République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA

Faculté des Sciences Exactes
Département Informatique

THÈSE

Présentée par

DJEDJIG Nabil

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud Computing

Thème

Negotiation Protocols to Establish a Trust Relationship in the Internet of Things

Soutenue le : 02/12/2021 Devant le Jury composé de :

Nom et Prénom	Grade		
Mr YAZID Mohand	MCA	Univ. de Bejaia	Président
Mr TANDJAOUI Djamel	Directeur de recherche	CERIST, Alger	Rapporteur
Mr BENCHAÏBA Mahfoud	Professeur	Univ. de USTHB	Examinateur
Mr ZAFFOUNE Youcef	MCA	Univ. de USTHB	Examinateur
Mme BOULAHROUZ Djamila	MCA	Univ. de Bejaia	Examinatrice
Mr ROMDHANI Imed	Professeur Associé	Univ. de Napier	Invité

Année Universitaire : 2020/2021

Dedication

To my parents, for their constant support and unceasing love

To my precious wife Faiza, for her affection, devotion and patience

To our princess daughter Maria for her loving little heart

To my sisters, brothers, and my family in-laws

To my whole family.

Acknowledgements

First and foremost, I thank Allah Almighty for giving me the strength, the courage and the knowledge to complete my doctoral degree.

I would like to express my deepest gratitude to my supervisor Research Director Djamel Tandjaoui, for his continuous support and guidance, monitoring, orientations and relevant remarks, which steered me in the right direction to complete my work.

I sincerely thank Dr Imed Romdhani for his support and valuable comments. It was a great pleasure and honour to work with him. He greeted me to his research laboratory at Napier University in Edinburgh, which helped me a lot to build a robust and useful research collaboration.

I would like to thank Dr YAZID Mohand (Universite de Bejaia) for agreeing to chair my thesis jury, as well as Prof. BENCHAIBA Mahfoud (USTHB), Dr ZAFFOUNE Youcef (USTHB) and Dr BOULAHROUZE Djamila (Universite de Bejaia) for honouring me with their acceptance as examiners in my thesis jury.

I would like to express my deepest gratitude to my parents for their support, love, prayers, patience, and sacrifices. It is to them that I owe all the success in my life. I would also like to extend my sincere thanks to my sisters, brothers, and my family in-laws for their support and encouragements. Big ups for my lovely daughter Maria. Finally, I accomplish this recognition by thanking my lovely wife for giving me more than enough love, understanding, and strength necessary to complete my work. Nothing could have made sense without having my wonderful wife at my side.

I would like to thank my friends -especially Abdelouahab AMIRA- for their appreciable help, their continuous encouragement and their uninterrupted moral support.

I would like to warmly thank my colleagues from CERIST and Edinburgh Napier University for having encouraged, motivated and above all supported and helped me before and during the preparation of my thesis.

And to be sure not to forget anyone, that all those, near or far, contributed by their advice, their encouragement or their friendship, to the result of this modest work, find here the expression of my deep recognition.

Contents

List of Figures

List of Tables

List of Abbreviations

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks

BI Beacon Interval

BMA Bad-mouthing attacks

BR Border Router

BSA Ballot-stuffing attack

CAP Contention Access Period

CFP Contention-Free Period

CoAP Constrained Application Protocol

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

DAG Directed Acyclic Graph

DAO Destination Advertisement Object

DAO-ACK Destination Advertisement Object Acknowledgement

DDoS Distributed Denial of Service

DIO DODAG Information Object

DIS DODAG Information Solicitation

DODAG Destination Oriented Directed Acyclic Graph

DoS Denial of Service

ERNT Extended RPL Node Trustworthiness

ETX Expected Transmission Count

EPC Electronic Product Code

FFD Full-Function Device

GTS Guaranteed Time Slots

ICMPv6 Internet Control Message Protocol for IPv6

ICT Information and Communication Technologies

IDS Intrusion Detection Systems

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IoT Internet of Things

IPv6 Internet Protocol version 6

LLNs Low-power and Lossy Networks

LR-WPANs Low-Rate Wireless Personal Area Networks

M2M Machine-to-Machine

MAC Media Access Control

MANET Mobile Ad Hoc Network

MRHOF Minimum Rank with Hysteresis Objective Function

MRTS Metric-based RPL Trustworthiness Scheme

NFC Near Field Communications

OF Objective Function

OF0 Objective Function Zero

OOA On-off attacks

OSA Opportunistic service attacks

PAN Personal Area Network

PCM Pan Coordinator Manager

PC Path Cost

PD Prisoner's Dilemma

QoS Quality of Service

RFD Reduced-Function Device

RFID Radio Frequency Identification

RNT RPL Node Trustworthiness

ROLL Routing Over Low-power and Lossy networks

RPL IPv6 Routing Protocol for Low-power and Lossy networks

RSN RFID sensor networks.

SD Superframe duration

SPA Self-promotion attacks

TDMA Time Domain Multiple Access

TOF Trust-based Objective Function

TPM Trust Platform Module

UDP User Datagram Protocol

WAN Wide Area Network

WPAN Wireless Personal Area Network

WSAN Wireless Sensor and Actuator Networks

WSN wireless sensor networks

Introduction

Background and Problem

The Internet of Things (IoT) represents a vision in which all physical objects (things) are interconnected and realise real-time interaction through the Internet, 3G, or WIFI networks. The IoT concept can likely be formed of various wireless technologies, such as Radio-Frequency Identification (RFID) tags, sensors, actuators, and mobile phones. In these technologies, computing and communication systems are seamlessly embedded [8]. For everyday human life, the IoT provides a rich set of advanced, intelligent and revolutionary applications and services such as healthcare, home automation, smart grid, automated transportation, environmental monitoring, and smart cities.

To allow effective communication in the IoT Low-Power and Lossy Networks (LLNs), while taking into consideration the heterogeneity of objects and applications, IoT systems have adopted the open standards of TCP/IP protocol suite as the networking solution. Nonetheless, because IoT-LLNs are different from common wired computer networks, the standards bodies IETF and IEEE introduced new stack and protocols to meet the requirements of such constrained IoT environment. Indeed, in IoT-LLNs, the perception layer is based on the IEEE 802.15.4 protocol, developed by the IEEE 802.15 Personal Area Network (PAN) Working Group that specifies the physical layer and media access control while focusing on low-cost and low-power transmissions between devices, making it a framework of choice for upper layers developed to address the IoT. Whereas, the network layer is based on the Routing Protocol for LLNs (RPL) that has been standardised by the IETF-ROLL Working Group for constrained environments such as the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), and is considered as the de facto routing protocol of the IoT [9]. Indeed, RPL takes into consideration limitations either in energy power or in computational capabilities of such networks.

However, the heterogeneity and dynamicity, as well as the scarcity of resources in IoT environments, make security a great challenge that needs to be addressed. Hence, the IoT applications are unlikely to fulfil a widespread diffusion until they provide strong security foundations, which will prevent the growth of malicious models, or at least mitigate their impact [10]. Like any network, cryptography and authentication mechanisms are used in IoT security. Providing strong authentication and cryptography mechanisms can help to mitigate several security issues for IoT. Cryptography and authentication techniques are used to exchange messages securely between nodes, and thus represent the first line of defence to counter external attacks. These mechanisms can detect and prevent external attacks, but they

cannot deal with internal attacks and adversarial nodes' problem within the network. Insider attackers can bypass these mechanisms by gaining access to shared keys and trigger several attacks against the IoT network. Therefore, a reliable IoT environment needs to be built based on effective trust management mechanisms for selecting trustworthy devices and objects to guarantee consistent data analysis, qualified services and enhanced information and network security [11].

Motivation

Trust management aims to build a secure relationship based only on trustworthiness between all participating nodes to design secure protocols while taking into consideration the IoT specificities. The trust management in the IoT networks is gaining importance as a key element to establish a secure environment. Nevertheless, although there are some initial proposals and implementations to manage security aspects in IoT networks [12], there is a lack of advanced trust models specially meant for this emerging paradigm, where the objects and communications characteristics raise new security risks. The trust management solutions deployed in specific networks such as WSN, MANET, and VANET are not yet adapted to cope with the IoT new scenarios, which require new trust relationships models [13] [14] [15]. We think that trust models for IoT have not yet addressed and studied enough in the state of the art, leading to different research opportunities, such as trust between the IoT layers, among devices (nodes, objects, things), and among applications and devices.

The fundamental problem that is being dealt with in this thesis is ***"How can we build trust in an IoT network?"***. In particular, both IEEE 802.15.4 and RPL face many security issues and are subject to several attacks, which can be handled by introducing new trust models. These trust models should enable the establishment of trusted and secure relationships among the participating objects during the network association process and to start and maintain a reliable routing. This trust management system will play an essential role in reliable data fusion and mining, qualified services with context-awareness, and enhanced user privacy and information security [16].

Contributions

In this thesis, we address the trust management issue in the context of IoT. Particularly, we propose trust management solutions on both Media Access Control (MAC) layer and the networking layer, precisely in RPL.

In the MAC layer, we propose a contribution that aims to address the MAC unfairness attacks against the IEEE 802.15.4 MAC layer, using a trust mechanism, while maintaining channel access to all participating nodes. This trust model focuses especially on the Guaranteed Time Slots (GTS) related attacks. In this approach, a Personal Area Network (PAN) Coordinator Manager (PCM) collaborates with PANs and Coordinators to identify malicious behaviour, calculate trust values for participating nodes, and exclude malicious nodes. To deal with the IEEE 802.15.4 MAC

GTS security, we propose two algorithms. The first allows a controlled association process, while the second permits GTS allocation based on nodes trustworthiness.

In the network layer, we propose three contributions. The first contribution introduces a new version of RPL, which enables routing by using trustworthiness between the different nodes. This model maps the nodes' behaviours to a trust metric and uses the obtained metric and the introduced trust-based objective function during RPL secure routing construction and maintenance. In the first trust model, named trust-based RPL, the nodes used the RPL Node Trustworthiness (RNT) metric and the Trust Objective Function (TOF) to select the parent having the greatest value of trustworthiness as a preferred parent. In the second contribution, we introduce the Metric-based RPL Trustworthiness Scheme (MRTS) to enhance the trust-based RPL model proposed previously. MRTS uses collaborative trustworthiness evaluation between the network's different nodes. It extended the RNT metric, namely ERNT, with new parameters and redefined TOF to deal with the trust inference problem by selecting the most trusted path from the source node to the Border Router (BR). We use an extended distributed Bellman-Ford algorithm to implement our scheme and the results show that the new scheme improves the security of RPL. In our third contribution, we improve MRTS by adding a new parameter in ERNT and redefining TOF for trust calculation. We evaluate MRTS performance by using the open-source Contiki-Cooja simulator. The simulations show good results regarding routing security, power consumption, packet delivery ratio, and throughput. Besides, we provide mathematical analyses of the MRTS routing and the ERNT metric. Finally, we present game-theory-based mathematical analyses and a simulation study of MRTS as a strategy for cooperation enforcement.

Thesis Organisation

This thesis is divided into two main parts. The first part is devoted to the study of the IoT context and the state of the art, while the second part consists of an in-depth description of our contributions.

The first part is composed of two chapters. Chapter 1 studies the IoT concept, the enabling technologies, and the IoT potential applications along with the most leading IoT challenges. Besides, it provides the different IoT architectures and a clear overview of the protocols accommodated in the current IoT standardised stack. Chapter 2 presents an overview of trust management in IoT and illustrates problems to trust management, such as interoperability and dynamicity, in addition to attacks against trust models. It overviews trust models classifications and introduces a new classification. Furthermore, it surveys existing trust management solutions in WSNs, IoTs and particularly for the RPL routing protocol. Finally, it elaborates a synthesis of trust-based solutions classification based on the IoT layers, trust evaluation models and trust-related attacks.

The second part of the thesis is organised into two chapters. Chapter 3 introduces our contribution to the Mac layer, which address the MAC unfairness attacks against the IEEE 802.15.4 MAC layer. Chapter 4 presents and validates our proposed contributions to the network layer.

Finally, we conclude the thesis with a summary of our contributions, the limitations observed and the possible perspectives of our research work. Besides, we cite the published scientific production, which represents the results of our research activities during this thesis.

Part I State of the Art

Chapter 1

The Internet of Things

This chapter presents a thorough background on IoT vision and characteristics, enabling technologies, architectures, applications, as well as their challenges.

1.1 IoT Definition and Vision

The Internet of Things (IoT) is a complex concept that makes it difficult to come up with a common definition. One simple vision of the IoT is that all physical objects (things) in a large dynamic distributed network system, can be identifiable (anything identifies itself), are interconnected and realise real-time interaction to provide or consume information by using intelligent interfaces. An object (or thing) can be defined as a real/physical or digital/virtual entity, which possesses a unique identifier, associated to at least one name and one address with some communication functionalities, such as the ability to be discovered and to accept incoming messages and reply to them [1] [17].

The IoT concept can likely be formed of various wireless technologies such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, and mobile phones. In these technologies, computing and communication systems are seamlessly embedded. In IoT, the smart objects communicate through wired or wireless technologies, such as the Internet, 3G, or WIFI networks [8].

Initially, IoT has been defined as the result of the convergence of the three main visions: things-oriented, internet-oriented, and semantic-oriented [1] [18], as shown in Figure 1.1.

- **Things oriented vision:** At the beginning, the definition of IoT derives from a "Things oriented", that consider only RFID tags as a thing, which identify any object using the Electronic Product Code (EPC) specification, such as the vision in [19]. Now, this vision is extended to use different wireless technologies, such as NFC, WSAN. These wireless technologies together with RFID are recognised as the components that will connect the real world with the digital world.
- **Internet-oriented vision:** In this vision, the IoT is seen as a global infrastructure which connects both virtual and physical generic objects and highlights the importance of including existing and evolving Internet and network

developments. In this context, the Internet Protocol (IP) is promoted as the network technology for connecting smart objects around the world. The IP address has to be simplified, and therefore any object could be addressable and reachable from any location. Examples of these visions are [20] [21].

- **Semantic oriented vision:** The considerable number of objects in the network generates massive quantities of data, which will present a big problem of representation, storage, interconnection, and organisation of information. Semantically oriented vision could exploit suitable modelling solutions to process meaningfully the sets of data generated by IoT, as described in [22].

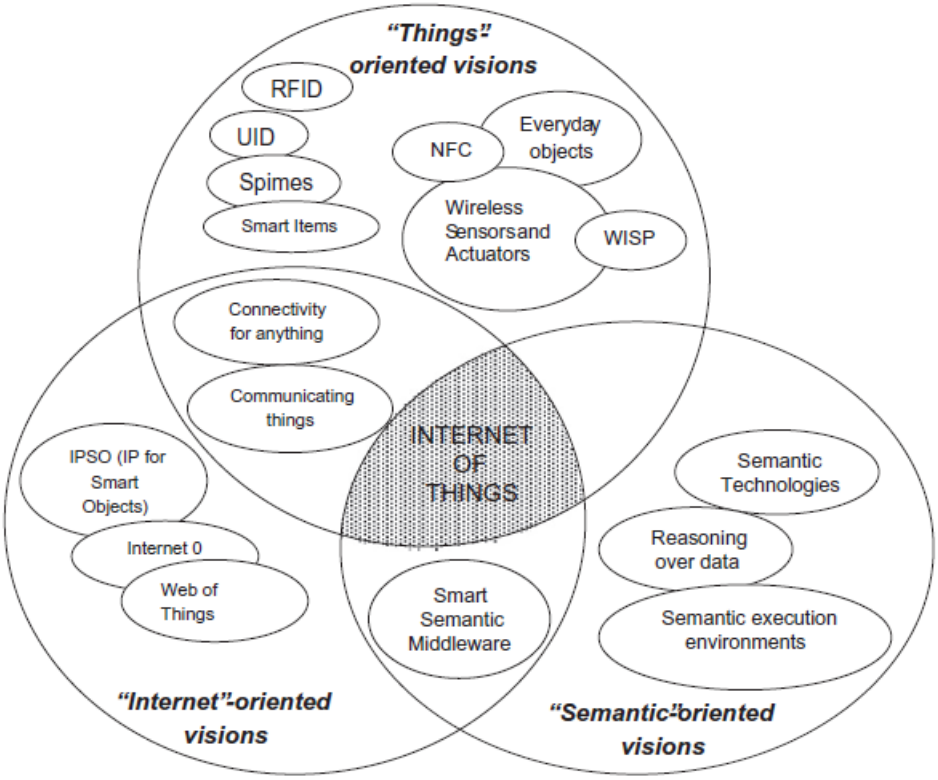


Figure 1.1: Convergence of three main visions of the Internet of Things [1].

It is difficult to understand what IoT really means, what constitutes its basic concepts, and what are the social, economical and technical implications that hinder in the deployment of IoT. Figure 1.2 illustrates the European research cluster of IoT (IERC) definitions.

1.2 IoT Characteristics

As highlighted in the section above, there is no common definition of the IoT. It is highly related to the vision of each academic or business entity. To get the IoT concept closer to reality, the following characteristics need to be addressed [17]:

- **Devices heterogeneity:** IoT is characterised by a large number of heterogeneous devices, which should have very different capacities from computation and communication. Thus, systems and protocols should be designed to support a high level of heterogeneity.

Figure 1.2: The Internet of Things Definitions.

- **Scalability:** Nowadays, the number of objects is increasing considerably. It is more than a billion objects. Therefore, architecture and systems must be able to support and manage scalability for identification and addressing, communication and networking, data and information management, and security management.

- **Resource Limitations:** IoT's objects have limited power, processing and storage capacities. Thus, systems and protocols need to be designed in a way to optimize and minimize the objects' energy, storage and computation usage as much as possible.

- **Self-organisation and self-healing capabilities:** IoT progresses in a smart distributed environment. Thus, systems and protocols should be designed to allow smart objects to autonomously respond to a wide range of different situations in a distributed manner.

- **Semantic interoperability and data management:** IoT will mainly consist of exchanging and analysing huge amounts of data. Therefore, systems and protocols should be designed to allow heterogeneous devices to transfer data and operate in an interoperable way.

- **Mobility:** Most of the smart devices and IoT actors are mobile. This characteristic causes several changing to the network conditions which makes it difficult to communicate with each other. Furthermore, not handling mobility can generate more security breach.

- **Security and Privacy:** IoT affects every aspect of human and business lives. In addition, IoT's devices generate a huge amount of data. As a consequence, IoT entities should be equipped with strong security and privacy policies. This includes securing the devices themselves (i.e. hardware), exchanged data and information, communications and networks, and endpoints.

1.3 Enabling Technologies

The ability of the IoT concept to interface with the physical realm is realised through the integration of several enabling technologies. The leading technologies are as follows.

1.3.1 Identification, sensing and communication technologies

Nowadays, communication technologies go forward rapidly. In this context, wireless technologies have played a key role in their development, to the point that almost every human uses at least one wireless technology. In the IoT, wireless technologies play a crucial role in data collection (sensing) and data communication. They offer the possibility of integrating several applications in different application areas such as e-health, intelligent transport systems, etc. In IoT communication technologies, heterogeneous objects could connect to provide specific smart services. However, the fact that objects are limited in term of size, energy consumption, and computation, the IoT nodes must operate with low power in the presence of lossy and noisy communication links. The two leading wireless communication technologies are Wireless Sensor Networks (WSN) and radio-frequency identification (RFID). In the following, we provide a comprehensive presentation of IoT enabling technologies that are: RFID, Sensors and Networks, Standards, Augmented intelligence, and Augmented behaviour.

1.3.1.1 RFID Technology

RFID technology is considered as an essential development in the embedded devices field, which is used to identify objects. RFID is the first technology used to realise the M2M concept [2]. An RFID system can be composed into two main components: several RFID Tags (transponders) and at least one RFID Readers (transceivers). The tag has a tiny microchip with memory to record information added to an object, persons or animals, and an embedded antenna. A unique identifier characterises the tag. The recorded information can automatically be used to provide the object's identity. The antenna is used to permit the communication of the chip with the reader by using radio waves. So, the tag can be seen as an electronic bar-code. The RFID reader conveys a request signal to the tag, the tag sends the tag's number to the reader by a reflected signal, as shown in Figure 1.3. The reader then transmits that number to the database to be identified. As a result, RFID systems could monitor objects in real-time, without the need of human; which permit the connection of the real world with the digital world. RFID systems can be used in a wide range of application, such as logistics, e-health, security, etc. [1] [2].

From a hardware point of view, "an RFID is a tiny microchip with an embedded antenna, which is used for both receiving the reader signal and transmitting the tag identity. The tag is manufactured in a package that can be used as an adhesive sticker" [1]. Example, Hitachi has developed a tag with dimensions 0.4 mm x 0.4 mm x 0.15 mm [1]. See Figure 1.4.

RFID tags can be classified into two types: passive and active. The passive tags do not have a battery. They use the power required to communicate their data from the request signal transmitted by a RFID reader. Nevertheless, the active tags have their own battery. They use their own power to read from a distance and to send information to the reader.

Figure 1.3: RFID system [2].

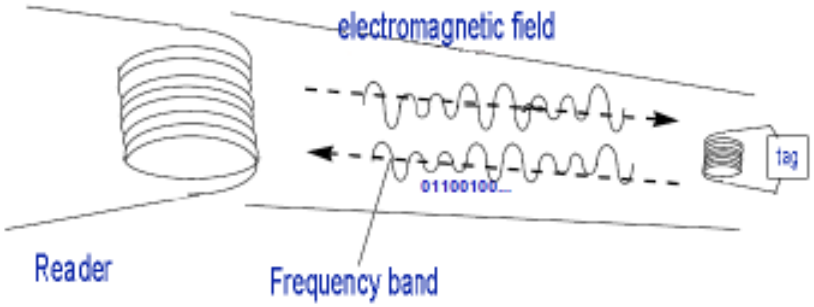


Figure 1.4: RFID tag and reader.

1.3.1.2 Sensors and Networks

Sensors are small sensing self-powered devices, which can collect information, process data, detect special events, communicate in a wireless multi-hop fashion through a wireless network, and thus, report the processed data of their sensing to a small number of particular nodes called sinks or base station, as shown in Figure 1.5. The technological complement to a sensor is an actuator, a device that converts an electrical signal into an action. By combining multiple sensors, each serving different purposes, it is possible to build complex value loops that exploit many different types of information. For example, sensors can collaborate with RFID systems to better track the status of things, such as their location, temperature, movements, etc. They can augment the awareness of a particular environment. Therefore, they act as a further bridge between the physical and digital world. Three primary factors are driving the deployment of sensor technology: price, capability, and size. As sensors get less expensive, smarter, and smaller, they can generate a wider range of data at a lower cost and can be used in a wider range of applications, such as environmental monitoring, e-health, intelligent transportation systems, military, surveillance, micro-surgery, and agriculture.

Sensors must communicate the sensed data to other locations for aggregation

and analysis. This typically involves transmitting data over a network. Sensors and IoT's devices (smartphones, laptops, tablets, etc.) are connected to networks using various networking devices such as hubs, gateways, routers, network bridges, and switches, depending on the application. For instance, when data have to be transferred over short distances (eg., inside a room), devices can use wireless personal area network (PAN) technologies such as Bluetooth and ZigBee in addition to wired connections through technologies such as Universal Serial Bus (USB). On another hand, when data have to be transferred over a relatively bigger area (e.g., an office), devices could use local area network (LAN) technologies such as Ethernet and fibre optics, or Wireless LAN networks technologies such as Wi-Fi. When data are to be transferred over a wider area beyond buildings and cities, a wide area network (WAN) such as Internet has to be used. Indeed, technologies such as 4G (LTE, LTE-A) and 5G are favourable for IoT applications, given their high data transfer rates. Technologies such as Bluetooth Low Energy and Low Power Wi-Fi are well suited for energy-constrained devices.

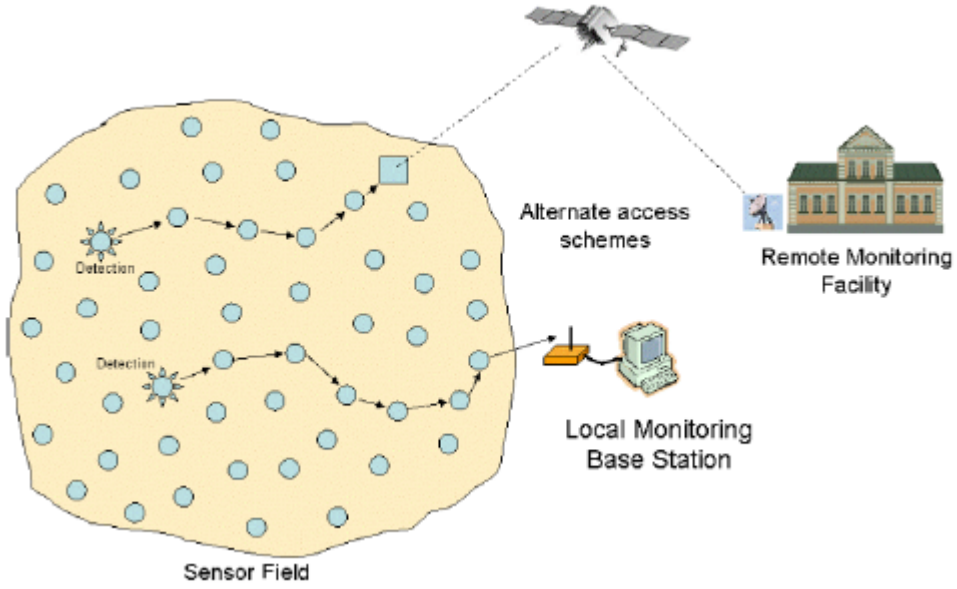


Figure 1.5: Wireless Sensor Network.

The integration of sensing technologies into passive RFID tags will enable a lot of new applications into the context of IoT [23]. Sensing RFID systems permit the building of RFID sensor networks "RSN", which consist of small RFID-based sensing and computing devices. RFID readers will be the sinks of data generated by sensing RFID tags. Furthermore, they will offer power for different network operations. As a result, the RFID sensor network can make the possibility of supporting sensing, computing, and communication capabilities in a passive system [24].

1.3.2 Middleware

The middleware is a software layer (or a set of sub-layers) or service programming interposed between the physical layer (i.e., hardware) and the application levels. It provides the required abstraction to hide the complexity of the different technologies involved in the lower layers. The middleware aims to develop specific applications enabled by the IoT infrastructures that can exempt developers from concerns that

are not directly related to the designed applications. And thus, it allows developers to save time and energies on issues concerning the management and the utilisation of the IoT infrastructures. By using middleware, devices and applications with different interfaces can exchange information and share resources with each other. The middleware becomes more critical in these recent years due to its role in the simplification of the development of new services and the integration of legacy technologies into new ones.

Researches on middleware for IoT can be divided into five categories [25]: 1) message-oriented middleware; 2) semantic Web-based middleware; 3) location-based service and surveillance middleware; 4) communication middleware; and 5) pervasive middleware.

- **The message-oriented middleware** provides reliable information exchange among various platforms, and communication protocols (e.g., AMQP, DDS, MQTT, and XMPP) [2] [26].
- **The semantic Web-based middleware** provides the interactions and interoperability among various sensor networks, such as the SoA-based middleware [27] and the task computing-based middleware [28].
- **The location-based service and surveillance middleware** integrates the locations of devices and other information to provide integrated value services [29].
- **The communication middleware** may provide reliable communications among heterogeneous devices and applications, such as the RFID-based middleware (Fosstrak [2], etc.), the sensor network-based middleware (TinyREST [30], etc.), and the supervisory control and data acquisition.
- **The pervasive middleware** provides services on multiple and heterogeneous platforms for computing environments [31].

Other researchers grouped middleware solutions based on their design approaches on seven groups [survey]: 1) event-based; 2) service-oriented; 3) VM-based; 4) agent-based; 5) tuple-spaces; 6) database-oriented; 7) application-specific.

1.3.3 Standards

Data collected by sensors are aggregated so that meaningful conclusions can be drawn. Aggregation is achieved through the use of various standards depending on the IoT application. Two types of standards relevant for the aggregation process are technology standards (such as network protocols, communication protocols, and data-aggregation standards) and regulatory standards (such as those related to security and privacy of data). Examples of networks and communications technology standards are the IEEE 802.15.4, the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), the Routing Protocol for Low-Power and Lossy Networks (RPL), and the Constrained Application Protocol (COAP) as will be described in Section V. ZigBee, Z-wave, Data distribution service (DDS), Advanced message queuing protocol (AMQP), and Extensible messaging and presence protocol (XMPP) are also networks, communication, or messaging standards. Besides,

multicast DNS (mDNS) can support the name resolution in IoT applications.

Extraction, Transformation, Loading (ETL) tools and Big data ETL tools such as Hadoop/MapReduce are examples of data-aggregation standards. For the regulatory standards, there is a need for clear regulations related to the collection, handling, ownership, use, and sale of the collected data. For example, the US Health Insurance Portability and Accountability Act (HIPAA) controls the protection of medical information collected by doctors, hospitals, and insurance companies. However, HIPAA does not control information collected through personal wearable devices.

1.3.4 Augmented intelligence

Data analysis refers to the ability to extract insight smartly from collected data by different machines to provide the required IoT's services. An analysis is driven by cognitive technologies and intelligent models that facilitate the use of these cognitive technologies to discover and use resources, and model information, aiming at making sense of the right decision to provide the exact service. Augmented intelligence technologies include descriptive analytics tools such as Tableau and SAS Visual Analytics, predictive analytics tools such as machine learning models, and prescriptive analytics tools that include optimisation techniques that are based on large data sets, business rules (information on constraints), and mathematical models. Computer vision, Natural-language processing, and Speech recognition cognitive technologies can be used for both predictive and prescriptive analytics.

1.4 IoT Applications and impact areas

The technology advances, the diverse needs of potential users, and the potentialities provided by the IoT make possible the deployment of a considerable number of applications. There are many areas and environments in which new applications would likely enhance the kind of our personal lives, corporations, and communities. These environments are now equipped with smart objects. Allowing these objects in our environment to communicate with each other and to process the information collected will deploy a vast range for unforeseen applications, as shown in Figure 1.6.

In the following, we present the most relevant applications domain:

1.4.1 Smart Transportation and Smart Logistics domain

Several means of transportation, such as cars, trains, buses, etc., are currently equipped with sensors, tags, actuators, and processing power. The roads, rails, and public spaces are also equipped with RFID tags and sensors to collect useful and vital information for good management of traffic. The use of the transportation domain aims to optimise the utilisation of physical city infrastructures such as network, power grid, etc. and enhance the citizens' quality of life. Cars and roads can exchange important information through an appropriate service. These pieces of information will be sent to the drivers and private transportations to help them to find a better route to avoid accidents and jam, which will save time and energy

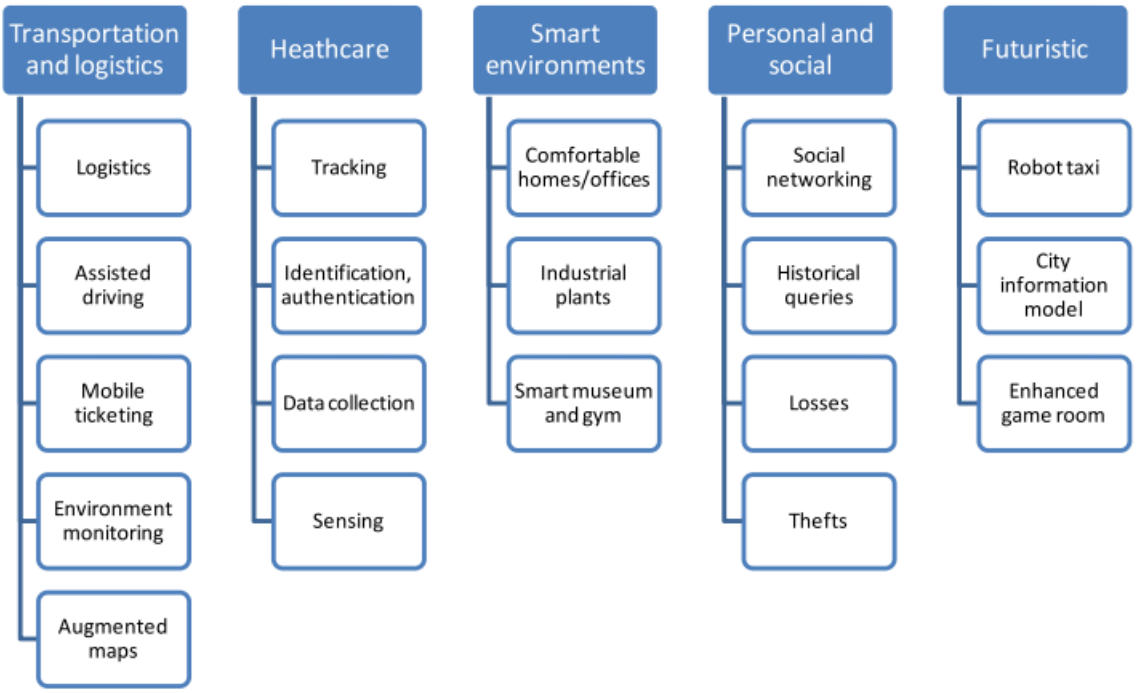


Figure 1.6: IoT Applications domains [1].

and route optimisation. As a result, the optimal route will save energy, time, money, and reduce the number of kilometres driven. Thus, contribute to reduced emission and less pollution. Furthermore, Several applications about transportation services can be equipped with an NFC tag. Therefore, the user can get information about several benefits from the web by using his mobile phones, such as touristic maps which allow him to save a lot of time and money. For instance, Saarika et al. [32] proposed an IoT-based smart parking system along with an intelligent signboard. Likewise, Sherly and Somasundareswari [33] proposed an IoT-based real-time traffic monitoring system that uses real-time traffic information for identifying and tracking vehicles, parking and roads.

There are many scenarios in smart logistics including logistics transportation, warehousing, loading/unloading, carrying, packaging, distribution processing, distribution, and information processing. For instance, logistics transportation is the most important economic activity among the components of logistics systems. It consists of shipping items from one place to another place using facilities and tools [34]. On another hand, logistics warehousing is a significant component in the logistics supply chain. It consists of activities that control, classify, and manage the inventory [35]. Indeed, authors in [36] illustrated potential implications of industry 4.0 on different logistics scenarios.

1.4.2 HealthCare domain

IoT applications play an essential role in the healthcare domain, which are used to assist patients in monitoring and tracking purposes. Patients carry or wear medical sensors to track their vital health parameters, such as blood pressure and body temperature. These sensors will collect and analyse data; if any unusual activity,

the sensor raises the alarm and transmit it to remote medical centres to perform the status of the patient, and then the staff medical will take the right actions for the patient. These pieces of information could be beneficial to medical assistance in the case of an emergency, the nearby hospital will be alerted. The intervention will be so fast, which means the life of the patient is saved and the hospitalisation cost is reduced [18]. Authors in [37] provide an analysis of the E-Health IoT domain from a different point of view. They highlight the growing importance of IoT technologies in the medical environment.

1.4.3 Smart Environment (home, office, plant) Domain

A smart environment makes the use of our environment easier and makes our life better and comfortable at home, in office, and industrial plants. A smart environment uses the intelligence of Smart embedded sensor technology to monitor critical parameters of the domain. Sensors and actuators are used to manage the equipment of home and office, which make life more relaxed and comfortable in several aspects. For example, management of energy via the control of home equipment such as washing machine, and air conditioners. Monitoring room lighting according to the time of the day. They are saving energy by switching off the electrical equipment when not needed [1]. For example, authors in [38] proposed an IoT-based smart system that: 1) provides faster detection of accidents by providing sensed information to the nearby hospitals and police station; 2) reduces the overall wastage of electricity by controlling street lights; 3) reduces the pollution of water bodies by sensing the ingredients of the water; 4) monitors the weather conditions through a smart mobile app; 5) detects theft by using video surveillance. Additionally, authors in [39] introduced an IoT-based smart air pollution monitoring system that triggers alarms to warn the surrounding people if the value of the measured pollutants exceeds a threshold. Besides, in [40], authors proposed an integrated smart environment based on IoT, where several sectors such as agriculture, security and emergency, banking, surveillance, meteorology, health care, education, and e-government, domestic appliances monitoring, traffic surveillance are integrated and the various objects and devices are connected using sensor networks and RFID technology. Furthermore, in industrial plants, RFID tags deploy all production parts. Smart environments help the monitoring of production in industrial plants automatically, which helps to save time, money, and energy to the production service [1].

1.4.4 Agriculture Domain

The agriculture domain is considered as a complex IoT paradigm, which uses IoT solutions and technologies to enable efficient resource management. It can help monitor the development of plants. These plants are equipped with RFID tags and sensors to be controlled and managed in case of a drastic or unexpected change in the evolution of plants due to temperature/humidity, by sending these anomalies to the reader to be shared across the Internet. The farmer or scientist can access this information from a remote place and take necessary actions [41]. [42], [43] and [44] are examples of works in the field of smart agriculture.

1.4.5 Personal and social domain

Several IoT applications are used in the personal and social domain. These applications allow the building of social relationships between persons in a different way. These applications are equipped with RFID and NFC devices. From these applications, we can cite social networking applications. These applications use social networking websites, such as Twitter, Facebook. It automatically updates all information that concerns the social activities of persons. In this concept, various objects can periodically tweet the readings, which can be quickly followed by friends from anywhere in real-time [1]. Many researchers explored the personal and social domain as can be seen in [45],[46], [47], and [48]

1.5 IoT Architecture

The most known and accepted model for IoT is the three-layer architecture composed of the Application, Network, and Perception Layers. Nevertheless, as presented in Section 1.3.3, there exist a pool of architecture models for IoT that added more abstraction to the basic IoT architecture. In follows, we present the three-layer and five-layer architectures as well as the SOA-based architecture (See Figure 1.7).

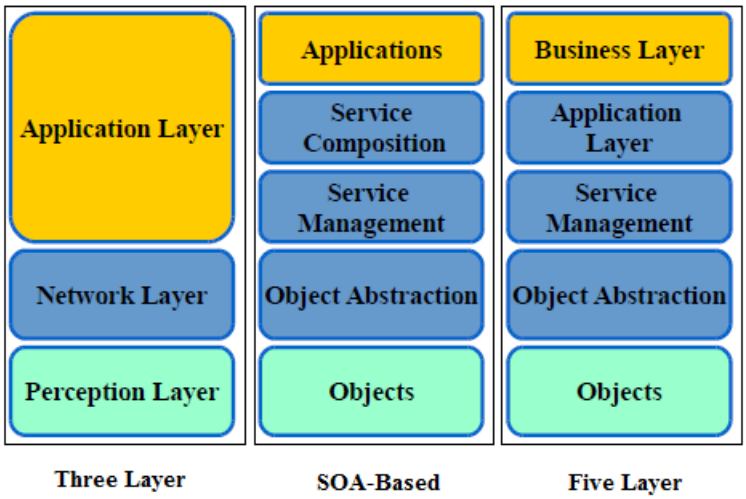


Figure 1.7: The Internet of Things Definitions.

1.5.1 Three-Layer Architecture

The three-layer architecture was introduced in the early stages of research in the area of IoT [49]. It has the three following layers:

- **Perception layer:** is also known as the objects, sensing or devices layer. It represents the physical layer where heterogeneous, resource-constrained and highly distributed IoT devices co-exist for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.
- **Network layer:** is also known as the wireless sensors network. It represents an intermediate layer which is used to connect to other smart things, network

devices, and servers, aggregate, process, and transmit sensed data from the perception layer to the application layer using wired and/or wireless communication technologies like WiFi, ZigBee, or LTE and GPRS.

- **Application layer:** is responsible for delivering application-specific services to the end-user. It provides services to manage, analyse and visualise measurements and outputs using users specific interfaces. It defines various applications in which the IoT can be deployed, such as, smart health, smart cities, and smart environment.

1.5.2 Five-Layer Architecture

The five layers in this architecture are objects, objects abstraction, services management, application, and business (see Figure 1.7) [2]. The role of the objects and application layers is the same as in the three-layer architecture. In follows, we present the function of the remaining layers.

- **Object Abstraction layer:** is known as the transport layer. It transfers the produced data from the objects layer to the service management layer and vice versa through, through various technologies such as RFID, 3G, GSM, UMTS, WiFi, Bluetooth Low Energy, infrared, ZigBee, etc. Furthermore, other functions like cloud computing and data management processes are handled at this layer.

