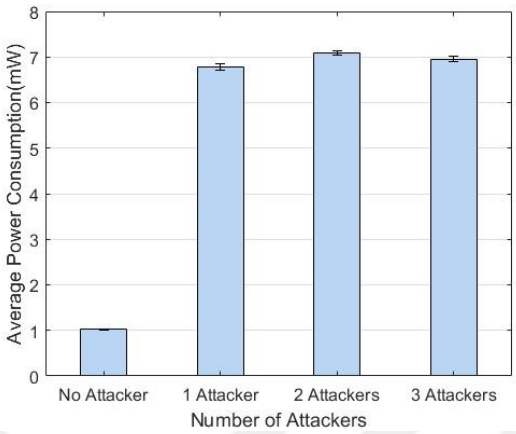

Figure 7.10 : Comparison of number of attackers in terms of average power consumption.

In this subsection, we provided the simulation results for three performance metrics with respect to number of attackers. We provided the averages of the results for single attacker, two attackers and three attackers. Although analyzing the averages of the results gives an overall idea, seeing the results for each attacking position may be helpful too. In addition to this, investigating whether position of the attackers can still give valuable information is needed as well. Therefore, on the following subsection, we provide individual simulation results and consider whether the position of the attacker(s) is/are still important for VNA.

7.2.4.2 Comparison of position of the attackers: packet delivery ratio

Packet delivery ratio results for single attacker, two attackers and three attackers are shown in Figure 7.11, Figure 7.12 and Figure 7.13 respectively. Closure of attacker(s) to the DODAG root is signified as the coloring scheme explained in Section 7.2.4 where, bright *red* shows the position being close to the DODAG root and light *green* shows the position being far from the root. No attacker case on the other hand is shown in *blue* with N/A label. In these figures, each attack position combination is shown at the x-axis, whereas y-axis shows the packet delivery ratio results in terms of percentages. For the sake of comparisons, we sorted the results in ascending order.

The simulation results for single attacker cases show that, the lowest delivery ratios were obtained for positions that are in the center of the network (i.e., nodes with IDs 6, 7, 10 and 11). The attacking positions that have the highest packet delivery ratios are positions 4, 13, 2, 5 and 16. Among these, positions of 2 and 5 are the closest positions to the DODAG root and positions 4, 13 and 16 are the positions both far from the root and located at the edges of the topology. For multiple attacker cases, we see a similar pattern where central attacking positions cause the lowest packet delivery ratios. However in our earlier study in Chapter 5, we had found out that VNA causes the lowest packet delivery ratios for the positions that are located far from the root. This was interesting for us and we investigated the underlying reason in Section 7.2.4.5.

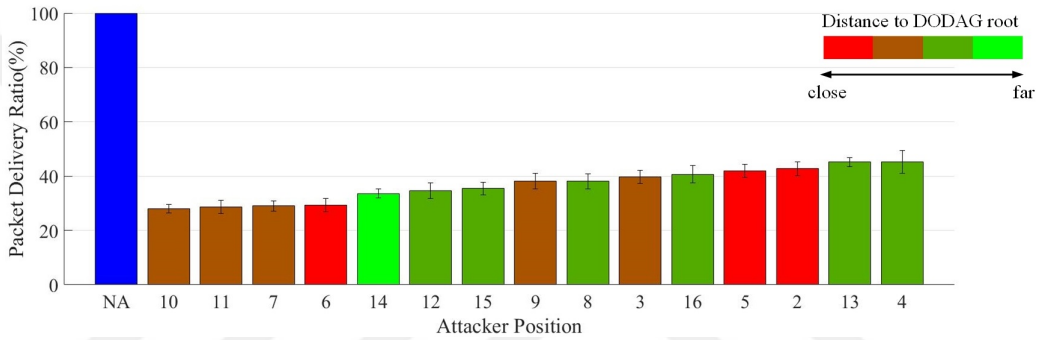

Figure 7.11 : Packet delivery ratio results w.r.t. attacking positions for single attacker case.

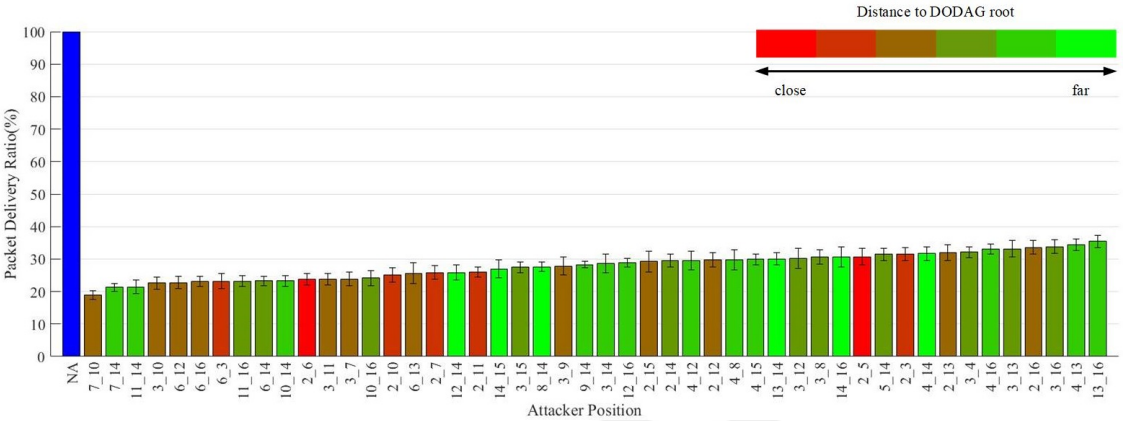

Figure 7.12 : Packet delivery ratio results w.r.t. attacking positions for two attackers case.

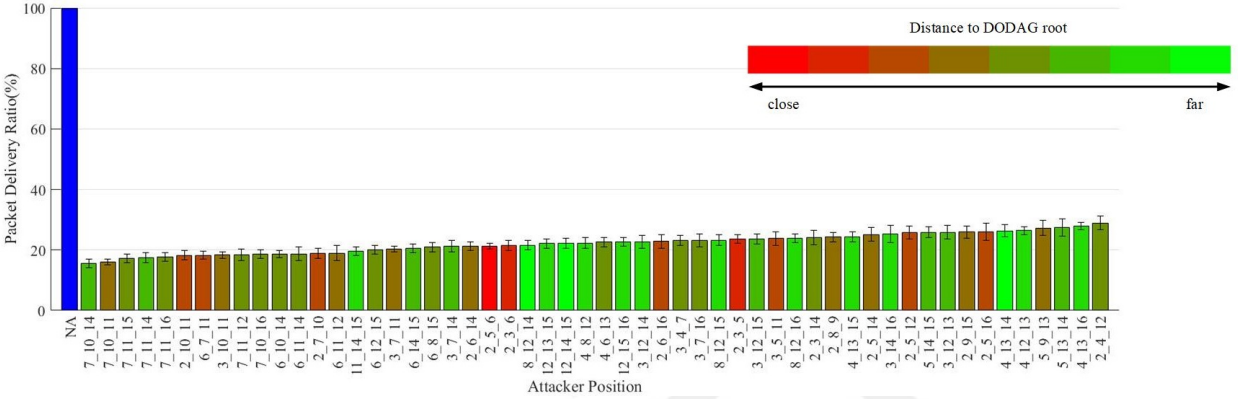

Figure 7.13 : Packet delivery ratio results w.r.t. attacking positions for three attackers case.

7.2.4.3 Comparison of position of the attackers: average network delay

Average network delay results for single attacker, two attackers and three attackers are shown in Figure 7.14, Figure 7.15 and Figure 7.16 respectively. Closure of attacker(s) to the DODAG root is signified as the coloring scheme explained in Section 7.2.4 where, bright *red* shows the position being close to the DODAG root and light *green* shows the position being far from the root. No attacker case on the other hand is shown in *blue* with N/A label. In these figures, each attack position combination is shown at the x-axis, whereas y-axis shows the average network delay results in terms of milliseconds. For the sake of comparisons, we sorted the average network delay results in descending order.

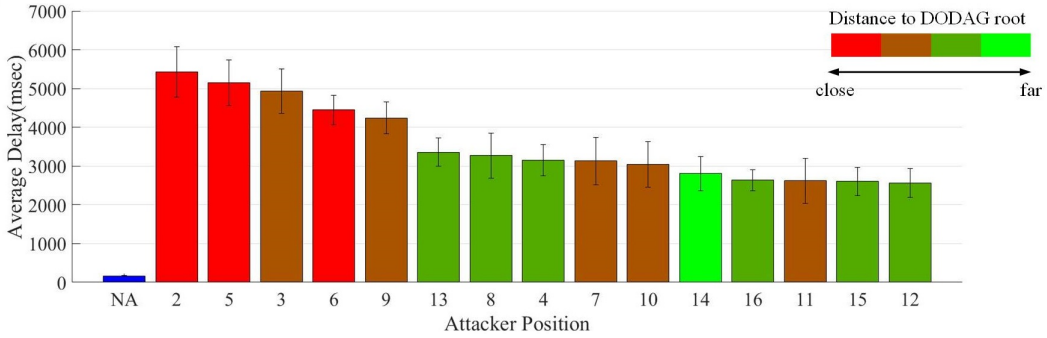

Figure 7.14 : Average network delay results w.r.t. attacking positions for single attacker case.

Simulation results depicted in Figure 7.14, Figure 7.15 and Figure 7.16 show that, effect of the attack gets stronger when attacker(s) get(s) closer to the root. In all of the figures, we see that red-colored bars are the majority of the bars that have the longest average network delay values. On the other hand, green bars are the majority for the lowest delay values. This is an interesting outcome since we had not found such a correlation between the position of the attacker and average network delay in our ealier study (i.e., our study in Chapter 5).

In terms of the distance between multiple attackers and the effect of the attack, we could not find a relation. We can not say whether attackers should be a neighbor of each other or far from each other in order to cause the longer delays.

In terms of attackers' placement, edge or center, we can say that attackers in the center tend to cause shorter delays than attackers at the edges.

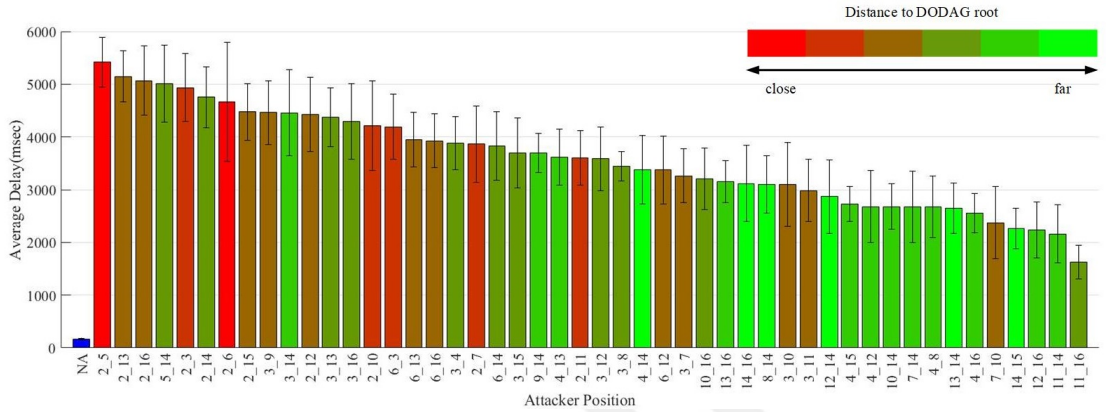

Figure 7.15 : Average network delay results w.r.t. attacking positions for two attackers case.

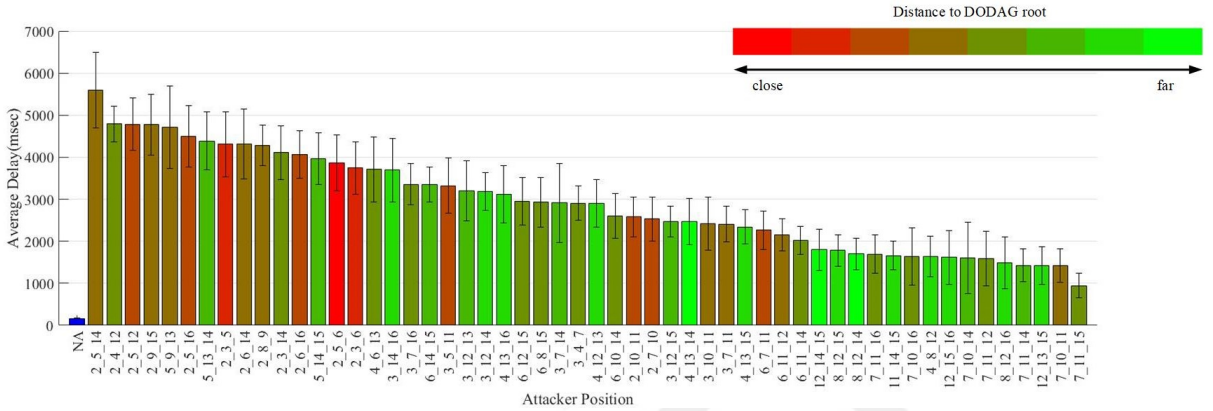

Figure 7.16 : Average network delay results w.r.t. attacking positions for three attackers case.

7.2.4.4 Comparison of position of the attackers: power consumption

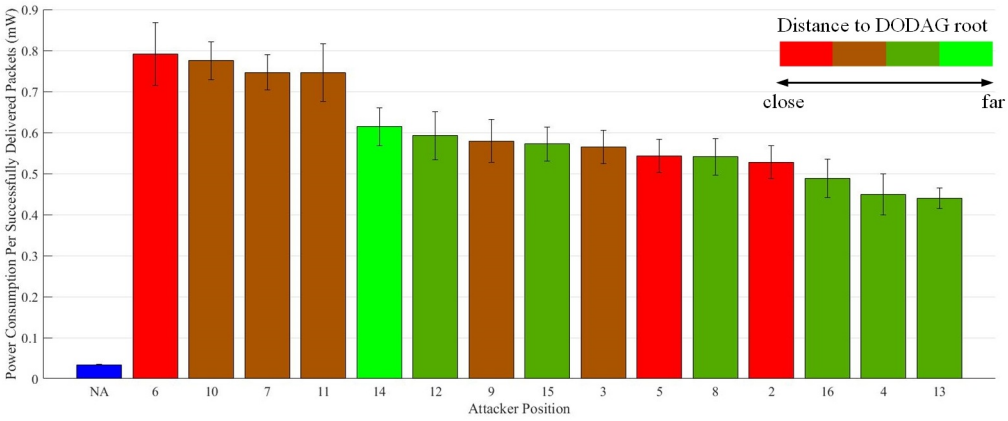

Figure 7.17 : Power consumption per successfully delivered packets w.r.t. attacking positions for single attacker case.