- **Service Management layer:** is also known as processing or Middleware layer. It pairs a service with its requester based on addresses and names. It enables the IoT application programmers to work with heterogeneous objects without consideration to a specific hardware platform. and deliver the required services over the network wire protocols.

- **Business layer:** is also known as the management layer. It manages the whole complete IoT system, including activities and services. It uses received data from the application layer to build business models, graphs, flowcharts, application, etc. Besides, this layer employs many technologies such as databases, cloud computing, and big data modules to support decision-making processes. Furthermore, it achieves the monitoring and management of the underlying four layers.

1.5.3 SOA-Based Architecture

The SOA-based architecture is a technique that conceives and deploys applications and structure projects according to an approach based on the principle of "services" [50]. A service-oriented architecture (SOA) is a set of communicating services based on common interfaces and standard protocols [51]. It can be used to decompose complex and monolithic systems into applications consisting of an ecosystem of more straightforward and well-defined. Thus, it adapts the applications to specific users' needs. Like the five-layer architecture, the SOA-based architecture is composed of five layers, as illustrated in Figure 1.7.

- **Object layer:** is the same as in the three-layer and five-layer architectures

- **Object Abstraction layer:** is it he same as in the five-layer architecture. The aim of this layer is matching the access of the different devices by using common language and procedure to the heterogeneity of the objects.

- **Service Management layer (Middleware):** provides the main functions, which are predicted to be offered for each object for their control in the IoT scenario. A basic set of services includes object dynamic discovery, status monitoring, and service configuration.

- **Service Composition layer:** is a common layer on top of an SOA-based middleware architecture. It offers the functionalities for the composition of single services provided by networked objects to build specific applications. The only notion used in this layer is services; there is no notion of devices.

- **Applications layer:** is on the top of the architecture. It exploits all the features of the middleware layer to export all the functionalities of the system to the final user (customer).

1.6 Communication Protocol Stack for Constrained IoT Systems

To allow the effective communication in IoT while taking into consideration the heterogeneity of objects and applications, IoT systems have adopted the open standards of TCP/IP protocol suite, originally developed for the wired global Internet, as the networking solution. Nonetheless, IoT networks differ from common wired computer networks. Hence, the standards bodies IETF and IEEE introduced new stack to meet the requirements of such constrained IoT environment. Figure 1.8 illustrates both the TCP/IP stack and the IoT protocol stack.

In the following, we will define the essential layers and protocols of the standardisation of IoT stack.

1.6.1 IEEE 802.15.4

IEEE 802.15.4 is a standard developed by the IEEE 802.15 Personal Area Network (PAN) Working Group. It defines low-power wireless embedded radio communications at 2.4 GHz, 915 MHz and 868 MHz. It is designed for the physical layer (PHY) and the Media Access Control (MAC) for low-rate wireless personal area networks (LR-WPANs) [52] [25]. IEEE 802.15.4 protocol aims to ensure a reliable data transfer, short-range operation, high message throughput, low cost and reasonable battery. Due to the characteristics of the IEEE 802.15.4, many wireless communication technologies and protocols, such as Zigbee [53], WirelessHART [54] [55] are based on the IEEE 802.15.4.

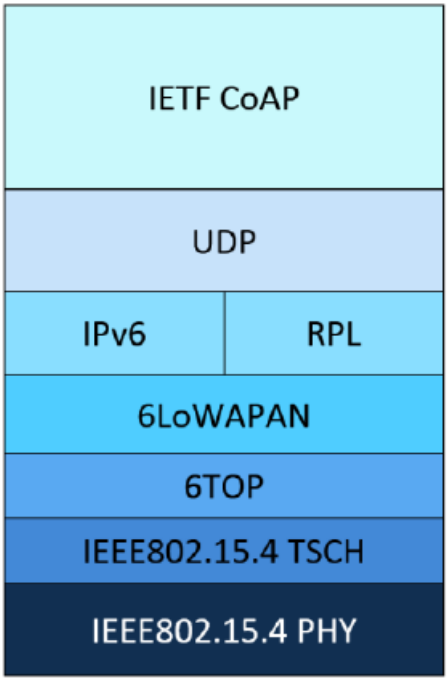


Figure 1.8: The IoT Stack [3].

1.6.2 IPv6 over Low -Power Wireless Personal Area Networks "6LoWPAN"

6LoWPAN is a compressed version of IPv6 that has been developed and standardised by the 6LoWPAN IETF Working group. It was designed to combine between IPv6 protocol (network layer) and IEEE 802.15.4 protocol (Mac layer). In fact, the IEEE 802.15.4 standard has been used as a baseline for 6LoWPAN communication development, and thus, the 6LoWPAN enables the transmission of IPv6 packets over IEEE 802.15.4 networks [25]. Due to the limited bandwidth (127 bytes) provided by the IEEE 802.15.4 protocol, the 6LoWPAN provides IPV6 header compression mechanisms of IPv6 datagrams to reduce the transmission overhead, fragmentation to meet the IPv6 Maximum Transmission Unit (MTU) requirement, and forwarding to link-layer to support multi-hop delivery [56]. 6LoWPAN removes many IPv6 overheads so that a small IPv6 datagram can be sent over a single IEEE 802.15.4 hop in the best case.

1.6.3 Constrained Application Protocol "COAP"

CoAP is an application layer protocol for IoT applications developed by the IETF Constrained RESTful Environments (CoRE) working group. It defines a web transfer protocol based on REST "REpresentational State Transfer" architecture on top of HTTP functionalities, which considers the various objects in the network as resources [57] [2] [58].Since most of the devices in IoT are resources constrained, HTTP cannot be used in IoT, because of its complexity. For that, CoAP was proposed to adapt some HTTP functions to meet the requirements for IoT. CoAP aims to enable resources constrained devices to achieve RESTful interactions [25], it aims to act as HTTP but compactly within the constrained networks. CoAP works on top of the unreliable UDP transport layer to provide a good interface for the standard Internet

services. When CoAP is used with 6LoWPAN as defined in RFC4944, messages fit into a single IEEE 802.15.4 frame to minimize fragmentation. Since CoAP is used in the IoT as an application protocol, end-to-end security between two applications can be provided with the Datagram Transport Layer Security (DTLS).

1.6.4 The Routing Protocol for Low-Power and Lossy Networks

The Routing Protocol for Low-Power and Lossy Networks (RPL) [59] has been designed and standardised for LLN networks, such as 6LoWPAN networks, and is recognised as the routing protocol of the Internet of Things (IoT). The existing routing protocols for LLNs can be categorised as reactive or proactive. In the reactive protocol, the route from the source to the destination is created on-demand. In contrast, in the proactive protocol, each node within the topology preserves a routing table that evaluates the routes [60] [61]. RPL is a proactive distance-vector routing protocol that constructs a logical representation of the network topology as a Directed Acyclic Graph (DAG). The DAG is composed of one or more Destination-Oriented DAGs (DODAGs) with one root per DODAG. As depicted in Figure 1.9, in each DODAG, nodes are connected to the Border Router (BR) - edge router/gateway. Each node in a DODAG has some parameters such as an IPv6 address, a parent (s) list, discovered neighbours' list, Rank, etc. A backbone link connects the BR to the Internet and other BRs.

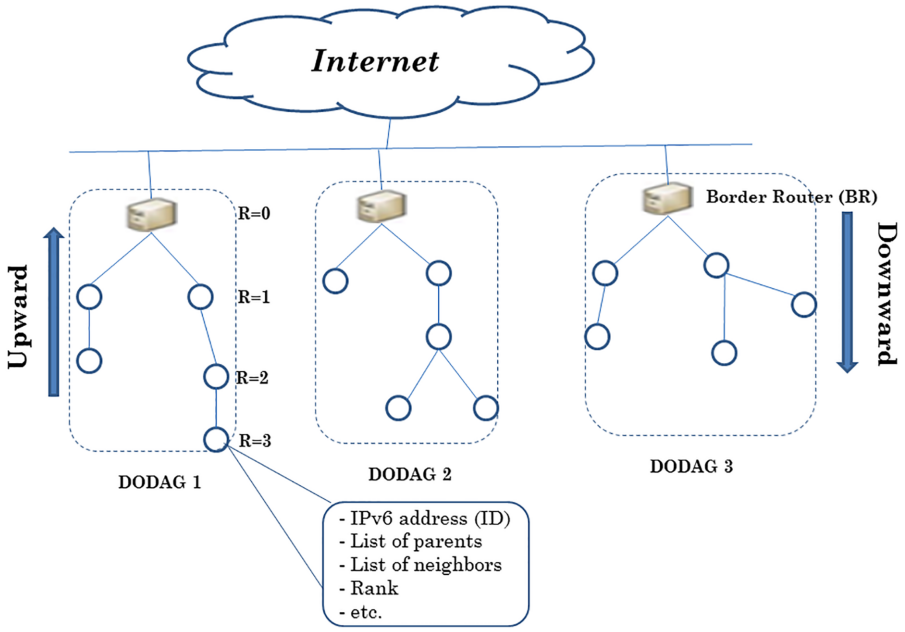


Figure 1.9: The RPL Topology.

1.6.4.1 RPL Control Packets

RPL supports point-to-point (between nodes), multipoint-to-point (from nodes to the BR), and point-to-multipoint (from the BR to nodes) traffics. RPL uses specific ICMPv6 (Internet Control Message Protocol for IPv6) messages dissemination

and a Trickle mechanism to construct and maintain the network topology. These messages are DODAG Information Object (DIO), DODAG Information Solicitation (DIS), and DODAG Destination Advertisement Object (DAO) control messages. The type of each message can be identified using the code field within the message format, as shown in Figure 1.10

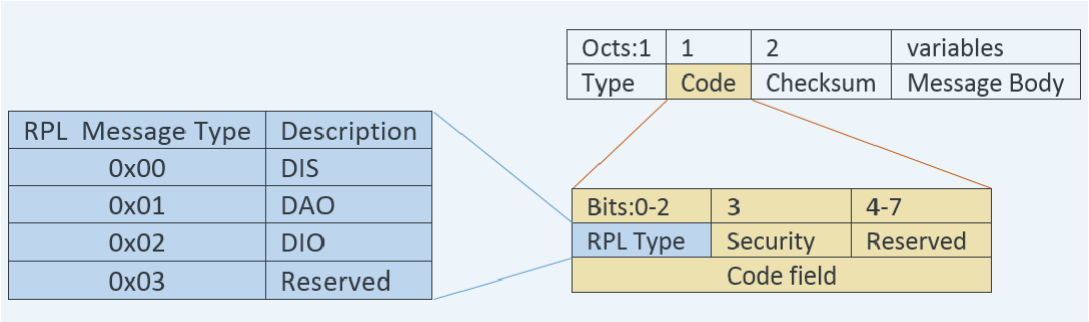


Figure 1.10: RPL ICMP messages.

- **DODAG Information Object (DIO):** The DIO carries information, which permits a node to find an RPL Instance, learn its configuration parameters, select a DODAG parent set, construct and maintain the DODAG. DIO message can contain options such as node/link metrics and constraints (i.e., node energy, hop count, throughput, latency, link colour, and ETX –Expected Transmission Count) [62]. Besides, DIO messages carry the Objective Function (OF) that nodes use to optimise path construction [63].

- **DODAG Destination Advertisement Object (DAO):** The DAO message is used to propagate destination information upwards along the DODAG. After joining the DODAG, the node advertises a DAO message to its neighbours to update their routing table. In the storing mode, the child node unicasts the DAO message to its preferred parent. Whereas, it unicasts the DAO message directly to the DODAG root in the non-storing. Like DIO message, DAO message can contain options such as node/link metrics and constraints [62].

- **DODAG Information Solicitation (DIS):** A new RPL node wishing to join a DODAG sends a unicast or multicast DIS message to request a DODAG Information Object from an RPL node.

- **DODAG Destination Advertisement Object Acknowledgement (DAO-Ack):** DAO-Ack is an optional control message that can be sent in the response to receiving a DAO message. The receiver of a DAO sends buck a unicast DAO-Ack to the DAO sender to acknowledge its willingness to act as a next-hop node towards the DODAG root.

1.6.4.2 RANK and DODAG Construction and Maintenance

To construct and maintain the DODAG topology, the Border router (BR) broadcasts an initial DIO message to all neighbour nodes (see Figure 1.12.a). This DIO

message contains the RPLInstanceID, the DODAG ID, the DODAG Version Number, Rank of the BR, the OF, Trickle timer, the metrics/constraints and, other parameters, as shown in Figure 1.11. The RPLInstanceID is used to identify the number of DODAGs (i.e., one or more), the DODAGID represents the RPL Instance, the DODAG Version Number indicates the number of times the DODAG has been reconstructed as part of a technique of free loop, and the Rank represents the individual position of a node with respect to the BR and other nodes.

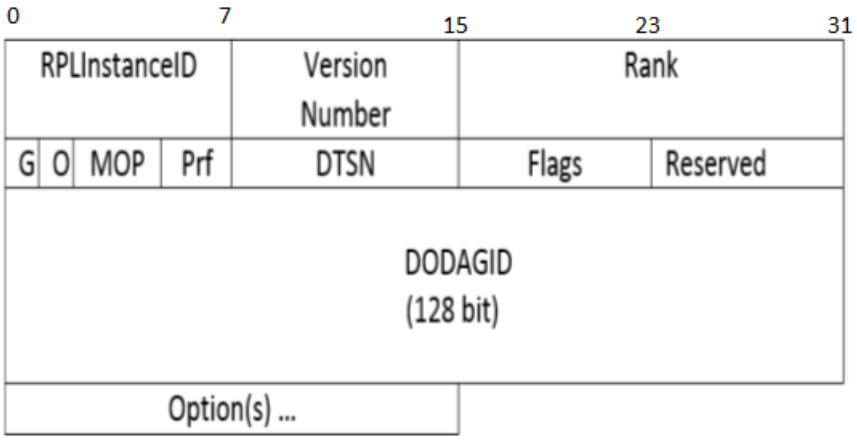


Figure 1.11: DIO Format.

To optimise DAG paths construction and calculate node's Rank, RPL uses a set of node/link routing metrics and constraints (i.e., node energy, hop count, throughput, latency, link colour, and ETX -Expected Transmission Count) [62]. Besides, RPL uses an Objective Function (OF) that defines how routing metrics and constraints are used to compute node's Rank with a monotonical increase fashion in a downward direction, wherein the DODAG root has the least Rank to guarantee loop-free topology [63]. Indeed, routing metrics/constraints, OF, Rank and other information are conveyed within the DODAG Information Object (DIO) messages.

When a node n receives DIO messages from its neighbours, it uses the information conveyed in these DIO messages to join a DODAG. The node selects a set of parents allowing it to reach the BR. Then, the node computes its Rank using Equation (1.1) [64] associated with the selected OF. Besides, it chooses a preferred parent, which ensures traffic routing to the BR. The node chooses the parent, which have the lowest Rank. Afterwards, the node generates and broadcasts its new DIO message with all updated information (i.e., Rank) to its neighbours, as shown in Figure 1.12.b. All neighbouring nodes will repeat the process until each one joins the DODAG. Figure 1.12 shows the DIO messages sequence into DODAG construction.

Once the construction is completed, the maintenance begins respecting a Trickle timer mechanism [59]. This timer regulates the transmission rate of DIO messages. Thus, in the steady-case, the trickle timer interval increases, and the transmission rate will be slowed. Otherwise, In case of inconsistencies (e.g., altered DIO messages, etc.) that involve changes in the topology, the nodes reset the Trickle timer

to a lower value, and hence, control messages transmission rate will be fastened. The nodes use Global Repair (GR) and Local Repair (LR) mechanisms to fix links and nodes failures, and other inconsistencies. Once GR or LG triggered, the nodes reset their trickle timers and update their respective parents' lists and Ranks.

$$\begin{cases} \text{Rank(N)} = \text{Rank(P)} + \text{Rank\_increase} \\ \text{Rank\_increase} = \text{step} * \text{MinHopRankIncrease} \end{cases} \tag{1.1}$$

The Rank values should increase monotonically from the BR towards the leaf nodes and decrease monotonically from the leaf nodes towards the BR. Furthermore, packets should be transmitted either upward towards the BR, or downward towards leaf nodes, respecting the Rank rule defined in [59]. Hence, when a node receives a packet upward, the sender must have a Rank higher than that node and vice versa, when a node receives a packet downward, the sender must have a Rank lower than that node. As well, it should be bounded by MinHopRankIncrease and MaxHopRankIncrease [59]. Therefore, to comply with the Rank monotonic property, the BR sets its Rank to MinHopRankIncrease. Then, each node N calculates its Rank Rank(N) as the sum of the Rank of its preferred parent (Rank(P)) and Rank_increase as in Equation 1.1.

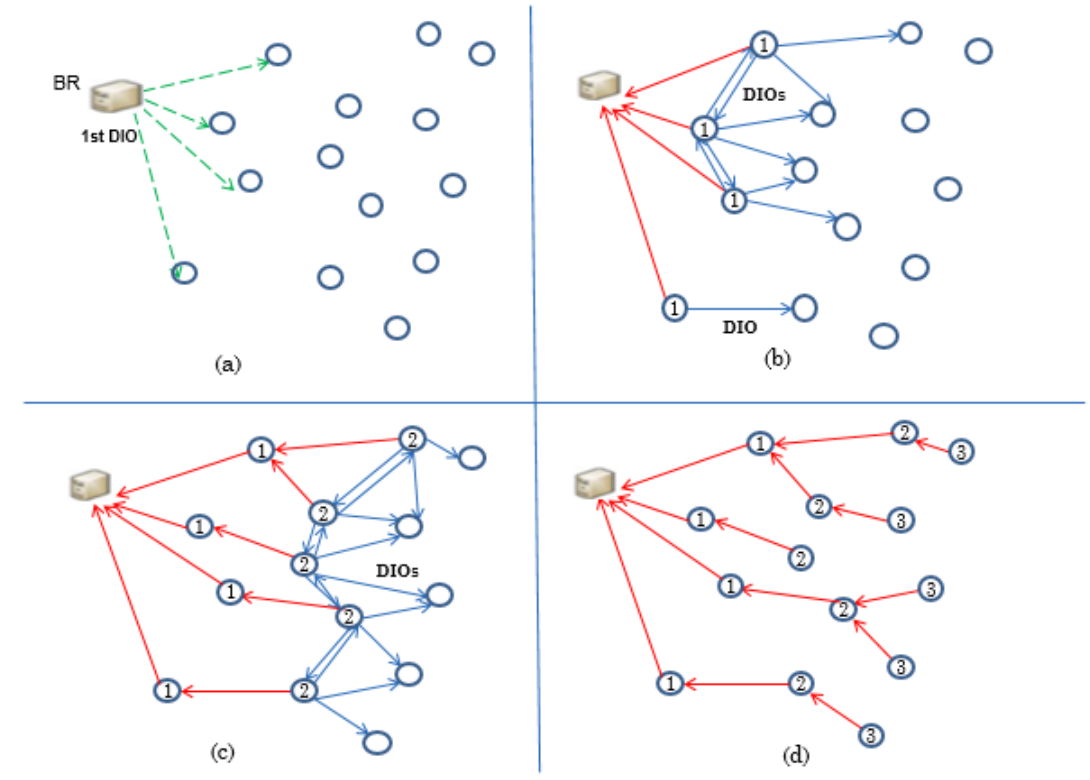


Figure 1.12: The sequence of DIOs in DoDAG RPL Construction.

1.6.4.3 Objective Functions

The objective function helps an RPL node to join the appropriate DODAG with its suitable version. The OF allows calculating the Rank based on some routing metrics and constraints such as hop-count, delay, energy, and so forth. Indeed, it aims to provide the RPL nodes the optimised route and topology construction within an RPL Instance by choosing the potential parents by translating one or more metrics into the rank value. An Objective Code Point (OCP) within the DIO Configuration option, precisely in the object filed, identifies the Objective Function. There exist two objective functions that have been standardised by IETF [59]: Objective Function Zero (OF0) and Minimum Rank with Hysteresis Objective Function (MRHOF). These two Objective Functions have been implemented in the standard RPL routing protocol as the default objective functions. The RPL standard does not impose any adoption of certain OF neither routing metrics and even splits the implementation of objectives functions from the fundamental specifications of RPL. Thus, the RPL specification left the door open for more improvements to any new objective functions.

1.6.4.3.1 Objective Function Zero (OF0)

The OF0 is the standard OF proposed and designed to enable interoperability between different implementations of RPL. The OF0 is simplified and does not use any routing metrics listed in [65], such as throughput, latency, link quality, and node energy, for the Rank definition. It uses only the hop count as a routing metric. Furthermore, in the OF0, the preferred parent is selected, considering the neighbour nodes' minimum Rank. Thus, the Rank will increase strictly from the node towards the sink monotonically.

1.6.4.3.2 Minimum Rank with Hysteresis Objective Function (MRHOF)

The MRHOF is more complicated than OF0. It was developed to select a path by emphasising the hysteresis. The MRHOF uses different node/link metrics to compute a node's Rank. The MRHOF takes into account only the metrics specified in RFC 6551. By default, the MRHOF uses the Expected Transmission Count (ETX) metric to evaluate the quality of the links among the nodes and select routes with high end-to-end throughput. The ETX is defined according to Equation (1.2).

$$\mathrm{ETX} = \frac{1}{\mathrm{D_f} * \mathrm{D_r}} \tag{1.2}$$

Where:

- D_f is the measured probability that a neighbour receives a packet.
- D_r is the probability that an acknowledgement packet is successfully received.
- D_f * D_r is the expected probability that a transmission is successfully received and acknowledged.

1.6.4.4 Routing Attacks against RPL

RPL is vulnerable to many attacks, which researchers have treated in the literature [66] [67]. Nowadays, several classifications for RPL threats exist. In our earlier study [66], we proposed two main classes: the novel RPL specification-based attacks and the existing routing attacks tailored to the context of RPL. The first class includes the rank, neighbour, and version number attacks, whereas the second consists of the hello flooding, selective forwarding, Sybil, wormhole, and Blackhole attacks. In the following, we give the definitions of the two attacks addressed in our contributions (see Section 4.3).

1.6.4.4.1 Rank Attacks (RA)

There exist several variants of the Rank attack, namely Decreased Rank Attack [68], Rank Attack [69], Worst Parent Attack [70], and Increased Rank Attack [71]. These attacks lead to generate loops in the network, exhaust node resources and congest the network. In this thesis, we give particular attention to the decreased rank attack. In this attack, the malicious node illegitimately advertises a better Rank equal to a lower Rank value inducing other nodes to select it as a parent. Once selected as a parent, the adversary node can trigger other attacks, such as the sinkhole or selective forwarding.

1.6.4.4.2 Blackhole Attack (BA)

In the BA attack, the malicious node drops all packets (control and data packets) routed through it [72] [73] [74]. In the literature, researchers consider this attack as a DoS attack. Indeed, the Blackhole attack is more dangerous if combined with Rank or sinkhole attacks since the attacker is in a position where normal nodes route colossal traffic through it. This attack increases the number of exchanged DIO messages, which leads to instability of the network, data packets delay, and consequently, resource exhaustion.

1.7 IoT challenges

Several issues at the different layers of the protocol stack have been addressed by the scientific community on sensor networks [75]. The main problems are energy efficiency (limitation of resource in WSN), scalability (the number of nodes can increase considerably), reliability (the network may be used in critical applications), and robustness (sensor nodes could be subject to failure) [75]. However, there are still challenges that need to be addressed to support IoT. The leading challenges are classified into three categories, the standardisation activity, the addressing and networking problems, and the security issues.

1.7.1 Standardisation activity

Standardising technology in IoT is indispensable to ensure better interoperability. At present, various producers use their own techniques, which will lead to inaccessible services. For this reason, Standards need to be designed to move from

"Intranet of Things" into the more complete "Internet of Things". In the context of IoT, several research works aim to improve, standardise, or update standards, to implement and deploy a complete standardisation of the IoT paradigm defined by the scientific community. Among them, we quote the Auto-ID laboratory [76] [77] [1], the European Commission [78] [1] and the European standards organisations (ETSI, CENELEC, etc.), the international organisations (ISO, ITU), and the standardisation consortia (IETF (Internet Engineering Task Force), EPCglobal, etc.). Besides, several contributions and workgroup are in progress, such as the European Telecommunications Standards Institute (ETSI) and the IETF working group [78]. 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) -previously cited- is an IETF working group [79]. The basic protocols making up the 6LoWPAN architecture have already been published. The IETF ROLL working group is particularly interested in RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [80].

According to the group of European projects CERP-IoT (the Cluster of European Research Projects) [81], the integration of different objects in wider networks, mobile or fixed, will facilitate their interconnection with the Internet of the future [1]. For this, the collaboration between standardisation and normalisation institutions, the various working and research groups and the industrial sector is necessary.

1.7.2 Addressing and networking problems

The IoT consists of a considerable number of heterogeneous objects, in different locations, which requires efficient addressing techniques and solutions. Nowadays, IPv6 addressing has been proposed for low power wireless communication nodes (6LoWPAN). IPv6 addresses are expressed on 16 bytes, which will generate 10^{38} addresses, which allows an address to be assigned to any communicating object in the network [1]. In particular, RFID tags use identifiers from 64 to 96 bits, standardised by EPCglobal. To allow the addressing of RFID tags into IPv6 networks, several works have been proposed on the integration of RFID tags into IPv6 networks [82] [83]. Also, adapted mechanisms are necessary to support mobility in different IoT scenarios. Another research issue is how to obtain IP addresses for connected objects. The concept of Object Name Service (ONS) has been introduced to provide description and reference to each object [84]. The ONS technology based on the EPCglobal standard has the same principle as Internet DNS (Domain Name Service, Domain Name System).

In traditional Internet, the protocol used at the transport layer for reliable communications is the Transmission Control Protocol (TCP) [85]. However, the TCP is inadequate for the IoT for several reasons [1], and then a new design of the transport layer is required for the IoT. Therefore, it will also need quality of service (QoS) support. Several works have been carried out, and others are in progress to support QoS in an IoT context [86] [87] [88]

Researchers have addressed the networking challenges from different points [89]: Small MTU, Multi-link subnet, Multicast efficiency, Mesh network routing, and Resource discovery.

1.7.3 Security issues

The IoT is exceptionally vulnerable to various possible attacks. The main problems related to security concern are Data confidentiality and Integrity, Identification, Authentication, and Access Control, Availability, Privacy, and Trust. Indeed, several complex security mechanisms and protocols exist for different networks; however, because of the constrained nature of IoT devices and networks only lightweight protocols can be used to keep the balance between maximising security and minimising resource consumptions.

1.7.3.1 Data Confidentiality

Data confidentiality represents a critical security issue in IoT because of the massive number of measurement devices (RFID, sensors, etc.) that could be integrated into the IoT. Data confidentiality aims to guarantee that the data is accessible, and it can be modified only by permitted entities and cannot be listened by not allowed entities. In the IoT context, the entities could be users or objects. For that, it is crucial to confirm that the data collected by a device will not divulge secure information to not permitted neighbouring devices. Several techniques have been proposed and improved in the literature; the best known are those who deal with key management in the IoT [90].

1.7.3.2 Data Integrity

Integrity ensures that the data cannot be forged or tampered by a third party during the data delivery within networks. It is necessary to provide accurate data for authorised users. Integrity is important for IoT because if IoT applications receive forged data or tampered data, erroneous operation status can be estimated and wrong feedback commands can be made, which could further disrupt the operation of IoT applications. To achieve acceptable integrity, enhanced secure data integrity mechanisms (false data filtering schemes, etc.) should be developed and applied [25].

1.7.3.3 Identification, Authentication, and Access Control

Furthermore, two essential aspects need to be addressed: the definition of an object authentication process and the definition of an access control mechanism. It is necessary to properly manage the identity and its authentication in the IoT while taking into consideration the limitations and characteristics of the IoT. The authentication of identity can be done between two objects, or between an object and a server or service provider [90]. It can confirm that the data conveyed in networks are authentic, and the devices or applications that request the data are also legitimate. For this, several research works are in progress to improve the solutions already proposed or to develop new effective mechanisms [2] [91]. Access control is essential in any system. A service can be particular for a particular object with rights and restrictions. Several policies of access control have been proposed, and other work is underway to offer better solutions [90] [2].

1.7.3.4 Availability

Because IoT networks are made of hostile devices, and regarding the sensitivity of IoT applications, it is important that devices, networks and offered data and services should be always available and work properly. An attack on availability can be conducted by triggering Distributed Denial of Service (DDoS) attacks caused by a huge number of distributed attackers or Denial of Service (DoS) attacks, such as traffic flooding/overload by a huge amount of messages to IoT servers and devices. In the context of security, intrusions and malicious activities should be detected. Intrusion Detection Systems (IDSs) and firewalls are used to ensure availability security.

1.7.3.5 Privacy

The IoT covers a lot of applications in different fields, and each application is used by several users or clients [90] [17]. The personal information of these users, such as their locations, interaction histories and preferences, should be protected to guarantee the protection of their privacy. The concept of privacy is deeply regarded as one of the essential security principles. Privacy defines the rules to ensure that only the corresponding user can control their data and that no other user can access or process the data. In the literature, several solutions aim to ensure the protection of the privacy of users [90] [90] [2].

1.7.3.6 Trust

Trust concept is profoundly considered as one of the essential security principles. It has been studied carefully in different fields and domains, such as social sciences, economics, philosophy, and cyberspace. The trust concept is defined differently in the literature, depending on the context of trust. The heterogeneity of IoT components in addition to the nature of communication channels and other characteristics, make IoT vulnerable to several security issues related to each layer of the IoT architecture. These vulnerabilities need to be addressed so that all participating entities in the IoT environment should be trustworthy. Several security solutions exist to ensure different requirements, such as confidentiality, integrity, access control, and privacy. Trust can guarantee the security mentioned above to be achieved during the interactions among various objects, different IoT layers, and different applications. In fact, in all cases, trust management systems have to detect non-trustworthy behaviour, isolate untrusted entities and zones, and redirect IoT functionalities to trusted zones.

1.8 Conclusion

In this chapter, we gave a complete definition, vision, and characteristics of IoT. The enabling technologies were presented. The IoT potential applications are also submitted with more focus on the near future trend of such applications and their impacts. Besides, different IoT architectures have been illustrated. We also provided a clear overview of the protocols accommodated in the current IoT standardised stack. The protocol RPL in the network layer has been given more attention. Moreover, we addressed the most leading IoT challenges and issues to be resolved. These

challenges are supported incredibly by the scientific community, especially security issues.

In the next chapter, we will focus on trust management issue in the context of IoT, by describing the security requirements for IoT, presenting trust related properties, overviewing and classifying existing trust models, describing the trust issues and related attacks, and finally giving a synthesis of the existing research works concerning trust models.

Chapter 2

Trust Management in the Internet of Things

In this chapter, we focus on trust management issue in the context of the Internet of things (IoT). The main idea of trust management is to create a relationship based on trustworthiness between all participating nodes. Thus, each node within the network should communicate only with trusted nodes. On the one hand, without trust between nodes, the communication cannot start. On the other hand, establishing and maintaining trust relationship between IoT components (objects, systems, etc.) is vital to ensure that the overall design is more efficient in terms of security.

2.1 Security Challenges For IoT

IoT has specific characteristics such as heterogeneity, connectivity and ubiquity, resources limitation, self-organisation, mobility, and scalability, in addition to the complexity of IoT networks due to the increasing number of devices connected to the Internet, as well as, the growing amount of data generated by these devices. It is evident from the above and the non-standardisation of IoT technologies that several challenges for securing IoT devices and networks rise and have to be addressed [2] [4]. The IoT applications generate an increasing amount of data that are targeted by both attackers and commercial competitors. Thus, it is necessary to ensure data confidentiality, integrity, and privacy. Besides, authentication, authorisation, access control, availability, privacy-preserving, and trust management should be integrated into all IoT systems.

- **Confidentiality:** ensures that data are revealed to only authorised entities that can access and modify them securely. In other words, the exchanged data within IoT applications should be hidden from intermediate and unauthorised entities. Due to IoT devices' resource-constrained nature, there is a trivial need of lightweight cryptographic cyphers that can provide optimal confidentiality. Researchers have surveyed lightweight cryptographic solutions that have been proposed recently [92].

- **Integrity:** ensures that data have not been changed in transit by an intermediary or a malicious entity. Hence, any change of exchanged data has to be imperatively detected. Like data confidentiality, data integrity can be provided

through cryptographic methods. Researchers recently used the blockchain features to ensure data integrity verification of data produced by IoT devices [93].

- **Authentication:** ensures that each entity in IoT is uniquely identified. Nodes can be replaced or copied, so, each node needs to identify itself, and mutual authentication between IoT entities is required. Consequently, any impersonating node should be detected. End nodes' authenticity needs to be addressed using software and/or hardware solutions [94]. Researchers surveyed and analysed different authentication protocols for IoT [95].

- **Authorisation and Access Control:** ensure that only authorised users access IoT entities. Because illegal access to IoT entities will put the network's security at stake, it is fundamental to disclose data and route information only to authorised parties. Several solutions have been proposed for IoT applications authorisation and access control [96] [97].

- **Availability:** states that despite the exposure to malicious attacks or failures due to IoT characteristics, IoT entities, networks and services should be always available and work properly. Recently, researchers focus on the availability issues for all IoT services and networks [98] [99] [100].

- **Privacy:** The information privacy goes from data privacy, location privacy, query privacy, to identity privacy. Information can include end-user identity, door lock password, time of turning off lights, blood pressure and heart rate, etc. The privacy ensures that IoT entities information is highly protected from the third party; for instance, by defining the rules under which data referring to individual entities may be accessed. According to Kumar and Patel (2014), privacy needs to be addressed in the IoT device itself, in storage, communication, and processing. In the literature, privacy is handled using several methods. For instance, using probability distribution technique [101], cryptography-based privacy-preserving method [102], or anonymise sensitive data method [103]. Lastly, researchers investigated using blockchain technology to preserve privacy [104] [105].

- **Trust:** Because of IoT characteristics, there is a need for architecting the IoT in a trustworthy manner allowing the ability to adapt to the unexpected security breaches automatically. In fact, trust management systems have to detect non-trustworthy behaviour, isolate untrusted entities and zones, and redirect IoT functionalities to trusted zones. In the following sections, the trust concept, definitions, properties and objectives are outlined. In addition, research studies on trust models and trust management in the context of IoT are provided.

2.2 Trust Definitions

Trust concept has been studied thoroughly in different fields and domain, such as social sciences, economics, philosophy, and cyberspace. Consequently, the trust concept is defined differently in the literature, depending on our views and the context of trust. Following some of the existing definitions:

1. **Definition 1:** Mayer, Davis, and Schoorman [106] defined trust as the willingness of a party to be vulnerable to the action of another party based on the expectation that the other will perform a particular action necessary to the trustor, irrespective to the ability to monitor or control that other party [107].

2. **Definition 2:** In online transactions, Kimery and McCord (2002) defined trust such as online trust is a customer's willingness and enables to accept an online transaction according to their positive and negative expectations on future online shopping behaviour [107].

3. **Definition 3:** Also, Corritore, Kracher, and Wiedenbeck (2003, page 740) [108] defined online trust as *"an attitude of confident expectation in an online situation of risk that one's vulnerabilities will not be exploited"*.

4. **Definition 4:** Chang, Dillon, and Hussain [109] defined trust as the belief that the trusting agent has in the trusted agent's willingness and capability to deliver a quality of service in a given context and in a given timeslot.

5. **Definition 5:** Buttyan and Hubaux [110]: Trust is about the ability to predict the behaviour of another Party.

6. **Definition 6:** Based on the analysis of several reports, Aljazzaf, Perry, and Capretz (2010, page 168) [111] defined trust such as: *"Trust is the willingness of the trustor to rely on a trustee to do what is promised in a given context, irrespective of the ability to monitor or control the trustee, and even though negative consequences may occur"*.

7. **Definition 7:** Daubert, Wiesmaier, and Kikiras [112] defined trust in the context of IoT as device trust, entity trust, and data trust; where trusted computing and computational trust could be used to establish device trust. Entity trust refers to the expected behaviour of participants, such as persons or services. Furthermore, trusted data may be derived from untrusted sources by aggregation or may be created from IoT services where data require trust assessment.

Trust definitions include several concepts such as dependency, confidence expectation, vulnerability, reliability, comfort, utility, context-specificity, risk attitude, and lack of control [111]. Accordingly, there is no standard definition of Trust; nevertheless, it is evident that trust management's primary goal is leveraging security by assisting in decision-making processes.

2.3 Trust Properties

As aforementioned, the trust concept has been seen and interpreted in many different ways and in different contexts, making it a very complicated idea. Because trust is influenced by many properties (attributes) that can be related or not to security, authentication, confidentiality, integrity, availability, and identity management could be considered as trust concepts and/or trust components. According to a different review, trust relates to other factors or attributes, such as goodness,

reliability, availability, ability, or other characters of an entity.

Different actors participate in trust management, where each actor plays one or several roles. Thus, an actor can be a trustor and/or a trustee, or even a third party that gives its opinion about another actor. Other actors can be service requesters, service providers and trusted third parties (credentials or gather feedbacks). To establish a trust relationship, a trustor must trust a trustee within a specific context. As already stated, several trust properties related to the different actors have to be considered when elaborating a robust trust management scheme. These properties can be summarised as follow [113] [114] [16]:

1. **Context properties:** A trust relationship is based on the context, which indicates all the information that define involved actors' situation. In other words, the purpose of the Trust, the environment of Trust (e.g., time and location), the evolved actors' role, and the Trust's risk are defined a priori. For example, a trustor can trust a trustee to forward a data packet in one context; however, the same trustor cannot trust a trustee to do other tasks in another context.

2. **Subjectivity properties:** These properties are trust factors that are difficult to measure and monitor. They are more involved in cognitive or social Trust.

   a. Trustor: confidence, (subjective) expectations or expectancy, subjective probability, willingness, belief, disposition, attitude, feeling, intention, faith, hope, trustor's dependence and reliance.

   b. Trustee: honesty, faith, goodness, motivations and benevolence.

3. **Objectivity properties:** These properties are trust factors that can be measured and monitored. These properties are more involved in the computational trust.

   a. Trustor: assessment, criteria or policies specified by the trustor to make a trust decision.

   b. Trustee: competence, ability, security, dependability, integrity, predictability, reliability, timeliness, reputation (observed behaviour), strength, availability.

2.4 Trust Management Objectives For IoT

The heterogeneity of IoT components and the nature of communication channels and other characteristics, make IoT vulnerable to several security issues related to each IoT architecture layer. These vulnerabilities need to be addressed as all participating entities in the IoT environment should be trustworthy. Besides, uncertainty and risk are critical issues for IoT deployment since entities could be untrusted, and thus, security could be easily broken. In this context, trust management is crucial for reliability, privacy and information security, allowing IoT users to be more specific and confident regarding IoT services. First, sensed and exchanged data need to be trusted; thus, securing these data can be considered trust management. Second, entities within an IoT network need to communicate using trusted

relationships; therefore, identity controls and authorisation systems must be established with build-in Trust management to share information reliably. Third, data and application have to be accessed from only trusted entities. Hence, access control solutions have to be established based on trustworthiness. In conclusion, identification, authentication and authorisation, as well as access control systems and other existing security protocol could be part of an extensive trust management system.

Yan et al. [16] highlighted objectives that trust management in IoT should respond to. These objectives are summarized as follow:

1. **Trust Relationship and Decision (TRD):** A trust relationship is based on the context, which indicates all the information that define the situation of involved actors. Hence, the trust relationship is not absolute. In other words, the purpose of trust, the environment of trust (e.g., time and location), the evolved actors' role, and the Trust's risk are defined a priori. For example, a trustor can trust a trustee to forward a data packet in one context; however, the same trustor cannot trust a trustee to do other tasks in another context.

2. **Data Perception Trust (DPT):** Reliability and trustworthiness of sensed and collected data should be ensured. In this context, the objective properties of the trustee should be considered in the physical sensing layer.

3. **Data fusion and mining trust (DFMT):** The sensed data should be processed and analysed in an accurately trustworthy way while ensuring reliability and privacy preservation.