Power consumption results per successfully delivered packets for single attacker, two attackers and three attackers are shown in Figure 7.17, Figure 7.18 and Figure 7.19 respectively.

For single attacker cases, highest power consumption per successfully delivered packets were obtained for the positions that are at the center of the topology, in other words for positions 6, 10, 7, 11. The lowest power consumption per successfully delivered packet values were obtained for the positions that are at the edges of the topology (i.e., 16, 13, 4). In terms of distance to the DODAG root, although results for the attacking positions 2 and 5 are lower than several positions, closer positions seem to cause more power consumption per successfully delivered packets values than the farther positions.

The simulation results for multiple attacker scenarios show a similar behavior to single attacker cases.This result again was interesting for us since we had not found such a correlation between the position of the attacker and average power consumption in our earlier study.

For power consumption per successfully delivered packets results, we cannot say that the distance between attackers has correlation with the effects of the attack.

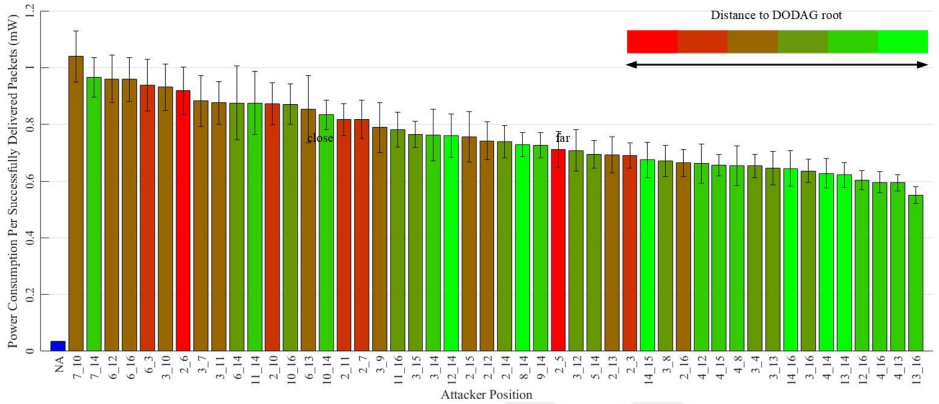

Figure 7.18 : Power consumption per successfully delivered packets w.r.t. attacking positions for two attackers case.

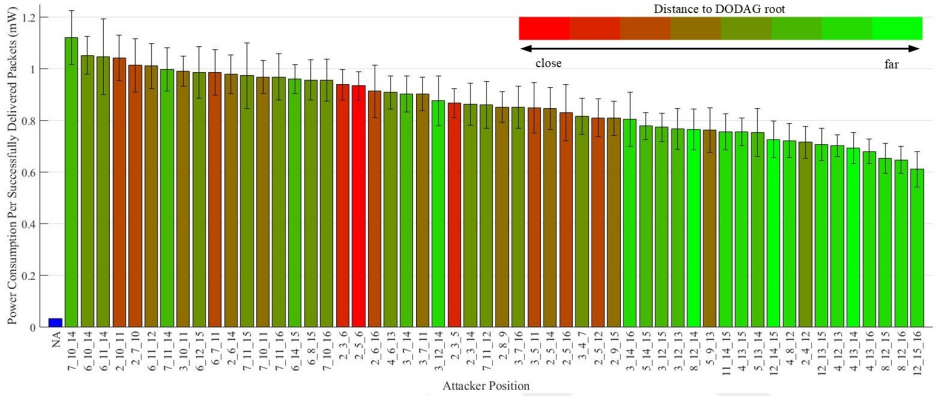

Figure 7.19 : Power consumption per successfully delivered packets w.r.t. attacking positions for three attackers case.

7.2.4.5 Investigation of obtaining different results

Since the obtained results of this study are different than our earlier study, we focused on the investigation of the underlying reasons. There exist topological differences between the studies. As a start, we suspected that the differences in simulation topologies could be the underlying reason. The earlier study was considering a single attacker in a factory environment. There were 44 nodes in the topology where, four of them were mobile nodes and the rest were static nodes. Among the static nodes, 26 of them were positioned on a grid and 14 nodes were positioned randomly. All of the nodes were emulated Tmote Sky motes. Nodes were running the RPL UDP application with Contiki 2.7 and were sending a temperature measurement to the DODAG root every minute. On the other hand, our current study has 16 nodes. All the nodes are static and they are placed on a grid topology. Nodes again run the RPL UDP application with the same data packet generation rate. However they accommodate the updated version of Contiki, namely Contiki 3.0.

In order to make sure that the main reason for different performance results stems from the topological differences, we decided to run two sets of simulations with the same topology but with different operating system versions. Hence, we constructed exactly the same simulation topology of our earlier study which had Contiki 2.7 with Contiki 3.0 in this case. We made sure that both simulations start with the same random seed, mobile nodes in both simulations have exactly the same mobility pattern and both simulations run for the same durations. For the basic comparison, we decided to compare the packet delivery ratio results. For this aim, we selected four positions in the topology for a single attacker where, two positions (14 and 40) are the closest positions to the DODAG root and the other two positions (19 and 25) are the farthest positions to the root. When we run the simulations for 10 times, we obtained the following results:

Table 7.4 : Comparison of Packet Delivery Ratio (%) results for Contiki 3.0 and Contiki 2.7.

Attacker Position	14	40	19	25
Packet Delivery Ratio for Contiki 3.0	33.29	29.57	32.97	39.16
Packet Delivery Ratio for Contiki 2.7	78.77	74.69	48.55	49.69

The obtained results show that the underlying reason for different results is not related to the topology. Because we did not obtain similar results for both of the simulations under the same topology, same random seeds, same mobility patterns and same durations.

After inspecting the topology, we considered reviewing our performance evaluation mechanism, especially the logging part. In all of our simulations with Contiki Cooja, we print the associated logs using C *prinf* functions. At the end of the simulations, a performance evaluation code developed by us reads the log files and calculates the performance metrics during that simulation. This is one of the drawback of Contiki Cooja since it does not provide performance results inherently. Before the performance evaluation of each study, we make sure that the logging mechanism runs correctly. When we compared these two studies in terms of performance metrics, we realized that significantly more performance metrics are logged in the latter study in comparison to the earlier one. *Packet delivery ratio*, *control message overhead*, *average network delay* and *average power consumption* , *routing loops*, *absence of a route*, *packet drops due to errors signaled in RPL header options of IPv6 packets*, *MAC layer retransmissions* and *change in rank* are measured in the multiple attacker simulations. However only the first four metrics (i.e., *Packet delivery ratio*, *control message overhead*, *average network delay* and *average power consumption*) are measured in the earlier study. We suspected that measuring many performance parameters might have caused an overhead which affects the simulation outcomes. In order to validate this idea, we started to remove the performance parameters from the simulations with Contiki 3.0 one by one, run the simulations again and compare the performance. What we realized after running tens of simulations with 95% confidence intervals is that, among all the performance parameters, logging the dropped fragments at the MAC layer affects the simulation results by 5%. We found out that our logging mechanism at the MAC layer does some math operations in order to print the fragment content in human readable character strings. And these math operations cause the node to be busy with calculations. We fixed this bug immediately. However, realizing this bug and fixing it did not make much difference for our investigations. Our analysis showed that the underlying reason for obtaining different results is not related to the logging mechanism.

As the final analysis, we started to compare the RPL implementations of Contiki versions 2.7 and 3.0. When we compared the implementations, we realized that the RPL implementation within the updated version has significant differences. There exist several differences in the RPL implementations of two versions, which are related to forwarding or dropping the packets, resetting DIO timer, applying global repair as a result of header verification errors, sending control messages in specific situations, etc. We concluded that the operating system updates caused us to obtain different results for RPL networks under the attack of VNA.

7.3 Mitigation of Multiple Attackers with Elimination and Shield Techniques

In this section, we analyze how effective our novel mitigation techniques, namely Elimination and Shield (Chapter 6) are against multiple version number attackers.

We performed simulations in order to analyze the performance of our novel mitigation techniques. For this aim, we selected a set of attacking positions for two and three attackers cases in the Grid Topology (Figure 7.3) and placed our mitigation techniques to each legitimate node. The attacking positions we selected are as follows:

- Two attackers positions: 2-6, 2-5, 4-13, 7-10, 11-14, 12-16,
- Three attackers positions: 2-5-6, 4-13-16, 7-10-14, 7-10-11, 12-15-16.

We selected these positions based on the strongest attacking positions for power consumption results for multiple attacker cases, which were given in Figure 7.12 and Figure 7.13, positions that are close to the DODAG root and positions that are at the edges of the topology.

The simulation results for the analysis of our mitigation techniques against multiple attackers are given in Figure 7.20. The figure shows the packet delivery ratio with respect to the selected two and three attacker positions based on the employed mitigation technique where RPL Under Attack shown in blue signifies no mitigation technique and RPL+Elimination and RPL+Shield signify elimination and shield techniques which are running on legitimate nodes respectively.

The simulation results clearly show that, our mitigation techniques are highly effective for multiple attacker scenarios. Shield technique again provides better results in

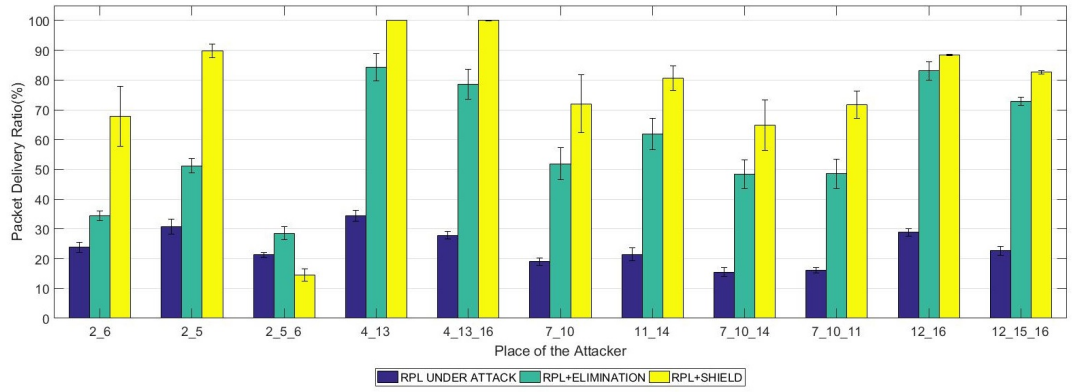

Figure 7.20 : Multiple attackers mitigation results for packet delivery ratio.

comparison to Elimination. The highest packet delivery ratios (100%) were obtained for attacking positions that are at the edges of the topology (i.e., positions 4-13 and 4-13-16).

The strongest attacking position for two attackers case was the position of 7-10 which had packet delivery ratio of 18% for RPL under attack. Another strong position was the position of 11-14 with packet delivery ratio of 21%. Elimination technique mitigates the effect of the attack and enables RPL network to have a packet delivery ratio of 51% under attack. Whereas Shield achieves a better result and by means of Shield, even for the strongest attacking position, we can reach to 72% packet delivery ratio. For the attacking position of 11-14, Elimination yields to 61% and Shield yields to 80% packet delivery ratios.

For the strongest attacking positions for three attackers cases, which are the positions of 7-10-14 (packet delivery ratio of 15%) and 7-10-11 (packet delivery ratio of 16%), Elimination and Shield help us to improve the packet delivery ratios. Elimination yields to 48% for both of the positions. However, Shield yields to 64% for the attacking position of 7-10-14 and 71% for 7-10-11.

Only one position combination causes our mitigation techniques not to be effective, which is shown as the position of 2-5-6 where three attackers are placed to the positions of 2, 5 and 6. This result is very reasonable considering the logic of our mitigation techniques. As we explained in Chapter 6, our mitigation techniques were formed based on two important points:

- For Version Number Attack, attacking positions that are far from the root have the strongest effect on the performance of the network,

- Legitimate Version Number Updates can be initiated only by the DODAG root and that is why such an information is disseminated through the network from root's side down to the leaf nodes' sides.

These important points formed our mitigation techniques. When we consider the attacking position of 2, 5 and 6 in our topology given in Figure 7.3, these three positions are the closest positions to the DODAG root. And they are the critical positions where no other node exists in the DODAG apart from these nodes that, any information to and from the DODAG root can pass through. In other words, this is the only attacking position combination that completely eliminates our mitigation techniques. Because they will always have better rank values than any other non-root node in the DODAG (disables the effect of Elimination technique) and they will always be the only parents that will act like a trustworthy parents for other nodes (disables the effect of Shield technique). This is the reason why our mitigation techniques seem to fail. Results show that Elimination still seems to improve the packet delivery ratio but Shield seems to cause lower delivery ratios. Since Shield is ineffective by design here and since it causes processing overhead to the nodes, it is reasonable for RPL network to have a lower packet delivery ratio with Shield than RPL without any mitigation technique here. Nevertheless, since this position combination (2-5-6) is the closest position to the DODAG root, it would be highly probable by any other security mechanism or network administrator to find out the attacking nodes.

Mitigation performance of Elimination and Shield techniques in this section may look lower than the performance obtained in Chapter 6. The underlying reason is related to the RPL configuration details in both studies. In our earlier study which we explained in Chapter 6, we needed to understand how our novel mitigation techniques will act against legitimate version number updates. However, there was a point with Contiki RPL implementation in which there was no mechanism for DODAG root in Contiki RPL to update the version number. Interestingly, although it was not changing the version number, if it receives a DIO with a newer version number, then it was updating the version number of the DODAG and starting global repair operation without even considering who updated the version number. In order to analyze the accuracy results of our mitigation techniques, as we mentioned in Chapter 6, we had modified the RPL implementation so that DODAG root initiates the version number updates with respect

to intervals with negative exponential distribution and it does not update the version number at all except for these update points. However in this study, we kept the default RPL implementation and integrated our Elimination and Shield mechanisms. When we consider these two RPL configurations, we can say that in the previous configuration less version number updates were being initiated by the DODAG root since it was not changing the VN except for the update points. However, in the latter configuration any malicious update which was not successfully filtered by our mitigation techniques can force the DODAG root to change the version number, thus start another network wide repair operation. Mitigation performance of Elimination and Shield techniques in this study are not as much high as the ones in Chapter 6 with this reason.

7.4 Conclusion of the Section

In this chapter we focused on analyzing the effect of multiple VN attackers. With this aim, we firstly ran simulations on an initial topology. Afterwards, we created a grid topology with 16 nodes and determined attacking positions for single, two and three attackers. Our simulations led us to the results that we had not observed in our earlier study.

In our earlier study, we had found out that the packet delivery ratio and the control packet overhead results were highly correlated to the location of the attacker. In terms of power consumption and average network delay, we had not realized such a correlation. These findings, except for the power consumption results, had also been verified by a predecessor study [10] (the predecessor study does not consider the power consumption results).