4. **Data transmission and communication trust (DTCT):** The sensed and processed data should be transmitted and communicated securely in a trustworthy way. Thus, trust-based routing and secure key management are required to address data transmission and communication trust objective.

5. **Privacy preservation (PP):** Users and data privacy are critical issues that need to be addressed to fulfil trust objectives.

6. **Quality of IoT services (QIoTS):** The quality of IoT services should be ensured while maintaining a high-security level.

7. **System security and robustness (SSR):** reliability against attacks and IoT system availability should be ensured to get users' Trust.

8. **Generality (G):** It is more desirable to have a generic trust management system, that is nor depending neither on the context nor on other specific requirements.

9. **Identity Trust (IT):** Trust depends on identity. Having identity enables building the history of the interactions related to that identity.

The different objectives relate to the different IoT architecture layers, which means that trust management should be ensured in all layers and needs crossing-layer support. Indeed, trust management systems for IoT should help detect malicious nodes by assisting other security protocols and mechanisms, such as authentication

mechanisms, Intrusion Detection Systems, key management systems, and privacy-related mechanisms. For instance, a node can use trust evaluations to revoke the keys of an untrusted node. Furthermore, nodes can also use trust to select which neighbour will collaborate to distribute a pair-wise key.

2.5 Trust Models Classifications

Trust management is considered as one solution to loT security issues. There is a need to distinguish between trust management and trust modelling. Indeed, trust modelling describes trust establishment and computation techniques. Consequently, trust models contribute to the specific development and realisation of trust management for IoT. According to Airehrour et al., (page 17) [4], *"Trust modelling is a useful practice of estimating the level of reliability among devices within a system. It pinpoints the concerns which could affect the trust of a system while helping to identify areas where a low value of trust could degrade a system's operational efficiency and usability."*.

Whereas, *"Trust management is a service mechanism that self-organising a set of items based on their trust status to take an informed decision."* [115].

Because trust definition depends on the properties mentioned above and on the context where, and what purpose it is going to be used, different trust models can exist. Trust models are made of a set of properties, rules and methods to forge trust among entities. Indeed, they depend on one or more forms of extraction, evaluation, and transmission of trust information, besides the mechanism used to decide. In the literature, several trust models have been proposed.

2.5.1 Trust Model of Airehrour et al.

Airehrour et al., [4] introduced different trust models based on methods used to evaluate trust. For instance, Bayesian statistics, game theory, entropy, fuzzy, probability, neural network, swarm intelligence, directed/undirected graph, arithmetic/weighting and Markov chain are the methods used to evaluate trust for secure routing. As depicted in Figure 2.1, for each approach, the authors associated a trust model.

2.5.2 Trust Model of Nunoo-Mensah et al.

Most existing trust models use analytical techniques to evaluate trust values; nevertheless, other methods such as evolutionary algorithms, ant colony–based algorithms, machine learning, and social networks have been used. For this reason, Nunoo-Mensah, Boateng, and Gadze [5] introduced new classes of trust models based on biologically inspired and socio–based trust methods. The authors classified trust models as socio-inspired, bio-inspired, and analytical. As shown in Figure 2.2, the analytical class includes the different methods presented in [4].

2.5.3 Trust Model of Moyano et al.

Another classification has been proposed by Moyano et al., [6], where the authors presented two classes: decision models and evaluation models (See Figure 2.3). The

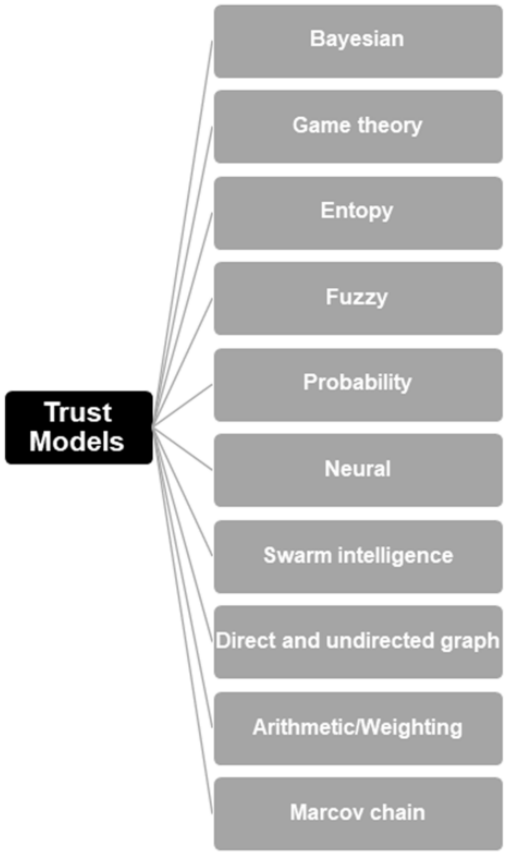


Figure 2.1: Trust Models according to Airehrour et al., [4]

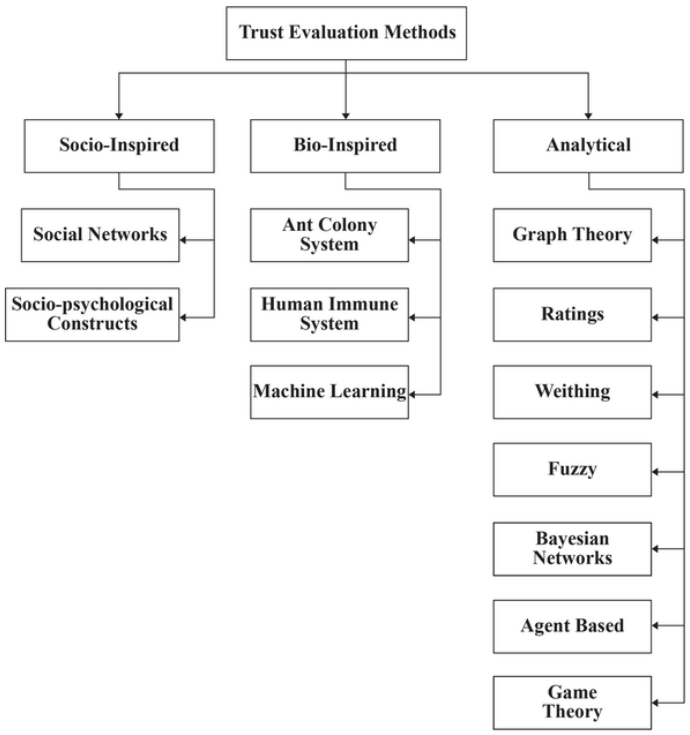


Figure 2.2: Trust Models according to Nunoo-Mensah et al., [5]

first class includes policy models and negotiation models, whilst the second class comprises propagation (flow) models, reputation models and behaviour models.

2.5.3.1 Trust Decision Models

These models bring unified solutions to access control decision and managing the authentication and authorisation process by making them into one task.

2.5.3.1.1 Policy models: These models aim to grant access to resources using the conditions predefined in policies.

2.5.3.1.2 Negotiation models: In these models, two entities perform a step-by-step, negotiation-driven exchange of credentials and policies until they decide whether to trust each other or not.

2.5.3.2 Trust Evaluation Models

These models are also known as computational trust models. Unlike the decision models, the evaluation models use measurement to quantify trust. They evaluate and quantify entities attributes such as reliability, honesty, and integrity to calculate trust value.

2.5.3.2.1 Behaviour models: In these models, each trust relationship is associated with a trust value indicating the degree of trust has the trustor in the trustee. The trust values are calculated using chosen trust metrics.

2.5.3.2.2 Propagation models: These models indicate the way an entity disseminates the trust information to other entities. Trust propagation can be distributed or centralised [116].

i. Distributed trust: Entities calculate and propagate the trust observation to other entities autonomously. The centralised entity is not necessary.

ii. Centralised trust: Entities cannot propagate trust observation. Only the centralised entity is responsible for the trust propagation.

2.5.3.2.3 Reputation models: In these models, different entities exchange trust information and collaborate to evaluate an entity. Thus, each entity takes into account the recommendation of others to evaluate another entity.

2.5.4 Trust Model of Guo et al.

Guo et al., [116] did not consider trust decision models; instead, they proposed a more detailed trust evaluation models classification. Indeed, five trust sub-models can be used in these models: trust composition (QoS trust, Social trust), trust propagation (Distributed, Centralized), trust aggregation (Belief Theory, Bayesian systems, Fuzzy logic, Weighted sum, Regression Analysis), trust update (Event-Driven, Time-Driven), and trust formation (Single-trust, Multi-trust). Guo et al., [116] proposed a classification of works in the literature based on combining the following different models.

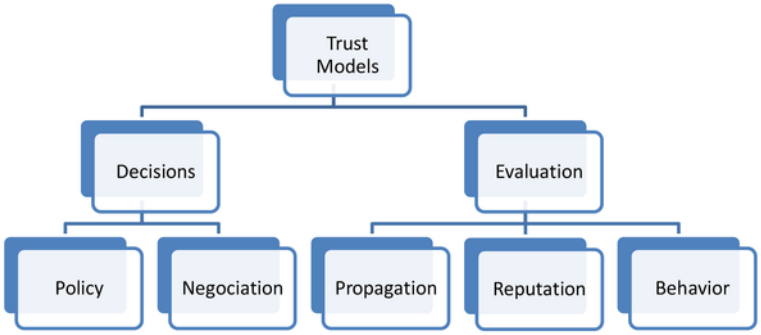


Figure 2.3: Trust Models according to Moyano et al., [6]

2.5.4.1 Composition models

In these models, entities must know what trust properties to use to calculate Trust. Composition models include Quality of Service (QoS) trust models and Social trust models.

2.5.4.1.1 QoS models: Refers to an entity's level of expectation that another IoT entity can achieve its functionalities properly. These models use some trust properties such as competence, cooperativeness, reliability, and task completion to measure trust values [14].

2.5.4.1.2 Social models: These models are the mapping of the social relationship between IoT entities owners. Thus, the map is used to assess if an IoT entity is trustworthy or not. Social trust uses some trust properties such as intimacy, honesty, privacy, centrality, and connectivity to measure trust values [14]. Social models refer more to subjective properties, whilst QoS models refer more to objective properties.

2.5.4.2 Propagation models

They are already defined in Section 2.5.3.2.2.

2.5.4.3 Aggregation models

These models indicate the best way to aggregate trust information evaluated by the entity itself (direct evaluation) or by other entities (indirect evaluation) [14]. There exist different trust aggregation techniques in the literature: weighted sum, belief theory, Bayesian inference, fuzzy logic, and regression analysis.

2.5.4.4 Update models

These models indicate when to update trust values. The trust information update can be performed after an event or transaction that can affect the QoS *(Event-driven)* or periodically *(Time-driven)*.

2.5.4.5 Formation models

These models indicate whither trust calculation is based on only one trust property *(Single-trust)* or based on multiple properties *(Multi-trust)*. Besides, formation

models should indicate what weights to put on social and QoS trust properties to form trust [14].

2.5.5 Proposed Classification

Based on [6] [116], Guo et al., [116] added new sub-models in trust evaluation models defined in [6]. In this thesis, the author believes that aggregation models can be considered as sub-model of reputation models. Indeed, reputation can be used as a means to determine whether an entity can trust another entity [117]. In this context, aggregation models have to be used to gather and aggregate recommendation from the indirect evaluation. Tormo, Mármol, and Pérez [118] used the weighted sum aggregation method in the reputation system. Furthermore, the others' belief that aggregation models can also be considered as a sub-model of formation models. In the case of multi-trust formation models where properties have to be combined, adequate aggregation methods such as weighted sum have to be used to evaluate trust. In other words, both formation and reputation models use aggregation models in some cases. In addition, evaluation models presented in [4] [5] are sub-models in aggregation models.

Figure 2.4 presents the different models with differences related to enlightenment brought in this thesis.

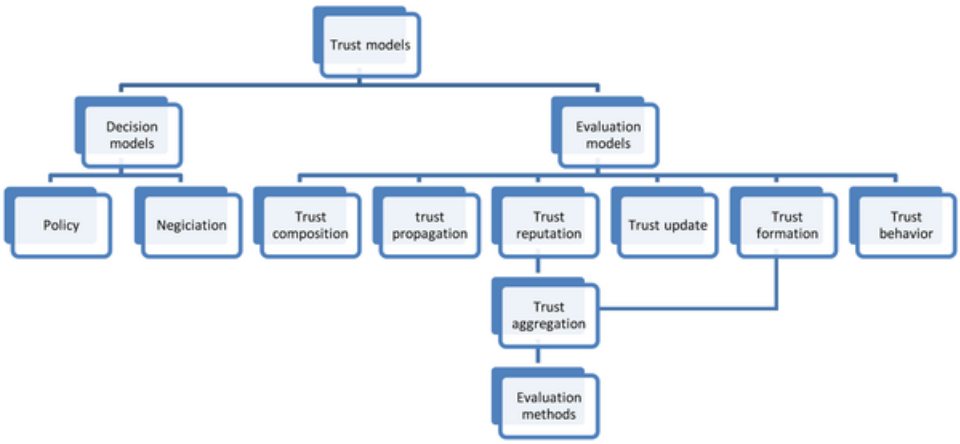


Figure 2.4: Trust Design Models

2.6 Trust Issues And Trust-Related Attacks

Although problems due to the complexity, characteristics and nature of the IoT, other issues raised. These latter are due to the wrong understanding of trust management's need and the complexity of the trust management system itself. The most relevant issues that can disrupt trust are interoperability and dynamicity [119].

- **Interoperability:** The heterogeneity is a crucial characteristic for IoT networks. Indeed, IoT's devices have several computational, storage and communication capabilities and use different protocols to communicate and cooperate. Precisely, different trust management protocols can be used in IoT

applications. For that reason, solutions must be provided to allow trust management systems (protocols) to communicate, interoperate and work together consistently despite their differences.

- **Dynamicity:** IoT environments are very dynamic. Any device can be connected or disconnected to/from the network, which causes several changes to the network conditions and disturbing the communication. In these conditions, the trust management system must update itself to adapt to any environmental changes.

Besides interoperability and dynamicity issues, another problem that can seriously disrupt a trust management system is its vulnerability to some attacks. Different attacks can be executed by a malicious node, such as forgery attacks, jamming attacks, replay attack, eavesdropping, Sybil attacks, denial of service attacks, black/sinkhole attacks, and slandering attacks [16] [116] [14]. However, these attacks are specially designed to create problems whenever nodes want to calculate trust value of some others to perturb trust systems [120] [90] [116]. Indeed, most of trust management systems rely on cooperation among distributed entities. Nonetheless, cooperation can easily be damaged by selfish behaviours and exploited by a malicious attacker to trigger the following trust-related attacks:

- **Self-promotion attacks (SPA):** In these attacks, the malicious node manipulates its reputation by providing good recommendations for itself [121] [116].

- **Bad-mouthing attacks (BMA):** The malicious node manipulates another trusted node's reputation by providing bad recommendations for it [116].

- **Ballot-stuffing attacks (BSA):** Also known as *Good-Mouthing Attacks*. In these attacks, some malicious nodes can cooperate to trigger the attack. One malicious node manipulates another malicious node's reputation by providing good recommendations for it [116].

- **Opportunistic service attacks (OSA):** The malicious node attempts to become opportunistic by providing a good service to keep its reputation high. The aim is to cheat with other malicious nodes to carry out bad mouthing and good mouthing attacks [116].

- **On-off attacks (OOA):** The malicious node provides a good and bad service alternatively. The aim is to keep its reputation good, and it can compromise the network by giving a good recommendation for malicious nodes or bad recommendation for trusted nodes. These attacks seem to be more difficult to detect.

Several solutions have been proposed in the literature to mitigate these attacks. For instance, authors in [122] [123] proposed trust management system to mitigate on-off attacks. The proposed solution also mitigates bad-mouthing and ballot-stuffing attacks [123]. Furthermore, researchers suggested self-organising maps to detect bad-mouthing attacks on trust reputation systems [124]. Some trust systems are initially designed to be resistant to the presented attacks like the one introduced in [125] [126] , which is immune to Bad-mouthing, Ballot-stuffing, and self-promoting

attacks.

According to Sun et al., [120], the following attacks also perturb trust systems:

- **Selective behaviour attack:** A malicious node could behave well from the point of view of most of its neighbours, and behave badly concerning the rest of the nodes. Thus, the average recommendation will remain positive, while it can cause damage to certain nodes.

- **Sybil attack and newcomer attack:** In the authentication and access control breaches, a node can create, emulate or impersonate different nodes in the network. Thus it can manipulate the recommendations and promote itself as a respected node. These attacks allow a malicious node to throw away its bad reputation by creating a new identity.

2.7 Trust Management In IoT

IoT networks security is of great importance in real-life applications. Considerable research works have been conducted to address security in the field of communications and networking. Special attention is on trust management as trust could be embedded in communication and network protocol designs. Besides, the need for cooperation and collaboration between participating nodes is critical in developing trust relationships as these determine the availability, dependability and secure operations of the network.

Trust management issues have been addressed for various computer, communication and information systems, such as Mobile Ad-Hoc Networks (MANET), social networking, Wireless Sensor Networks (WSNs), and recently the Internet of Things. In most works, researchers consider that IoT objects are only wireless sensors. Hence, this chapter overviews some of the trust management solutions in WSNs.

Two categories of Trust Models (TMs) exist for WSNs: Ordinary WSNs TMs (OTMs) and Cluster-based WSNs TMs (CTMs). The authors in [127] proposed two sub-categories of OTMs. The first sub-category is called Node TMs, where the models are classified as centralised or distributed. In centralised models, all nodes trust values calculations are made by a centralised base station [128]. However, these models consume high energy and are not suitable for most WSNs. In distributed models, the nodes themselves make trust values computations [129] [130] [131]. Nevertheless, these latter involve huge memory and computation complexities. The second sub-category is called Data TMs [132] [133] [134]. In these models, nodes' trust values evaluation is based on data inconsistencies or erroneous data processing. Yet, Data TMs are not used to secure data. For the CTMs, Shaikh et al., [135] proposed a hybrid (distributed/centralised computation) group-based TM, where a single trust value is attributed to a group of nodes. This model protects against malicious and selfish nodes. However, it is based on an unrealistic assumption and is not protected against TMs attacks (bad-mouthing attack, on-off attack, etc.). Zhou et al. proposed a hybrid TM that monitors the different nodes' behaviour and detects invalid data from compromised and faulty nodes [136]. The

drawback of this model is the cluster head, which is vulnerable to malicious attacks.

In [13] [14], the authors proposed a hierarchical trust management protocol based on both QoS trust (energy, unselfishness) and social trust (intimacy, honesty) properties. These different metrics used formation, reputation (aggregation) and update models to compute the trust level of a node. The protocol uses direct observations based on nodes' knowledge and indirect recommendations from the network's nodes to update trust values. This approach exploits a clustering approach to cope with a large number of heterogeneous sensor nodes. Further, it handles selfish and malicious sensor nodes for survivability and intrusion tolerance. However, the different metrics used in this protocol are calculated using the energy as a parameter. Consequently, if a normal node is surrounded by selfish nodes, it will consume more energy, and it can be considered non-trusted while it is trusted. Besides, since the protocol uses indirect recommendations, it can be vulnerable to good and bad-mouthing attacks.

This chapter classifies the current research works on trust management for IoT regarding the IoT architecture layers. As presented in Chapter I, there exist several proposals regarding the IoT architecture. The most common one is the three-layer architecture composed of the sensor layer, the network (core) layer and the application layer [2]. The sensor layer is composed of physical devices such as RFID, sensors and actuators. The most used standard in the field of IoT for this layer is the IEEE 802.15.4 standard. It specifies a sub-layer for Medium Access Control (MAC) and a physical layer (PHY) for low-rate wireless private area networks (LR-WPAN). Data and information are collected, processed, and transmitted to a base station via wireless channels in this layer. The network layer is an intermediary layer that collects data and information from the sensor layer and routes them to the application layer using wired and wireless communication networks like WiFi, ZigBee, LTE and GPRS. This layer covers underlying technologies such as the IPv6 Over Low Power Wireless Area Networks (6LoWPAN) and the Routing Protocol for Low-Power and Lossy Networks (RPL) for managing and routing collected data. Both sensor and network layers are hosted on constrained and power-limited IoT devices. The application layer aggregates and analyses received data from the network layer to provide the final users' requested services. It covers underlying technologies such as Constrained Application Protocol (CoAP) and is hosted on powerful devices due to its complex and enormous computational needs.

The Three IoT layers are subject to several specific attacks and threats. Since making a trust mechanism for the whole IoT is very difficult, the trust mechanism can be established for each of the three IoT mentioned above layers. This solution allows controlling and handling the trust for each layer depending on special purpose: sensors self-organising for the sensor layer, efficient and secure routing of data packets and control messages for the network layer, and multi-service for the application layer, as depicted in Figure 2.5.

Different surveys have classified trust management models for IoT. Yan et al., [16] used the context, and the objective and subjective properties of the trustees and the trustors to categorise trust models. According to the authors, trust management should be widely applied to various IoT systems. Trust should be ensured vertically

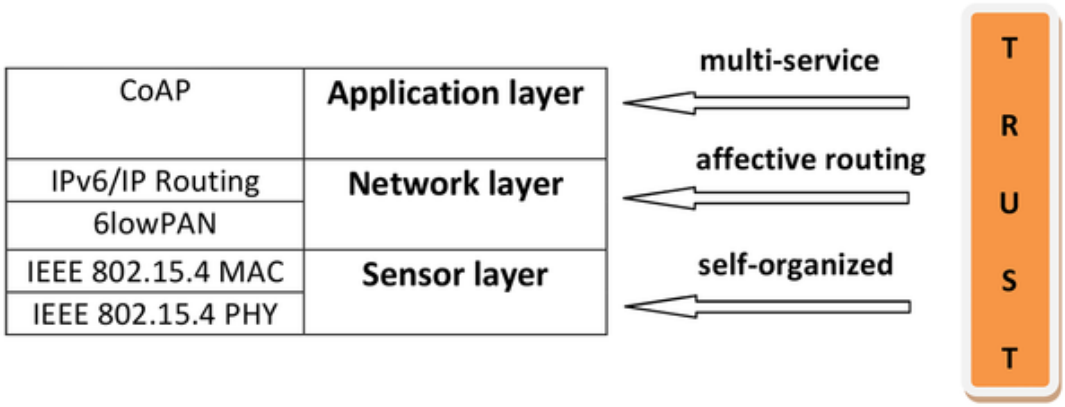


Figure 2.5: Layered trust for IoT

in all IoT layers through different security aspects combining decision models and evaluation models. Hence, the global model should include trust evaluation between the entities in all layers, system and entities reliability and availability, privacy and key management, and trust routing and Quality of IoT Services (QIoTS).

Also, Sicari et al., [90] surveyed trust solutions for IoT. According to the authors, trust is a complex notion used in various contexts with different meanings. They classified trust models into four categories: Social networking models, Fuzzy methods models, Cooperative approach models, and Identity based method models.

Furthermore, Guo et al., [116] presented a classification of trust computation models for service management in service-oriented IoT systems. This classification contains eight classes based on five trust design dimensions: trust composition (QoS trust, social trust), trust propagation (distributed, centralized), trust aggregation (belief theory, Bayesian systems, fuzzy logic, weighted sum, regression analysis), trust update (event-driven, time-driven), and trust formation (single-trust, multi-trust). Besides, the authors presented trust-related attacks, which can perturb the trust computation models: self-promotion attacks (SPA), bad-mouthing attacks (BMA), ballot-stuffing attacks (BSA), opportunistic service attacks (OSA), and on-off attacks (OOA).

2.7.1 Trust Management In MAC Layer

The openness and the resource-constrained nature of IoT networks make channel access more vulnerable to serious security problems. The IEEE 802.15.4 Media Access Control (MAC) layer faces the risk of Denial of Service (DoS) attacks from malicious nodes, aiming to degrade the network performance or even make it down. Furthermore, IEEE 802.15.4 MAC layer is subject to attacks known as MAC unfairness attacks where an attacker attempts to get a dominating position and hold unfair advantages over the other nodes.

David and de Sousa Jr [137] presented a new attack used by malicious nodes to control Guaranteed Time Slots (GTS) access (unfairness attack problem) and perform DoS attack. This attack can be used in both beacon-enabled and non-

beacon-enabled modes. The authors proposed to counter this attack by using a Bayesian trust model based on collected MAC sub-layer data. In this model, the coordinator evaluates the trust value of the node by using the node's behaviour metrics. The coordinator uses previous node information and compares it with information about node received from other nodes (recommendation) to evaluate the node's trust value.

2.7.2 Trust Management In Network Layer

Authors in [138] [139] proposed a trust management model for Social IoT (SIoT). This work is the continuity to previous ones in [13] [14]. This model uses social relationships attributes: honesty, cooperativeness, and community-interest to calculate the nodes' trust level. Further the problem from [13] [14], this model is based on social relationships (social IoT environment), and hence cannot be used in a wide range of IoT applications.

Chen et al., [15] proposed Community of Interest dynamic hierarchical trust management (COI-HiTrust) protocol that integrates mobility in trust evaluation. In COI-HiTrust, trust protocol parameter settings can be dynamically adjusted in response to environments changes. The authors used COI-HiTrust in the context of a MANET network. Also, authors in [140] proposed a 3-tier cloud-cloudlet-device hierarchical trust-based service management protocol (IoT-HiTrust) that eliminates the problems of the protocol suggested in [138] and [139]. The authors used cloud servers to stock many recommendations for each trustee node.

2.7.2.1 *Trust Management In Routing Protocol*

The works mentioned above introduced trust management solutions in the broad context of IoT. Recently, some works have been conducted on trust management for the IoT IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). For instance, Karkazis et al., [141] [142] introduced the Packet Forwarding Indication (PFI) metric to build trust knowledge as a trust-related metric for RPL. In this approach, each node transmits a packet to one of its neighbours and listens whether this neighbour forwards the packet or not. Then, it calculates the probability for the packet to travel along the path successfully. The drawback of this approach is that each node takes a decision based only on its knowledge. Thus, if this node misbehaves, it will choose a failing path rather than a trusted one.

To secure communications in an RPL-based network, authors in [143] proposed to use the classical security mechanisms of a Trusted Platform Module (TPM). This approach uses TPM to establish the trustworthiness of nodes before exchanging keying material. Furthermore, it provides a secure method to exchange group keys used to secure control messages. However, if a node becomes infected or misbehaves after establishing its trustworthiness and exchanging group keys, it remains trustworthy against other nodes. Moreover, the trustworthiness establishment is done only for exchanging keys securely and not for routing.

Airehrour et al. [144] [145] proposed SecTrust-RPL: a trust-aware RPL routing protocol to secure RPL from routing attacks. In SecTrust, the trust calculation

process evaluates a node's trustworthiness based on direct and indirect packet forwarding behaviour between linked and 2-hops nodes, respectively. Although SecTrust uses indirect trust observation, a node recommendation depends only on the neighbour of its indirectly linked neighbours (the parent of its parent). In other words, the indirect trust of a node is calculated based only on one recommendation of the intermediate neighbour, which makes it vulnerable to Bad-mouthing and Good-mouthing attacks.

Khan, Ullrich, Voyiatzis, and Herrmann [146] proposed a centralised trust-based model for managing the reputation of every node participating in RPL-based network. In this model, each node relies on packets routed across the network to calculate direct trust for other nodes, thus elaborating positive and negative experiences with other nodes. The gathered trust information is then transmitted to a central entity (6LBR), which evaluates the network nodes' interactions and gives them a global reputation. This solution uses only direct observation (direct trust), making it vulnerable to a single point of failure.

Lahbib et al. [147] proposed LT-RPL, Link reliable and Trust aware model for RPL protocol. In their approach, the nodes send their node ID, neighbour ID, remaining energy percentage, the packet forwarding ratio, the Packet Reception Ratio(PRR), the Packet Error Rate (PER), the Expected Transmission Count (ETX), the transmission delay as well as the entity time to a trust manager, periodically. The trust manager stores trust-related data and evaluates the trust of each node. Authors did not explain how they calculated the recommendations. The proposed trust mechanism can counter Greyhole and Blackhole attacks; however, there is neither an IDS nor a mechanism to detect other attacks such as Version Number and Sybil attacks. Besides, since the trust manager handles the storage and computation tasks, this solution is vulnerable to a single point of failure.

Medjek et al. [148] introduced a new trust-based Intrusion Detection System scheme for RPL, named T-IDS. T-IDS is a distributed, cooperative and hierarchical IDS, which can detect novel intrusions by comparing network behaviour deviations. In T-IDS, each node monitors and collaborates with his peers to detect intrusions and report them to a 6LoWPAN Border Router (6LBR). In this work, authors added a new timer and extensions to RPL messages format to deal with mobility, identity and multicast issues. To off-load security-related computation and storage, authors equipped each node with TPM.

Seyyed and Fereidoon [149] proposed DCTM-IoT, a Dynamic and Comprehensive Trust Model for IoT which have a multi-dimensional vision of trust. The authors integrated several parameters in trust calculation, such as packet forwarding indicator, ETX, energy, and mobility. Besides, the nodes calculate trust using direct and indirect observations from neighbours. The authors claim that their programmed code size (48.28 Kbytes) is less than the objects' (Tmote Sky) available memory (48 Kbytes) which is not valid. From one side, too much information (historical) are used in DCTM-IoT and need to be stored and handled, thus making the solution not lightweight for constrained objects. From another side, the authors did not present nor how they integrated their model to RPL (recommendations, trust propagation,

new objective function), neither how they used the detection of attacks with the trust calculation process.

Kiran et al. [150] proposed a trust-based DDOS attack detection approach. The authors used packet frequency within a time interval as a trust indicator. The root node calculates the data frequency rate and maintains lists of nodes that crossed the rate for several intervals. Each time a node appears in a list, its trust value diminishes. When the trust value falls under a threshold, the root classifies the node as malicious and sends its identity to other nodes. The nodes receiving the malicious node identity discard it from the routing operation. From one side, this approach can only detect DDOS attack. From another side, the fact that it is centralised makes it vulnerable to a single point of failure.

2.7.3 Trust Management In Application Network

There exist several application security issues; for instance, information access and user authentication, information privacy, destroy and track of data stream, IoT platform stability, middleware security, management platform, and so on. Many works were proposed in the literature to overcome application layer breach, among them trust models.

Saied, Olivereau, Zeghlache, and Laurent [151] proposed a centralised context-aware trust model to manage collaboration between nodes with different context and resource capacities. In this model, a node intending to set up a collaborative service sends a trustworthiness request to a central trust manager. The trust manager collects trustworthiness information on the nodes depending on a given context. It then outputs recommendations on nodes to the requesting node. The requesting node relies on the recommended nodes' collaborative service and assesses the quality of each service provision from each assisting node. Finally, the trust manager performs self-updates by learning from past operations to improve future operations.

In [115] [152], the authors proposed a layered trust model using fuzzy set theory and a formal semantics-based language. In this model, there exist service requesters and a service provider. The IoT is considered as a service provider and is composed of three layers: the sensor layer, the core layer, and the application layer. The trust management scheme includes three steps: trust information extraction specific for each layer, trust transmission to the next layer, and finally trust decision-making, transmitted to the service requester.

Truong, Lee, Askwith, and Lee [153] suggested a trust evaluation model platform that they considered as Trust as a Service (TaaS) for SIoT environment. They defined a mathematical formula to aggregate QoS and social attributes in a specific context to decide on the trustworthiness of a given service provider.

2.7.3.1 *Trust Management In The Cloud Computing*

Unlike IoT devices, Cloud computing has virtually unlimited capabilities in terms of storage and processing power. Besides the computational and storage capabilities, IoT and Cloud have several complementary aspects. IoT is a pervasive environment

composed of real-world things with limited reachability, that have the Internet as a point of convergence and are sources of Big data. On the other side, the Cloud is a centralised environment composed of virtual resources with ubiquitous reachability, that use the Internet to deliver services and manage Big data [154].

Cloud servers are usually used to hold the IoT applications and sensed data. These servers facilitate the flow between IoT data collection and data processing. Botta et al. [154] surveyed IoT applications that used cloud services to be significantly improved. The growth of IoT devices will be followed by the increase of Cloud services for these devices. Nevertheless, this use of cloud technologies raises several security breaches that have to be addressed. Dabbagh and Rayes [155] discussed some attacks against the cloud; for instance, hidden-channel attacks, VM (Virtual Machine) migration attacks, theft-of-service attack, and VM escape attack. Gonzales, Kaplan, Saltzman, Winkelman, and Woods [156] listed several attacks that can be targeted against the Cloud, based on the exploited node security breach; for instance, VM CPU timing side-channel attack, software-defined networking attack, nested virtualisation attack, and so on.

Indeed, common security challenges are related to the lack of trust in data security and privacy, and the service provider. In this context, several works have been proposed. Kantarci and Mouftah [157] proposed a Mobility-Aware and trustworthy Crowdsourcing (MACS) framework in a cloud-centric IoT architecture providing Sensing-as-a-Service (SaaS) to a smart city management platform for public safety. SaaS enables collecting sensed data through numerous smart devices equipped with various types of sensors based on pay-as-you-go. MACS collects information about users and calculates their trustworthiness over time based on their reputation. The trustworthiness values are then used to reduce the payments made to the malicious users who aim at disinformation at the smart city management authority.

Wu, Yang, and He [158] addressed the problem of VM live migration when the hypervisor is untrusted. The authors proposed a framework named SMIG to provide enhanced security protection of user data during migration from an untrusted hypervisor to a trusted one. In their solution, each node integrates a Trusted Platform Module (TPM), and a list of trusted hypervisors is maintained in a table named the Integrity Validation Table (IVT). This table is created and updated by the Region Critical Trusted Computing Base (TCB). The IVT determines when to start the migration and where to migrate VMs. The authors suggested proposed a new target (trusted hypervisor) determination protocol (TDP) to secure communications between service providers and the Region Critical TCB. They introduced a dynamic integrity measurement mechanism that uses collected security data information to detect whether the hypervisor is compromised or not. The authors stated that their solution is vulnerable to other network-related attacks.

Gonzales et al. [156] proposed a Cloud Computing System (CCS) reference architecture and a cloud security assessment called Cloud-Trust. Cloud-Trust relies on conditional probabilities representing the probability that realises if other CCS components have already been compromised. The calculated probability takes into consideration information from IAM and SIEM. Cloud-Trust defines trust zone

(TZ) as a combination of network segmentation and identity and access management (IAM) controls. The authors used a Bayesian network model to construct an acyclic directed graph using the attack paths for different CCS components. They evaluated their solutions for different security configurations. The authors concluded that a highly secured CCS architecture is the one that uses firewalls to inspect packets and block IP ports and protocols. These firewalls must include host-based Intrusion Detection Systems (IDSs), keystroke logging, reverse web proxy servers, DMZs, IAM servers, security incident event managers (SIEMs), and other detection and protection systems. Hence, CCS architectures with more security controls have a lower probability of successful APT infiltration. Even if the attack itself is not detected, the compromised CCS component can be deleted.

Fogs are a miniaturised version of Clouds that provide virtualised computing resources (VM) to bring the computing capabilities to the 6LBR. These resources are shared between the IoT constrained devices to off-load data processing. The new paradigm named Fog computing is itself subject to several threats such as authentication and trust issues, risks related to VM higher migration, DoS Attacks, and privacy issues. In this context, trust management scheme has to be introduced. For example, the trust solution proposed for the Cloud can be adapted (i.e., lightweight solution) to the context of Fog.

Table 2.1 summarises the work proposed in the literature according to the three architecture layers (MAC layer, Network layer and Application layer) and to the trust-related attacks.

2.8 Synthesis

This chapter discussed some trust models proposed in the literature. There is no one unique model as the concept of trust is used in many different contexts and with different meanings. Most existing trust models indeed use no a unique definition of Trust. We presented a comparative study of trust-based approaches according to the trust models involved in trust evaluation. From our point of view, building up trust in a volatile and dynamic IoT environment still a big challenge. Indeed, with the technological evolution of IoT networks, trust management will be much more complicated since the number of entities and service providers is dramatically increasing; the relationships user–service provider is becoming transient; in addition to the fact that entities play multiple roles where users can become service providers.

Table 2.3 summarises the different trust solutions and their respective used models. Table 2.2 summarises notations used in Table 2.3.

2.9 Conclusion

In this chapter, we have outlined the essential background to facilitate the understanding of Trust (i.e., definitions, actors, properties, and objectives). We illustrated problems and obstacles to trust management, such as interoperability and dynamic-

Table 2.1: A Summary of Trust-based Models With Respect to IoT Layers and Trust-Based Attacks

Models	**Architecture layer**	**Attacks**		
		SPA	**BMA**	**BSA**
[137]	MAC layer	✓	-	-
[138] [139]	Network layer	✓	-	-
[15] [140]	Network layer	✓	-	-
[141] [142]	Network layer (RPL)	✓	-	-
[143]	Network layer (RPL)	-	-	-
[145] [144]	Network layer (RPL)	✓	-	-
[146]	Network layer (RPL)	-	-	-
[147]	Network layer (RPL)	-	-	-
[149]	Network layer (RPL)	-	-	-
[150]	Network layer (RPL)	-	-	-
[151]	Application layer	✓	✓	✓
[115]	MAC, N and A layers	-	-	-
[152]	MAC, N and A layers	-	-	-
[153]	Application layer	-	-	-
[157]	Application layer	-	-	-
[158]	Application layer	-	-	-
[156]	Application layer	-	-	-

Table 2.2: Notations

Notation	**Description**
QoS	Trust Quality of service
Soc	Trust Social
Dis	Distributed
Cen	Centralised
Wig	Wight sum
Fuz	Fuzzy
Bay	Bayesian
E	Event-Driven
T	Time-Driven
Sin	Single-trust
Mul	Multi-trust
SPA	Self-promotion attacks
BMA	Bad-mouthing attacks
BSA	Ballot-stuffing attacks

ity, in addition to attacks against trust models. The chapter overviewed trust models classifications, and then a new classification has been introduced. Furthermore, we surveyed existing trust management solutions in WSNs, IoTs and particularly the

Table 2.3: Synthesis of Trust-Based Solutions in IoT Networks

Work	**Composition**		**Propagation**		**Aggregation**			**Update**	**Formation**		**Behaviour**	**Reputation**
	QoS	Soc	Dis	Cen	Wig	Fuz	Bay	E/T	sin	mul		
[137]	✓			✓			✓	T				
[138] [139]	✓	✓	✓	✓	✓			E/T		✓	✓	✓
[15] [140]	✓	✓	✓	✓	✓			E/T		✓	✓	✓
[141] [142]	✓		✓		✓			T	✓			✓
[143]		✓	✓					E/T	✓			
[145] [144]	✓		✓		✓			T	✓			✓
[146]	✓			✓	✓			E/T	✓		✓	✓
[147]	✓			✓	✓			E/T	✓		✓	✓
[149]	✓			✓	✓			E/T	✓		✓	✓
[150]	✓			✓	✓			E/T	✓		✓	✓
[151]	✓			✓	✓			E	✓		✓	✓
[115] [152]	✓			✓		✓			✓			
[153]	✓	✓		✓	✓		✓	E		✓		✓
[157]	✓	✓		✓	✓			T	✓		✓	✓
[158]		✓		✓	✓			T		✓	✓	

RPL routing protocol. We elaborated a synthesis of trust-based solutions classification based on the IoT layers, trust evaluation models and trust-related attacks.

The following chapters present our contributions and the work that we have carried out to answer the research problem. We will eventually evaluate and compare our models with other trust models to prove the effectiveness and security of our schemes and also rank it compared to other trust models.

Part II Contributions

Chapter 3

Trust Management in MAC Layer

This chapter introduces a new algorithm for trust management in the Media Access Control (MAC) layer. The aim is to address the MAC unfairness attacks against such layer. The proposed trust model focuses especially on the Guaranteed Time Slots (GTS) related attacks.