In this study we had single, two and three attackers. We performed two sets of comparisons. In the first set of comparisons, we investigated the effect of increasing the number of attackers for three performance metrics. We realized that, increasing the number of attackers affects only the packet delivery ratios. In the second set of comparisons, for three performance metrics, we considered each attacker class separately. For average network delay we realized that affect of the attack gets stronger for the position combinations that are close to the DODAG root. For packet delivery ratio results and power consumption per successfully delivered packets, attackers at the center of the network caused the highest packet losses. These results were interesting

for us since our earlier study and another study had not realized such affects. Therefore, we conducted analysis to find out the underlying reasons. We considered topological differences, logging mechanisms and operating system changes. Finally, we concluded that, OS updates are the underlying reason for us to obtain different results for an RPL under VNA.

In the last part of the chapter, we evaluated the performance of our mitigation techniques against multiple attackers based on packet delivery ratio results. We found out that our mitigation techniques successfully mitigate the effect of multiple attackers.

8. CONCLUSIONS

In this thesis, we studied security of the IoT from the DoS Attacks point of view. We specifically focused on a subclass of the IoT, namely the LLNs in which resource-constrained devices form a network to connect to the Internet in a lossy environment. The IEEE and the IETF proposed several standards and protocols for LLNs which bear the characteristics of such networks in mind. For the PHY and MAC layer, IEEE 802.15.4 standard was proposed. Since IPv4 address space is already consumed, standardization organizations considered to employ the IPv6 addressing scheme for devices in the LLNs. In order to route the IPv6 packets within the LLNs, the RPL was proposed by the IETF. Since the maximum transmission unit is 127 bytes in the IEEE 802.15.4-based standards, the IETF proposed an adaptation layer under the name of the 6LoWPAN, which provides header compression mechanisms in order to fit IPv6 packets into the MAC layer fragments. On top of the network layer, standardization organizations considered to use the UDP, instead of the TCP for the LLNs due to the resource-constraints. For the application layer and above, the CoAP and the CBOR were proposed by the IETF.

As we study the security of the LLNs, we analyzed the standardization efforts to secure the LLN-based IoT networks. Considering the pros and cons of the security solutions, resource-constraints, QoS requirements, implementation challenges, deployment environments and users, the Internet connectivity and the D/DoS attacks targeting the existing computer networks, we believed that there is a need for the LLNs to be secured against such attacks. Hence, we researched the D/DoS attacks which may target the LLNs. Our research revealed that, the LLNs can be the target of D/DoS attacks which can be either new (due to new protocols and standards) or old (attacks that have been targeting the existing networks). We analyzed the works that analyze the effects of the attacks, cryptography-based security solutions, protocol security studies and intrusion detection and mitigation systems that are specifically designed for the LLNs.

After our analysis, we decided to focus on the RPL protocol and the RPL VN attack. In VNA, an attacker maliciously changes the VN, thus initiates an illegitimate global repair operation. However, in an ordinary RPL network, global repair operation can be started only by the DODAG root. Our motivation to choose the VNA among the other attacks were manifold. First of all, it was a pretty new DoS attack and it was not studied much in the literature. Since it was new, there was no detection or mitigation study or any in-depth analysis. It was also an interesting attack for us that, it did not need thousands of attackers, but only a single attacker was enough to affect the whole network.

In order to analyze the effect of the VNA, we created a realistic topology considering the IETF's routing requirements documents for the LLNs. Using the Contiki Cooja, we simulated a single attacker at several positions within the RPL DODAG. Our topology had mobile nodes and hence, mobile attackers. We employed a probabilistic attacker model to identify the best setting for an attacker, not only in terms of the position, but also the attacking probability. Our analysis with respect to packet delivery ratio, RPL control message overhead, average network delay and average power consumption metrics showed that, control message overhead and packet delivery ratio results are highly correlated to the position of the attacker. Effect of the attack gets stronger as the attacker gets farther from the DODAG root. In terms of the attacking probability and the effect of the attack, we figured out that, performance of the network gets worse as the probability of attacking increases. Simulation results for mobile attackers show that, mobile attackers have nearly the same detrimental effects on the network as the best attacking positions for a static attacker.

Our analysis with a single attacker revealed that, the VNA is a very serious threat for the RPL-based LLNs. It misuses the global repair operation of the RPL protocol and forces the complete network to be rebuilt over and over again. During this time, an excess amount of RPL control messages are generated, majority of the application data packets are lost, average delay and power consumption are severely increased. Results showed that, effect of the attack gets more detrimental when the attacker is far from the root. Which implies that, the leaf positions of the DODAG are the best positions for an attacker to start an illegitimate VN update. However, the RPL specification indicates that, the VN update can only be initiated by the DODAG root,

which is completely the opposite direction. These two observations led us to come up with a new and very basic mechanism which would allow the VN updates coming from the direction of the DODAG root and stop the VN updates coming from the opposite direction (i.e., from the direction of leaf nodes). We considered to use the rank parameter in order to understand the position of nodes relative to each other. We named it as the *elimination* technique, which checks the sender of the incoming DIO with a greater VN. If the sender has a worse rank than the receiver node, it shows that it is positioned at the lower portions of the DODAG. Hence, the DIO is discarded since such an update should not come from that portions of the DODAG. We believed that the elimination technique would be very efficient to mitigate the effect of the VNA consisting of a single attacker. Since it eliminates the VN updates started from the direction of leaf nodes (positions at which VNA is the strongest) towards the DODAG root, the strongest attacking positions could be eliminated by this way. However, if the attacker is not far from the root, in other words, if the attacker is an intermediate node within the DODAG, then it can still affect the lower portions of the DODAG since the elimination technique will allow it. In order to overcome this issue and propose a complete solution, we came up with the *shield* mechanism that is based on the elimination technique. In shield, every node works on its own and keeps a list of its neighbors that have a better rank value than itself in a table, which we call as *ShieldList*. If it receives a DIO with a greater VN (which signifies that a VN update is taking place), it firstly checks whether the DIO sender is in its ShieldList or not. If it does not exist within the table, then it discards the DIO. Because, such a DIO can come only from the direction of the DODAG root and possible senders are already listed in the ShieldList table. However, if the DIO sender exists within the table, then comes a trust mechanism, in which the receiving node expects other neighboring nodes (nodes within the ShieldList table) to announce such an update. It waits until majority of the ShieldList entries claim a VN update. Hence, it does not trust a single neighbor claiming a VN update, but trusts more neighbors announcing the same update. Thus, it can provide a solution not only against the attackers at the leaf nodes, but also attackers at the intermediate positions. Our simulation results show that, by means of the proposed novel mitigation techniques, the delay caused by the attacker can be shortened up to 87%, the average power consumption can be reduced up to 63%, the control message overhead can be lowered up to 71% and the data packets

delivery ratio can be increased up to 86%. The proposed techniques, while allowing the ordinary RPL operation, trade off the mitigation performance against the resource overheads and thus allow network administrators to choose the right scheme for their RPL network.

In the last part of our thesis study, we focused on multiple VN attackers. We performed two sets of comparisons. In the first set of comparisons, we investigated the effect of increasing the number of attackers. In the second set of comparisons, we analyzed the effect of multiple attackers positions. Simulation results showed that, increasing the number of attackers affects only the packet delivery ratios. In terms power consumption per successfully delivered packets, the lowest values were obtained for the positions that are at the edges of the topology. In terms of distance to the DODAG root, closer positions seem to cause more power consumption per successfully delivered packets values than the farther positions. For packet delivery ratio and average network delay results, attackers at the center of the network caused the highest packet losses and longest delays. These results were interesting for us since our earlier study and a predecessor study had not realized such affects. Therefore, we conducted analysis to find out the underlying reasons. We considered topological differences, logging mechanisms and operating system changes. We concluded that, OS updates are the underlying reason for us to obtain different results for an RPL network under the VNA. Finally, we evaluated the performance of our mitigation techniques against multiple attackers based on packet delivery ratio results. We found out that our mitigation techniques successfully mitigate the effect of multiple attackers.

All along our thesis study, we used Contiki OS and its simulator Cooja for implementations and performance analysis. Contiki is an open source operating system developed specifically for the IoT devices. It provides the implementations of the standardized protocol stack as well as libraries (e.g., data structures, timers, memory management) for application developers. Cooja simulator comes with Contiki, which employs the emulations of TI MSP-430-based microcontrollers. Using Cooja, it is possible to quickly simulate an IoT network consisting of IoT nodes with Contiki application images. However, there is one drawback with Contiki and Cooja. It is troublesome to evaluate the performance of your custom IoT application. Network simulators (e.g., NS, OPNET) usually have built-in libraries or modules that help a

researcher to create traffic types (i.e., constant bit rate, Poisson, self-similar traffic). However in Contiki, we have to implement it to the application source code manually. Secondly, Contiki does not provide built-in performance evaluation support, which again is usually provided by the network simulators. In Contiki, we have to put C *printf* statements to generate logs, which we process the logs at the end of the simulations to calculate the performance. This may not be too difficult for some metrics, such as packet delivery ratio, average delay. However, questions such as:

- Where do we see the dropped packets? At which layer of the protocol stack?
- What was the reason of a packet to be dropped at a specific layer?
- Can we see the packets that are still in the queues of the nodes?
- Why did the node send that control message?
- Why do we get different results for different versions of the operating system?

are quite difficult to answer when we are working with Contiki. In addition to this, since we print the log messages for every node, the simulation times get longer as the number of parameters that are logged increase. Moreover, not only the logging parameters affect the simulation times, but also the number of nodes within the simulation. Therefore, it takes very long to run simulations that have nodes more than 36 nodes in the topology. This was the actual reason of us for considering alternative ways to reduce the number of simulations for performance evaluations. Nevertheless, we had to use Contiki because no alternative OS was both providing the open source implementations of the protocol stack and simulator at the same time.

In this thesis, we started with studying the security of the LLNs and D/DoS attacks targeting such networks. Then we focused on the routing layer and a DoS attack that targets the RPL-based LLNs, namely the VNA. We analyzed the effect of it, we came up with two novel mitigation techniques to reduce the effects of it. Finally, we studied multiple VNAs. However, this is not the end of the journey for secure IoT networks. There exists several D/DoS attacks targeting the LLNs and the VNA is only one of them. We have to consider other attacks too. At this point, IDSes come into the mind. However, existing IDSes proposed for IoT target only a few attacks. When we

consider the categorization of the attacks with respect to the layers, we can see that application layer and above were not studied intensively for the LLNs. So, only the insider attackers were studied. However, since IoT networks will have the Internet connection, we can not ignore the outsider attackers. In addition to this, all of the studies in this respect aim to protect the IoT networks from the attackers. However, we have to think of scenarios where compromised IoT networks are used to apply D/DoS attacks. Therefore, we need to investigate what we can do in such scenarios. All in all, security of IoT networks is still a crucial question with many unexplored topics of research.

REFERENCES

[1] **Evans, D.** (2011). The Internet of Things How the Next Evolution of the Internet Is Changing Everything, `http://www.cisco.com/web/about/ac79/docs/innov/IoT\_IBSG\_0411FINAL.pdf`.

[2] **Al-Fuqaha, A.**, **Guizani, M.**, **Mohammadi, M.**, **Aledhari, M. and Ayyash, M.** (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys Tutorials*, *17*(4), 2347–2376.

[3] **Networks, A.**, (2018), Digital Attack Map Top daily DDoS attacks worldwide, http://www.digitalattackmap.com/, Retrieval date: Jan. 2018.

[4] (2016), DDoS on Dyn Impacts Twitter, Spotify, Reddit, https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/, Retrieval date: Dec. 2016.

[5] **UC Berkeley Center for Long-Term Cybersecurity** (2016), *Cybersecurity Futures 2020*, .

[6] **Winter, T.**, **Thubert, P.**, **Brandt, A.**, **Hui, J.**, **Kelsey, R.**, **Levis, P.**, **... and Alexander, R.**, (2012), RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550 (Proposed Standard).

[7] **Palattella, M.**, **Accettura, N.**, **Vilajosana, X.**, **Watteyne, T.**, **Grieco, L.**, **Boggia, G. and Dohler, M.** (2013). Standardized Protocol Stack for the Internet of (Important) Things, *IEEE Communications Surveys Tutorials*, *15*(3), 1389–1406.

[8] **SICS** (2006), *A Sensor Network Simulator for the Contiki OS*, T2006:5.

[9] (2015), Contiki: The Open Source OS for the Internet of Things, `http://www.contiki-os.org/`.

[10] **Mayzaud, A.**, **Sehgal, A.**, **Badonnel, R.**, **Chrisment, I. and Schonwälder, J.**, (2014). A Study of RPL DODAG Version Attacks, Monitoring and Securing Virtualized Networks and Services, volume8508 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp.92–104.

[11] **Khan, R.**, **Khan, S.U.**, **Zaheer, R. and Khan, S.** (2012). Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges, *2012 10th International Conference on Frontiers of Information Technology*, pp.257–260.

[12] **Wu, M.**, **Lu, T.J.**, **Ling, F.Y.**, **Sun, J. and Du, H.Y.** (2010). Research on the architecture of Internet of Things, *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, volume 5, pp.V5–484–V5–487.

[13] **Yang, Z.**, **Yue, Y.**, **Yang, Y.**, **Peng, Y.**, **Wang, X. and Liu, W.** (2011). Study and application on the architecture and key technologies for IoT, *2011 International Conference on Multimedia Technology*, pp.747–751.

[14] **Brandt, A.**, **Buron, J. and Porcu, G.**, (2010), Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826 (Informational), `http://www.ietf.org/rfc/rfc5826.txt`.

[15] **Martocci, J.**, **Mil, P.D.**, **Riou, N. and Vermeylen, W.**, (2010), Building Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5867 (Informational), `http://www.ietf.org/rfc/rfc5867.txt`.

[16] **Pister, K.**, **Thubert, P.**, **Dwars, S. and Phinney, T.**, (2009), Industrial Routing Requirements in Low-Power and Lossy Networks, RFC 5673 (Informational), `http://www.ietf.org/rfc/rfc5673.txt`.

[17] **Kettimuthu, R. and Muthukrishnan, S.** (2005). Is Bluetooth suitable for large-scale sensor networks?, *ICWN*, Citeseer, pp.448–454.

[18] **Krco, S.** (2002). Bluetooth based wireless sensor networks–implementation issues and solutions, *URL: http://www. telfor. org. yu/radovi/4019. pdf*.

[19] (2011). IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low Rate Wireless Personal Area Networks, *IEEE Std. 802.15.4-2011*.

[20] **Hui, J. and Thubert, P.**, (2011), Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, RFC 6282 (Proposed Standard).

[21] **Shelby, Z.**, **Hartke, K. and Bormann, C.**, (2014), The Constrained Application Protocol (CoAP), RFC 7252 (Proposed Standard).