3.1 Problem Statement

The research community considers the IEEE 802.15.4 standard as one of the enabling technologies for the short-range, low rate, and wireless communications. Hence, the IEEE 802.15.4 is most suitable for IoT. Indeed, it is the de-facto standard to define the physical and MAC layer for IoT networks [8]. Because the MAC layer is the basis of interconnecting IoT nodes, it is targeted by several attacks that have been widely surveyed [159] [160]. The IoT networks' vulnerability makes channel access security a serious problem. The IEEE 802.15.4 MAC layer faces the risk of attacks from malicious nodes, attempting to get a dominating location and hold unfair advantages over the other nodes. In this contribution, we address MAC unfairness attacks, especially GTS related attacks where attackers attempt to bypass the MAC priority. In these attacks, a malicious node cheats to get higher priority than legitimate nodes to maximise the channel access utilisation [161]. Most of MAC security solutions proposed in the literature are based on cryptography mechanism to deal with confidentiality and authentication issues.

Nevertheless, these solutions cannot handle MAC unfairness attacks. Indeed, embedding minor changes in the IEEE 802.15.4 standard will make it more secure against this type of attacks. We propose a new MAC-trust-based model to handle MAC unfairness attacks while maintaining channel access to all participating nodes. In our scheme, a Personal Area Network (PAN) Coordinator Manager (PCM) cooperates with PANs and Coordinators to detect malicious behaviour, calculate trust values for participating nodes, and keep up a blacklist of malicious nodes. Our model modifies Guaranteed Time Slots allocation policies according to nodes' trust values. Each time the trust decreases, the number of slots allocated to the node decreases too until no priority is assigned to the node.

In the following, we present a background of IEEE 802.15.4 GTS MAC process and related attacks, and then we introduce our proposed model.

3.2 Background

3.2.1 IEEE 802.15.4 PROTOCOL

IEEE 802.15.4 is a standard developed by the IEEE 802.15 Personal Area Network Working Group. It is designed for the physical and the MAC layers for low-rate wireless personal area networks (LR-WPANs) [52] [25]. In this document, we use the term PAN instead of WPAN. The IEEE 802.15.4 protocol aims to ensure a reliable data transfer, short-range operation, low-cost and reasonable battery life during the process. Relying on its efficiency, many recently specified upper layer-networking stacks are built on top of IEEE 802.15.4, such as the 6LowPAN, ZigBee, and WirelessHART. The IEEE 802.15.4 has a maximum transmission rate of 250 Kb/s and a maximum transmission unit (MTU) of 127 bytes; nonetheless, only up to 116 bytes are available for an upper-layer protocol.

The IEEE 802.15.4 supports two different types of devices (nodes) according to their capabilities and available resources: full-function device (FFD) and reduced-function device (RFD) [7]. An FFD can be a PAN coordinator, a coordinator (router), or an ordinary device, which can communicate with both RFDs and FFDs. The (PAN) coordinator is the main controller of the network and is responsible for managing the network operations, such as association/disassociation of the devices and their medium access coordination. It may be mains powered. In contrary, an RFD is a simple device with restricted resources that cannot be a PAN coordinator or a coordinator. It can only communicate with a (PAN) coordinator, and it is intended just for elementary applications. The RFD awakes when it is required only to mitigate the power consumption to the least level.

The IEEE 802.15.4 MAC defines two transmission modes; non-beacon-enabled and beacon-enabled. The non-beacon-enabled mode uses unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to control the medium access distributively. In the beacon-enabled mode, there are two channel access methods: the CSMA/CA in the contention access period (CAP), and the Time Domain Multiple Access (TDMA) using synchronisation beacons and Guaranteed Time Slots (GTS) in the contention-free period (CFP).

An IEEE 802.15.4 network can be built as star topology (3.1-a), peer-to-peer topology (3.1-b), and cluster-tree topology (3.1-c). In each topology, at least one FFD is required to serve as the network's coordinator.

- The star topology contains at least one FFD and some RFDs. Once an FFD is activated, it can create its network and become the PAN coordinator. This last is located at the topology's centre, and all the traffic between devices (i.e., FFDs and RFDs) must pass through it. Besides, the PAN coordinator is responsible for initiating, routing, and terminating the network's communication.

- The peer-to-peer topology contains one PAN coordinator and other nodes that can be FFDs and RFDs. In this topology, each node can communicate with other nodes within its radio communication range.

- The cluster-tree topology contains a PAN coordinator, a cluster head, and nodes (i.e., FFDs and RFDs). It uses the two above-cited topologies. This topology can allow nodes to communicate with other nodes outside of their radio communications range via the clusters.

To form a topology, the IEEE 802.15.4 specified a mechanism that a PAN coordinator executes for channel selection when starting a new network and a procedure, called association procedure, allowing devices to join a network.

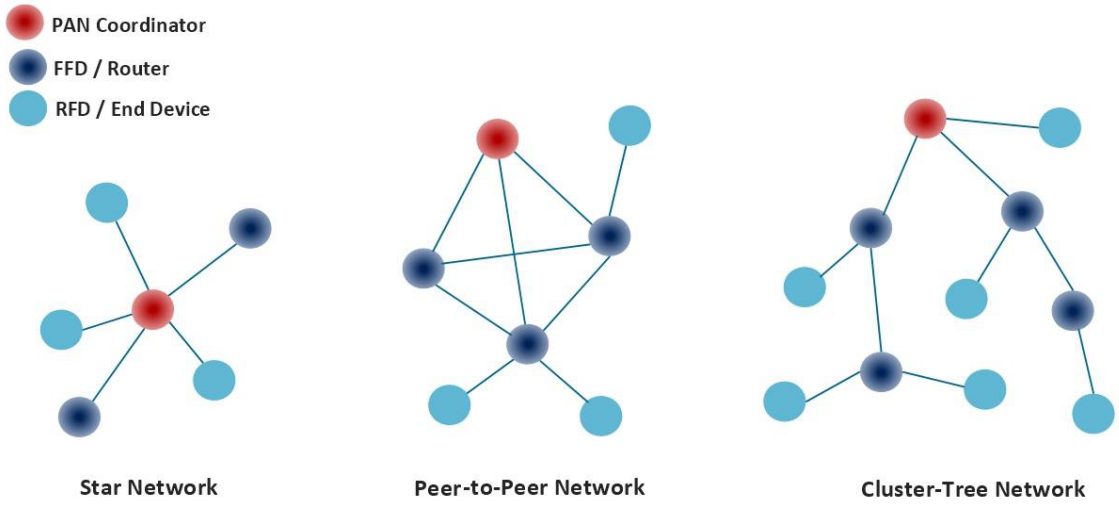


Figure 3.1: Topologies in IEEE 802.15.4 Networks: (a) Star topology, (b) Peer-to-peer topology, (c) cluster-tree topology [7].

3.2.2 The IEEE 802.15.4 Beacon-Enabled Transmission

As stated in the previous section, IEEE 802.15.4 networks can operate on beacon or non-beacon enabled modes. In our model, we focus on the beacon-enabled mode. In this mode, the superframe is started with a beacon frame transmitted by the PAN coordinator and is delimited by two beacons. The beacons are used to identify the PAN coordinator and to synchronise the devices within the network. As depicted in Figure 3.2, the superframe is fully defined using a beacon interval (BI) and a superframe duration (SD), where BI refers to the time between the two beacons. The superframe may have an active period also referred to as SD and an optional inactive (i.e., idle) period. Devices can go to the low-power or sleep modes during the idle period, and thus save power.

On the other hand, each active period (SD) of the superframe is divided into 16 equally size slots. Besides, SD is divided into a Contention Access Period (CAP) and a Contention Free Period (CFP), as illustrated in Figure 3.2 and Figure 3.3. The nodes compete for medium access using slotted CSMA/CA within the CAP during SD. Hence, when a device wishes to transfer data to a coordinator, it listens for the network beacon. When the beacon is found, it synchronises to the superframe structure and transmits the data frame to the coordinator using slotted CSMA/CA algorithm. Likewise, the nodes that require dedicated bandwidth or low-latency

transmission can optionally activate the CFP upon requesting guaranteed time slots (GTSs) [162].

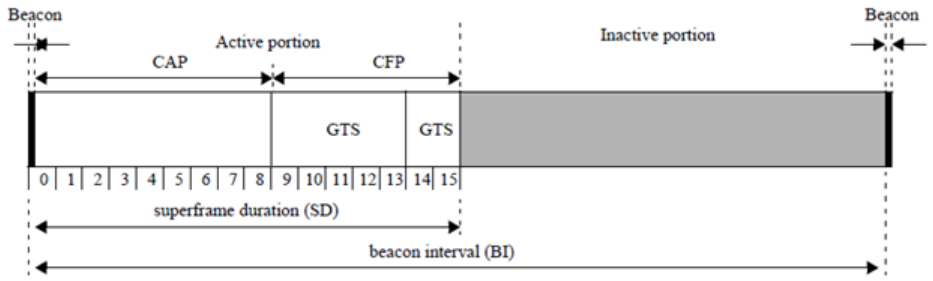


Figure 3.2: IEEE 802.15.4 superframe structure

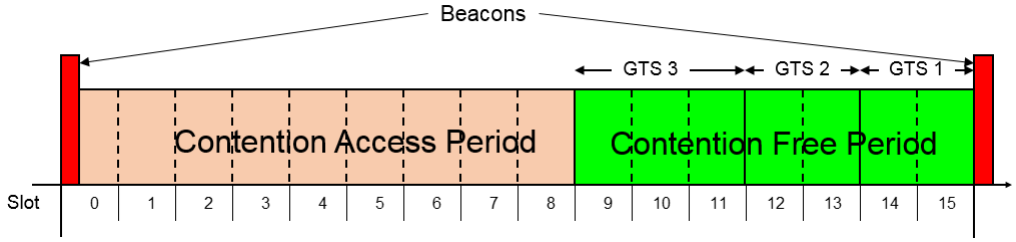


Figure 3.3: Structure of the active periods with GTSs

The PAN coordinator reserves GTS within the CFP of each superframe duration to provide real-time guaranteed channel access to in-network nodes for delay-sensitive applications. The PAN coordinator allocates and deallocates GTS on a First-come, First-serve basis [162], as depicted in Figure 3.4. It may allocate up to seven GTS at the one time. A node sends a GTS request frame during the CAP to request GTS from the coordinator. The node waits for the response of the coordinator in the next beacon. After that, the coordinator either accepts or rejects the request based on the current resource capacity available in the superframe. Once the coordinator grants a GTS request from a node, it reserves the GTS for that node during the CFP. On receiving beacon transmitted by the PAN coordinator, each node tries to send its packet using the superframe. Nodes that do not succeed in accessing the channel discard the packet and generate a new packet at the next superframe.

3.2.3 Guaranteed Time Slot (GTS) Attacks

There exist many attacks against the IEEE 802.15.4 MAC layer. This section gives particular attention to the GTS MAC attacks.

Malicious nodes extract slots information from the beacon sent by the (PAN) coordinators. Then, they trigger different MAC attacks, such as the link-layer-jamming, node-specific flooding, back-off manipulation, Clear Channel Assessment (CCA) manipulation, acknowledgement (ack) attacks, and other attacks. Furthermore, the GTS MAC channel-sharing scheme is vulnerable to malicious nodes that misbehave and break the standard communication rules to capture the channel with

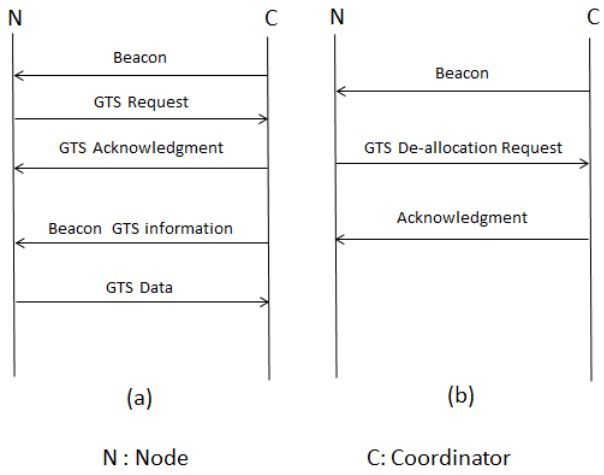


Figure 3.4: (a) GTS allocation process, (b) GTS de-allocation processes.

higher priority utilisation. Researchers have defined three IEEE 802.15.4 GTS attacks' classes: spoofing identities based attacks, fabricated identities based attacks, and interference-based attacks.

3.2.3.1 Spoofing Identities Based GTS Attacks

In these attacks, the malicious node spoofs existent identities in the PAN network to trigger the following GTS attacks [163].

3.2.3.1.1 Dos against Data Transmissions during CFP (DDTC)

To trigger DDTC, the adversary starts by passively eavesdropping on network traffic to collect information about the identifiers (IDs) of legitimate nodes and their allocated GTS. The malicious node can then use this collected information to spoof the IDs and send GTS deallocation requests on their behalf to the PAN coordinator. This leads to the termination of channel access rights previously granted to the attacked nodes during their assigned GTS [160] [137] [164].

3.2.3.1.2 False Data Injection (FDI)

On the contrary to the DDTC attack, in the FDI attack, the adversary collects information about the IDs of legitimate nodes that have not yet been allocated any GTS during the CFP period. It spoofs these IDs, pretends to be one of the unallocated, and sends a GTS allocation request on their behalf to the PAN coordinator. Consequently, it injects false traffic into the network during its stolen assigned GTS [160] [137] [164].

3.2.3.2 Fabricated Identities Based GTS Attacks

In this class of attacks, a malicious node can use its own or other non-existing IDs to conduct the following attacks [165].

3.2.3.2.1 Dos against GTS Requests (DGTSR)

In the DGTSR attack, a malicious node collects information about the GTS list that contains allocated and free GTS. As a result, the adversary keeps sending GTS allocation requests to the PAN coordinator until all seven slots in the GTS list are replete [160] [137] [164].

3.2.3.2.2 Stealing Network Bandwidth (STB)

In the STB attack, the malicious node triggers the previous DGTSR attack with the addition that it also injects false traffic into the network during the assigned GTS [160] [137] [164]. This attack is hard to detect than the DGTSR attack because the PAN coordinator does not drop the allocated slots. Indeed, it recognises that these later are being used for transmitting traffic.

3.2.3.3 Interference based GTS Attacks

In this class of attacks, one or two malicious nodes collect information about the beginnings and ends of GTS slots that the PAN coordinator has assigned to legitimate nodes. They (it) then use(s) link-layer-jamming to create interference during the assigned slots with the intent of corrupting ongoing transmissions as in next sub-sections [164] [163].

3.2.3.3.1 One or Two Intelligent Attackers (OTIA)

In OTIA attacks, one intelligent attacker corrupts the GTS slot's first superframe slot or the entire superframe slots contained within the GTS slot. Alternatively, two intelligent malicious nodes could collaborate. The first one attacks the communication with the largest GTS slot length, whereas the second one attacks with the second-largest GTS slot length.

3.2.3.3.2 One or Two Random Attackers (OTRA)

In OTRA attacks, one or two malicious nodes attack(s) the GTS slot of one or two randomly selected communication(s). In the case of two attackers, it is possible to target the same communication due to the random nature of the attack.

3.3 THE PROPOSED MODEL

As presented above, IEEE 802.15.4 MAC GTS is subject to several attacks that can harm its functionalities. In the present chapter, we propose two algorithms that aim to enhance the IEEE 802.15.4 MAC GTS security. The first algorithm allows verifying the association process, whereas the second one permits to allocate GTS dynamically for real-time applications based on nodes trustworthiness.

The GTS period in the IEEE 802.15.4 MAC is adjustable by beacon parameters (i.e., BeaconOrder-BO and SuperframeOrder-RO) [162]. In our model, the GTS period is initially set using BO and RO. After the first GTS request, the GTS period

is recalculated and reallocated based on nodes trust values.

Three entities (actors) participate in the proposed model: (1) one Pan Coordinator Manager, denoted PCM, (2) at least one PAN Coordinator and Coordinators denoted C_i, (3) nodes (sensors) with identifiers, N_j. Coordinators and PANs are full-function devices, whereas nodes can be FFDs or reduced-function devices. The PCM keeps in a table (i.e., database) the list of all coordinators and PANs and the list of all nodes within the network. Indeed, for each Coordinator C_i, PCM maintains the list of nodes associated with it, the trust value, denoted T_{Nj}, of each node N_j, and the number of GTS request frames, denoted NB_{Nj}, for each node N_j. The PCM monitors GTS across the entire network by keeping the history of all stationary and mobile nodes. For security consideration, we assume each node N_j is associated with only one C_i at time t.

3.3.1 Controlled MAC Association

As already said, each node is allowed to be associated with only one PAN/Coordinator at one-time t. Hence, each time a node sends an association request to a PAN or a Coordinator, this later sends an association control request to the PCM. Then, the PCM checks in its database the state of the node. Two cases could occur:

1. The node does not exist in the database, which means it is not associated with any PAN/Coordinator. In this case, the PCM sends an Association Control Acknowledgement, and the PAN/Coordinator can associate this node.

2. The node is already associated with one PAN/Coordinator. In this case, the PCM sends a Request status to the PAN/Coordinator associating the node. Two cases could occur:

   - The node became orphan because it lost the connexion with the PAN/Coordinator. In this case, the PCM sends an Association Control Acknowledgement and updates its database.
   - The node is associated correctly to the PAN/Coordinator. Thus, the PCM blacklists the node and sends an Association Control Notification to all PAN/Coordinators.

Figure 3.5 and Algorithm 1 summarises the controlled association process.

3.3.2 Adaptive Allocation GTS MAC

Initially, at the first association, all the network's nodes are fully trusted. Thus, trust values of all nodes are set to one (i.e., $T_{Nj}=1$). Besides, the number of GTS request frames for each node is set to zero (i.e., $NB_{Nj}=0$). The maximum number of GTS request frames allowed within a period T for each node is set as a threshold, denoted TH.

After being successfully associated with the PAN/coordinator, each node N_j sends GTS request frames through which it asks the PAN/coordinator to assign it a number of GTS as defined by BO and RO. Once the PAN/Coordinator receives

Algorithm 1 Controlled Association Algorithm

Input: One PCM; a number of coordinators M; a number of nodes N; each node N_j is associated to only one coordinator $C_i \in M$;
N_j sends Association Request to C_i;
C_i sends Association Control Request (Nj,C_i) to PCM
PCM checks if $N_j \in C_h$ ($h \neq i$)
If $N_j \in C_h$ ($h \neq i$) **do**

PCM sends Request status to C_h ($h \neq i$)

If N_j is associated **do**

$N_j = 0$;

Blacklists N_j;

Sends Association Control notification (N_j) to C_i;

Sends Disassociation notification (N_j) to C_h;

Ci sends Disassociation notification to Nj;

Else If N_j is orphan **do**

PCM sends Association Control Acknowledgement (N_j) to C_i;

C_i sends Association Acknowledgement to N_j;

END If

Else If $N_j \notin C_h$ **do**

PCM sends Association Control Acknowledgement (N_j) to C_i;

C_i sends Association Acknowledgement to Nj;

END If

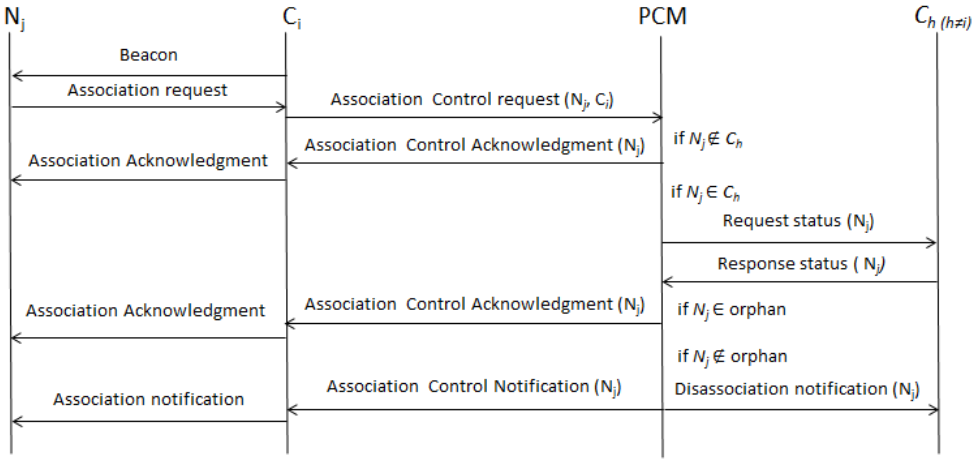


Figure 3.5: Controlled Association process.

the request from N_j, it increments the number of requests for that node NB_{Nj} (i.e., $NB_{Nj}= NB_{Nj}+1$) and sends the node identifier N_j and NB_{Nj} to the PCM. On receiving N_j and NB_{Nj}, the PCM checks if NB_{Nj} equals the threshold TH. If NB_{Nj} equals TH, the PCM sets T_{Nj} to zero, blacklists N_j and sends GTS notification to all PAN/Coordinators. If NB_{Nj} is less than TH, the PCM calculates the new trust value T_{Nj} according to Equation 3.1 and sends GTS Acknowledgement with the node identifier N_j, the number of GTS request frames NB_{Nj} and the new trust value T_{Nj} for this node to the PAN/Coordinator.

$$T_{Nj} = 1 - NB_{Nj}/TH \tag{3.1}$$

For the first GTS request, the PAN/Coordinator acknowledges the nodes and allocates them a number of GTS equal to the number of requested GTS. Afterwards, the allocation is done according to nodes' trust value as follow.

The PAN/Coordinator splits GTS to three sub-GTS: GTS1 (2 slots), GTS2 (2 slots) and GTS3 (3 slots) [162]. It splits the trust interval onto three sub-intervals: [1, 2/3],]2/3, 1/3], and]1/3, 0[.

- If the new calculated trust value T_{Nj} is in [1, 2/3], the PAN/Coordinator allocates the node a number of GTS equal to the number of requested GTS (i.e., up to seven slots).
- If T_{Nj} is in]2/3, 1/3], the PAN/Coordinator allocates the node a number of GTS up to five slots (i.e., GTS3+GTS2). Even though the number of requested GTS is greater than five, the node will be assigned a maximum of five slots.
- If T_{Nj} is in]1/3, 0[, the PAN/Coordinator allocates the node a number of GTS up to three slots (i.e., GTS3). Even if the number of requested GTS is greater than three, the node will be assigned a maximum of three GTS.

If the PAN/Coordinator receives GTS request from two or more nodes at the same time, instead of allocating GTS on a First-come, First-serve basis, the PAN/Coordinator allocates GTS on a trust basis. Which means, the first served is the node with the

greatest trust value.

The allocation process is repeated while T has not expired. Once T has expired, PAN/Coordinators and PCM reset NBNj to zero and TNj to one. Figure 3.6 and Algorithm 2 summarise the proposed Adaptive GTS Allocation process.

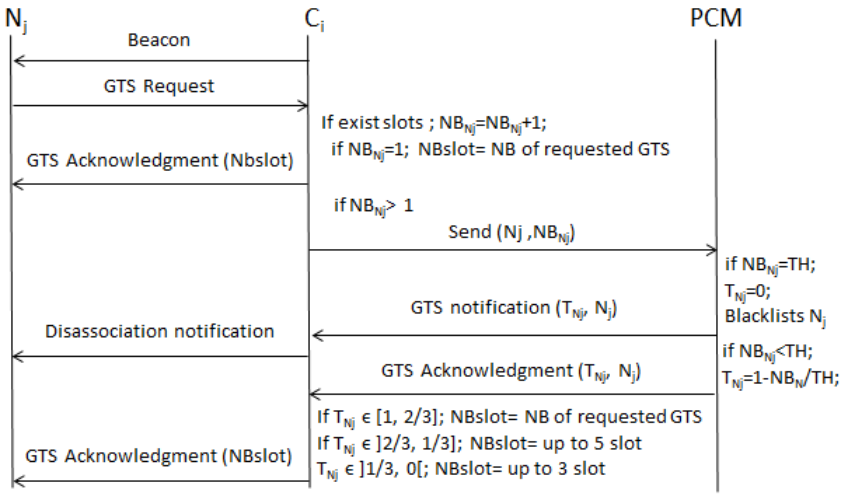


Figure 3.6: Trust-based GTS Allocation process.

3.4 Conclusion

In this chapter, we presented our contribution to the MAC layer. We introduced a trust-based defence and dynamic GTS allocation method to prevent and detect some MAC unfairness attacks in beacon-enabled IoT 802.15.4 networks. We proposed two algorithms to verify the association process and allocate GTS dynamically for real-time applications based on node trustworthiness. Also, we submitted a new central entity to IEEE 802.15.4 topology to act as a global neighbour discovery proxy. This new entity (PCM) caches all nodes' new identity and monitors local GTS allocation based on nodes' behaviour. This new approach can handle easily mobile nodes.

In the next chapter, we will present our contributions to the Network layer, precisely to handle and assess trust in the Routing Protocol for Low-Power and Lossy Networks (RPL). We will eventually evaluate and compare our models with other trust models to prove the effectiveness and security of our schemes and rank it compared to other trust models.

Algorithm 2 Trsut-based GTS Allocation Algorithm

Input: One PCM; a number of coordinators M; a number of nodes N; each node N_j is associated to only one coordinator $C_i \in M$; the trust value of each node $N_j \in N$ is set to $T_{Nj} = 1$; the number of request to GTS from each node N_j is set to $NB_{Nj} = 0$; TH= Threshold;(TH : a maximum number of requests to GTS);
While (T) **do**

N_j sends GTS Request to C_i;

If exist slots;

C_i calculates $NB_{Nj} = NB_{Nj} + 1$;

If $NB_{Nj} = 1$ **do**

NB_{slots}= NB.GTS.Req;

Sends GTS Acknowledgement (Nbslot) (NB_{slot});

END If

If $NB_{Nj} > 1$ **do**

Sends (N_j, NB_{Nj}) to PCM;

If NB_{Nj}= TH **do**

T_{Nj}=0;

Blacklists N_j;

Sends GTS notification (T_{Nj}, N_j) to C_i;

updates its list;

C_i sends an Disassociation notification to N_j;

END If

If $NB_{Nj} < TH$ **do**

T_{Nj}=1- NB_{Nj}/TH;

Sends GTS Acknowledgement (T_{Nj}, N_j) to C_i;

If $T_{Nj} \in [1, 2/3]$ **do** NB_{slot}= NB.GTS.Req;

If $T_{Nj} \in]2/3, 1/3]$ **do** NB_{slot} = up to 5 slot of NB.GTS.Req;

If $T_{Nj} \in]1/3, 0]$ **do** NB_{slot} = up to 3 slot of NB.GTS.Req;

Sends GTS Acknowledgement (NB_{slot});

END If

END If

END While

Chapter 4

Trust Management in Network Layer (RPL)

In this chapter, we present our contributions to the Network layer in chronological order. We focus in particular on the Routing Protocol for Low-Power and Lossy Networks (RPL). We start with our first proposition, entitled "Trust-based RPL for the Internet of Things" [166]. This new version of RPL aims to strengthen RPL security by adding a new trust metric based on nodes behaviours, called "RPL Node Trustworthiness". The trust metric is calculated by the collaboration of different neighbouring nodes in the network.

Then, we present the second contribution entitled, "New trust metric for the RPL routing protocol" [167]. In this contribution, we propose a new scheme for RPL named "Metric-based RPL Trustworthiness Scheme (MRTS)". MRTS is an enhancement of our first work, in which we deal with the trust inference problem. MRTS addresses trust issue during the construction and maintenance of routing paths from each node to the BR (Border Router). To handle this issue, we extend DIO (DODAG Information Object) message by introducing the trust-based metric ERNT (Extended RPL Node Trustworthiness) and the objective function TOF (Trust Objective Function).

Finally, we present our third contribution, entitled "Trust-aware and cooperative routing protocol for IoT security" [168]. This latter is a revision of our previous work [167] where new elements and components are added to enhance MRTS performance in terms of security, lifetime and routing Quality of Service (QoS). Besides, this work extends the work in [167] with a simulation validation and a mathematical analysis.

4.1 Trust-based RPL for the Internet of Things

The classical security provided by a Trusted Platform Module (TPM) [143] (see Chapter 2, Section 2.7.2.1) is not sufficient alone to manage trust in RPL, as it uses cryptography and authentication in only exchanging group keys used to secure control messages. To address the limitations of the TPM solution, a new RPL construction approach is introduced to strengthen trustworthiness between in-network nodes. This approach uses a new metric based on nodes' behaviours called "RPL Node Trustworthiness: RNT". The RNT trust metric allows a node to communi-

cate only with trusted nodes. Indeed, different neighbouring nodes collaborate in calculating RNT and thus contribute to the decision-making. In this solution, the TPM is integrated as a co-processor used to offload all security computations and processing. The proposed approach is based on two complementary points:

- It uses cryptography and authentication methods provided by TPM. These methods ensure the first line of trust between nodes to exchange control messages securely.
- It introduces nodes' behaviours to ensure that nodes participating in the construction and the maintenance of the RPL topology are still trustworthy.

4.1.1 Trusted Platform Module

The Trusted Platform Module "TPM" is a small, low cost, hardware security device, intended to perform critical security services for a client machine. This hardware component securely stores information, such as digital keys, certificates, and passwords. These pieces of information are used to authenticate the platform. Indeed, it solves messages signing, keys generation and storage problems, and can store platform measurements that ensure trustworthiness [169].

In the proposed trusted RPL, we use TPM as a co-processor embedded to each node within the network. Firstly, TPM is used to secure the control messages exchanged during RPL construction (i.e., authentication and cryptography). Before exchanging control messages, a trustworthiness relation is established between nodes, and then keying material exchange is done safely using RSA keys. Hence, TPM provides node authentication, as described in [143]. Secondly, TPM is used to offload all security computations and processing (i.e., cryptography, and RNT computations and storage).

4.1.2 Trust Metric Parameters

In this approach, a combination of three parameters is used to evaluate nodes' trustworthiness: honesty, energy, and unselfishness. The combination is flexible and adjustable as it allows to add or remove behavioural components specific to a given IoT application.

4.1.2.1 Honesty

The honesty parameter signals whether a node is malicious or not. Hence, the node i evaluates the node j behaviour to decide if the node j is compromised or not. To this end, some approaches use intrusion detection systems (IDS) based on a set of anomaly detection rules [170][171].

4.1.2.2 Energy

The energy of the node is a QoS trust component. It refers to the level of expectation of node i that the node j has sufficient energy to achieve its functionalities. The energy trust between node i and node j is the remaining energy (ER) percentage

of the node j estimated by the node i and vice versa. In IoT, the nodes consume mainly their energy while receiving and sending packets.

4.1.2.3 Selfishness

A selfish node is a node that intends to limit its resource expenditure while attempting to consume the resources of others. The selfishness of a node can be calculated as a distributed and collaborative score. By using techniques such as overhearing and snooping [172], the node i evaluates the node j during a period P and decides if the node j is selfish or not.

4.1.3 The RPL Node Trustworthiness (RNT) Metric

The RPL specification introduced a set of link and node routing metrics and constraints that can be used by RPL nodes in path calculation [62]. Nevertheless, it left open the way for the proposal of other metrics. Indeed, RPL has many serious vulnerabilities which can be exploited by attackers. To overcome these issues, we focus on trust when constructing the best path. Hence, the proposed trust routing metric will be carried within the DAG Metric Container (DAG-MC) object of the DIO message presented in Figure 4.1 [62] and evaluated as follows.

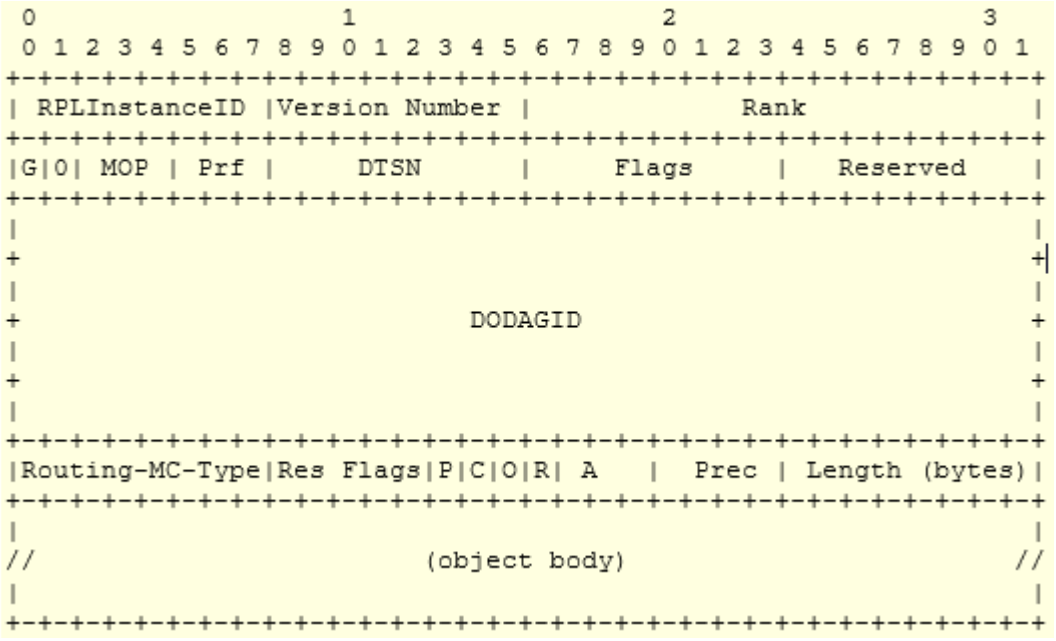


Figure 4.1: Example of DIO Message with a DAG Metric Container Option

4.1.3.1 DAG Metric Container

Multiple routing metrics and constraints could be carried within the DAG Metric Container option defined in [59]. A routing metric is a quantitative value that is used by the objective function to evaluate the path cost. Link and node metrics are usually, but not always, additive. The best path to the BR is the path that satisfies all supplied constraints and that has the lowest cost with respect to the specified metrics. In the presence of constraints, it is also called the shortest constrained

path. Routing metrics and constraints can appear in any order in the DAG-MC and have a common format as depicted in Figure 4.1.

4.1.3.2 RNT representation DIO message

We relied on the RFC [62] to define the RNT object. The RNT object is integrated within the object body of DAG-MC. It is used to provide information related to nodes trustworthiness and may be used as a recorded metric and a constraint. Figure 4.2 illustrates the generic format of an RNT object body. When RNT is used as a constraint, the BR indicates a trust's threshold that nodes must use to select their parents among their neighbours. In the case where RNT is used as a recorded metric, each node participating in the construction of RPL inserts RNT sub-objects reflecting its neighbours' trust values.

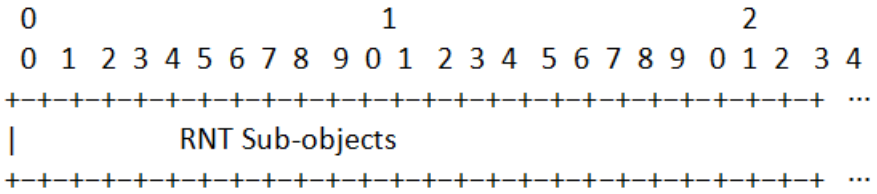


Figure 4.2: RNT Object Format

The format of an RNT Sub-Object is as depicted in Figure 4.3. The RNT Sub-Object's fields are defined as follows.

- Flags. The 5 Flags' bits remain unused. They must be initialised to zero by the sender and must be ignored by the receiver.
- T (1 bit). Flag of one bit indicating the node type. It is used only if the RNT object is used as a metric. When a node calculates the trustworthiness of its neighbours, it compares the values with the threshold set by the BR and then set the T flag as follows:

  T=1: Designates a trusted node when the trust value, noted C_T, is greater or equal to the threshold.

  T=0: designates an untrusted node when the trust value C_T is lower than the threshold.

- I (1 bit). The I flag is used only if the RNT object is used as a constraint.

  I=1: indicates that untrusted nodes can be included in the parents' list.

  I=0: indicates that untrusted nodes must be excluded from the parents' list.

- C_T (8 bits). When used as a constraint, this 8 bits field defines a threshold, which allows a node to decide if a neighbour can be trusted or not. Then it sets the T flag of this neighbour. When used as a metric, it represents the calculated trustworthiness of neighbour node indicated in the field "N".
- N (168 bits). The variable-length N field represents the identifier of the evaluated neighbour j or of the node i itself. It can be an IPv6 address or a TPM_ID.

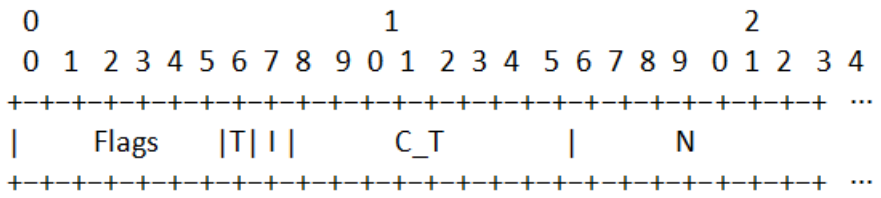


Figure 4.3: RNT Sub-Object Format

4.1.3.3 RNT evaluation

As presented in Chapter 2, several application-specific methods exist to quantify trust relationships between nodes, such as belief theory, Bayesian systems, Fuzzy logic, and weighted sum. Because RPL's nodes have limited storage and processing capacities, we chose the weighted sum method to evaluate nodes' trustworthiness. In this approach, we have taken as a foundation the work of Bao et al. [14], which is flexible and can be adjusted by adding or removing behavioural components specific for a given application. Hence, trust is calculated based on the collaboration of neighbouring nodes and using nodes behaviours components: selfishness, energy, and honesty. Besides, in this mechanism, a node's trust value is a combination of both direct observation and indirect recommendations.

4.1.3.3.1 Direct Trust Evaluation

Each node i evaluates the trust value of its one-hop neighbour node j at time t using Equation 4.1 [14]. The time t could correspond to several events, such as the sending of DIO message, a local repair or a global repair.

$$\begin{cases} T_{ij}(t) = w_1 T_{ij}^{honesty}(t) + w_2 T_{ij}^{energy}(t) + w_3 T_{ij}^{unselfishness}(t) \\ \mathrm{w_1 + w_2 + w_3 = 1} \end{cases} \tag{4.1}$$

$T_{ij}(t)$ takes values between 0 and 1 (i.e., $T_{ij}(t) \in [0, 1]$). 1 refers to complete trust, and 0 indicates no trust. w_1, w_2 and w_3 associated with the three trust components honesty, energy and unselfishness, respectively. These weights depend on the context and are defined by the network administrator. Each component $\mathrm{T^X} \in \{honesty, energy, unselfishness\}$ trust value is evaluated according to Equation 4.2 [14].

$$\mathrm{T_{ij}^X(t) = (1 - \alpha) T_{ij}^X(t - \Delta t) + \alpha T_{ij}^{X,direct}(t)} \tag{4.2}$$

Where Δt is the trust update interval corresponding to the DIO trickle timer; and $\alpha \in [0, 1]$ means that trust evaluation will rely more on direct or more on old observations. Nodes store their old observations in the TPM co-processor.

4.1.3.3.2 Indirect Trust Evaluation

After calculating the direct trust for each neighbour j, the node i uses the trust values received within the DIO messages (i.e., in the RNT objects) from its neighbours k (recommendations received from k recommenders) at time t to calculate

the final trust value of the node j, as in Equation 4.3. The final trust value is the average of the direct trust value calculated according to Equation 4.1, and all recommendations received for that neighbour j in RNT objects. The obtained result represents the final trust value for that neighbour, and it will be used to select the set of parents and the preferred parent.

$$\mathrm{T_{jFinal}} = \frac{\sum_{\mathrm{k=1}}^{\mathrm{m}} \mathrm{T_{kj}}}{\mathrm{m}} \tag{4.3}$$

$k \in \{i, (neighbours\ of\ i\ \cap neighbours\ of\ j)\}$ and m represents the number of the node from which node i received trust values for its neighbour j plus itself (i.e., the trust value calculated by the node i for the node j).

Using Equation 4.4, the node i calculates its new trust value. This latter is the average of all received trust values for itself. Where $k' \in \{neighbours\ of\ i\}$ and n represents the number of neighbour nodes from which node i received its trust values.

$$\mathrm{T_i} = \frac{\sum_{\mathrm{k'=1}}^{\mathrm{n}} \mathrm{T_{k'i}}}{\mathrm{n}} \tag{4.4}$$

4.1.4 Trust-based Objective Function (TOF)

The Trust-based Objective Function (TOF) defines how nodes use RNT metric and constraint to select parent and calculate Rank. During the construction of RPL topology according to TOF, the BR includes the RPL-Trustworthiness constraint in the DIO message. Furthermore, each node, including the BR includes a metric of the same type. This metric conveys the trust values of the direct nodes (i.e., neighbours). All the trust values of the node's neighbours are used to check whether or not the constraint is satisfied. In the following, we describe the process of the new OF.

4.1.4.1 Trust-based Path Cost Computation

To ensure routing security, it is desirable to avoid selecting untrusted nodes or nodes with low trust values as routers. Thus, the support for constraint-based routing is needed. In such cases, the routing protocol may compute a longer path to satisfy the security requirement of the network.

According to TOF, each node i computes the path cost in terms of trustworthiness metric from the node i to the BR through j for each reachable neighbour j. The path cost is the trust value T_{jFinal} calculated according to Equation 4.3. The path involving nodes with higher final trust values will be selected. However, the path involving untrusted nodes should be avoided. Consequently, the selected path can be the longest but remains the most secure.

4.1.4.2 Parent Selection

After computing the trust values for all the candidate neighbours, the node i selects a set of parents satisfying the constraint (i.e., nodes having trust values greater or equal to the threshold). After that, it chooses the parent having the greatest (i.e.,

best) trustworthiness as a preferred parent. If some potential parents have the same trustworthiness values, the node i will choose the one having the lowest Rank value as preferred parent.

4.1.4.3 Rank Calculation

The Rank should monotonically decrease when moving upward to the BR, and monotonically increase when moving down to the leaf nodes. As well, it should be bounded by MinHopRankIncrease and MaxHopRankIncrease [59]. Therefore, to comply with the Rank monotonic property, the BR sets its Rank to MinHopRankIncrease. Afterwards, each node N calculates its Rank *R(N)* as the sum of the Rank of its preferred parent (*R(P)*) and *rank_increase*. This latter is computed using Equation 4.5, where T_{jFinal} is the trust value of the selected parent.

$$\begin{cases} \mathrm{R(N)} = \mathrm{R(P)} + \mathrm{rank\_increase} \\ \mathrm{rank\_increase} = \mathrm{step} + \mathrm{MinHopRankIncrease} \\ \mathrm{step} = \mathrm{T_{jFinal}} * 100 \end{cases} \tag{4.5}$$

4.1.4.4 RNT Calculation

When constructing RPL, the BR declares itself as a Floating root (i.e., BR has no preferred parent) [59]. It broadcasts a first DIO message conveying the RNT threshold and its one-hop neighbours' RNT metric values. On receiving DIO messages, each neighbour i of the BR (1) selects the BR as a preferred parent, (2) calculates the rank according to Equation 4.5, (3) evaluates $\mathrm{T_{jFinal}}$ of the respective neighbours, and $\mathrm{T_i}$ of itself, and (4) broadcasts DIO message conveying the RNT objects. The process is repeated; each node i receiving DIO messages (1) evaluates $\mathrm{T_{jFinal}}$ of its neighbours j and $\mathrm{T_i}$ of itself, (2) selects the preferred parent having the greatest trustworthiness, (3) calculates the Rank (Equation 4.5), and (4) broadcasts DIO message conveying the RNT objects to its neighbours.

4.1.5 Illustrative example

This section presents an illustrative example that shows the process of our approach. We illustrate a network of 13 nodes as drawn in Figure 4.4 and a trust value threshold TH=0.5. Each node calculates the trust values of its neighbours according to Equation 4.1. Table 4.1 depicts the obtained values (T_{ij}), which represents the initial trust values. Table 4.2 shows the new trust values after several iterations and Figure 4.5 illustrates the network after the execution of the RPL construction algorithm based on RNT.

First Iteration: The BR broadcasts a first DIO message to its neighbours, which contains the trust values presented in Table 1. As the BR declares itself as a Floating root, nodes N1, N2, N3 and N4 select it as their parent. Then, they calculate the new trust values according to Equations 4.3 and 4.4.

Second Iteration: Nodes N1, N2, N3 and N4 broadcast their respective DIOs messages to their respective neighbours. Each DIO message conveys the respective

Table 4.1: The Calculated Trust Values

i \ j	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12
BR	0.6	0.8	0.5	0.7								
N1		0.4			0.5	0.7						
N2	0.6		0.5			0.7	0.6					
N3		0.6					0.7					
N4								0.8				
N5	0.5					0.7			0.6	0.4		
N6	0.7	0.5			0.4					0.7		
N7		0.7	0.6							0.6	0.6	
N8				0.5								0.6
N9					0.6							
N10					0.6	0.6	0.5					
N11							0.5					
N12								0.6				

Table 4.2: The New Trust Values from the Different Iterations

i \ j	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	iterations
N1	0.6	0.6			0.5	0.7							First
N2	0.6	0.6	0.5			0.7	0.6						
N3		0.7	0.5				0.7						
N4				0.7				0.8					
N5	0.55				0.45	0.7			0.6	0.4			Second
N6	0.63	0.56			0.45	0.7				0.7			
N7		0.66	0.53				0.65			0.6	0.6		
N8				0.6				0.8				0.6	
N9					0.52				0.6				Third
N10					0.5	0.66	0.57			0.66			
N11							0.57				0.6		
N12								0.7				0.6	

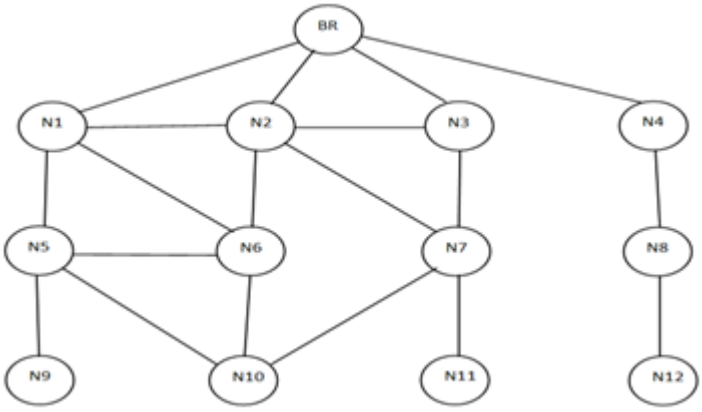


Figure 4.4: Network of 13 Nodes

trust values presented in Table 4.2. On receiving trust values in DIO messages, the nodes N5, N6, N7 and N8 calculate the new trust values according to Equations 4.3 and 4.4. They select their parents according to (Section 4.1.4.2) using TH=0.5 as a constraint.

- N5 receives a DIO message from N1. The trust value of N1 is greater than the threshold, hence N5 chooses N1 as a parent.
- N6 receives DIOs messages from N1 and N2. The trust values of N1 and N2 are greater than the threshold, so, N6 selects them as a set of parents. Then, it chooses N1 as a preferred parent because its trust value is greater than the one of N2.
- N7 receives DIOs messages from N2 and N3. As the trust values of N2 and N3 are greater than the threshold, N7 selects them as a set of parents. Then, it chooses N2 as a preferred parent because its trust value is the greatest.
- N8 receives a DIO message from N4. The trust value of N4 is greater than the threshold, then N8 selects N4 as a parent.

Third Iteration: Nodes N5, N6, N7 and N8 broadcast their DIOs messages to their neighbours. Each DIO message conveys the trust values presented in Table 4.2. Nodes N9, N10, N11 and N12 receive the trust values and calculate the new ones according to Equations 4.3 and 4.4. They select their parents according to (Section 4.1.4.2) using TH=0.5 as a constraint.

- N9 receives a DIO message from N5. The trust value of N5 is greater than the threshold, then N9 selects N5 as a parent.
- N10 receives DIOs messages from N5, N6 and N7. Because the trust values of N5, N6 and N7 are greater than the threshold, N10 chooses them as a set of parents. Then, it selects N6 as a preferred parent since its trust value is greater than the one of N5 and N7.
- N11 receives a DIO message from N7. The trust value of N7 is greater than the threshold, and thus N11 chooses N7 as a parent.

- N12 receives a DIO message from N8. The trust value of N8 is greater than the threshold, and hence N12 selects N8 as a parent.

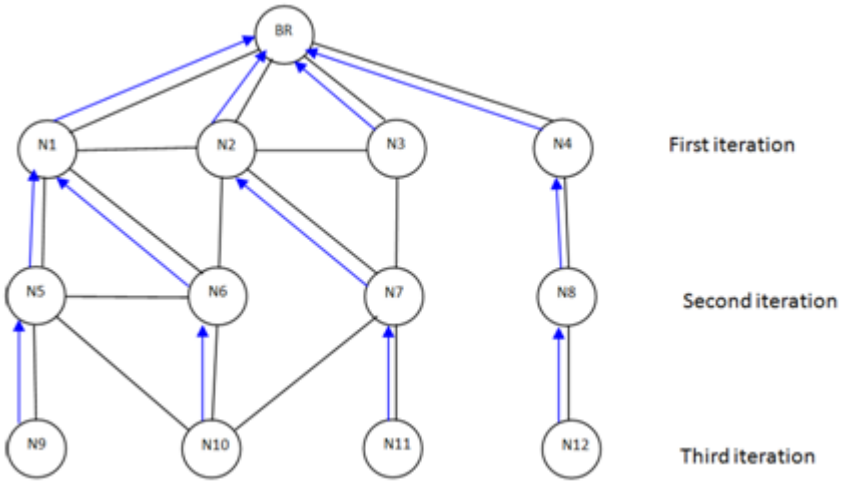


Figure 4.5: The network after RPL construction

4.1.6 Discussion

The new RPL uses the node's trust values as the main routing metric. The introduced trust metric is mainly based on the node's behaviour, which is important in establishing a reliable trust mechanism. Indeed, the new RPL allows the network self-organisation based on nodes trust status. Nevertheless, in the proposed solution, a node within the network selects a path according to its direct neighbours (i.e., a node selects the parent having the greatest trust value). This approach did not consider the trust value along the selected path (i.e., trust inference problem). The path may not be the most secure. To address this issue, we propose, in the next section, a new scheme for RPL named "Metric-based RPL Trustworthiness Scheme (MRTS)". The goal of MRTS is to enhance RPL security and deal with the trust inference problem. The MRTS addresses trust issue during the construction and maintenance of routing paths from each node to the BR (Border Router).

4.2 New Trust Metric for the RPL Routing Protocol

The present introduces a new scheme of RPL that supports the definition of collaborative security [173]. This scheme, named "Metric-based RPL Trustworthiness Scheme (MRTS)" uses collaborative trustworthiness evaluation between the network's different nodes. The MRTS enhances RPL routing security by calculating and choosing the most trusted path from the source node to the border router during the construction and maintenance of routing paths. To this end, MRTS extends the RNT metric defined in Section 4.1 to a new trust metric named "Extended RPL Node Trustworthiness (ERNT)" and redefined the trust-based objective function (TOF). ERNT represents each node's trust values within the network, and TOF

demonstrates how ERNT is mapped to path cost.

The following section, we will depict MRTS functioning. Similarly to previous approach (See Section 4.1), we propose to complement the MRTS scheme by built-in security in the nodes themselves. Therefore, we use a hardware security chip, named "Trust Platform Module (TPM)", as a co-processor embedded to each node within the network (see Section 4.1.1).

4.2.1 Trust Metric Parameters

Like the first contribution (Section 4.1.2), the MRTS also uses a combination of three parameters to evaluate nodes' trustworthiness: honesty, energy, and unselfishness.

4.2.2 Extended RPL Node Trustworthiness Metric

The Extended RPL Node Trustworthiness (ERNT) metric represents a quantitative and dynamic routing metric exchanged between nodes through DIO messages. It is used to evaluate each node's trustworthiness within the network and quantify the paths' costs.

4.2.2.1 ERNT representation in RPL DIO message

The DODAG Information Object (DIO) carries information on the metrics and the objective function to use while constructing RPL. To implement the MRTS scheme, we introduce an ERNT object in the DAG Metric Container [59] of the DIO message.

As depicted in Figure 4.6, the ERNT object contains several ERNT sub-objects. MRTS uses the ERNT object both as a constraint and as a recorded metric depending on the 'C' flag on the DAG Metric Container. The BR uses an ERNT sub-object as a constraint to indicate a Threshold. Nodes must use this Threshold to include or eliminate nodes that are not trusty. Likewise, the BR and other nodes use ERNT sub-object as a recorded metric. This latter represents a scalar, which determines the trustworthiness as well as the path cost. Every node participating in the construction of RPL inserts ERNT sub-objects (i.e., records). One of the ERNT sub-objects conveys the trust value of the node itself (i.e., T_i from Equation 4.7). The second one conveys the path cost through the preferred parent of the node itself. The others convey the trust values of the node's neighbours (i.e., T_{jFinal} from Equation 4.6). The format of the ERNT Sub-Object is as depicted in Figure. 4.6. In fact, unlike the RNT Sub-Object mentioned in Section 4.1.3, some fields have been modified.

- NID filed represents a node identifier. It is similar to the N field defined in Section 4.1.3.2.
- NT filed represents a node trustworthiness. It is similar to the C_T filed in Section 4.1.3.2.
- P is a 1-bit flag that indicates if the node NID is a preferred parent. If P is set to 1 (P =1) then the node NID is the preferred parent of the neighbour j. On

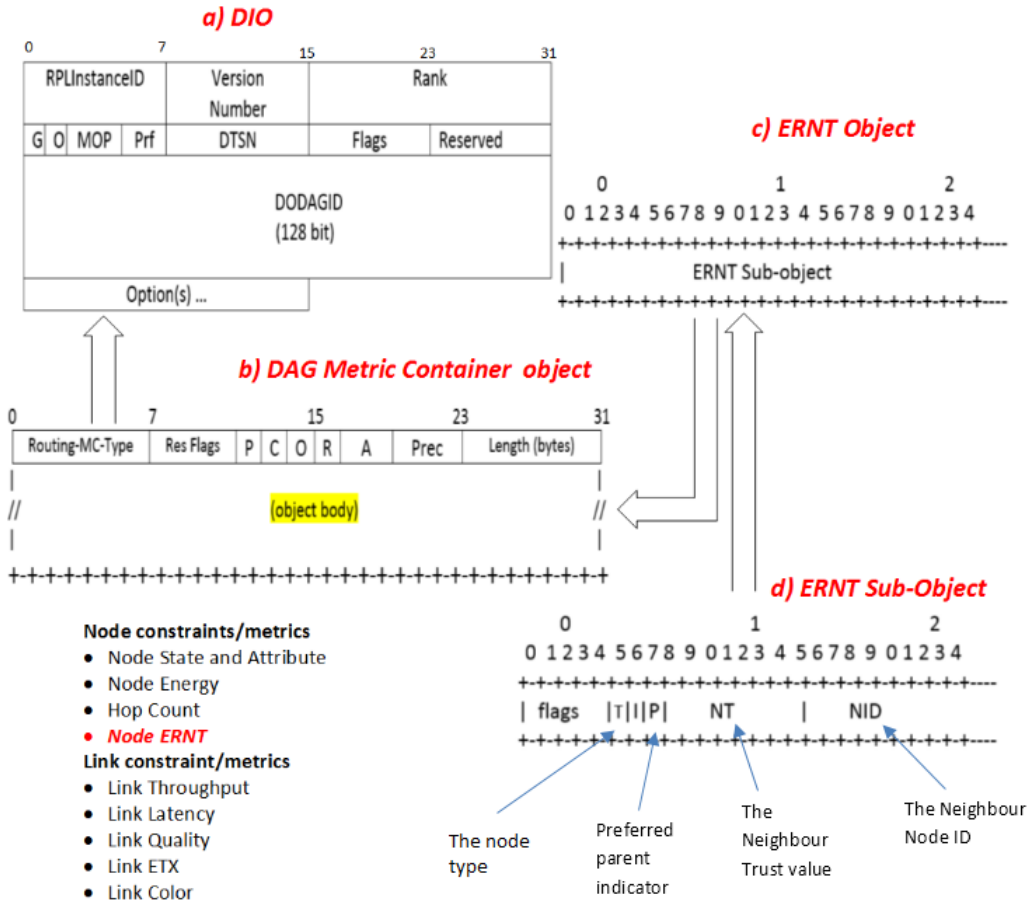


Figure 4.6: ERNT Object and ERNT Sub-objects within the DIO DAG-Metric-Container

the other hand, if P is set to 0 (P =0) then the node NID is not the preferred parent of the neighbour j.

- I is a 1-bit flag that is set by the BR to include or exclude nodes from the parent list as presented in Section 4.1.3.2.
- T is a 1-bit flag that indicates the status of a node (trusted or untrusted) according to its trust value as presented in Section 4.1.3.2.
- Flags are 5 unused bits as defined in Section 4.1.3.2.

4.2.2.2 ERNT Evaluation

The assessment of the nodes' trustworthiness according to the MRTS mechanism goes through two stages: direct observation and indirect recommendations.

4.2.2.2.1 Direct Trust Evaluation

Each node i evaluates the trust value, $T_{ij}(t)$, of its 1-hop neighbour node j at time t using Equation 4.1 defined in Section 4.1.3.3.1. Furthermore, each component $T_{ij}^{X}(t)$, $X \in \{honesty; energy; unselfishness\}$, is evaluated according to Equation 4.2 defined in Section 4.1.3.3.1.

4.2.2.2.2 Indirect Trust Evaluation

The node i computes T_{jFinal} for each neighbour node j according to Equation 4.6. Indeed, T_{jFinal} is the average of the direct trust value (i.e., $T_{ij}(t)$) calculated, as direct trust, using Equation 4.1, and all trust values (ERNTs) (i.e., T_{kj}) received for that neighbour j. T_{jFinal} is used to calculate paths' costs and select a set of parents and the preferred parent. If the node i receives recommendations for nodes that are not one-hop neighbours, it will ignore them.

$$T_{jFinal} = \frac{T_{ij}(t) + \sum_k T_{kj}}{m} \tag{4.6}$$

In Equation 4.6, k is the intersection between the neighbours' set of i and the neighbours' set of j (i.e., $k = N[i] \cap N[j]$), and m represents the number of nodes from which node i received trust values (ERNTs) for that neighbour j plus itself.

The node i calculates its trust value T_i using Equation 4.7. Unlike in Section 4.1.3.3, each node absolutely trusts itself, which means $T_{ii} = 1$. Consequently and according to Equation 4.7, T_i is calculated as the average of T_{ii} plus all received trust values (ERNTs) for the node i itself (i.e., $T_{k'i}$).

$$T_i = \frac{1 + \sum_{k'} T_{k'i}}{m'} \tag{4.7}$$

In Equation 4.7, k' is the set of node' i neighbours (i.e., $k' = N[i]$) and m' represents the number of neighbours of the node i plus the node i itself.

4.2.3 Trust Objective Function

The trust objective function (TOF) defines how nodes in MRTS use ERNT metric and constraint to select the preferred parent and to calculate Rank. Besides, TOF states how ERNT is transformed into path cost, and how this path cost is translated into node Rank. As it can be seen in next sections, TOF allows to find most trusted paths -paths with best trust values- and avoid untrusted paths -paths with untrusted nodes-.

4.2.3.1 MRTS Path Cost

To reach the destination (BR), each node i computes the Path Cost, $PC_{i,BR}$ (PC_i for short), through every reachable potential parent j. PC_i is a scalar value representing node's characteristics along the end-to-end path.

There exist several ways to compute the path cost using a trust metric [174][175][176]. It is known as a trust inference problem. One way is to select the strongest path, determined by the path with the highest minimum value, and take the lowest value on that path [175][176]. A second way is to choose the strongest path, determined by the path with the highest product of all values on the path, and take the product of all values on that path as path cost [174][176]. A third way is to select the strongest path, determined by the path with the weighted average of the trust values' minima along the disjoint paths [176]. Another solution could be to choose the strongest path, determined by the path with the highest average value, and take the average

value on that path as path cost.

To meet the MRTS routing requirements of consistency, optimality, and loop-freeness [177][176], we used the first solution to define the path cost, PCi, as the minimum of on-path nodes' trust values from the source node i to the destination BR, as in Equation 4.8. According to Equation 4.9, PC_i is calculated as the minimum value between the potential parent path cost PC_j and T_{jFinal} for that parent j, recursively. When the BR sets the flag I to 0, the topology formed by RPL must avoid all non-trusted nodes, and thus avoid paths with non-trusted nodes.

$$PC_i = \min_{j \in \{SOP\} \& T_{jFinal} \geq Threshold} T_{jFinal} \tag{4.8}$$

$$PC_i = \min_{j \in \{SOP\} \& T_{jFinal} \geq Threshold} PC_j, T_{jFinal} \tag{4.9}$$

4.2.3.2 MRTS Parent Selection

After the node i evaluated the trust values (ERNTs) for all candidate neighbours j, if the Threshold in the ERNT constraint object is not satisfied (i.e., the trust value is less than the Threshold), the advertising node will not be selected as a parent by the node processing the DIO message. If the constraint is satisfied, the processing node i adds the advertising node to its set of parents. Afterwards, the node evaluates the path cost through each potential parent according to Equation 4.9. Finally, it selects the parent who is in the path having the greatest path cost as preferred parent. The best path is the one with the highest minimum value, as outlined in Equation 4.10. As a result, among the candidate paths, the selected path can be the longest but remains the most secure (See Figure 3.5). If some candidate paths have the same path costs, then the processing node will select as a preferred parent the one having the lowest rank.

$$PC_i = \max \min_{j \in \{SOP\} \& T_{jFinal} \geq Threshold} PC_j, T_{jFinal} \tag{4.10}$$

In the following, we present two examples illustrating path cost calculation and parent selection. In the examples, we note paths from N4 to BR as P^1=<N4,N3,N1,BR> and P^2=<N4,N2,BR>. The node N4 receives DIO messages from nodes N3 and N2. It evaluates T_{N3}, T_{N2}, and calculates their respective paths' costs.

Example 1: Using Equation 4.9, $PC^1_{N4} = 0.6$ and $PC^2_{N4} = 0.5$.

According to Equation 4.10, $PC_{N4} = PC^1_{N4}$ ($PC^1_{N4} > PC^2_{N4}$), Consequently, N4 will choose P^1 for routing, as shown in Figure 4.7.

Example 2: Using Equation 4.9, $PC^1_{N4} = 0.6$ and $PC^2_{N4} = 0.7$.

According to Equation 4.10, $PC_{N4} = PC^2_{N4}$ ($PC^2_{N4} > PC^1_{N4}$), As a result, N4 will choose P^2 for routing, as shown in Figure 4.8.

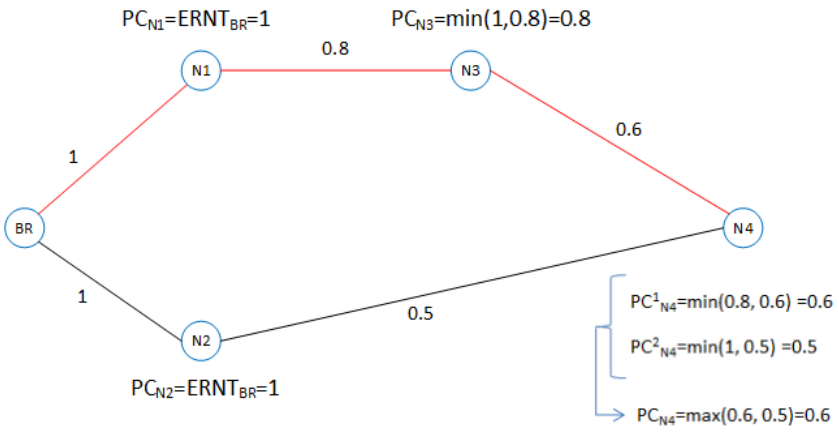


Figure 4.7: Node N4 Choosing the Longest But Most Trusted Path

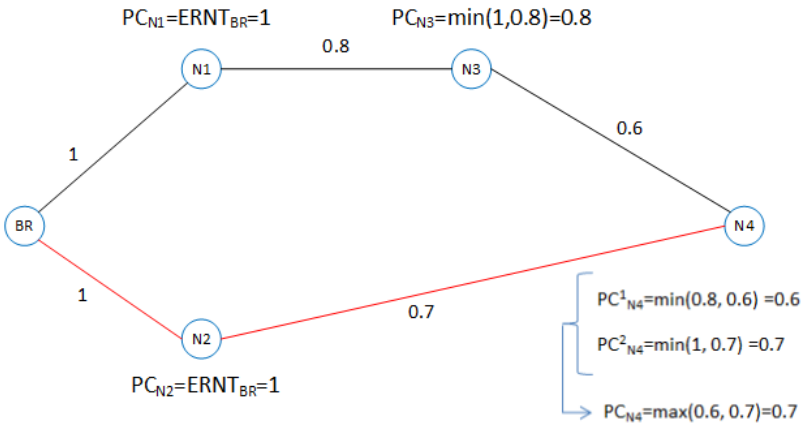


Figure 4.8: Node N4 choosing the Shortest and Most Trusted Path

4.2.3.3 MRTS Rank Calculation

MRTS differs from the secure-RPL scheme presented previously in the Rank calculation (Section 4.1.4.3). While the previous scheme uses Equation 4.5 and relies on the preferred parent's final trust value to evaluate the Rank, in MRTS, each node i calculates its Rank R(i) using Equation 4.11. In this equation, PC_i is the path cost through the preferred parent *PP*.

$$\begin{cases} R(i) = R(PP) + rank\_increase \\ rank\_increase = (1/PC_i) * 100 \end{cases} \tag{4.11}$$

4.2.3.4 MRTS Calculation

As it will be presented in follows, the previous secure-RPL scheme differs from MRTS in the trust metric calculation because the former relies on the preferred parent trust value, whereas the latter relies on the path cost.

When constructing RPL, the BR declares itself as a Floating root [59]. It broadcasts a first DIO message conveying the ERNT Threshold and the ERNT metric values of its one-hop neighbours. On receiving DIO messages, each neighbour i of

the BR (1) selects the BR as a preferred parent, (2) calculates the rank according to equation 4.11, (3) sets T_{BR} to 1 (i.e., the BR is fully trusted), (4) evaluates T_{jFinal} of the respective neighbours and T_i of itself, and (5) broadcasts a DIO message conveying the ERNT object. In this case, each neighbour i of the BR inserts an ERNT sub-object (as in Figure 4.6) with P=1, NT=1 and NID=BR, which means that the BR is the preferred parent and its trust value is equal to 1. This trust value represents the Path Cost PC_i (i.e., at the first stage, $PC_i = ERNT_{BR} = 1$).

The process is repeated until the construction of the whole topology as follows. Each node i receiving DIO messages (1) evaluates T_{jFinal} of its neighbours j and T_i of itself, (2) calculates PC_i through each potential parent j to the BR according to Equation 4.9, (3) selects the preferred parent having the best path cost according to Equation 4.10, (4) calculates the Rank using Equation 4.11, and (5) broadcasts a DIO message conveying the ERNT object. In this case, every node i inserts an ERNT sub-object with P=1 and NT= PC_i, which means that the node NID is the preferred parent of i and the path cost through it is equal to PC_i as in Equation 4.10.

4.2.4 MRTS Evaluation

RPL is a distance-vector routing protocol, and it uses the Bellman-Ford algorithm to calculate path cost [142]. In a weighted directed graph $G(V, E)$, this algorithm computes shortest paths from a source node to a destination node. It can handle graphs in which some of the edge weights are negative numbers.

To evaluate MRTS, we implemented a Linux-based paths simulator. In the first part of the simulator, we implemented the standard RPL using the ETX metric. In the second part, we implemented MRTS based on a distributed extended version of Bellman-Ford algorithm using ERNT metric -MRTS's Bellman-Ford algorithm- (See Algorithm 3).

In this evaluation, MRTS-based network is defined as a directed weighted graph $G(V, E)$, where V is the set of nodes and E is the set of edges representing links between neighbouring nodes. Each edge, $e = (i, j)$ is associated to a positive weight corresponding to T_{jFinal}, where $e \in E$, $i, j \in V$, and T_{jFinal} is the final trust evaluation of node i for its neighbour node j. Hence, as inputs to Algorithm 3, we have the network graph G, the function to calculate paths costs f, the source node s, and the destination BR.

We specify that in our evaluation we considered an initial representation of a network of 13 nodes represented in Figure 4.9 and Figure 4.10, where the BR is the only destination, and the trust Threshold is set to TH=0.5.

Figure 4.9 displays a network where values on the edges represent ETX-link values. While Figure 4.10 depicts a network where values on the edges represent mutual T_{ij} evaluations of neighbouring nodes (from Equation 4.1). It is obvious from the evaluations in Figure 4.10 that the node N1 is untrusted.

Algorithm 3 MRTS(G,f,s,BR)

Input: G, f, s, BR
Output: PC[], p[]
function BellmanFord(G(V, E), s, BR)

1. Step 1: Initialise graph
   foreach vertex $v \in V$ **do**

   PC[v]$\leftarrow$ inf

   p[v]$\leftarrow$NIL

   end foreach
   PC[s]$\leftarrow$0
   p[s]$\leftarrow$s

2. Step 2: Relax edges repeatedly
   for i **from** 0 **to** $|V| - 1$ **do**

   foreach $(u, v) \in E$ **with weight** $(1 - \mathrm{T}_{\mathrm{vFinal}})$ **do**

   if $((v \in Parent[u])$ **and** $(\mathrm{T}_{\mathrm{vFinal}} \geq Threshold))$ **do**

   if $(PC[v] > max(PC[u], (1 - \mathrm{T}_{\mathrm{vFinal}})))$ **do**

   $PC[v] \leftarrow max(PC[u], (1 - \mathrm{T}_{\mathrm{vFinal}}))$

   p[v] $\leftarrow$ p[u]$\oplus$(u,v)

   end if

   end if

   end foreach

   end for

Return PC[], p[]

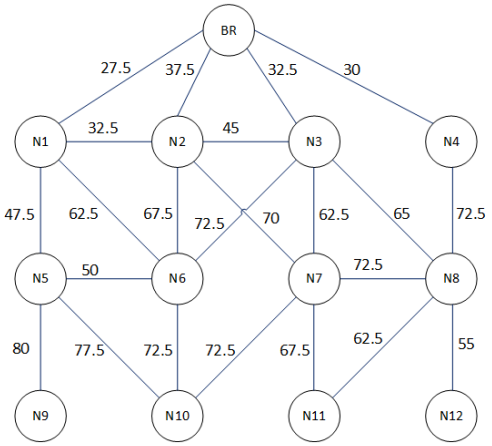


Figure 4.9: Network using *ETX* values

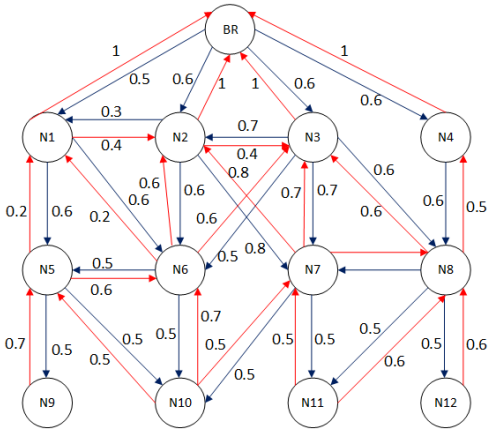


Figure 4.10: Network using T_{ij} values

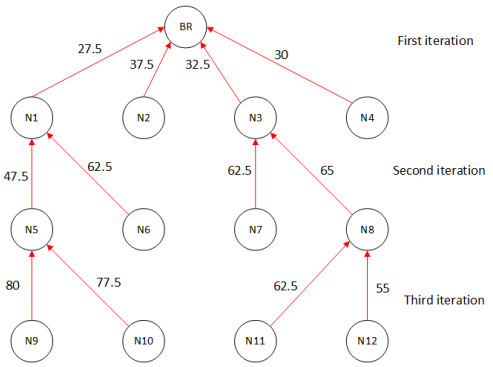


Figure 4.11: RPL topology using *ETX* values

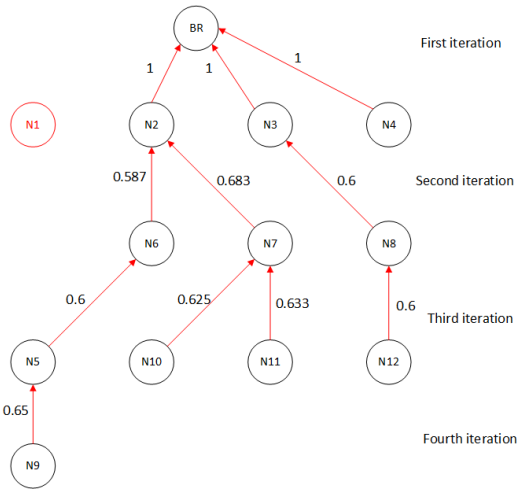


Figure 4.12: RPL topology using Trust values

Figure 4.11 represents the network topology of RPL using the Bellman-Ford algorithm with the ETX metric. Whereas, Figure 4.12 illustrates the network construction using ERNT and MRTS (Algorithm 3), where values on edges represent T_{jFinal} evaluations of neighbouring nodes using Equation 4.6.

Comparing Figure 4.12 and Figure 4.11, we notice clearly that the routing topology has completely changed where we have four levels instead of three, respectively. This is due to the fact N1 is untrusted. So, in the case of using ETX (i.e., standard RPL in Figure 4.11), each node selects the path with minimum total ETX and thus can forward packets through the non-trusted node N1. Consequently, in a network of 13 nodes, there are four nodes (i.e., N5, N6, N9, N10) that use the wrong paths, which represents a third of the network. However, when we use MRTS to construct the routing topology (see Figure 4.12), N1 is avoided, and the selected paths are more secure.

We extended our simulations to a network of 51 nodes. We notice that for standard RPL every time the number of malicious nodes increases, the probability of selecting routing paths including malicious nodes also increases. Nevertheless, in MRTS construction, every time the number of malicious nodes increases, the topology changes while avoiding malicious nodes. Hence, malicious nodes could not participate in the network operations and trigger attacks because they had already been avoided.

4.2.5 Discussion

In this part, we proposed a new trust management scheme to secure routing in RPL. This new trust-based RPL, namely, MRTS is based on a distributed and collaborative trust model, where nodes' behaviours are used to evaluate nodes' trust values. The trust value is named ERNT trust metric. After ERNT collaborative evaluation, MRTS considers only the trusted nodes in routing. The node's path cost in MRTS aids the routing discovery process to set up secure routing paths.

Compared with standard RPL, the MRTS's routing algorithm shows better performance in term of trustworthiness, according to results of the experiments. MRTS allows secure network self-organisation based on nodes trust status.

Our work is based on a distributed trust computation model. It relies on the collaboration between nodes to compute the trust and decide which forwarding path to select while handling trust inference problem. Furthermore, our model considers two trust metrics: QoS trust (Energy and Selfishness) and Social trust (honesty). Besides, the proposed scheme can deal easily with different attacks such as Self-promotion, Bad-mouthing, and Ballot-stuffing attacks. This is because each node uses several evaluation values received from different neighbouring nodes to calculate the trust value of a specific node. Hence, even if a node i transmits a bad or a good fake evaluation for another node j or itself, the values received from other neighbouring nodes will counter this evaluation (node's i evaluation).

Nevertheless, no simulation evaluates MRTS concerning energy consumption,

routing and security overheads. To address these issues, we propose in the next section our third contribution. This contribution extends our work with a simulation validation and an in-depth mathematical analysis.

4.3 Trust-aware and Cooperative Routing Protocol for IoT Security

We summarise our contributions as follows.

1. We present a revision of MRTS introduced in Section 4.2, considering all the modifications or additions. We added the ETX (Expected Transmission Count) metric as a new parameter for trust calculation. We changed the trust metric flag's functionality to make MRTS more flexible (i.e., secure and non-secure modes). We modified the parent selection process to extend nodes' energy.
2. We evaluate MRTS performance and give simulation results.
3. We provide a mathematical analysis of the MRTS routing and the Extended RPL Node Trustworthiness (ERNT) metric.
4. We perform a mathematical analysis and a simulation study of MRTS as a strategy for cooperation enforcement using game theory concepts.

4.3.1 Metric-based RPL Trustworthiness Scheme

One issue of MRTS as it is defined in Section 4.2 is that it relies only on node's metrics (i.e., energy, honesty and selfishness) to select the best path to route traffic. If nodes are honest and not selfish, the energy will be the primary metric to choose the parent. Consequently, some nodes along the selected trusted path will consume more energy than other nodes, resulting in unbalanced energy consumption. Furthermore, MRTS in our previous contribution does not consider link metrics, thus reducing the routing quality, such as the packet delivery ratio. Nevertheless, link metrics are essential to assess the reliability of the route. This routing property is vital for IoT applications since reliable routes provide a high delivery ratio. Researchers proposed several methods to estimate the reliability of a route, such as the Received Signal Strength (RSS), the Link Quality Level (LQL), and the expected number of retransmissions (ETX) [178].

The ETX metric estimates the average number of transmissions and retransmissions required to send a packet to a neighbour. ETX is one of the most widely used link metrics to enhance the RPL performance, where the lower the ETX, the better the quality of the link. In addition, there exist several lightweight implementations of ETX for the RPL routing protocol. Given the aforementioned arguments, and to balance the energy consumption of the nodes while promoting routes with higher packets delivery ratio, we extended MRTS with the ETX metric.

Besides, In the present work, we changed the use of the flag T (See Figure 4.13) to indicate the security status of the network. Thus, when T is set to 1, the active

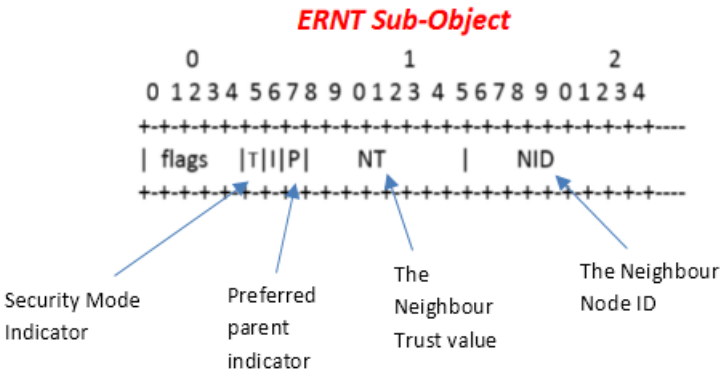


Figure 4.13: The New ERNT Sub-objects

mode is enabled, and the nodes perform the security check, whereas, when T is set to 0, the passive mode is enabled, and the nodes do not perform any security check.

4.3.2 Trust Metric Parameters

MRTS uses a combination of four parameters to evaluate nodes' trustworthiness: selfishness, honesty, ETX, and energy.

4.3.2.1 Energy

The energy trust between node i and node j is the remaining energy (ER) percentage of the node j estimated by the node i and vice versa. In IoT, the nodes consume mainly their energy while receiving and sending packets. There exist different approaches to calculate the energy. According to the energy model in [179], the energy consumed by a node i sending k bits data to the node j, denoted by E_{mt}, is calculated using Equation 4.12. E_{elec} is the electronics energy (i.e., the energy required for the transmitter as well as the receiver circuitry), E_{amp} is the energy dissipation for transmitting amplifier, and d is the distance from the node i to the node j. The energy consumed by the node j receiving the k bits data, denoted by E_{mr}, is calculated according to Equation 4.13. In RPL topology, every node communicates with its neighbours and sends data with the power level corresponding to the communication range of the node. Therefore, d is equal to the communication range.

$$E_{mt}(i) = k * (E_{elec} + E_{amp} * d^2) \tag{4.12}$$

$$E_{mr}(i) = k * E_{elec} \tag{4.13}$$

Initially, $ER(i)(t)$ is equal to the maximum energy E_{max}, which means at $t = 0, ER(i)(0) = E_{max}$. The energy spent by a node i is the sum of the energy consumed in message transmission, and the energy consumed in message reception. Thus, the node i calculates its remaining energy as shown in Equation 4.14.

$$ER(i) = ER(i) - (E_{mt}(i) + E_{mr}(i)) \tag{4.14}$$

Each node reports its residual energy to its neighbours periodically. The energy trust value $T_{ij}^{ER} \in [0, 1]$ is equal to the ratio $ER_{ij}(t)$ and E_{max} as shown in Equation 4.15, where $ER_{ij}(t) = \min(ER_{reported}(t), ER_{estimated}(t))$ and $ER_{estimated}(t) = ER(j)(t)$.

$$\mathrm{T}_{\mathrm{ij}}^{\mathrm{ER}}(\mathrm{t}) = \frac{\mathrm{ER}_{\mathrm{ij}}(\mathrm{t})}{\mathrm{E}_{\mathrm{max}}} \tag{4.15}$$

4.3.2.2 Selfishness

The selfishness of a node can be calculated as a distributed and collaborative score. By using techniques such as overhearing and snooping [172], the node i evaluates the node j during a period P and decides if the node j is selfish or not. Assume that an application requires minimum energy denoted by $\mathrm{E}_{\mathrm{min}}$. If ER(i)(t) is greater than $\mathrm{E}_{\mathrm{min}}$, the node i behaves correctly. In contrary, if ER(i)(t) is less or equal to $\mathrm{E}_{\mathrm{min}}$, it does not take part in forwarding packets any longer and uses, for example, its energy for transmissions of its packets, which implies it is more likely to become selfish. Therefore, during the trust calculation phase, MRTS allows some degree of selfishness for the nodes to save their resources. According to this approach, the nodes find a trade-off between energy and selfishness.