[22] **Bormann, C. and Hoffman, P.**, (2013), Concise Binary Object Representation (CBOR), RFC 7049 (Proposed Standard), `https://www.rfc-editor.org/rfc/rfc7049.txt`.

[23] **Kent, S. and Seo, K.**, (2005), Security Architecture for the Internet Protocol, RFC 4301 (Proposed Standard), `https://www.rfc-editor.org/rfc/rfc4301.txt`, updated by RFCs 6040, 7619.

[24] **Rescorla, E. and Modadugu, N.**, (2006), Datagram Transport Layer Security, RFC 4347 (Proposed Standard), `https://www.rfc-editor.org/rfc/rfc4347.txt`, obsoleted by RFC 6347, updated by RFCs 5746, 7507.

[25] **Raza, S.**, **Duquennoy, S.**, **Chung, T.**, **Yazar, D.**, **Voigt, T. and Roedig, U.** (2011). Securing communication in 6LoWPAN with compressed IPsec, *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pp.1–8.

[26] **Kurose, J.F. and Ross, K.W.** (2012). *Computer Networking: A Top-Down Approach (6th Edition)*, Pearson.

[27] **Arce, I.**, **Clark-Fisher, K.**, **Daswani, N.**, **DelGrosso, J.**, **Dhillon, D. and Kern, Christoph ... Schoenfield, B.** (2014). Avoiding the top 10 software security design flaws, *Technical report, IEEE Computer Societys Center for Secure Design (CSD)*.

[28] **Acar, Y.**, **Backes, M.**, **Fahl, S.**, **Kim, D.**, **Mazurek, M.L. and Stransky, C.** (2017). How Internet Resources Might Be Helping You Develop Faster but Less Securely, *IEEE Security Privacy*, *15*(2), 50–60.

[29] **Schneier, B.** (2016). Stop Trying to Fix the User, *IEEE Security Privacy*, *14*(5), 96–96.

[30] **Wallgren, L.**, **Raza, S. and Voigt, T.** (2013). Routing Attacks and Countermeasures in the RPL-Based Internet of Things, *International Journal of Distributed Sensor Networks*, *2013*, 11.

[31] **Aris, A.**, **Oktug, S.F. and Yalcin, S.B.O.** (2015). Internet-of-Things security: Denial of service attacks, *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, pp.903–906.

[32] **Amin, Y.M. and Abdel-Hamid, A.T.** (2016). A Comprehensive Taxonomy and Analysis of IEEE 802.15.4 Attacks, *Journal of Electrical and Computer Engineering*, *2016*, 1–12.

[33] **Razzak, F.** (2012). Spamming the Internet of Things: A Possibility and its probable Solution, *Procedia Computer Science*, *10*, 658 – 665.

[34] **Sokullu, R.**, **Dagdeviren, O. and Korkmaz, I.** (2008). On the IEEE 802.15.4 MAC Layer Attacks: GTS Attack, *SENSORCOMM '08. Second International Conference on Sensor Technologies and Applications, 2008.*, pp.673–678.

[35] **Hummen, R.**, **Hiller, J.**, **Wirtz, H.**, **Henze, M.**, **Shafagh, H. and Wehrle, K.** (2013). 6LoWPAN Fragmentation Attacks and Mitigation Mechanisms, *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '13, pp.55–66.

[36] **Le, A.**, **Loo, J.**, **Chai, K.K. and Aiash, M.** (2016). A Specification-Based IDS for Detecting Attacks on RPL-Based Network Topology, *Information*, *7*(2), `http://www.mdpi.com/2078-2489/7/2/25`.

[37] **Raza, S.**, **Wallgren, L. and Voigt, T.** (2013). SVELTE: Real-time intrusion detection in the Internet of Things , *Ad Hoc Networks*, *11*(8), 2661 – 2674.

[38] **Weekly, K. and Pister, K.** (2012). Evaluating sinkhole defense techniques in RPL networks, *2012 20th IEEE International Conference on Network Protocols (ICNP)*, pp.1–6.

[39] **Pongle, P. and Chavan, G.** (2015). Article: Real Time Intrusion and Wormhole Attack Detection in Internet of Things, *International Journal of Computer Applications*, *121*(9), 1–9.

[40] **Kasinathan, P.**, **Pastrone, C.**, **Spirito, M. and Vinkovits, M.** (2013). Denial-of-Service detection in 6LoWPAN based Internet of Things, *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp.600–607.

[41] **Le, A.**, **Loo, J.**, **Lasebae, A.**, **Vinel, A.**, **Chen, Y. and Chai, M.** (2013). The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks, *IEEE Sensors Journal*, *13*(10), 3685–3692.

[42] **Aris, A.**, **Oktug, S.F. and Yalcin, S.B.O.** (2016). RPL version number attacks: In-depth study, *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp.776–779.

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

[44] **Sehgal, A.**, **Mayzaud, A.**, **Badonnel, R.**, **Chrisment, I. and Schonwalder, J.** (2014). Addressing DODAG inconsistency attacks in RPL networks, *Global Information Infrastructure and Networking Symposium (GIIS), 2014*, pp.1–8.

[45] **Mayzaud, A.**, **Sehgal, A.**, **Badonnel, R.**, **Chrisment, I. and Schönwälder, J.** (2015). Mitigation of topological inconsistency attacks in RPL-based low-power lossy networks, *International Journal of Network Management*, *25*(5), 320–339, `http://dx.doi.org/10.1002/nem.1898`, nem.1898.

[46] **Renuka Venkata Ramani, C.**, (2016), Two way Firewall for Internet of Things.

[47] **Cho, E.J.**, **Kim, J.H. and Hong, C.S.**, (2009). Attack Model and Detection Scheme for Botnet on 6LoWPAN, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.515–518.

[48] **Le, A.**, **Loo, J.**, **Luo, Y. and Lasebae, A.** (2011). Specification-based IDS for securing RPL from topology attacks, *2011 IFIP Wireless Days (WD)*, pp.1–3.

[49] **Misra, S.**, **Krishna, P.V.**, **Agarwal, H.**, **Saxena, A. and Obaidat, M.S.** (2011). A Learning Automata Based Solution for Preventing Distributed Denial of Service in Internet of Things, *Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, ITHINGSCPSCOM '11, pp.114–122.

[50] **Kasinathan, P.**, **Costamagna, G.**, **Khaleel, H.**, **Pastrone, C. and Spirito, M.A.** (2013). DEMO: An IDS Framework for Internet of Things Empowered by 6LoWPAN, *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, pp.1337–1340.

[51] **Amaral, J.P.**, **Oliveira, L.M.**, **Rodrigues, J.J.P.C.**, **Han, G. and Shu, L.** (2014). Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, *2014 IEEE International Conference on Communications (ICC)*, pp.1796–1801.

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

[53] **Sedjelmaci, H.**, **Senouci, S.M. and Al-Bahri, M.** (2016). A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology, *2016 IEEE International Conference on Communications, ICC 2016*.

[54] **Mayzaud, A.**, **Badonnel, R. and Chrisment, I.** (2016). Detecting Version Number Attacks Using a Distributed Monitoring Architecture, *Proc. of IEEE/IFIP/In Assoc. with ACM SIGCOMM International Conference on Network and Service Management (CNSM 2016)*, pp.127–135.

[55] **Mayzaud, A.**, **Sehgal, A.**, **Badonnel, R.**, **Chrisment, I. and Schönwälder, J.** (2016). Using the RPL protocol for supporting passive monitoring in the Internet of Things, *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp.366–374.