We consider the following two types of packets to calculate selfishness:

- **Control packets:** if the node drops control packets, it is considered as malicious even if it has low energy levels and its trust selfishness is set to 0. It is not tolerated to drop control packets since they are necessary for the maintenance of the routing topology.
- **Data packets:** we consider two cases.
  - ***Normal energy levels:*** if the node drops data packets and its remaining energy is greater than the required minimum energy for the application execution (i.e., $\mathrm{ER}_{\mathrm{ij}}(\mathrm{t}) > \mathrm{E}_{\mathrm{min}}$), the count of selfishness, denoted N, is incremented (i.e., $N = N + 1$). If N reaches the selfishness threshold, denoted $\mathrm{T}_{\mathrm{selfish}}$, the node is considered as selfish.
  - ***Low energy levels:*** if the node drops data packets because it has low energy levels, the count number of selfishness (i.e., N) is not incremented, which means the node is not considered as selfish.

The node's selfishness is calculated using Equation 4.16, where N is reset at the end of the period P.

$$T_{ij}^{Selfish,new}(t) = \begin{cases} 0 & \text{if } N(t) \geq T_{selfish} \\ 1 - (\frac{N(t)}{T_{selfish}}) & \text{else.} \end{cases} \tag{4.16}$$

4.3.2.3 Honesty

Some approaches use intrusion detection systems (IDS) based on a set of anomaly detection rules to estimate nodes' honesty [170][171]. In MRTS, each node i implements an IDS to monitor and detect malicious behaviours. If the IDS triggers an alert against a node j, the monitoring node i considers the node j dishonest and attributes to it an honesty-trust-value of 0, as in Equation 4.17.

$$T_{ij}^{Honesty,new}(t) = \begin{cases} 0 & \text{if node j misbehaves} \\ 1 & \text{else.} \end{cases} \tag{4.17}$$

4.3.2.4 ETX

ETX is a QoS trust component. *"The ETX of a path is the expected total number of packet transmissions (including retransmissions) required to successfully deliver a packet along that path"* [180]. It is a reliability metric used to enable routing protocols to find high-throughput routes, and thus to reduce energy consumption. To calculate $T_{ij}^{ETX}(t)$, ETX(t) is firstly normalised to $[0, 1]$ using the Min-Max-Normalisation method in Equation 4.18, where $ETX_{min} = 0$ and $ETX_{max} = 255$ as normalised in ContikiRPL implementation [181]. Then, Equation 4.19 is applied.

$$ETX(t) = \frac{ETX(t) - E_{min}}{E_{max} - E_{min}} = \frac{ETX(t)}{E_{max}} \tag{4.18}$$

$$T_{ij}^{ETX}(t) = 1 - ETX(t) \tag{4.19}$$

4.3.3 Trust Evaluation

4.3.3.1 Direct Trust

Direct trust is calculated, as depicted in Equation 4.20, where w_1, w_2, w_3 and w_4 are weights associated with honesty, selfishness, energy, and ETX parameters, respectively.

$$\begin{cases} T_{ij}^{Direct}(t) = w_1 T_{ij}^{Honesty}(t) + w_2 T_{ij}^{Selfish}(t) + w_3 T_{ij}^{ER}(t) + w_4 T_{ij}^{ETX}(t) \\ w_1 + w_2 + w_3 + w_4 = 1 \end{cases} \tag{4.20}$$

Equation 4.2 defined in Section 4.1.3.3.1, is used to evaluate behavioural parameter $X \in \{Honesty; Selfish\}$. Since the remaining energy reflects the ability of a node to achieve its functionalities and ETX reflects the status of the link, the trust calculation for both relies only on new observations, as presented in Section 4.3.2.1 and Section 4.3.2.4, respectively.

4.3.3.2 Indirect Trust

Indirect trust is evaluated using Equation 4.6 defined in Section 4.2.2.2.2.

4.3.4 Trust Propagation and Update

4.3.4.1 Trust Propagation

In MRTS, the nodes exchange, share and update trust pieces of information through the quantitative and dynamic RPL Node Trustworthiness metric; ERNT. The BR uses an ERNT sub-object as a constraint to indicate the trust threshold, noted T_{Trust}, that nodes must use to include or eliminate nodes that are not trustworthy. Besides, the BR and each node participating in the construction of RPL and following MRTS uses ERNT as a recorded metric, by inserting one ERNT sub-object (i.e., a record) for each calculated final trust value, as well as the path cost. Indeed, the path cost value represents the preferred parent's trust value.

4.3.4.2 Trust Update

MRTS updates trust values either periodically or reactively. The periodic trust update is time-driven, where MRTS uses the trickle timer for sending DIO messages as a regulator, while the reactive trust update is event-driven, where MRTS uses global repair and local repair events as triggers. In the proposed solution, when the IDS rises an alarm (i.e., it detects an attack) or if the $T_{Selfish}$ is reached, the local repair or global repair is triggered. Otherwise, the trickle timer regulates the update.

When a node i receives DIO messages from its neighbours, it uses the information conveyed in these DIO messages to update its routing table. It calculates the trust values of its neighbours using the direct assessments and recommendations received in DIO messages (according to Section 4.3.3). It then selects a set of trusted parents allowing it to reach the BR. It calculates the path cost through each potential parent and selects as a preferred parent the one with the highest path cost value (according to Section 4.3.5.1), which ensures the most trusted and reliable traffic routing to the BR. Finally, it generates and broadcasts a new DIO message containing the calculated trust values for each of its neighbours. All the neighbouring nodes repeat the process until the DODAG is reconstructed.

Once the construction is completed, the maintenance begins following the Trickle timer. The timer regulates the transmission rate of the control messages. In the stable state, the trust update interval of the trickle timer increases, and the transmission rate will be slowed, which signifies fewer control messages, and less computation (i.e., the network consumes less energy, memory and CPU). Otherwise, if there are inconsistencies (e.g., attack detection, selfish behaviour detection, and a new node joining the DODAG), which involve changes in the topology, the Trickle timer will be reset to a lower value, and the transmission rate will be fastened, which implies more control messages and more computation.

To reduce the computation cost in terms of energy consumption caused by trust update overheads, MRTS smooths out a small path cost (i.e., trust) increase or decrease. In the proposed solution, we consider a hysteresis threshold of 0.15 to avoid frequent parent changes, which helps maintain stability and conserve energy.

4.3.5 Attacker Isolation and Parent Selection

4.3.5.1 Parent Selection

TOF implements both node isolation and parent selection procedures. It is composed of two steps: the topology initialisation step (i.e., neighbour discovery) and the context-aware adaptive security execution step. The nodes execute the first step at deployment because they do not know their neighbours and thus could not evaluate their trustworthiness regarding honesty and selfishness. Since at deployment, all nodes have the same initial energy, the only parameter to use to construct the RPL topology is ETX along the path. We used ContikiRPL built-in function to calculate ETX. The preferred parent is the one with the minimum ETX value, where ETX is calculated as the sum of ETX along the path (i.e., from the parent node to the BR).

After the initialisation phase, every node knows its neighbours. If secure mode is not activated (i.e., T flag set to 0 in the ERNT sub-object), the only parameter to use is ETX as in the first step, and the nodes use TOF to find best paths by selecting parents with minimum ETX values. Otherwise, if the secure mode is activated, every node evaluates the path cost, selects a set of parents having trust value greater or equal to the threshold T_{Trust}, and selects its preferred parent. The way to calculate the path cost,to select a set of parents, and to select the preferred parent is already defined in Section 4.2.3

The node changes its current preferred parent with a new preferred parent only if the path cost through this new parent is higher than the currently selected parent by at least the hysteresis threshold of 0.15. In contrary to the previous definition in Section 4.2, if some candidate paths have the same path costs, the node i will choose as the preferred parent, the one having the higher remaining energy.

4.3.5.2 Attacker Isolation

Several methods exist to isolate untrusted nodes from participating in network operations. In MRTS, each node maintains a blacklist with the collaboration of the IDS. Once a node is classified as untrusted, it is added to that blacklist. As a result, normal nodes ignore all data and control packets coming from the blacklisted nodes and do not consider them, any more, in routing decision.

Algorithm 4 summarises the overall functioning of MRTS using ERNT, and Table 4.3 presents different notations used to describe MRTS.

4.3.6 MRTS Evaluation

In this section, we validate MRTS based on: i) a performance study via simulation, ii) a mathematical analysis of MRTS routing's requirements, and iii) a mathematical analysis of the ERNT routing metric.

4.3.6.1 Simulation Study

4.3.6.1.1 Simulation Settings

For our simulations, we used the lightweight and open-source Contiki 2.7/Cooja simulator [181]. We simulated a network of 30 nodes with one BR placed in the centre and 29 Sky mote (TelosB) sender nodes placed randomly around the BR. Each Sky mote is powered by an 8 MHz, 16-bit Texas Instruments MSP430 microcontroller with 10 kBytes of RAM and 48 kBytes of flash memory. Three of the 29 nodes are attackers planted randomly within the network that trigger Rank or Blackhole attacks. We executed the simulations ten times with three different topologies, and we averaged the outputs of the simulations.

We set the trust threshold T_{Trust} to 0.5 and α to 0.75. Because all four factors are equally important to select secure routes that respect good QoS, initially, we set the weights w_1, w_2, w_3 and w_4 equally to 0.25. Since in this study we focus on the security issues for RPL routing, during the simulation, if the IDS detects a node

Algorithm 4 MRTS Decision Process

Require: $NodesList, NeighboursList, T_{Trust}, T_{Selfish}$, $w_1, w_2, w_3, w_4, P, \alpha$
Ensure: $PreferredParent, Rank$
if $NeighboursList = \varnothing$ **then**
Construct the topology according to MRHOF-RPL in ContikiRPL implementation [181]
else

while 1 **do**
if $ERNT.T = 0$ (passive mode) **then**
Construct the topology according to MRHOF-RPL
else $\{ERNT.T = 1$ (active mode)$\}$

for all $j \in NeighbourList$ **do**
(Calculate Direct Trust)
Activate Promiscuous mode, watchdog mechanism, and IDS
$\mathrm{ER_{ij}(t)} \Leftarrow \min(\mathrm{ER_{reported}(t)}, \mathrm{ER_{estimated}(t)}$
$\mathrm{T_{ij}^{ER}(t)} \Leftarrow \frac{\mathrm{ER_{ij}(t)}}{\mathrm{E_{max}}}$

$\mathrm{ETX_j(t)} \Leftarrow \frac{\mathrm{ETX_j(t)}}{\mathrm{E_{max}}}$
$\mathrm{T_{ij}^{ETX}(t)} \Leftarrow 1 - \mathrm{ETX_j(t)}$
$\mathrm{T_{ij}^{Selfish,new}(t)} \Leftarrow 1 - (\frac{\mathrm{N(t)}}{\mathrm{T_{selfish}}})$
$\mathrm{T_{ij}^{Honesty,new}(t)} \Leftarrow \{0, 1\}$
Execute equation 4.20
Update Trust Table ($T_{ij}^{Honesty}(t), T_{ij}^{Selfish}(t), T_{ij}^{Direct}(t)$)
end for

for all $j \in NeighbourList$ **do**
(Calculate Indirect Trust using recommendations)
Execute equation 4.6, where $T_{kj}^{recom}(t) = ERNT.NT$

Update Trust Table ($T_{ij}(t)$)
Update ParentList ($T_{ij}(t) \geq T_{Trust}$)
end for
From ParentList, Select $T_{ij}(t)$ with greater PC_i
Update Rank if ($PC_i - PC_{Actual-Parent} > 0.15$)
Build DIO with calculated values and forward
end if
end while
end if
return $PreferredParent$, $Rank$

Table 4.3: Terminology

Notation	Description
MRTS	Metric-based RPL Trustworthiness Scheme
ERNT	Extended RPL Node Trustworthiness: trust object that conveys trust values and related information, where T field for setting the security mode, P flag to indicate the parent status (path cost), NT flag to indicate the trust value, and NID to indicate the node identifier
TOF	Trust Objective Function
ETX	Expected Transmission Count
ER	Remaining Energy
$\mathbf{T_{ij}}^{Direct}(t)$	Measures Direct Trust of node i towards node j at time t
$\mathbf{T_{kj}}^{Recom}(t)$	Recommendation of node k towards node j at time t received in ERNT objects
$\mathbf{T_{ij}}(t)$	Measures Trust of node i towards node j at time t using direct trust and recommendations
$\mathbf{T_{ij}}^{X}(t)$	Measures Trust of node i towards node j at time t for the component $X \in \{honesty, selfish, energy, ETX\}$. These correspond to $\mathbf{T_{ij}}^{Honesty}(t)$, $\mathbf{T_{ij}}^{Selfish}(t)$, $\mathbf{T_{ij}}^{ER}(t)$, and $\mathbf{T_{ij}}^{ETX}(t)$
$\mathbf{w_1}$, $\mathbf{w_2}$, $\mathbf{w_3}$, $\mathbf{w_4}$	weights associated to honesty, selfishness, energy, and ETX parameters, respectively
$\mathbf{ER_{ij}}(t)$	Remaining Energy assessment of node i toward node j at time t
$\mathbf{T_{Selfish}}$	Selfishness threshold. The number of time a node is allowed to not forwarding other nodes packets
$\mathbf{T_{Trust}}$	Trust threshold
P	Monitoring period for selfishness assessment
$\mathbf{PC_i}$	Measures the path cost of the node i. This corresponds to the minimum of on-path nodes' trust values from the source node i to the destination BR
SOP	Set Of Parent
MRHOF-RPL	The Minimum Rank with Hysteresis Objective Function. The objective function that selects routes that minimize ETX

as malicious, the normal nodes will adjust the weights associated to the malicious node by setting w_1 to 1, and w_2, w_3 and w_4 equally to 0. Likewise, if a node detects another node as selfish, the normal node will adjust the weights associated with the selfish node by setting w_2 to 1, and w_1, w_3 and w_4 equally to 0. Table 4.4 shows the simulation parameters.

We used both time-driven and event-driven update mechanisms. The computation process is triggered according to the trickle timer (time-driven) and if the IDS sends an alert or if the $T_{Selfish}$ is reached (event-driven).

We studied the MRTS performance and compared it to MRHOF-RPL and

SecTrust-RPL (SecTrust for short). We calculated the average packet delivery ratio (APDR) (%), the average energy consumption (AEC), the average rank changes (ARC), and the throughput. APDR corresponds to the ratio of the packets delivered to the total packets sent. AEC corresponds to the average energy consumption by all nodes within the network. ARC is the average number of parent switches. Finally, the throughput is calculated as the product of the average number of delivered packets for all simulations' topologies, the size of the packets and the integer 8 (i.e., to convert bytes into bits), divided by the total simulation's time of 3600 seconds.

Table 4.4: Simulation Parameters

Parameter	Value
Simulator	Cooja-Contiki 2.7
Simulation time	1h
Number of nodes	30
Network area	100m*100m
Range of nodes	RX:50%, TX:50m, interference:60m
Radio medium model	UDGM: Distance Loss
Traffic rate	1 packet sent every 10 seconds
Number of attacker nodes	3
Attacks	Rank/Blackhole
w_1, w_2, w_3 and w_4	0.25
T_{Trust}	0.5
α	0.75

4.3.6.1.2 Simulation Results

Rank Changes: Figure 4.14 shows the average rank changes rate for MRHOF-RPL, SecTrust, and MRTS under Rank and Blackhole attacks. As the simulation progresses the average frequency of rank changes for MRHOF-RPL under both attacks is high. For instance, under the Blackhole attack, the results show 300 rank changes in the first 30 minutes and up to 460 rank changes at the end of the simulation. Under the Rank attack, 120 rank changes the first 30 minutes and up to 380 rank changes after 60 minutes. To control the topology instability caused by both attacks, the nodes go through a high rate of parent changes inducing a high rate of rank changes.

Even though SecTrust shows a significant improvement regarding the network stability over MRHOF-RPL, MRTS shows better results leading to more stability. Under the Blackhole attack, MRTS showed 60 rank changes in the first 30 minutes and up to 80 rank changes at the end of the simulation. Under Rank attack, MRTS showed 50 rank changes in the first 30 minutes and up to 40 rank changes at the end of the simulation. MRTS performs better than SecTrust because nodes collaborate to detect and isolate attackers quickly, which helps to maintain the stability of the network.

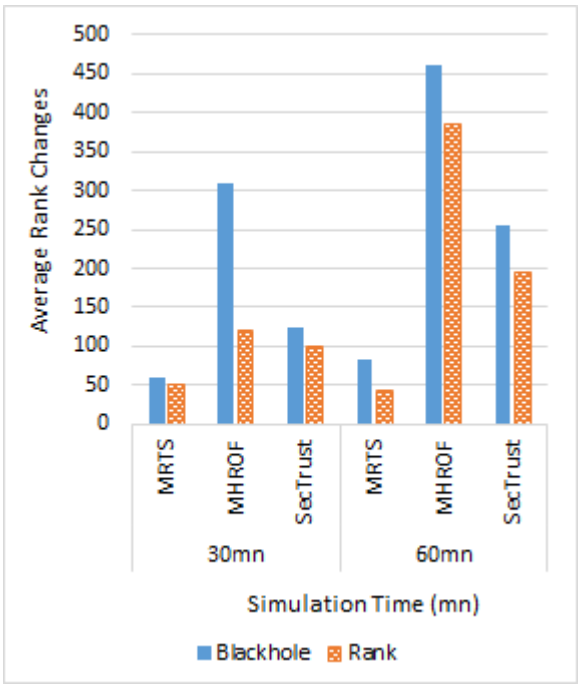


Figure 4.14: Comparison of the Average Node Rank Changes under Blackhole and Rank Attacks in MRTS, MRHOF-RPL, and SecTrust Simulations.

Packet Delivery Ratio: In addition to network congestion and packet collision, it can be observed from Figure 4.15 that the effects of Blackhole and Rank attacks on packet delivery ratio for MRHOF-RPL are disastrous (25-40%). Several causes can explain the observations. For instance, when a normal node selects a malicious node as a preferred parent to forward its packets, the latter may delete control packets making the topology unstable and unavailable. In contrary, MRTS maintained the packet delivery ratio quite high (up to 90%) since it uses IDS to detect attacks and provides a new routing scheme to isolate malicious nodes and maintain a secure topology. As a result, attacks on MRHOF-RPL cause significant damages compared to MRTS. Besides, MRTS shows better packet delivery ratio compared to SecTrust. Indeed, since MRTS reduces the rank changes rate, it provides a more stable network over SecTrust, and consequently, reduces packet loss.

Energy Consumption: In MRHOF-RPL network, some nodes consume more energy than others do because they tend more often to be selected as a preferred parent relying on their ETX; this is an issue since the higher energy cost of the chosen parents affects the entire network's lifetime. As depicted in Figure 4.14 and Figure 4.16, when MRHOF-RPL network is under attacks, nodes consume more energy because of the topology instability and rank changes rate (i.e., due to parent changes). We explain the instability of the network by the fact that MRHOF-RPL does not have any mechanism to handle attacks. From Figure 4.16, we notice that during the first 20-30 minutes, MRHOF-RPL and SecTrust consumed lower energy than MRTS. After some time, MRTS performed better because energy consumption is much more balanced among different nodes. The performance of MRTS in terms of energy consumption is because MRTS takes into account the remaining energy for each node in routing decision. Indeed, under attacks, MRTS consumes the most

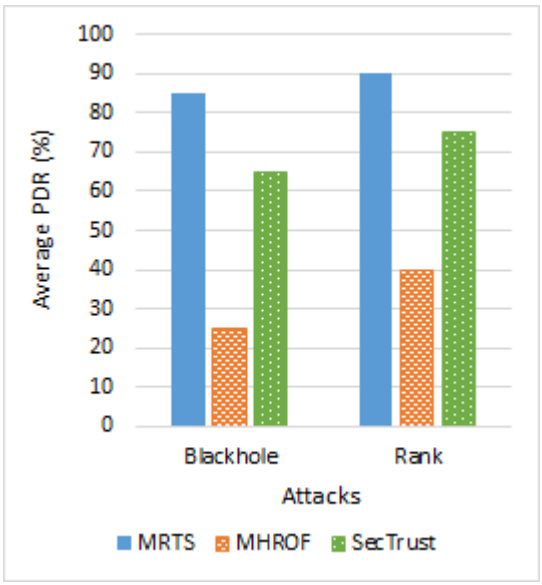


Figure 4.15: Comparison of the Average Packet Delivery Ratio Measurements Between MRTS, MRHOF-RPL, and SecTrust under Blackhole and Rank Attacks.

energy in calculation and DIO transmissions, but once the malicious nodes are detected and isolated, the topology becomes more stable, and the energy consumption rate decreases. Furthermore, as already stated, if two candidate parents have the same trust value, the node selects the one having the highest remaining energy.

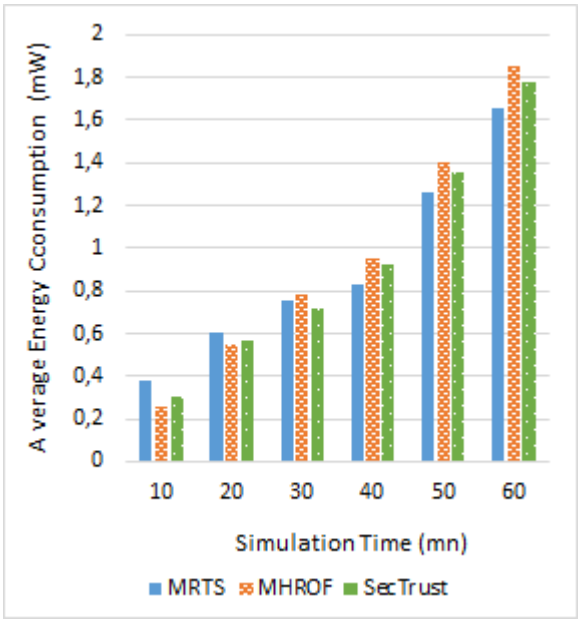


Figure 4.16: Comparison of the Average Energy Consumption over Time Between MRTS, MRHOF-RPL, and SecTrust.

Throughput: Figure 4.17 shows that the throughput for MRHOF-RPL under both Blackhole and Rank attacks is largely decreased compared to SecTrust and MRTS. In fact, in MRHOF-RPL, the throughput of nodes that are a child of parents that triggered Blackhole or Rank attacks is equal to zero since their packets never reach

their destination (i.e., the border router). Contrarily, in SecTrust and MRTS, the attacks are detected and malicious nodes are isolated. As a consequence, all nodes' throughput is always higher than zero, which implies that the overall throughput increases. Similarly to the packet delivery ratio, the throughput in MRTS is better compared to SecTrust because MRTS provides more stable network over SecTrust, thus reducing packet loss and increasing throughput.

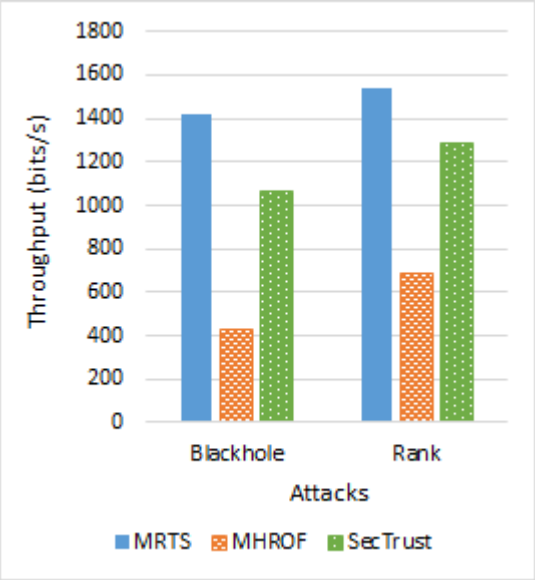


Figure 4.17: Comparison of the Average Throughput Measurements Between MRTS, MRHOF-RPL, and SecTrust under Blackhole and Rank Attacks during 3600 Second Simulations Time.

4.3.6.2 MRTS Requirements Analysis

In this section, we validate MRTS relying on the study of Yang et al. [177]. According to the researchers, a routing protocol needs three requirements to operate properly: consistency, optimality and loop-freeness. Besides, a routing protocol consists of two components: a path calculation algorithm and a packet-forwarding scheme.

To forward packets to the border router, RPL can use either hop-by-hop forwarding scheme or source routing. Furthermore, since RPL is a distance-vector routing protocol, it uses the Bellman-Ford algorithm to calculate path cost [16]. In a weighted directed graph, this algorithm computes the shortest paths from a source node to a destination node. In following, we demonstrate that MRTS combining ERNT, MRTS's Bellman-Ford algorithm [167], and either hop-by-hop forwarding scheme or source-routing meets the three routing requirements.

4.3.6.2.1 Network Model

MRTS based network is defined as a directed weighted graph $G = (V, E)$, where V is the set of nodes and E is the set of edges representing links between neighbouring nodes. Each edge, $e = (i, j)$ is associated to a positive weight corresponding to T_{ij},

where $e \in E$, $i, j \in V$, and T_{ij} is the trust evaluation of node i for its neighbour node j. Traffic within the network is multipoint-to-point, where all source nodes send data to a single destination node (BR). Thus, traffic is transmitted upward from source nodes u_n to a destination u_1, where u_1 is the BR. A path p from u_n to u_1 is denoted as $p < u_n, u_1 >= (u_n, u_{n-1}, ..., u_1)$. The expression, $next(p, u_k)$, denotes the next-hop of u_k on $p < u_n, u_1 >$, where $k = n, n-1, ..., 1$. In MRTS, $next(p, u_k)$ represents the preferred parent (i.e., 1-hop: PP_k) that u_k stores in its routing table and uses to forward traffic (i.e., $next(p, u_k) = PP_k$). The path calculation algorithm of MRTS routing protocol (MRTS's Bellman-Ford algorithm (see Section 4.2.4 Algorithm 3)) is represented by a function MRTS(G,f,s,BR) that returns a path $p < s, BR >$ from the source node s to the BR. In addition, we denote $SP(p < u_n, u_1 >, v, u_1)$ the sub-path of path $p < u_n, u_1 >$ between on-path nodes v and u_1. In our network model, we consider two kinds of nodes: trusted and untrusted nodes. The first category represents legitimate nodes, with trust values $T_{ij} \geq T_{Trust}$. The second one represents nodes with trust values $T_{ij} < T_{Trust}$.

4.3.6.2.2 Consistency

According to [177], a routing protocol is consistent if whatever a node along a given path, the packet forwarding decision is consistent with each node in that path. In other words, if a node u_n decides to forward packets through the path $p < u_n, u_1 >= (u_n, u_{n-1}, ..., u_1)$, other on-path-$p < u_n, u_1 >$ nodes should forward packets through p-subpaths. It seems obvious that MRTS is consistent. For source routing, on-path nodes forward packets relying on packets headers, and thus, routing consistency is systematically satisfied. For hop-by-hop routing, because in MRTS the routing tree is constructed from the BR to leaf nodes, each intermediate node is, in fact, a preferred parent of each sending node along a given path. Hence, if a node selects a preferred parent as its next-hop ($next(p, u_n) = u_{n-1}$), it selects automatically the sub-path $SP(p < u_n, u_1 >, u_{n-1}, u_1) = p < u_{n-1}, u_1 >= (u_{n-1}, ..., u_1)$ through that parent u_{n-1} to forward packets. The process is recursive for each node on the sub-path.

4.3.6.2.3 Optimality

A routing protocol is optimal if it routes the traffic along the best path for every pair of nodes within the network [177]. MRTS uses *MRTS(G,f,s,BR)* path calculation algorithm allowing a source node s to calculate path cost, denoted PC^x, for each path $p^x < s, BR >$ to the BR, where x is the number of paths. For instance, if we consider three possible paths p^1, p^2, and p^3, with $PC^1 = 0.5$, $PC^2 = 0.8$, and $PC^3 = 0.7$ then, according to Section 4.2.3, s will select p^2 as the best path to route the traffic.

4.3.6.2.4 Loop-freeness

A routing protocol is said to be loop-free if it does not create any packet forwarding loop. Besides, if the routing protocol uses hop-by-hop routing and Bellman-Ford algorithm to calculate valid paths, consistency is sufficient to ensure loop-freeness

[177]. According to our analysis, MRTS is consistent, and consequently, it is loop-free. From the other hand, after the selection of the best path, the source node calculates its rank using the rank of its selected parent. If a node receives a DIO message from another node having a rank value greater than its rank value, the receiving node will discard the DIO message, and thus, will never select a child as a parent; therefore, loops will be avoided. In the case of source-routing, a source node can eliminate loops.

4.3.6.3 ERNT Mathematical Model

To ensure the proper operation of a given routing protocol (i.e., consistency, optimality and loop-freeness), authors in [177] identify the properties that a routing metric should have: isotonicity and monotonicity. The monotonic property could ensure the convergence of the routing algorithm, while the isotonicity property essentially affects the order of the paths weights and could ensure their convergence is optimal for distance vector protocols like RPL. As explained in [177][175][176], we represent a routing metric as an algebra on top of a quadruplet $(S, \oplus, f, \leq)$, where S is the set of all paths, f is a function that maps a path to a path cost, $\leq$ represents a total order of costs, and $\oplus$ is the path concatenation operation. According to the proved Lemma 2 in [177] *"if for every source $s \in V$, destination $d \in V$ the path weight structure $(S, \oplus, f, \leq)$ is left-isotonic and left-monotonic, there exists a lightest path from s to d such that all its sub-paths with source s are also lightest paths. Such a lightest path is called a D-lightest path".*

For ERNT metric, the algebraic path weight structure is $(S, \oplus, f(max), \leq)$ where $f(max) = max(p)$ is the maximum trust value $(1 - \mathrm{T_{ij}}(t))$ along with the path p. The order relation for ERNT is $\leq$, to minimise the trust of different paths p^x. Based upon the proved Theorems 7, 8 and 9 in [177], and given that MRTS uses MRTS's Bellman-Ford algorithm for path calculation (see Section 4.2.4 Algorithm 3), it is enough to demonstrate that ERNT is left isotonic and left monotonic for calculating lightest paths. In the case of source-routing, it is enough to demonstrate that ERNT is left isotonic.

4.3.6.3.1 Isotonicity

A routing metric is left isotonic if the order relation between two paths is preserved if a common third path prefixes them. Formally, the algebraic structure $(S, \oplus, f, \leq)$ is left isotonic if $\forall a, b, c \in S, f(a) \leq f(b) \Rightarrow f(c \oplus a) \leq f(c \oplus b)$. This means, $(ERNT, \oplus, f(max), \leq)$ is left isotonic if $\forall a, b, c \in S, max(a) \leq max(b) \Rightarrow max(c \oplus a) \leq max(c \oplus b)$.

Let us consider two given paths a and b with $max(a) \leq max(b)$. For a given prefixed path c, there exist three cases:

1. If $max(c) \leq max(a) \leq max(b)$:
$$\left.\begin{array}{l} max(c \oplus a) = max(a) \\ max(c \oplus b) = max(b) \end{array}\right\} \Longrightarrow max(c \oplus a) \leq max(c \oplus b)$$

2. If $max(a) \leq max(c) \leq max(b)$:
$$\left.\begin{array}{l} max(c \oplus a) = max(c) \\ max(c \oplus b) = max(b) \end{array}\right\} \Longrightarrow max(c \oplus a) \leq max(c \oplus b)$$

3. If $max(a) \leq max(b) \leq max(c)$:
$$\left.\begin{array}{l} max(c \oplus a) = max(c) \\ max(c \oplus b) = max(c) \\ max(c \oplus a) = max(c \oplus b) \end{array}\right\} \Longrightarrow max(c \oplus a) \leq max(c \oplus b)$$

4.3.6.3.2 Monotonicity

A routing metric is left monotonic if the path cost does not decrease when prefixed by another path. Formally, $(S, \oplus, f, \leq)$ is left monotonic if $\forall a, c \in S, f(a) \leq f(c \oplus a)$. This means, $(ERNT, \oplus, f(max), \leq)$ is left monotonic if $\forall a, c \in S, max(a) \leq max(c \oplus a)$.

For a given prefixed path c, there exist two cases:

1. If $max(a) \leq max(c)$:
$$\left.\begin{array}{l} max(c \oplus a) = max(c) \\ max(a) \leq max(c) \end{array}\right\} \Longrightarrow max(a) \leq max(c \oplus a)$$

2. If $max(c) \leq max(a)$:
$$\left.\begin{array}{l} max(c \oplus a) = max(a) \end{array}\right\} \Longrightarrow max(a) \leq max(c \oplus a)$$

4.3.6.4 Discussion

Our solution is flexible as it depends on the context according to (1) weights associated with trust calculation, (2) the ERNT metric object flags, and (3) trust-related thresholds.

The nodes executing MRTS can dynamically modify the weights, w_i ($i \in \{1, 2, 3, 4\}$), according to rules specified by the context. Hence, our solution allows a trade-off between security effectiveness (i.e., honesty and selfishness) and QoS requirements (i.e., energy efficiency and ETX) depending on the weights' dynamic changes defined by every IoT application. For instance, in our experimental study, we set the weights initially to an equal value, and then we adjusted them according to security breach detection (i.e., honesty and selfishness). In another case, w_1 and w_2 could be greater than w_3 and w_4. If the energy-saving is the most important, w_3 could have the greatest value. In another case, an application could switch to the active (i.e., secure) mode and set w_1 and w_2 to greater values if the energy state is greater than a threshold (E_{min}), and switch to the passive (i.e., non-secure) mode if the energy state is less than a threshold.

Even though some degree of selfishness is allowed for nodes to save their resources, in the present study, we do not consider reintegration of isolated nodes once classified as untrusted. In some cases, the reintegration and backup of untrusted nodes can be required (e.g., to support fault tolerance). Nevertheless, considering a reintegration mechanism raises new problems regarding the effectiveness of the

solution (e.g., how to prevent nodes from abusing the reintegration mechanism). In this context, if the BR sets the flag I in the ERNT object to 1, the parent selection process allows the selection of untrusted nodes in the parents' set, whereas, if the flag I is set to 0, the parent selection process does not allow the insertion of untrusted nodes in the parents' set. This flexibility can be seen as a reintegration mechanism, in which the border router switches the flag I between 1 and 0 according to the necessity of the application context (e.g., fault tolerance). Nonetheless, rules need to be designed to handle this reintegration carefully.

In the simulation study, we noticed good results regarding energy consumption. However, our solution could not always get the right balance between energy-efficiency and security issues. $T_{selfish}$ and T_{Trust} threshold parameters' setting is a trade-off issue, and no value can fit all scenarios and criteria. For instance, in a scenario in which security is of great concern, T_{Trust} value needs to be as high as possible while $T_{selfish}$ value needs to be as small as possible. This way, malicious and selfish nodes would be isolated quickly. Nevertheless, the topology could be more frequently unstable, leading to more energy consumption. Besides, if T_{Trust} is too big, many nodes might be isolated, and consequently, the proper functioning of the network can be affected.

Since MRTS is built upon the RPL protocol, it inherits both its advantages and disadvantages. Indeed, the developers of Contiki-ng[1] confirmed that because of supporting new functionalities from standards and Internet drafts, the implementation of RPL becomes more complex and thus gets a large ROM footprint. In MRTS, each node maintains the list of all its neighbours with the necessary information to calculate trust values. Hence, if the network scales up, the neighbours' list will increase, and obviously, nodes will need more storage capacity. In fact, the storage limitation of LNN objects is still a big challenge, especially for large scale routing. As argued by Xiyuan et al. [182], the challenge is to find a balanced solution that reduces the memory overhead risk while improving the utilisation of the node capacity.

The collaborative isolation gives MRTS many advantages. On the one hand, it permits to reduce false positives. On the other hand, since all neighbours cooperate in the evaluation of a given node, even if two successive nodes misbehave, MRTS can detect and isolate them.

Table 4.5 gives a comparison of MRTS with other solutions used to secure RPL from routing attacks.

4.3.7 MRTS: A Strategy For Cooperation Enforcement

Following MRTS, the network operations' participation is conditioned by the trust value of each node; this means if TOF classifies a node as untrusted, it will be discarded from the network. As a consequence, there is no advantage for a smart, rational intruder to misbehave since it will be discarded from the network. Thus,

[1]The contiki OS for Next Generation IoT Devices that implements new functionalities to enhance RPL. Readers can reach the documentation online at https://github.com/contiki-ng/contiki-ng/wiki

the nodes in the network could achieve effective cooperation, and MRTS could be seen as a stable strategy of the interactive nodes within a repeated game. The network will then obtain service of higher security and trust between the cooperating nodes. In this section, we introduce our system model and explain how we mapped MRTS into a strategy for the iterated Prisoner's Dilemma (PD)[2]. Finally, we analyse the MRTS strategy formally and compare it to other strategies using simulation software regarding cooperation promotion and evolution. Then, we analyse the MRTS strategy formally and compare it to other strategies using simulation software regarding cooperation promotion and evolution. Table 4.6 presents the different notations used to analyse the cooperation enforcement characteristic of MRTS.

4.3.7.1 System Model

We used the non-zero-sum non-cooperative iterated PD game as the conceptual foundation for modelling interaction between the nodes of an MRTS-based network and the trust decision making process for each node, which finally results in cooperation (i.e., trusted) or defection (i.e., untrusted). From the security point of view, cooperating and defecting nodes correspond to the nodes executing the network's operation correctly or misbehaving, respectively. In this work's context, a misbehaving node is either a selfish node, an intruder that triggers attacks against the network, a node with not enough energy, or/and a node with a bad ETX. These pieces of information are abstract for mathematical modelling. Every two players engaged in the game (i.e., decision process to cooperate or defect; trust or untrust) play simultaneous PD moves in every stage of the game. After every stage, the players reveal all information (i.e.,trust values) about the previous stage. In the first stage, all players cooperate (trust), and then intruder nodes will defect (untrust) while normal nodes either will choose to cooperate or defect according to other players' moves in previous stages. We define MRTS as:

1. Cooperate on the first move;
2. In each period observe the past opponent's actions and count the number of defections, which corresponds to its trust evaluation: nbD;
3. If $nbD < Threshold$ Cooperate else Defect for the remainder of the game.

In Section 4.3.7.2, we give the equilibrium analysis of the proposed MRTS strategy and compare it with other known strategies. The equilibrium tells us about the most rational choice for each player in the game in a particular situation, and the network follows that by either isolating misbehaving nodes or not.

4.3.7.2 MRTS Strategy Analysis

Defection is the equilibrium in the one-shot PD game. Likewise, in a finite repeated PD, the only equilibrium is to defect, and it represents the Sub-game Perfect Equilibrium (SPE). However, it is not the only equilibrium in an Infinitely repeated PD

[2]The PD is a non-cooperative game with imperfect information that can be applicable in many domains. The PD can be extended to a multi-player or a repeated game and it is the basis for many models used to analyse the performance of networks' routing protocols. For more details please refer to [185][186][187].

(IPD). Indeed, it is possible to cooperate as an equilibrium because players can anticipate future rewards and punishments. There can be different equilibria in repeated games [186].

According to the authors in [185], [186], and [187], each player i has a repeated game strategy $s_i = (s_i^0, s_i^1, ..., s_i^T)$, where each s_i^t is history-dependent. The game is repeated T periods (i.e., stages), and T can be infinite ($T = \infty$). Formally, we represent each MRTS strategy of a player i as a sequence of history-dependent stage-game strategies as in Equation 4.21, where, C: Cooperate, D: Defect, $(C,C)^t$: means (C, C) repeated t times, and $(D,C)^{nbD}$: means (D, C) repeated nbD times.

$$s_i^t(h^t) = \begin{cases} C & \text{if } t = 0 \text{ or } h^t = ((C,C)^t) \\ C & \text{if } h^{nbD} = ((D,C)^{nbD}) \text{ and } nbD < Threshold \\ D & \text{if } h^{nbD} = ((D,C)^{nbD}) \text{ and } nbD \geq Threshold \end{cases} \tag{4.21}$$

Is it an equilibrium for two players to play MRTS for this iterated PD game? We consider the game as a two phases' game: the cooperation phase and the defection phase. In the cooperation phase, no player has defected previously, and hence both players are cooperating. In MRTS, the defection phase is divided into two sub-phases: the defection-cooperation phase and the defection-defection phase. In the defection-cooperation phase, opponent defects either alternatively or continuously, whereas the player cooperates until the number of defections is equal or greater than a defined threshold. In the defection-defection phase, the number of opponent's defections exceeds the threshold, and thus defection-defection is played forever. We check if in any of these phases of the game a player will need to deviate from the MRTS strategy, assuming that the other player is adopting the MRTS strategy.

It is assumed that the repeated game environment is stationary [186], and thus the payoff matrix is the same in every period. In this analysis, we use the PD payoff matrix[3] from Table 4.7 and formulas in the **Annexe A** [186] to calculate repeated game payoffs [4].

4.3.7.2.1 Cooperation Equilibrium

For IPD, there are infinitely many equilibria. It is possible to have an equilibrium in which both players always cooperate; this is what we are going to present in the following. If both players cooperated on the first-period $t = 0$, at period $t = 1$, the history is $h^1 = (C, C)$, and they both play cooperatively again. As a consequence, at period $t = 2$, the history is $h^2 = ((C, C), (C, C))$ and so on; resulting in an infinite path of (C, C). Thus, assuming that cooperation is an equilibrium, we calculate the repeated game Equilibrium Payoff (EP) to each player as in Equation 4.22, where

[3]If both players cooperate they both get a cooperation reward (R). However, if only one cooperates then the cooperating player gets a sucker score (S) whereas the defecting player receives a selfish temptation salary (T). Finally, if they both defect they both get a selfish punishment (P).

[4]http://virtualperfection.com/gametheory/5.2.InfinitelyRepeatedGames.1.0.pdf

δ[5] is the discount factor that takes values in [0,1]. Thus, the average equilibrium payoff is: $\overline{EP} = R$

$$\mathrm{EP} = \sum_{\mathrm{t}=0}^{\mathrm{n}} \delta^{\mathrm{t}}\mathrm{R} = \mathrm{R} + \mathrm{R}\delta + \mathrm{R}\delta^2 + \mathrm{R}\delta^3 + ... + \mathrm{R}\delta^{\mathrm{n}} \tag{4.22}$$

The following question arises: can any player gain from deviating from cooperation given that other players are accurately following it? Several cases can occur: 1) defecting k times from the first period, 2) defecting k times from the x^{th} period, and 3) defecting k times extended on several periods, where $k > 0$. Remember that by following the MRTS strategy, if player one (1) defects, player two (2) will cooperate as long as the number of defections is less than the threshold. In other words, (MRTS, MRTS) strategy at period-(t+1) depends not only on what is played at period-t but also on previous plays.

Case 1)-1: if player 1 defects in this first period and continues defecting k times ($Threshold = k$), he/she will have a payoff of T for the first k defections and then a payoff of P for the remainder of the game. Hence, the payoff for the defecting player 1 (i.e., Defection Payoff: DP) is calculated as in Equation 4.23. Thus, the average defection payoff is: $\overline{DP} = T(1-\delta^k) + P\delta^k$.

$$\begin{aligned} \mathrm{DP} &= \textstyle\sum_{\mathrm{t}=0}^{\mathrm{k}-1} \delta^{\mathrm{t}}\mathrm{T} + \sum_{\mathrm{t}=\mathrm{k}}^{\mathrm{n}} \delta^{\mathrm{t}}\mathrm{P} \\ &= \mathrm{T} + \mathrm{T}\delta + ... + \mathrm{T}\delta^{\mathrm{k}-1} + \mathrm{P}\delta^{\mathrm{k}} + ... + \mathrm{P}\delta^{\mathrm{n}} \end{aligned} \tag{4.23}$$

A player will continue cooperating according to the MRTS strategy if the following condition holds: $\overline{EP} \geq \overline{DP}$ (i.e., $R \geq T(1-\delta^k) + P\delta^k$), and thus if inequality 4.24 holds.

$$\delta \geq \left(\frac{\mathrm{T}-\mathrm{R}}{\mathrm{T}-\mathrm{P}}\right)^{\frac{1}{\mathrm{K}}} \tag{4.24}$$

Case 1)-2: if player 1 cooperates, then defects in the x^{th} period, and continues defecting k times, he/she will have a payoff of R for the x first periods, a payoff of T for the k defection times, and then a payoff of P for the remainder of the game. Hence, the payoff is calculated as in Equation 4.25. Thus, the average defection payoff is: $\overline{DP} = R(1-\delta^x) + T(\delta^x - \delta^{x+k}) + P(\delta^{x+k} - \delta^n)$

$$\begin{aligned} \mathrm{DP} = &\textstyle\sum_{\mathrm{t}=0}^{\mathrm{x}-1} \delta^{\mathrm{t}}\mathrm{R} + \sum_{\mathrm{t}=\mathrm{x}}^{\mathrm{x}+\mathrm{k}-1} \delta^{\mathrm{t}}\mathrm{T} + \sum_{\mathrm{t}=\mathrm{x}+\mathrm{k}}^{\mathrm{n}} \delta^{\mathrm{t}}\mathrm{P} \\ &= \mathrm{R} + \mathrm{R}\delta + ... + \mathrm{R}\delta^{\mathrm{x}-1} \\ &\quad + \mathrm{T}\delta^{\mathrm{x}} + ... + \mathrm{T}\delta^{\mathrm{x}+\mathrm{k}-1} \\ &\quad + \mathrm{P}\delta^{\mathrm{x}+\mathrm{k}} + ... + \mathrm{P}\delta^{\mathrm{n}} \end{aligned} \tag{4.25}$$

Similarly to case 1)-1, a player will continue cooperating according to MRTS strategy if $\overline{EP} \geq \overline{DP}$ (i.e., $R \geq R(1-\delta^x) + T(\delta^x - \delta^{x+k}) + P(\delta^{x+k} - \delta^n)$), and thus if inequality 4.24 holds.

[5]The discount factor allows to bound the stage-game payoffs and thus allows the infinite sum of the weighted payoffs to be finite.

We conclude that cooperation is an equilibrium (i.e., every player will be willing to cooperate forever) as long as the condition 4.24 holds. In other words, if the value of the discount factor δ is as in inequality 4.24, the deviation is not profitable (i.e., we mean by deviation k deviation times). Indeed, in the case of defection and for sufficiently patient players, there is a trade-off of getting a good payoff for k-defection-stages and then suffering for the rest of the time. As already stated in the literature, higher δ means more patience from a player, more care for the future, a higher chance of surviving into the next stage, and consequently enabling greater cooperation.

In this analysis, we do not present the case where the player defects k times extended on several periods. Nevertheless, the biggest gain a player can have by defecting is to play defection at the k-first-periods since the discount factor decreases more and more with time. In other words, the defection equilibrium payoff (DP) for any choice of defection periods is weakly less than the cooperation equilibrium payoff (CP) for $\delta \geq \left(\frac{T-R}{T-P}\right)^{\frac{1}{K}}$. Consequently, the player willingness is to cooperate rather than to defect. The simulation results in Section 4.3.7.3 demonstrate that the cooperation is maintained regardless of defection positions (periods). What matters is the number of defections throughout the game.

4.3.7.2.2 Defection Equilibrium

Can any player gain from deviating from defection strategy, given that other players are accurately following it? Two cases could occur:

Case 2)-1: if nbD through h^{t} is greater than or equal to the threshold, then play (D, D). If both players arrive at a sub-game of mutual defection forever (D, D) (i.e., after "$k = threshold$" defection times), this sub-game consists of the IPD. Playing the stage-game Nash equilibrium (D, D) of a game that is being infinitely repeated (in this case, PD) is an equilibrium itself [186]. Thus, if the two players are defecting forever, the best response for both of them is to continue defecting forever, and no one will need to deviate.

Case 2)-2: if nbD through h^{t} is less than the threshold, then play (D, C). According to MRTS, the punishment phase is reached when nbD equals the threshold, and it corresponds to Case 2)-1. However, if player 2 deviates from playing C before reaching the threshold, he/she plays (D, D) and Case 2)-1 applies, where both players play the equilibrium path (D, D) forever. Hence, it is best for him/her to deviate from this phase of the strategy. On the other hand, if player 1 deviates from playing D, he/she plays (C, C), and thus both players play the equilibrium path (C, C) forever. As a consequence, the best response for him/her is to deviate from this strategy phase.

MRTS is a complex strategy that can be an SPE[6] or not. Under the condition of threshold equal to 1, it is equivalent to the Spiteful strategy, making it an SPE. From the MRTS strategy point of view, the whole period where a player defects while the other player cooperates and the number of defections is less than a threshold

[6]Sub-game Perfect Equilibrium [186].

is equivalent to the cooperation period (i.e., (C, C)). Thus, the limit when nbD reaches the threshold corresponds to the one-shot deviation. In other words, the overall defection period can be reduced to the one-shot deviation, equivalently to Spiteful. Nevertheless, when playing (D, C) or (C, D), if there was a deviation and the game enters the punishment phase (D, D) forever, no player will want to deviate again since this guarantees a minimum gain of P for both players. Likewise, if there was a deviation and the game enters the cooperation phase (C, C) forever, no player will want to deviate again since this guarantees a gain of R for both players. Similar to the tit-for-tat strategy, these two beneficial deviations imply that MRTS is not an SPE.

4.3.7.3 Simulation Results with Perfect Vs. Imperfect Monitoring

Axelrod [188][189][187] was the first to organise computer tournaments to numerically detect strategies that would favour cooperation among players in the iterated PD. To this end, the authors used an ecological evolution algorithm for finding optimal and robust strategies. At the beginning of the execution, the population size is the same for each strategy. A round-robin tournament is executed, inducing the population of bad strategies to decrease, whereas good strategies obtain new players. Because the game theory is developed based on the understanding that all involved players are rational, when a defecting player discovers that the benefit of cooperating players is higher than defecting ones, it will change its strategy to get a higher benefit. Therefore, the proportion of players within the different strategies is changing with time. Thus, the process is iterated until the population does not change anymore. *"At the end, the good strategy is the one which stays alive in the population for the longest possible time, and in the biggest possible proportion".*

In this section, we present a numerical analysis of the MRTS strategy's performance in terms of cooperation and cooperation evolution among nodes. We compare MRTS to other known strategies; always cooperate (all_c), always defect (all_d), tit-for-tat, Spiteful, and soft-major. In the tit-for-tat strategy, each player starts the game by cooperating, and for all the future stages, each player copies the opponent's move from the previous stage. In the Spiteful strategy, each player starts by cooperating and continues to cooperate as long as everyone has cooperated previously. In the soft-major strategy, each player cooperates, and then plays the opponent's majority move. If the opponent's cooperation and defection moves are equal, the player cooperates. Both all_c and all_d strategies are history-independent.

We consider two cases: perfect and imperfect monitoring and use for the simulation a software introduced in [190]. We implemented the MRTS strategy and added it to the list of strategies in [190]. As inputs, each strategy begins the simulation with a population of 100 players and competes in a round-robin tournament. Besides, we used the payoff matrix in Table 4.7, where $P = 1$, $T = 5$, $S = 0$, and $R = 3$.

4.3.7.3.1 Perfect Monitoring

As depicted in Figure 4.18, the MRTS strategy won the tournament equally with

tit-for-tat, Spiteful, and soft-major. This achievement implies that MRTS is an Evolutionary Stable Strategy (ESS) of the IPD game. It is equivalent to the three other strategies as an evolutionary strategy to favour and enforce cooperation among players.

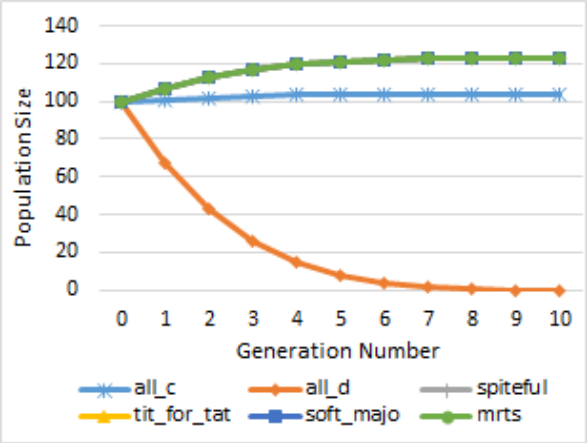


Figure 4.18: Comparison Between MRTS and Five Strategies Under Perfect Monitoring.

Figure 4.19 shows that MRTS is as good as other strategies to promote cooperation and cooperation evolution since it was ranked eighth out of 38 strategies involved in the simulation, with a size of 228 players. These results allow us to conclude that the MRTS punishment strategy enforces participating nodes' cooperation and prompts smart adversary nodes to become honest and cooperative.

4.3.7.3.2 Imperfect Monitoring

It is unrealistic to model real-world scenarios with the assumption of a noise-free environment, for instance, in the case of errors due to IDS monitoring tools, link quality, and promiscuous mode. The software introduces noise when playing the game to simulate imperfect monitoring. When we make the imperfect monitoring assumption, different results appear. Indeed, for a defection threshold of 10, the MRTS strategy performs better than the other strategies when the noise percentage is in-between 4% and 10%. It is the most evolutionary stable strategy (ESS) of the IPD game, as drawn in Figure 4.20a and Figure 4.20b. We notice from Figure 4.20c that when the noise is significant (25%), the soft-major strategy performs better than other strategies. Nevertheless, MRTS still performs better than Spiteful and tit-for-tat strategies.

As depicted in Figure 4.20d, when the noise is around 45% to 50% even if MRTS loses players, it stabilises after 90 generations with a population of 55 players. Consequently, MRTS performs better than other strategies such as tit-for-tat, which dies after 33 generations, and Spiteful, which dies after 65 generations. However, when the noise exceeds 50%, the MRTS strategy disappears after 149 generations but is still better than tit-for-tat, which disappears after 34 generations, and Spiteful, which disappears after 67 generations.

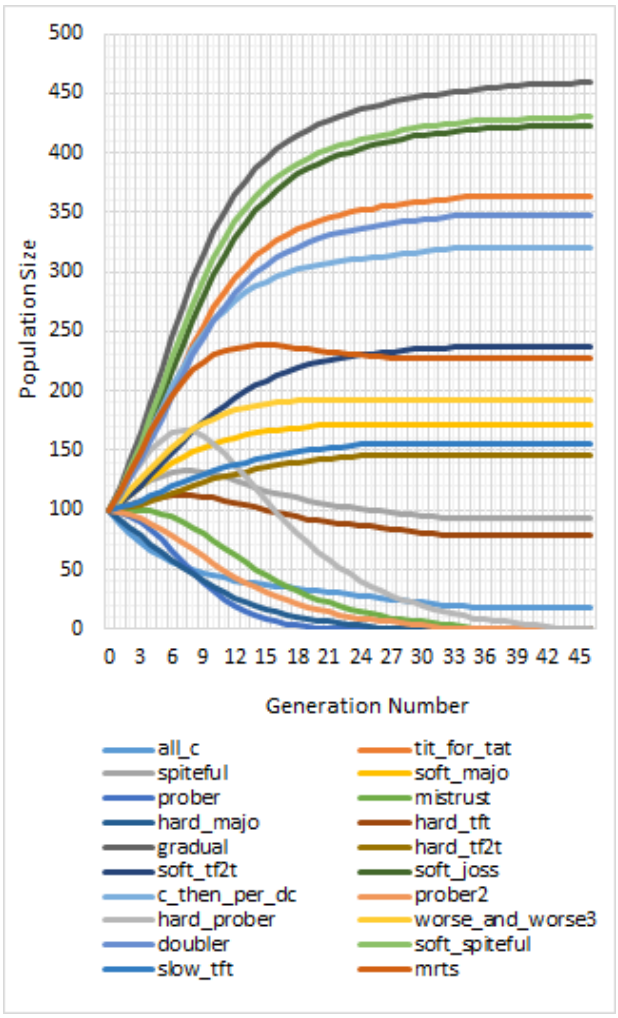
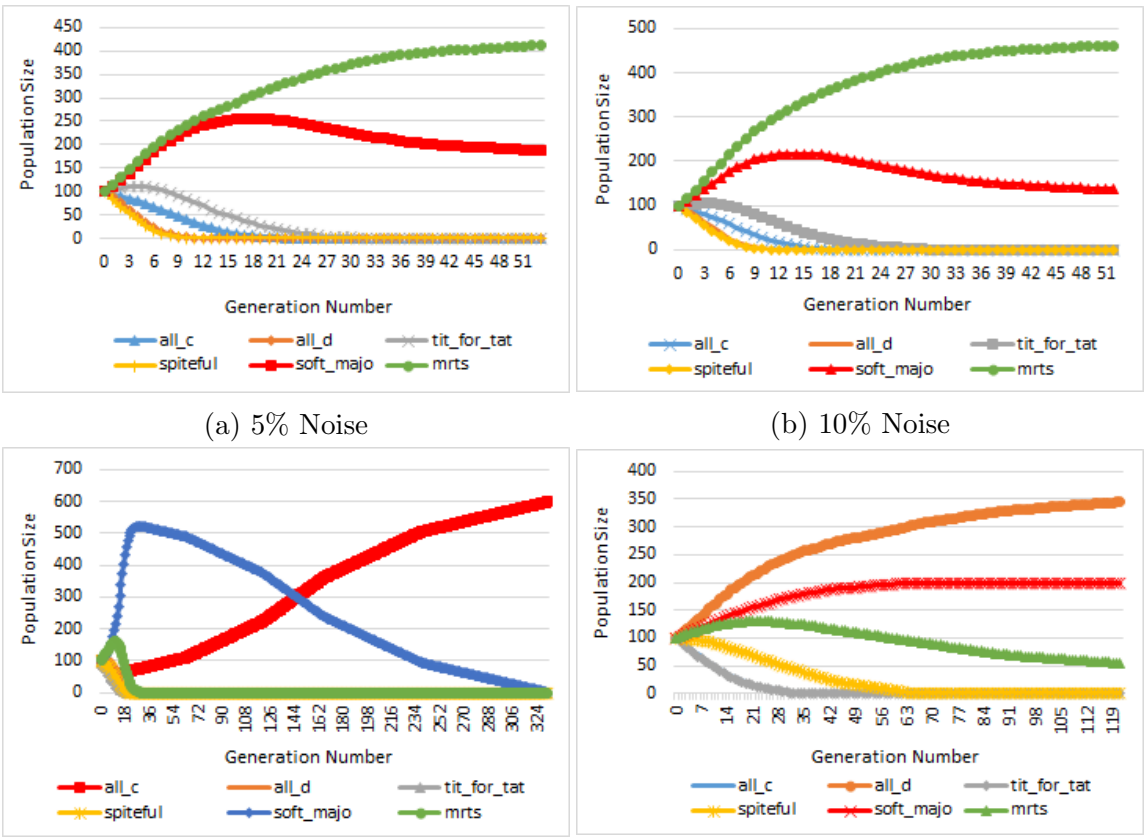


Figure 4.19: MRTS Compared to 19 Different Strategies Under Perfect Monitoring.

We can explain the results above as follows. By adopting the MRTS strategy, a node bases its decision on whether to trust (cooperate) or not (defect) relying on observations made on its opponents' past moves and a predefined defection threshold. Thus, the trust measure evaluated by the node takes into account more than one observation. Hence, the behaviour of the MRTS strategy does not depend on the noise percentage. Instead, it depends on the misperception noise itself and the threshold. Two cases could occur: i) in the first case, the misperception noise is cooperation instead of defection, and the threshold is not reached, consequently, the cooperation phase is extended, and thus MRTS performs better. ii) in the second case, the misperception noise is defection instead of cooperation, and the threshold is reached quickly; therefore, the cooperation phase is shortened, and thus MRTS disappears faster. In other words, the MRTS strategy will maintain the equilibrium state of cooperation when there exist small defection deviations, and the threshold is not reached, or when case i) occurs.

4.3.8 Discussion

We presented MRTS: a cooperation-trust-based routing mechanism for RPL. According to MRTS, at every hop of an RPL routing path, the child node selects the



(a) 5% Noise

(b) 10% Noise

(c) 25% Noise

(d) 50% Noise

Figure 4.20: MRTS Under Imperfect Monitoring.

node with a higher trust value, more energy, and better link quality as its preferred parent. We demonstrated through simulations that MRTS using multi-criteria based trust as routing metric (ERNT) is an efficient routing scheme to reduce the network security risks and maintain its performance and stability. Indeed, results showed that MRTS has low energy consumption and high packet delivery ratio, resulting from its capacity to detect and isolate attacks and its energy balanced topology mechanism. Furthermore, we proved that ERNT fulfils the isotonic and monotonic properties, hence allowing MRTS-based routing protocol to satisfy the requirements of consistency, optimality, and loop-freeness.

Furthermore, we translated MRTS into a strategy using game theory concepts. The MRTS strategy punishes and isolates the uncooperative (i.e., untrusted) nodes and enforces the network's security by enforcing nodes to cooperate rather than to cheat. We analysed the cooperation evolution of the MRTS strategy and demonstrated mathematically and with a simulation that the MRTS strategy is an evolutionary stable strategy, and under perfect monitoring, it is equivalent to the tit-for-tat and the Spiteful strategies to favour and enforce cooperation among nodes.

4.4 Conclusion

In this chapter, we presented our contributions, evaluated each contribution, and validated our proposed trust scheme in the context of IoT. Each contribution is a

succession of the previous one.

As future work, we plan to experiment and evaluate the MRTS performance in a real testbed and large-scale networks. Furthermore, we plan to extend MRTS with more mobility criteria and test its functionalities against different trust thresholds and routing attacks.

Table 4.5: Synthesis of Security Solutions for RPL ... (First part)

Works	Technique	Collaboration	Attacks	Disadvantages	Validation
[69]	IDS	No	Rank, sinkhole, DIS, and neighbour	More overhead because of the amount of information needed by the cluster head. Less accuracy detection after 10 minutes of execution. Cannot deal with mobile nodes	Simulation (Cooja)
[170]	IDS	No	Sinkhole, Rank, selective forwarding	High false detection rate and lack of DIO synchronisation. Cannot deal with mobile nodes	Simulation (Cooja)
[171]	IDS	Yes	Wormhole attack	Consumes energy. Cannot deal with mobile nodes	Simulation (Cooja)
[183]	IDS	No	HelloFlood, sinkhole, wormhole	Small network of 8 nodes	Python (Machine Learning)
[184]	IDS	No	Rank, Version number, HelloFlood	Fitting time too long	Python (Deep Learning)
[142]	Trust	No	Selective-forwarding	The nodes take decision based only on their own knowledge. If the node misbehaves, it will choose a failing path rather than a trusted one. Uses only one parameter (packet forwarding) to calculate trust	Simulation (J-Sim)
[146]	Trust	No	Network-level-attacks	Uses only direct trust. Vulnerable to bad-mouthing and good-mouthing attacks. Uses only one parameter (packet forwarding) to calculate trust. No specific attacks addressed. Vulnerable to a single point of failure	Simulation (MATLAB)
[145] **[144]**	Trust	No	Rank, Sybil, Blackhole	Uses only one parameter (packet forwarding) to calculate trust. Vulnerable to bad-mouthing and good-mouthing attacks	Simulation (Cooja)

Table 4.5: Synthesis of security solutions for RPL ... (Second part)

Works	Technique	Collaboration	Attacks	Disadvantages	Validation
[167]	Trust	Yes	Routing attacks	No simulation analysis to verify the effectiveness of the model. No specific attacks addressed	Simulation (Linux-C-Based)
[147]	Trust	Yes	greyhole, blackhole	How to calculate recommendations. Neither IDS nor a mechanism to detect other attacks	Simulation (Cooja)
[150]	Trust	Yes	DDOS	Uses only one parameter (data frequency) to calculate trust. Detect only DDOS attack. Vulnerable to a single point of failure	Simulation (Cooja)
[149]	Trust and IDS	Yes	Blackhole, Sybil and Rank	The programmed code size does not fit into the objects' (Tmote Sky) available memory. Massive and complex computation with too much information to use and store. No details on the integration method of the model to RPL. No details on the use of the IDS with the model	Simulation (Cooja)
New MRTS	IDS and Trust	Yes	Blackhole, Rank	Although the solution presents good performance regarding packet delivery, energy consumption, rank change, and throughput, further investigation is needed to prove its effectiveness in large networks	Simulation (Cooja) and Mathematical analysis

Table 4.6: MRTS Cooperation Enforcement Notations

Notation	Description
PD	Prisoner's Dilemma
IPD	Iterated Prisoner's Dilemma
SPE	Sub-game Perfect Equilibrium
ESS	Evolutionary Stable Strategy
nbD	Defection Count
k	Defection Threshold
C	Cooperate (Cooperation, Trust)
D	Defect (Defection, Untrust)
$\mathbf{(X,Y)}^Z$	(X,Y) strategy repeated Z times
$\mathbf{h^t}$	Game history at period t
$\mathbf{s_i}$	Repeated game strategy of the player i. $s_i = (s_i^0, s_i^1, ..., s_i^T)$
$\mathbf{s_i^T}$	Stage-game strategy (i.e., game strategy of the player i at the period T)
EP	Equilibrium Payoff
$\overline{\mathbf{EP}}$	Average Equilibrium Payoff
DP	Defection Payoff
$\overline{\mathbf{DP}}$	Average Defection Payoff
R	Cooperation reward if both players cooperate
S	Sucker score if only one player cooperates
T	Selfish temptation salary for the defecting player
P	Selfish punishment if both player defect
δ	Discount factor. To bound the stage-game payoffs and allow the infinite sum of the weighted payoffs to be finite.

Table 4.7: Prisoner's Dilemma payoff matrix

Player 1 \ Player 2	Cooperate (C)	Defect (D)
Cooperate (C)	(R,R)	(S,T)
Defect (D)	(T,S)	(P,P)

Conclusion

The Internet of Things (IoT) provides a rich set of advanced, intelligent and revolutionary applications and services such as healthcare, home automation, smart grid, automated transportation, environmental monitoring, and smart cities. Indeed, the core components of the IoT concept are the (smart) resource-constrained objects that sense data and the underlying communication technologies used to form the IoT networks. Nevertheless, one of the main issues faced by these IoT networks is infiltration from malicious nodes (objects) that can disrupt or stop the network operations and the services provided by the IoT applications. Consequently, using IoT needs to have a strong security mechanism that can identify and prevent malicious nodes from participating in the IoT operations at all levels. Particularly, ensuring trust management in IoT is crucial and very important for the appropriate functioning of its applications and services.

The main goal of this thesis is to enhance the security of the current standard protocols of IoT by introducing new trust mechanisms. Subsequently, in this thesis, we have performed and presented a study of the IoT concept, the enabling technologies, the different fields of applications, and the different architectures. Furthermore, we have provided an overview of the protocols accommodated in the current IoT standardised stack, as well as the main IoT challenges, particularly the security ones. In addition, we have given a state of the art on trust management in IoT, its problems and obstacles, its models' classifications, and the existing trust management solutions in IoT, and principally in the current standard routing protocol of IoT, namely, the Routing Protocol for Low-Power and Lossy Networks (RPL). This study allowed us to introduce a new classification of trust in the IoT context and propose new trust management approaches, which are the subject of our research.

We have proposed contributions on both the MAC layer, based on the IEEE 802.15.4 MAC standard, and the network layer, based on RPL. In the MAC layer, we have proposed a new MAC-trust-based model to handle MAC unfairness attacks while maintaining channel access to all participating nodes. The results showed that the model could handle easily untrusted nodes. From the other hand, in the network layer, we have proposed a new RPL version that permits routing by using trustworthiness between the different participating nodes during RPL topology construction and maintenance. This model maps a set of nodes' behaviours to a trust-based routing metric and implements a new trust-based objective function that are used to select the most trusted parent. The first model has been improved, thus emerging a new scheme of RPL, named Metric-based RPL Trustworthiness Scheme (MRTS), which deals with the trust inference problem of the first model. The inference drawback has been overcome by selecting the most trusted path from the source node to

the Border Router (BR), instead of selecting the most trusted parent. The simulation results showed that the new scheme improves the security of RPL. The final contribution is another enhancement of the second model (MRTS). In the third model, we have added new trust components and redefined the objective function to be more efficient in calculating the trust and the selection of parents. We have implemented and evaluated the model using extensive simulations and mathematical analyses. The MRTS performance evaluation showed good results concerning routing security, power consumption, packet delivery ratio, and throughput. Besides, the mathematical analyses have proven that MRTS meets the requirements of consistency, optimality, and loop-freeness and that the proposed trust-based routing metric has the isotonicity and monotonicity properties required for a routing protocol. In addition, game theory mathematical analyses and evolutionary simulation results have shown that MRTS, as a strategy, is an efficient approach in promoting the stability and the evolution of IoT networks.

As perspectives for our research, we will consider the new research directions in the field to enhance our contributions. We plan to experiment and evaluate the MRTS performance in large-scale networks and a real testbed to compare the obtained results with our simulations. Furthermore, we intend to extend MRTS with more criteria, such as mobility, and test its functionalities against different trust thresholds and routing attacks.

List of Publications

The following is a list of publications made while working on this thesis.

International communication 1

- Authors: Faiza Medjek, Djamel Tandjaoui, Mohammed Riyadh Abd-meziem, Nabil Djedjig
- Title: Analytical evaluation of the impacts of Sybil attacks against RPL under mobility
- Conference: The 12th International Symposium on Programming and Systems (ISPS)
- Location/date: Algiers (Algeria), 28-30/04/2015
- Preceedings: Pages 1-9
- Link: https://ieeexplore.ieee.org/document/7244960

International communication 2

- Authors: Nabil Djedjig, Djamel Tandjaoui, Faiza Medjek
- Title: Trust-based RPL for the Internet of Things
- Conference: The 20th IEEE Symposium on Computers and Communications (ISCC)
- Location/date: Larnaca (Cyprus), 6-9/07/2015
- Preceedings: Pages 962-967
- Link: https://ieeexplore.ieee.org/document/7405638

International communication 3

- Authors: Nabil Djedjig, Djamel Tandjaoui, Faiza Medjek, Imed Romdhani
- Title: New Trust Metric for the RPL Routing Protocol
- Conference: The 8th International Conference on Information and Communication Systems (ICICS)
- Location/date: Irbid (Jordan), 4-6/04/2017
- Preceedings: Pages 328-335
- Link: https://ieeexplore.ieee.org/document/7921993

International communication 4

- Authors: Faiza Medjek, Djamel Tandjaoui, Imed Romdhani, Nabil Djedjig
- Title: A trust-based intrusion detection system for mobile rpl based networks
- Conference: EEE International Conference on Internet of Things (iThings) and

IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)
- Location/date: Exeter (UK), 21-23/06/2017
- Preceedings: Pages 735-742
- Link: https://ieeexplore.ieee.org/document/8276832

International communication 5

- Authors: Nabil Djedjig, Djamel Tandjaoui, Imed Romdhani, Faiza Medjek
- Title: Trust-Based Defence Model Against MAC Unfairness Attacks for IoT
- Conference: The Thirteenth International Conference on Wireless and Mobile Communications (ICWMC)
- Location/date: Nice (France), 23-27/07/2017
- Preceedings: Pages 119-122
- Link: https://www.thinkmind.org/

International communication 6

- Authors: Faiza Medjek, Djamel Tandjaoui, Imed Romdhani, Nabil Djedjig
- Title: Performance evaluation of rpl protocol under mobile sybil attacks
- Conference: IEEE Trustcom/BigDataSE/ICESS
- Location/date: Sydney, NSW (Australia), 1-4/08/2017
- Preceedings: Pages 1049-1055
- Link: https://ieeexplore.ieee.org/document/8029554

Book chapter 1

- Authors: Nabil Djedjig, Djamel Tandjaoui, Imed Romdhani, Faiza Medjek
- Title: Trust Management in the Internet of Things
- Book Title: Security and Privacy in Smart Sensor Networks
- Pages 122-146
- Link: https://www.igi-global.com/chapter/trust-management-in-the-internet-of-things/203785

Book chapter 2

- Authors: Faiza Medjek, Djamel Tandjaoui, Imed Romdhani, Nabil Djedjig
- Title: Security Threats in the Internet of Things: RPL's Attacks and Countermeasures
- Book Title: Security and Privacy in Smart Sensor Networks
- Pages 147-178
- Link: https://www.igi-global.com/chapter/security-threats-in-the-internet-of-things/203786

International journal

- Authors: Nabil Djedjig, Djamel Tandjaoui, Faiza Medjek, Imed Romdhani
- Title: Trust-aware and cooperative routing protocol for IoT security
- Journal: Journal of Information Security and Applications

- Publisher: Elsevier
- Volume: 52
- Pages: 102467
- DOI: 10.1016/j.jisa.2020.102467
- Link: https://www.sciencedirect.com/science/article/pii/S2214212619306751

Appendix A

Game Theory Concepts

Game theory [185][186][187] is a mathematical tool used to describe and study conflicting and cooperative interactions with formalised games. Game theory enables decision makers involved in the game finding the best actions according to their different objectives. The entities involved in the game are known as *players*. There are one, two or N (N greater than two) players in a game. Each player can perform an action (i.e., a move) from the set of the player's possible actions, called the action space. The players are motivated by the so-called *payoff* or *utility function*. The payoff represents the gain/costs from having taken certain actions in the game.

Each player has one or more *strategies*, which are plans of action that specify which action to take based on prior knowledge. Indeed, the players' payoff are affected not only by their own strategies, but also by the other players as well. There exist two kind of strategies: *pure* and *mixed*. A pure strategy is a decision rule, which is always used by a player to determine the particular move he/she will make for any situation. On the other hand, a mixed strategy is a probability distribution over the set of actions. It is a mixture over the set of all pure strategies that allows a player to select randomly a pure strategy.

One solution of the game is the *Equilibrium*. An equilibrium is a combination of strategies leading to a maximum payoff that results in a state of the game where no player has any incentive to deviate from it. The most well-known equilibrium is the Nash Equilibrium (NE).

Games can be classified in different ways. There exist strategic/static/one-shot games and extensive/dynamic/repeated games. In the former, players choose their actions simultaneously from a predefined set of strategies. In other words, the players have only one move as a strategy. In the latter, players play a finite or infinite sequence of stage/one-shot games. The strategy space is a combination of pure strategies, and each player will determine its strategies taking into account all history strategies. The payoff of a player is the discounted sum of the payoffs he/she got in each period from playing the game.

Furthermore, games can be classified as cooperative and non-cooperative, of perfect information or of imperfect information, of complete information or of incomplete information, and finally, zero-sum or non-zero-sum.

- In cooperative games, each player acts taking into account a group-wide policy of the coalition the player belongs to, which means he/she does not only depend his/her actions on a self-interested perception of the game. On the other side, non-cooperative games are self-centric, so that each player pursues its own interests, which are partly conflicting with other players. Hence, each player acts independently without collaborating or communicating with any other players in the game.

- In games of perfect information, each player knows the history of all other players' actions before he/she performs his/her action. However, in games of imperfect information, at least one player does not know the history of all other players' actions.

- In games of complete information, each player knows the payoff functions of all other players. Nevertheless, in games of incomplete information at least one player does not know the payoff functions of all other players.

- In a zero-sum game the gain of one player represents the loss of the other player, and thus the summation of the utilities of all players is zero in every outcome of the game. Contrariwise, in a non-zero-sum game one player's gain (or loss) does not necessarily result in the other player's loss (or gain) which whereby the summation of the utilities of all players in every outcome of the game is different of zero (i.e., more than or less than zero) [185][186][187].

A.1 The Prisoner's Dilemma (PD) game

The Prisoner's Dilemma (PD) game is a very popular example in game theory. The game has two players who are suspects accused of a crime. There is not enough evidence to charge accusations, so the prosecution authority proposes to each one of them a reduction of the penalty if he confesses/cooperates. Hence, the players need to decide simultaneously either to cooperate or defect. Especially, if they both cooperate they both get a cooperation reward (R). However, if only one cooperates then the cooperating player gets a sucker score (S), whereas the defecting player receives a selfish temptation salary (T). Finally, if they both defect they both get a selfish punishment (P). The PD is a non-cooperative game with imperfect information that can be applicable in many domains. The PD can be extended to a multi-player or a repeated game, and it is the basis for many models used to analyse the performance of networks' routing protocols. Table A.1 summarizes the utilities of the two players in each case. In order to have a dilemma inequality (1) must be satisfied. In other words, temptation must be better than cooperation, which must be better than punishment, which must be better than the sucker. Furthermore, to avoid PD solution to influence strategies by giving too much importance to temptation regarding cooperation inequality (2) must be satisfied [185][186][187].

$$T > R > P > S \tag{A.1}$$

$$R > \frac{S + T}{2} \tag{A.2}$$

Table A.1: Prisoner's Dilemma payoff matrix

Player 1 \ Player 2	Cooperate (C)	Defect (D)
Cooperate (C)	(R,R)	(S,T)
Defect (D)	(T,S)	(P,P)

A.2 Repeated Game Concept

Many realistic scenarios need to be modelled as finitely/infinitely repeated games, where each stage-game is repeated finitely/infinitely, and neither player knows when the game will end. In this case, the player must make his/her decisions based on the mixed strategy, which will provide the best average payoff over time. This section relies on Ratliff's [186] and Axelrod's [187] works to present repeated game concept. It is referred to actions and strategies for moves in one-stage game and moves in repeated game, respectively. Repeated game can be formally defined as follows.

Definition 1: A stage-game denoted G is defined as, $G = (N, S, U)$, where N denotes the set of players, S the set of strategies and U the set of payoff functions. The actions, mixed strategies and payoffs of the stage-game G are denoted $\mathrm{a_i}$, α_i and $\mathrm{g_i}$, respectively. Each player $i \in N$ chooses an action $\mathrm{a_i}$ (i.e., a move) to play from an action space $\mathrm{A_i}$. Thus, the space of action profiles is $A = \mathrm{X_{i \in N} A_i}.A$

Definition 2: In the repeated game, known also as Supergame [1], the same stage-game G is repeated a finite or infinite number of times. In other words, G is repeated T periods (i.e., stages) where T can be infinite ($T = \infty$). The first period is t=0 and the last period is $t \in [1, \infty[$. If T=0 the repeated game corresponds to the one-shot game G. Playing the same stage-game in each period means that the environment for the repeated game is stationary (i.e., independent of time and history). In other words, from one side, the set of actions available to each player in any period of the game (i.e., in any stage-game) is the same regardless of which period it is, and regardless of what actions have taken place in the past. On the other side, the payoffs to the players from the stage-game in any period depend only on the action profile for G, which was played in that period, and this stage-game payoff to a player for a given action profile for G is independent of which period it is played.