[56] **Saeed, A.**, **Ahmadinia, A.**, **Javed, A. and Larijani, H.** (2016). Intelligent Intrusion Detection in Low-Power IoTs, *ACM Trans. Internet Technol.*, *16*(4), 27:1–27:25, `http://doi.acm.org/10.1145/2990499`.

[57] **Aris, A. and Oktug, S.F.** (2017). Poster: State of the Art IDS Design for IoT, *International Conference on Embedded Wireless Systems and Networks (EWSN) 2017*.

[58] **Levis, P.**, **Clausen, T.**, **Hui, J.**, **Gnawali, O. and Ko, J.**, (2011), The Trickle Algorithm, RFC 6206 (Proposed Standard), `https://www.rfc-editor.org/rfc/rfc6206.txt`.

[59] **Tsao, T.**, **Alexander, R.**, **Dohler, M.**, **Daza, V.**, **Lozano, A. and Richardson, M.**, (2015), A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs), RFC 7416 (Informational), `http://www.ietf.org/rfc/rfc7416.txt`.

[60] (2012). IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low Rate Wireless Personal Area Networks Amendment 1: MAC Sublayer, *IEEE Std. 802.15.4e*.

[61] **Nguyen, K.T.**, **Laurent, M. and Oualha, N.** (2015). Survey on secure communication protocols for the Internet of Things, *Ad Hoc Networks*, *32*, 17 – 31, internet of Things security and privacy: design methods and optimization.

[62] **Arış, A.**, **Oktuğ, S.F. and Voigt, T.**, (2018). Security of Internet of Things for a Reliable Internet of Services, Autonomous Control for a Reliable Internet of Services: Methods, Models, Approaches, Techniques, Algorithms, and Tools, Springer International Publishing, Cham, pp.337–370, https://doi.org/10.1007/978-3-319-90415-3_13.

[63] **Dohler, M.**, **Watteyne, T.**, **Winter, T. and Barthel, D.**, (2009), Routing Requirements for Urban Low-Power and Lossy Networks, RFC 5548 (Informational), http://www.ietf.org/rfc/rfc5548.txt.

[64] Moteiv Corporation, (2006). Tmote Sky: Datasheet, http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.

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

[66] **Mayzaud, A.**, **Badonnel, R. and Chrisment, I.** (2016). A Taxonomy of Attacks in RPL-based Internet of Things, *International Journal of Network Security*, *18*(3), 459 – 473,, https://hal.inria.fr/hal-01207859.

[67] **Otoum, S.**, **Kantarci, B. and Mouftah, H.T.** (2017). Detection of Known and Unknown Intrusive Sensor Behavior in Critical Applications, *IEEE Sensors Letters*, *1*(5), 1–4.

[68] **Otoum, S.**, **Kantarci, B. and Mouftah, H.** (2018). Adaptively Supervised and Intrusion-Aware Data Aggregation for Wireless Sensor Clusters in Critical Infrastructures, *2018 IEEE International Conference on Communications (ICC)*, pp.1–6.

[69] **Zarpelão, B.B.**, **Miani, R.S.**, **Kawakani, C.T. and de Alvarenga, S.C.** (2017). A survey of intrusion detection in Internet of Things , *Journal of Network and Computer Applications*, *84*, 25 – 37.

[70] **Heurtefeux, K.**, **Erdene-Ochir, O.**, **Mohsin, N. and Menouar, H.** (2015). Enhancing RPL Resilience Against Routing Layer Insider Attacks, *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, pp.802–807.

[71] **Ma, G.**, **Li, X.**, **Pei, Q. and Li, Z.** (2017). A Security Routing Protocol for Internet of Things Based on RPL, *2017 International Conference on Networking and Network Applications (NaNA)*, pp.209–213.

[72] **Kallapur, P.V.**, **Ranjan, N.**, **Vidyarthi, R.**, **Anshuman and Singh, V.** (2017). Enhanced variant of RPL for improved security, *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp.2302–2306.

[73] **Perrey, H.**, **Landsmann, M.**, **Ugus, O.**, **Wählisch, M. and Schmidt, T.C.** (2016). TRAIL: Topology Authentication in RPL, *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, Junction Publishing, USA, pp.59–64.

[74] **Nikravan, M.**, **Movaghar, A. and Hosseinzadeh, M.** (2018). A Lightweight Defense Approach to Mitigate Version Number and Rank Attacks in Low-Power and Lossy Networks, *Wireless Personal Communications*, https://doi.org/10.1007/s11277-017-5165-4.

[75] **Ahmed, F. and Ko, Y.B.** (2016). A Distributed and Cooperative Verification Mechanism to Defend against DODAG Version Number Attack in RPL, *Proceedings of the 6th International Joint Conference on Pervasive and Embedded Computing and Communication Systems - Volume 1: PEC, (PECCS 2016)*, INSTICC, SciTePress, pp.55–62.

[76] (2017), TinyOS RPL, https://github.com/tinyos/tinyos-main/tree/master/tos/lib/net/rpl, Retrieval date: Dec. 2017.

[77] (2017), RIOT RPL, https://github.com/RIOT-OS/RIOT/blob/master/sys/net/gnrc/routing/rpl/gnrc_rpl_dodag.c, Retrieval date: Dec. 2017.

[78] (2017), Contiki RPL, https://github.com/contiki-os/contiki/blob/release-3-0/core/net/rpl/rpl-dag.c, Retrieval date: Dec. 2017.

[79] CISCO, (2015). Routing Protocol for LLN (RPL) Configuration Guide, cisco ios release 15m t edition.

CURRICULUM VITAE

Name Surname: Ahmet Arış

Place and Date of Birth: Fethiye, Turkey, 1 October 1986

E-Mail: arisahmet@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2009, Bahçeşehir University, Faculty of Engineering and Natural Sciences, Computer Engineering
- **M.Sc.:** 2012, Istanbul Technical University, Graduate School of Science, Engineering and Technology, Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2015–2018 Research and Teaching Assistant at Istanbul Technical University, Faculty of Computer and Informatics Engineering
- 2017 Visiting Researcher at SICS, Sweden

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Arış, A.**, Örs Yalçın, S. B., and Oktuğ, S. F., March 2019. New Lightweight Mitigation Techniques for RPL Version Number Attack, *Elsevier Ad Hoc Networks,* vol. 85, pp. 81-91, 2019. DOI: 10.1016/j.adhoc.2018.10.022.
- **Arış, A.**, Oktuğ, S. F., and Voigt, T. May 2018. Security of Internet of Things for a Reliable Internet of Services, *Springer, Autonomous Control for a Reliable Internet of Services, Lecture Notes in Computer Science, vol 10768* vol. 10768, pp. 337-370, 2018. DOI: 10.1007/978-3-319-90415-3_13.
- **Arış, A.**, and Oktuğ, S. F., February 2017. Poster: State of the Art IDS Design for IoT, *International Conference on Embedded Wireless Systems and Networks (EWSN), Junction Publishing, USA* pp. 196-197, ISBN: 978-0-9949886-1-4
- **Arış, A.**, Oktuğ, S. F., and Örs Yalçın, S. B., April 2016. RPL Version Number Attacks: In-depth Study, *2016 IEEE/IFIP Network Operations and Management Symposium,* 2016. DOI: 10.1109/NOMS.2016.7502897.
- **Arış, A.**, Oktuğ, S. F., and Örs Yalçın, S. B., May 2015. Internet of Things Security: Denial of Service Attacks, *2015 23nd Signal Processing and Communications Applications Conference (SIU), IEEE* 2015. DOI: 10.1109/SIU.2015.7129976.