Definition 3 -History and stage-game strategies for period-t: In the presented repeated game, each player observes the realised actions at the end of each period-t. Thus, the action profile played in period-t is denoted as in Equation A.3, where $\mathrm{a}^t_{i(i=1,\ldots,n)}$ is the stage-game action, the player i chooses in period-t.

$$\mathrm{a^t} = (\mathrm{a^t_1}, \mathrm{a^t_2}, ..., \mathrm{a^t_n}) \tag{A.3}$$

Thanks to observation of realised actions, the history h^t of all the actions (i.e., action profiles) played before the period-t is denoted as in Equation A.4, where the

[1] A repeated game is a supergame in which the same (ordinary) game is played at each iteration. Supergame is the original name for situations where the same game is played repetitively. Repeated game is used for more general models.[186]

first period is $t = 0$ with the null history h^0, and the last period if one exists is T. h^t can be also defined as the concatenation of history h^{t-1} with the action profile a^{t-1}.

$$\mathrm{h}^{\mathrm{t}} = (\mathrm{a}^0, \mathrm{a}^1, ..., \mathrm{a}^{\mathrm{t}-1}) \tag{A.4}$$

The history of the whole repeated game is denoted as in Equation A.5. In addition, the set A^t of all possible histories h^t for period t is the t-fold Cartesian product of the space of action profiles A such as in Equation A.6.

$$\mathrm{h}^{\mathrm{T}+1} = (\mathrm{a}^0, \mathrm{a}^1, ..., \mathrm{a}^{\mathrm{T}}) \tag{A.5}$$

$$\mathrm{A}^{\mathrm{t}} = \prod_{\mathrm{j}=0}^{\mathrm{t}-1} \mathrm{A} \tag{A.6}$$

Definition 4 -Strategies in the Repeated Game: Strategies in the repeated game are functions of history. This will allow conditioning the players' stage-game action choices in later periods upon actions taken earlier by other players. Indeed, player i would play a_i^t (in definition 3) in period-t if the previous play had followed the history h^t. This means it is defined as the function s_i^t such as in Equation A.7, where, s_i^t: $\mathrm{A}^t \rightarrow \mathrm{A}_i$ is the strategy of the player i at period-t for the stage-game, which takes a history h^t as its argument. Hence, the period-t stage-game strategy profile (i.e., strategy profile for the period-t) s^t is defined as the n-tuple of individuals' stage-game strategy profiles such as in Equation A.8. Whereas, the strategy of the player i for the repeated game s_i is defined as the $(T+1)$-tuple of history-contingent of the player i stage-game strategies, such as in Equation A.9. Hence, a strategy profile s for the whole repeated game is either the n-tuple profile of players' repeated-game strategies, such as in Equation A.10ion of stage-game strategy profiles, one for each period, such as in Equation A.11.

$$\mathrm{a}_{\mathrm{i}}^{\mathrm{t}} = \mathrm{s}_{\mathrm{i}}^{\mathrm{t}}(\mathrm{h}^{\mathrm{t}}) \tag{A.7}$$

$$\mathrm{s}^{\mathrm{t}} = (\mathrm{s}_1^{\mathrm{t}}, \mathrm{s}_2^{\mathrm{t}}, ..., \mathrm{s}_{\mathrm{n}}^{\mathrm{t}}) \tag{A.8}$$

$$\mathrm{s}_{\mathrm{i}} = (\mathrm{s}_{\mathrm{i}}^0, \mathrm{s}_{\mathrm{i}}^1, ..., \mathrm{s}_{\mathrm{i}}^{\mathrm{T}}) \tag{A.9}$$

$$\mathrm{s} = (\mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_{\mathrm{n}}) \tag{A.10}$$

$$\mathrm{s} = (\mathrm{s}^0, \mathrm{s}^1, ..., \mathrm{s}^{\mathrm{T}}) \tag{A.11}$$

Definition 5 -Payoffs in the Repeated Game: Each player has a utility function defined over the outcomes of G, g_i: $\mathrm{A} \rightarrow \mathrm{R}$, where A is the space of action profiles. For instance, the outcomes for the two-player PD game are presented in the form of a payoff matrix as in Table A.1. The payoff to the player i from the period-t stage-game when each player executes his/her stage-game strategy profile $\mathrm{s}^t(\mathrm{h}^t)$ is as in Equation A.12. Thus, from stage-to-stage, each player receives a payoff u_i^t. For each player, the repeated game payoff will be an additive function of his/her stage-game

payoffs. Nevertheless, if the game is played infinitely, there is an infinite number of payoffs to be summed. To ensure the summation do not lead to an infinite number, the discounting of future payoffs relatively to earlier payoffs is introduced.

The discount factor $\delta \in (0,1)$ can be an expression of time preference and uncertainty about the length of the game as $\delta = \mathrm{e}^{-r\Delta}$, where Δ is the rate of time preference and Δ is the length of the stage. The discount factor allows to bound the stage-game payoffs, and thus allows the infinite sum of the weighted payoffs to be finite. Therefore, the payoff to the player i for the complete repeated game, when the repeated game strategy profile s (see definition 4) is played is a function of the strategies followed in each stage-game, and is defined as in Equation A.13 and Equation A.14.

$$\mathrm{u_i^t} = \mathrm{g_i}(\mathrm{s^t}(\mathrm{h^t})) \tag{A.12}$$

$$\mathrm{u_i}(\mathrm{s}) = \sum_{\mathrm{t}=0}^{\infty} \delta^{\mathrm{t}} \mathrm{g_i}(\mathrm{s^t}(\mathrm{h^t})) \tag{A.13}$$

$$\mathrm{u_i}(\mathrm{s}) = \sum_{\mathrm{t}=0}^{\infty} \delta^{\mathrm{t}} \mathrm{u_i^t} \tag{A.14}$$

It is more convenient to normalise the repeated-game payoffs to be on the same scale as the stage-game payoffs. To this end, the discounted payoff sum $\mathrm{u_i}(s)$ is multiplied by the term (1-δ). As a result, the average discounted value is defined as in Equation A.15.

$$\overline{\mathrm{u}}_{\mathrm{i}}(\mathrm{s}) = (1-\delta) \sum_{\mathrm{t}=0}^{\infty} \delta^{\mathrm{t}} \mathrm{u_i^t} \tag{A.15}$$

Indeed, the discounted sum of a player's payoffs from a repeated game is a geometric series, such as $\delta + b\delta + b\delta^2 + b\delta^3 + ... + b\delta^{\mathrm{T}}$. By applying geometric series rules, the sum will be as in Equation A.16 and Equation A.17, where $-1 < \delta < 1$.

$$\delta + \mathrm{b}\delta + \mathrm{b}\delta^2 + \mathrm{b}\delta^3 + ... + \mathrm{b}\delta^{\mathrm{T}} = \sum_{\mathrm{t}=0}^{\mathrm{T}} \mathrm{b}\delta^{\mathrm{t\text{-}1}} \tag{A.16}$$

$$\sum_{t=0}^{T} b\delta^{\mathrm{t\text{-}1}} = \begin{cases} \frac{b(1-\delta^{\mathrm{T}+1})}{1-\delta} & if \quad \mathrm{T} \quad finite \\ \frac{\mathrm{b}}{1-\delta} & if \quad \mathrm{T} \quad infinite \end{cases} \tag{A.17}$$

Repeated game play: When the repeated game starts, there is no past play, which means no history h^0. Each player i executes its stage-game strategy $\mathrm{a}_i^0 = \mathrm{s}_1^0$, and gets a payoff $\mathrm{u}_i(s) = \mathrm{u}_i^0$. This first (zero-*th*) period play generates the history $\mathrm{h}^1 = (\mathrm{a}^0)$, where $\mathrm{a}^0 = (\mathrm{a}_1^0, \mathrm{a}_2^0, ..., \mathrm{a}_n^0)$. This history is then revealed to the players so that each player can condition its period-1 play upon the period-0 play. Consequently, each player chooses its $t = 1$ stage-game strategy $\mathrm{s}^1(\mathrm{h}^1) = (\mathrm{s}_1^1(\mathrm{h}^1), ..., \mathrm{s}_n^1(\mathrm{h}^1))$. Therefore, in the $t = 1$ stage-game, the stage-game strategy profile $\mathrm{a}^1 = \mathrm{s}^1(\mathrm{h}^1)$ is played and a payoff $\mathrm{u}_i(s) = \mathrm{u}_i^0 + \delta \mathrm{u}_i^1$ is gotten. Once a^1 played, the history is updated again such as $\mathrm{h}^2 = (\mathrm{a}^0, \mathrm{a}^1)$. This history is then revealed to the players so that each player can choose its period-2 play (i.e., stage-game strategy) $\mathrm{a}^2 = \mathrm{s}^2(\mathrm{h}^2)$ generating a payoff $\mathrm{u}_i(s) = \mathrm{u}_i^0 + \delta \mathrm{u}_i^1 + \delta^2 \mathrm{u}_i^2$, and so on for all stages. At the end, $\mathrm{h}^{(T+1)}$ is known as the

path generated by the repeated game strategy profile s.

Definition 6 -Equilibrium in Repeated Games: Infinite repetition can be the key for obtaining behaviour in the stage games, which could not be equilibrium behaviour if the game were played once or a known finite number of times.

Bibliography

[1] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer networks 54 (15) (2010) 2787–2805.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE communications surveys & tutorials 17 (4) (2015) 2347–2376.

[3] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann, K. K. Leung, A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities, IEEE Wireless Communications 20 (6) (2013) 91–98.

[4] D. Airehrour, J. Gutierrez, S. K. Ray, Secure routing for internet of things: A survey, Journal of Network and Computer Applications 66 (2016) 198–213.

[5] H. Nunoo-Mensah, K. O. Boateng, J. D. Gadze, The adoption of socio-and bio-inspired algorithms for trust models in wireless sensor networks: A survey, International Journal of Communication Systems 31 (7) (2018) e3444.

[6] F. Moyano, C. Fernandez-Gago, J. Lopez, A framework for enabling trust requirements in social cloud applications, Requirements Engineering 18 (4) (2013) 321–341.

[7] I. S. Association, et al., Ieee standard for low-rate wireless networks, IEEE Std 802 (2016) 4–2015.

[8] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (7) (2013) 1645–1660.

[9] T. Winter, Rpl: Ipv6 routing protocol for low-power and lossy networks (2012).

[10] C. M. Medaglia, A. Serbanati, An overview of privacy and security issues in the internet of things, in: The internet of things, Springer, 2010, pp. 389–395.

[11] K.-D. Chang, J.-L. Chen, A survey of trust management in wsns, internet of things and future internet., KSII Transactions on Internet & Information Systems 6 (1) (2012).

[12] T. Xu, J. B. Wendt, M. Potkonjak, Security of iot systems: Design challenges and opportunities, in: 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2014, pp. 417–423.

[13] F. Bao, I.-R. Chen, M. Chang, J.-H. Cho, Hierarchical trust management for wireless sensor networks and its application to trust-based routing, in: Proceedings of the 2011 ACM Symposium on Applied Computing, ACM, 2011, pp. 1732–1738.

[14] F. Bao, R. Chen, M. Chang, J.-H. Cho, Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, Network and Service Management, IEEE Transactions on 9 (2) (2012) 169–183.

[15] R. Chen, J. Guo, Hierarchical trust management of community of interest groups in mobile ad hoc networks, Ad Hoc Networks 33 (2015) 154–167.

[16] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for internet of things, Journal of network and computer applications 42 (2014) 120–134.

[17] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad hoc networks 10 (7) (2012) 1497–1516.

[18] D. Singh, G. Tripathi, A. J. Jara, A survey of internet-of-things: Future vision, architecture, challenges and services, in: 2014 IEEE world forum on Internet of Things (WF-IoT), IEEE, 2014, pp. 287–292.

[19] M. Presser, A. Gluhak, The internet of things: Connecting the real world with the digital world, eurescom mess@ ge–the magazine for telecom insiders, vol. 2, 2009 (2012).

[20] D. Culler, S. Chakrabarti, I. Infusion, 6lowpan: Incorporating ieee 802.15. 4 into the ip architecture, IPSO Alliance, White paper (2009).

[21] N. Gershenfeld, R. Krikorian, D. Cohen, The internet of things, Scientific American 291 (4) (2004) 76–81.

[22] I. Toma, E. Simperl, G. Hench, A joint roadmap for semantic technologies and the internet of things, in: Proceedings of the Third STI Roadmapping Workshop, Crete, Greece, Vol. 1, 2009, pp. 140–53.

[23] G. Marrocco, C. Occhiuzzi, F. Amato, Sensor-oriented passive rfid, in: The Internet of Things, Springer, 2010, pp. 273–282.

[24] A. Sample, M. BUETTNER, B. GREENSTEIN, et al., Revisiting smart dust with rfid sensor networks, The Seventh ACM VVorkshop on Hot Topics in Networks (2008).

[25] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet of Things Journal 4 (5) (2017) 1125–1142.

[26] M. A. Chaqfeh, N. Mohamed, Challenges in middleware solutions for the internet of things, in: 2012 international conference on collaboration technologies and systems (CTS), IEEE, 2012, pp. 21–26.

[27] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. M. S. De Souza, V. Trifa, Soa-based integration of the internet of things in enterprise services, in: 2009 IEEE international conference on web services, IEEE, 2009, pp. 968–975.

[28] A. Gómez-Goiri, D. López-de Ipiña, A triple space-based semantic distributed middleware for internet of things, in: International Conference on Web Engineering, Springer, 2010, pp. 447–458.

[29] A. Roxin, C. Dumez, N. Cottin, J. Gaber, M. Wack, Transportml: A middleware for location-based services collaboration, in: 2009 3rd International Conference on New Technologies, Mobility and Security, IEEE, 2009, pp. 1–6.

[30] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, K. Kim, Tinyrest-a protocol for integrating sensor networks into the internet, in: Proc. of REALWSN, 2005, pp. 101–105.

[31] A. Mukherjee, D. Saha, C. Biswas, Present scenarios and future challenges in pervasive middleware, in: 2006 1st International Conference on Communication Systems Software & Middleware, IEEE, 2006, pp. 1–5.

[32] P. Saarika, K. Sandhya, T. Sudha, Smart transportation system using iot, in: 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), IEEE, 2017, pp. 1104–1107.

[33] J. Sherly, D. Somasundareswari, Internet of things based smart transportation systems, International Research Journal of Engineering and Technology 2 (7) (2015) 1207–1210.

[34] Y.-y. Tseng, W. L. Yue, M. A. Taylor, et al., The role of transportation in logistics chain, Eastern Asia Society for Transportation Studies, 2005.

[35] T. Brockman, 21 warehousing trends in the 21st century, IIE solutions 31 (7) (1999) 36–41.

[36] E. Hofmann, M. Rüsch, Industry 4.0 and the current status as well as future prospects on logistics, Computers in industry 89 (2017) 23–34.

[37] N. Scarpato, A. Pieroni, L. Di Nunzio, F. Fallucchi, E-health-iot universe: A review, management 21 (44) (2017) 46.

[38] M. A. Pradhan, S. Patankar, A. Shinde, V. Shivarkar, P. Phadatare, Iot for smart city: Improvising smart environment, in: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), IEEE, 2017, pp. 2003–2006.

[39] A. Alshamsi, Y. Anwar, M. Almulla, M. Aldohoori, N. Hamad, M. Awad, Monitoring pollution: Applying iot to create a smart environment, in: 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), IEEE, 2017, pp. 1–4.

[40] A. Vimal Jerald, S. Rabara, T. Bai, Internet of things (iot) based smart environment integrating various business applications, International Journal of Computer Applications 128 (8) (2015) 32–37.

[41] R. C. Andrew, R. Malekian, D. C. Bogatinoska, Iot solutions for precision agriculture, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2018, pp. 0345–0349.

[42] S. Prathibha, A. Hongal, M. Jyothi, Iot based monitoring system in smart agriculture, in: 2017 international conference on recent advances in electronics and communication technology (ICRAECT), IEEE, 2017, pp. 81–84.

[43] N. Suma, S. R. Samson, S. Saranya, G. Shanmugapriya, R. Subhashri, Iot based smart agriculture monitoring system, International Journal on Recent and Innovation Trends in computing and communication 5 (2) (2017) 177–181.

[44] P. P. Ray, Internet of things for smart agriculture: Technologies, practices and future direction, Journal of Ambient Intelligence and Smart Environments 9 (4) (2017) 395–420.

[45] B. Guo, D. Zhang, Z. Wang, Z. Yu, X. Zhou, Opportunistic iot: Exploring the harmonious interaction between human and the internet of things, Journal of Network and Computer Applications 36 (6) (2013) 1531–1539.

[46] Y. Saleem, N. Crespi, M. H. Rehmani, R. Copeland, D. Hussein, E. Bertin, Exploitation of social iot for recommendation services, in: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), IEEE, 2016, pp. 359–364.

[47] B. Afzal, M. Umair, G. A. Shah, E. Ahmed, Enabling iot platforms for social iot applications: vision, feature mapping, and challenges, Future Generation Computer Systems 92 (2019) 718–731.

[48] R. K. Lenka, A. K. Rath, Z. Tan, S. Sharma, D. Puthal, N. Simha, M. Prasad, R. Raja, S. S. Tripathi, Building scalable cyber-physical-social networking infrastructure using iot and low power sensors, IEEE Access 6 (2018) 30162–30173.

[49] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, H.-Y. Du, Research on the architecture of internet of things, in: 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Vol. 5, IEEE, 2010, pp. V5–484.

[50] T. Melliti, Interopérabilité des services web complexes: Application aux systèmes multi-agents, Ph.D. thesis, Paris 9 (2004).

[51] M. P. Papazoglou, D. Georgakopoulos, Introduction: Service-oriented computing, Communications of the ACM 46 (10) (2003) 24–28.

[52] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, et al., Transmission of ipv6 packets over ieee 802.15. 4 networks, Internet proposed standard RFC 4944 (2007) 130.

[53] W. Kluge, F. Poegel, H. Roller, M. Lange, T. Ferchland, L. Dathe, D. Eggert, A fully integrated 2.4-ghz ieee 802.15. 4-compliant transceiver for zigbee™ applications, IEEE Journal of Solid-State Circuits 41 (12) (2006) 2767–2775.

[54] A. N. Kim, F. Hekland, S. Petersen, P. Doyle, When hart goes wireless: Understanding and implementing the wirelesshart standard, in: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, IEEE, 2008, pp. 899–907.

[55] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, W. Pratt, Wirelesshart: Applying wireless technology in real-time industrial process control, in: 2008 IEEE Real-Time and Embedded Technology and Applications Symposium, IEEE, 2008, pp. 377–386.

[56] J. W. Hui, D. E. Culler, Extending ip to low-power, wireless personal area networks, IEEE Internet Computing 12 (4) (2008) 37–45.

[57] C. Bormann, A. P. Castellani, Z. Shelby, Coap: An application protocol for billions of tiny internet nodes, IEEE Internet Computing 16 (2) (2012) 62–67.

[58] W. Gao, J. Nguyen, W. Yu, C. Lu, D. Ku, Assessing performance of constrained application protocol (coap) in manet using emulation, in: Proceedings of the International Conference on Research in Adaptive and Convergent Systems, 2016, pp. 103–108.

[59] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, Rpl: Ipv6 routing protocol for low-power and lossy networks, RFC 6550, Internet Engineering Task Force (2012).

[60] P. Samar, M. R. Pearlman, Z. J. Haas, Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks, IEEE/ACM Transactions On Networking 12 (4) (2004) 595–608.

[61] L. Villasenor-Gonzalez, Y. Ge, L. Lament, Holsr: a hierarchical proactive routing mechanism for mobile ad hoc networks, IEEE Communications Magazine 43 (7) (2005) 118–125.

[62] J. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing metrics used for path calculation in low power and lossy networks, RFC 6551, Internet Engineering Task Force (2012).

[63] P. Thubert, Objective function zero for the routing protocol for low-power and lossy networks (rpl), RFC 6552, Internet Engineering Task Force (2012).

[64] O. Gnawali, P. Levis, The minimum rank with hysteresis objective function, RFC 6719 (2012).

[65] J. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing metrics used for path calculation in low-power and lossy networks, in: RFC 6551, IETF, 2012, pp. 1–30.

[66] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, Security threats in the internet of things: Rpl's attacks and countermeasures, in: Security and Privacy in Smart Sensor Networks, IGI Global, 2018, pp. 147–178.

[67] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, M. Richardson, A security threat analysis for the routing protocol for low-power and lossy networks (rpls), Tech. rep. (2015).

[68] A. Dvir, T. Holczer, L. Buttyan, Vera-version number and rank authentication in rpl, in: Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, IEEE, 2011, pp. 709–714.

[69] A. Le, J. Loo, K. Chai, M. Aiash, A specification-based ids for detecting attacks on rpl-based network topology, Information 7 (2) (2016) 25.

[70] A. Le, J. Loo, Y. Luo, A. Lasebae, The impacts of internal threats towards routing protocol for low power and lossy network performance, in: Computers and Communications (ISCC), 2013 IEEE Symposium on, IEEE, 2013, pp. 000789–000794.

[71] A. Mayzaud, R. Badonnel, I. Chrisment, A taxonomy of attacks in rpl-based internet of things, International Journal of Network Security 18 (3) (2016) 459–473.

[72] K. Chugh, L. Aboubaker, J. Loo, Case study of a black hole attack on lowpan-rpl, in: Proc. of the Sixth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), Rome, Italy (August 2012), 2012, pp. 157–162.

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

[74] Z. J. Haas, L. Yang, M.-L. Liu, Q. Li, F. Li, Current challenges and approaches in securing communications for sensors and actuators, in: The Art of Wireless Sensor Networks, Springer, 2014, pp. 569–608.

[75] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422.

[76] J. Sung, T. S. Lopez, D. Kim, The epc sensor network for rfid and wsn integration infrastructure, in: Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), IEEE, 2007, pp. 618–621.

[77] C. Floerkemeier, R. Bhattacharyya, S. Sarma, Beyond the id in rfid, in: The Internet of Things, Springer, 2010, pp. 219–227.

[78] S. Haller, S. Karnouskos, C. Schroth, The internet of things in an enterprise context, in: Future Internet Symposium, Springer, 2008, pp. 14–28.

[79] A. Nilssen, Security and privacy standardization in internet of things, eMatch (2009).

[80] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, A. R. RPL, Ipv6 routing protocol for low-power and lossy networks, RFC6550 of IETF (2012).

[81] M. Weyrich, C. Ebert, Reference architectures for the internet of things, IEEE Software 33 (1) (2015) 112–116.

[82] S.-D. Lee, M.-K. Shin, H.-J. Kim, Epc vs. ipv6 mapping mechanism, in: The 9th international conference on advanced communication technology, Vol. 2, IEEE, 2007, pp. 1243–1245.

[83] Y.-W. Ma, C.-F. Lai, Y.-M. Huang, J.-L. Chen, Mobile rfid with ipv6 for phone services, in: 2009 IEEE 13th International Symposium on Consumer Electronics, IEEE, 2009, pp. 169–170.

[84] T. A. DICTATE, Epcglobal object name service (ons) 1.0.

[85] V. Cerf, Y. Dalal, C. Sunshine, Specification of internet transmission control program, Tech. rep., RFC 675, December (1974).

[86] R. Duan, X. Chen, T. Xing, A qos architecture for iot, in: 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, IEEE, 2011, pp. 717–720.

[87] E. Mingozzi, G. Tanganelli, C. Vallati, B. Martínez, I. Mendia, M. Gonzalez-Rodriguez, Semantic-based context modeling for quality of service support in iot platforms, in: 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2016, pp. 1–6.

[88] S. Peros, H. Janjua, S. Akkermans, W. Joosen, D. Hughes, Dynamic qos support for iot backhaul networks through sdn, in: 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, 2018, pp. 187–192.

[89] W. Shang, Y. Yu, R. Droms, L. Zhang, Challenges in iot networking via tcp/ip architecture, Technical Report NDN-0038. NDN Project (2016).

[90] S. Sicari, A. Rizzardi, L. A. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Computer networks 76 (2015) 146–164.

[91] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, L. Shu, Authentication protocols for internet of things: a comprehensive survey, Security and Communication Networks 2017 (2017).

[92] B. J. Mohd, T. Hayajneh, A. V. Vasilakos, A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues, Journal of Network and Computer Applications 58 (2015) 73–93.

[93] C. Machado, A. A. M. Fröhlich, Iot data integrity verification for cyber-physical systems using blockchain, in: 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC), IEEE, 2018, pp. 83–90.

[94] T. Abera, N. Asokan, L. Davi, F. Koushanfar, A. Paverd, A.-R. Sadeghi, G. Tsudik, Things, trouble, trust: on building trust in iot systems, in: Proceedings of the 53rd Annual Design Automation Conference, 2016, pp. 1–6.

[95] T. Yang, G. Zhang, L. Liu, Y. Yang, S. Zhao, H. Sun, W. Wang, New features of authentication scheme for the iot: A survey, in: Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things, 2019, pp. 44–49.

[96] H. Kim, E. A. Lee, Authentication and authorization for the internet of things, IT Professional 19 (5) (2017) 27–33.

[97] M. Karimibiuki, E. Aggarwal, K. Pattabiraman, A. Ivanov, Dynpolac: Dynamic policy-based access control for iot systems, in: 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE, 2018, pp. 161–170.

[98] B. Volochiy, V. Yakovyna, O. Mulyak, Queueing networks for availability and safety assessment of the iot data service, in: 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Vol. 1, IEEE, 2017, pp. 393–396.

[99] M. A. López-Peña, J. Díaz, J. E. Pérez, H. Humanes, Devops for iot systems: Fast and continuous monitoring feedback of system availability, IEEE Internet of Things Journal 7 (10) (2020) 10695–10707.

[100] W. B. Qaim, O. Ozkasap, Draw: Data replication for enhanced data availability in iot-based sensor systems, in: 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE, 2018, pp. 770–775.

[101] Y. Huo, C. Yong, Y. Lu, Re-adp: real-time data aggregation with adaptive-event differential privacy for fog computing, Wireless Communications and Mobile Computing 2018 (2018).

[102] Y. Huo, C. Hu, X. Qi, T. Jing, Lodpd: a location difference-based proximity detection protocol for fog computing, IEEE Internet of Things Journal 4 (5) (2017) 1117–1124.

[103] J. Mao, W. Tian, J. Jiang, Z. He, Z. Zhou, J. Liu, Understanding structure-based social network de-anonymization techniques via empirical analysis, EURASIP Journal on Wireless Communications and Networking 2018 (1) (2018) 279.

[104] A. Dorri, M. Steger, S. S. Kanhere, R. Jurdak, Blockchain: A distributed solution to automotive security and privacy, IEEE Communications Magazine 55 (12) (2017) 119–125.

[105] K. M. Hossein, M. E. Esmaeili, T. Dargahi, et al., Blockchain-based privacy-preserving healthcare architecture, in: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), IEEE, 2019, pp. 1–4.

[106] R. C. Mayer, J. H. Davis, F. D. Schoorman, An integrative model of organizational trust, Academy of management review 20 (3) (1995) 709–734.

[107] A. Arabsorkhi, M. S. Haghighi, R. Ghorbanloo, A conceptual trust model for the internet of things interactions, in: 2016 8th International Symposium on Telecommunications (IST), IEEE, 2016, pp. 89–93.

[108] C. L. Corritore, B. Kracher, S. Wiedenbeck, On-line trust: concepts, evolving themes, a model, International journal of human-computer studies 58 (6) (2003) 737–758.

[109] E. Chang, T. S. Dillon, F. K. Hussain, Trust and reputation relationships in service-oriented environments, in: Third International Conference on Information Technology and Applications (ICITA'05), Vol. 1, IEEE, 2005, pp. 4–14.

[110] L. Buttyan, J.-P. Hubaux, Security and cooperation in wireless networks: thwarting malicious and selfish behavior in the age of ubiquitous computing, Cambridge University Press, 2007.

[111] Z. M. Aljazzaf, M. Perry, M. A. Capretz, Online trust: Definition and principles, in: 2010 Fifth International Multi-conference on Computing in the Global Information Technology, IEEE, 2010, pp. 163–168.

[112] J. Daubert, A. Wiesmaier, P. Kikiras, A view on privacy & trust in iot, in: 2015 IEEE International Conference on Communication Workshop (ICCW), IEEE, 2015, pp. 2665–2670.

[113] Z. Yan, S. Holtmanns, Trust modeling and management: from social trust to digital trust, in: Computer security, privacy and politics: current issues, challenges and solutions, IGI Global, 2008, pp. 290–323.

[114] Z. Yan, C. Prehofer, Autonomic trust management for a component-based software system, IEEE Transactions on Dependable and Secure Computing 8 (6) (2010) 810–823.

[115] J. P. Wang, S. Bin, Y. Yu, X. X. Niu, Distributed trust management mechanism for the internet of things, in: Applied Mechanics and Materials, Vol. 347, Trans Tech Publ, 2013, pp. 2463–2467.

[116] J. Guo, R. Chen, J. J. Tsai, A survey of trust computation models for service management in internet of things systems, Computer Communications 97 (2017) 1–14.

[117] N. Djedjig, D. Tandjaoui, I. Romdhani, F. Medjek, Trust management in the internet of things, in: Security and Privacy in Smart Sensor Networks, IGI Global, 2018, pp. 122–146.

[118] G. D. Tormo, F. G. Mármol, G. M. Pérez, Dynamic and flexible selection of a reputation mechanism for heterogeneous environments, Future Generation Computer Systems 49 (2015) 113–124.

[119] C. Fernandez-Gago, F. Moyano, J. Lopez, Modelling trust dynamics in the internet of things, Information Sciences 396 (2017) 72–82.

[120] Y. Sun, Z. Han, K. R. Liu, Defense of trust management vulnerabilities in distributed networks, IEEE Communications Magazine 46 (2) (2008) 112–119.

[121] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Computing Surveys (CSUR) 42 (1) (2009) 1.

[122] Y. Chae, L. C. DiPippo, Y. L. Sun, Trust management for defending on-off attacks, IEEE Transactions on Parallel and Distributed Systems 26 (4) (2014) 1178–1191.

[123] O. B. Abderrahim, M. H. Elhedhili, L. Saidane, Dtms-iot: A dirichlet-based trust management system mitigating on-off attacks and dishonest recommendations for the internet of things, in: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), IEEE, 2016, pp. 1–8.

[124] Z. Banković, J. C. Vallejo, D. Fraga, J. M. Moya, Detecting bad-mouthing attacks on reputation systems using self-organizing maps, in: Computational Intelligence in Security for Information Systems, Springer, 2011, pp. 9–16.

[125] F. Bao, R. Chen, J. Guo, Scalable, adaptive and survivable trust management for community of interest based internet of things systems, in: 2013 IEEE eleventh international symposium on autonomous decentralized systems (ISADS), Citeseer, 2013, pp. 1–7.

[126] R. Chen, J. Guo, Dynamic hierarchical trust management of mobile groups and its application to misbehaving node detection, in: 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, IEEE, 2014, pp. 49–56.

[127] G. Han, J. Jiang, L. Shu, J. Niu, H.-C. Chao, Management and applications of trust in wireless sensor networks: A survey, Journal of Computer and System Sciences 80 (3) (2014) 602–617.

[128] T. K. Kim, H. S. Seo, A trust model using fuzzy logic in wireless sensor network, World academy of science, engineering and technology 42 (6) (2008) 63–66.

[129] S. Ganeriwal, L. K. Balzano, M. B. Srivastava, Reputation-based framework for high integrity sensor networks, ACM Transactions on Sensor Networks (TOSN) 4 (3) (2008) 15.

[130] H. Chen, H. Wu, X. Zhou, C. Gao, Agent-based trust model in wireless sensor networks, in: Eighth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD 2007), Vol. 3, IEEE, 2007, pp. 119–124.

[131] Z. Yao, D. Kim, Y. Doh, Plus: Parameterized and localized trust management scheme for sensor networks security, in: 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, IEEE, 2006, pp. 437–446.

[132] J. Hur, Y. Lee, H. Youn, D. Choi, S. Jin, Trust evaluation model for wireless sensor networks, in: The 7th International Conference on Advanced Communication Technology, 2005, ICACT 2005., Vol. 1, IEEE, 2005, pp. 491–496.

[133] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, W.-C. Lee, Using sensorranks for in-network detection of faulty readings in wireless sensor networks, in: Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access, ACM, 2007, pp. 1–8.

[134] L. Gomez, A. Laube, A. Sorniotti, Trustworthiness assessment of wireless sensor data for business applications, in: 2009 international conference on advanced information networking and applications, IEEE, 2009, pp. 355–362.

[135] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, Y.-J. Song, Group-based trust management scheme for clustered wireless sensor networks, IEEE transactions on parallel and distributed systems 20 (11) (2008) 1698–1712.

[136] Y. Zhou, T. Huang, W. Wang, A trust establishment scheme for cluster-based sensor networks, in: 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2009, pp. 1–4.

[137] B. M. David, T. de Sousa Jr, A bayesian trust model for the mac layer in ieee 802.15. 4 networks, in: I2TS 2010-9th International Information and Telecommunication Technologies Symposium, 2010.

[138] F. Bao, I.-R. Chen, Trust management for the internet of things and its application to service composition, in: IEEE WoWMoM 2012 Workshop on the Internet of Things: Smart Objects and Services, IEEE, 2012, pp. 1–6.

[139] R. Chen, F. Bao, J. Guo, Trust-based service management for social internet of things systems, IEEE transactions on dependable and secure computing 13 (6) (2015) 684–696.

[140] R. Chen, J. Guo, D.-C. Wang, J. J. Tsai, H. Al-Hamadi, I. You, Trust-based service management for mobile cloud iot systems, IEEE Transactions on Network and Service Management 16 (1) (2018) 246–263.

[141] P. Karkazis, H. C. Leligou, L. Sarakis, T. Zahariadis, P. Trakadas, T. H. Velivassaki, C. Capsalis, Design of primary and composite routing metrics for rpl-compliant wireless sensor networks, in: 2012 International Conference on Telecommunications and Multimedia (TEMU), IEEE, 2012, pp. 13–18.

[142] P. Karkazis, I. Papaefstathiou, L. Sarakis, T. Zahariadis, T.-H. Velivassaki, D. Bargiotas, Evaluation of rpl with a transmission count-efficient and trust-aware routing metric, in: Communications (ICC), 2014 IEEE International Conference on, IEEE, 2014, pp. 550–556.

[143] S. Seeber, A. Sehgal, B. Stelte, G. D. Rodosek, J. Schönwälder, Towards a trust computing architecture for rpl in cyber physical systems, in: Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), IEEE, 2013, pp. 134–137.

[144] D. Airehrour, J. Gutierrez, S. K. Ray, Securing rpl routing protocol from blackhole attacks using a trust-based mechanism, in: 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, 2016, pp. 115–120.

[145] D. Airehrour, J. A. Gutierrez, S. K. Ray, Sectrust-rpl: A secure trust-aware rpl routing protocol for internet of things, Future Generation Computer Systems (2018).

[146] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, P. Herrmann, A trust-based resilient routing mechanism for the internet of things, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ACM, 2017, p. 27.

[147] A. Lahbib, K. Toumi, S. Elleuch, A. Laouiti, S. Martin, Link reliable and trust aware rpl routing protocol for internet of things, in: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), IEEE, 2017, pp. 1–5.

[148] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, A trust-based intrusion detection system for mobile rpl based networks, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2017, pp. 735–742.

[149] S. Y. Hashemi, F. S. Aliee, Dynamic and comprehensive trust model for iot and its integration into rpl, The Journal of Supercomputing (2018) 1–30.

[150] V. Kiran, S. Rani, P. Singh, Trust based defence system for ddos attack detection in rpl over internet of things, INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY 18 (12) (2018) 239–245.

[151] Y. B. Saied, A. Olivereau, D. Zeghlache, M. Laurent, Trust management system design for the internet of things: A context-aware and multi-service approach, Computers & Security 39 (2013) 351–365.

[152] L. Gu, J. Wang, B. Sun, Trust management mechanism for internet of things, China Communications 11 (2) (2014) 148–156.

[153] N. B. Truong, H. Lee, B. Askwith, G. M. Lee, Toward a trust evaluation mechanism in the social internet of things, Sensors 17 (6) (2017) 1346.

[154] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of cloud computing and internet of things: a survey, Future generation computer systems 56 (2016) 684–700.

[155] M. Dabbagh, A. Rayes, Enhanced cloud demand prediction for smart data centers, uS Patent App. 14/868,224 (Mar. 30 2017).

[156] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, D. Woods, Cloud-trust? a security assessment model for infrastructure as a service (iaas) clouds, IEEE Transactions on Cloud Computing (3) (2017) 523–536.

[157] B. Kantarci, H. T. Mouftah, Mobility-aware trustworthy crowdsourcing in cloud-centric internet of things, in: 2014 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2014, pp. 1–6.

[158] T. Wu, Q. Yang, Y. He, A secure and rapid response architecture for virtual machine migration from an untrusted hypervisor to a trusted one, Frontiers of Computer Science 11 (5) (2017) 821–835.

[159] S. M. Sajjad, M. Yousaf, Security analysis of ieee 802.15. 4 mac in the context of internet of things (iot), in: Information Assurance and Cyber Security (CIACS), 2014 Conference on, IEEE, 2014, pp. 9–14.

[160] Y. M. Amin, A. T. Abdel-Hamid, A comprehensive taxonomy and analysis of ieee 802.15. 4 attacks, Journal of Electrical and Computer Engineering 2016 (2016) 4.

[161] C. Balarengadurai, S. Saraswathi, Comparative analysis of detection of ddos attacks in ieee 802.15. 4 low rate wireless personal area network, Procedia Engineering 38 (2012) 3855–3863.

[162] IEEE, Local and metropolitan area networks—specific requirements—part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans), IEEE Standard for Information Technology (2006).

[163] R. Sokullu, O. Dagdeviren, I. Korkmaz, On the ieee 802.15. 4 mac layer attacks: Gts attack, in: Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on, IEEE, 2008, pp. 673–678.

[164] S. Saleem, S. Ullah, K. S. Kwak, A study of ieee 802.15. 4 security framework for wireless body area networks, Sensors 11 (2) (2011) 1383–1395.

[165] C. P. O'Flynn, Message denial and alteration on ieee 802.15. 4 low-power radio networks, in: New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on, IEEE, 2011, pp. 1–5.

[166] N. Djedjig, D. Tandjaoui, F. Medjek, Trust-based rpl for the internet of things, in: 2015 IEEE Symposium on Computers and Communication (ISCC), IEEE, 2015, pp. 962–967.

[167] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, New trust metric for the rpl routing protocol, in: Information and Communication Systems (ICICS), 2017 8th International Conference on, IEEE, 2017, pp. 328–335.

[168] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, Trust-aware and cooperative routing protocol for iot security, Journal of Information Security and Applications 52 (2020) 102467.

[169] M. Singh, S. Chauhan, Operating system based compliance validation of trusted computing.

[170] S. Raza, L. Wallgren, T. Voigt, Svelte: Real-time intrusion detection in the internet of things, Ad hoc networks 11 (8) (2013) 2661–2674.

[171] P. Pongle, G. Chavan, Real time intrusion and wormhole attack detection in internet of things, International Journal of Computer Applications 121 (9) (2015).

[172] S. Marti, T. J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM, 2000, pp. 255–265.

[173] J.-M. Seigneur, Collaborative Computer Security and Trust Management, IGI Global, 2009.

[174] J. Golbeck, Trust on the world wide web: a survey, Foundations and Trends in Web Science 1 (2) (2006) 131–197.

[175] G. Theodorakopoulos, J. S. Baras, On trust models and trust evaluation metrics for ad-hoc networks, IEEE Journal on selected areas in Communications 24 (LCA-ARTICLE-2007-016) (2006) 318–328.

[176] C. Zhang, X. Zhu, Y. Song, Y. Fang, A formal study of trust-based routing in wireless ad hoc networks, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[177] Y. Yang, J. Wang, Design guidelines for routing metrics in multihop wireless networks, in: INFOCOM 2008. The 27th conference on computer communications. IEEE, IEEE, 2008, pp. 1615–1623.

[178] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: A survey, ACM Transactions on Sensor Networks (TOSN) 8 (4) (2012) 34.

[179] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on wireless communications 1 (4) (2002) 660–670.

[180] D. S. J. De Couto, High-throughput routing for multi-hop wireless networks, Ph.D. thesis, Massachusetts Institute of Technology (2004).

[181] N. Tsiftes, J. Eriksson, A. Dunkels, Low-power wireless ipv6 routing with contikirpl, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, 2010, pp. 406–407.

[182] X. Liu, Z. Sheng, C. Yin, F. Ali, D. Roggen, Performance analysis of routing protocol for low power and lossy networks (rpl) in large scale networks, IEEE Internet of Things Journal 4 (6) (2017) 2172–2185.

[183] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, I. Ahmedy, Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol, IEEE Access 6 (2018) 16623–16638.

[184] F. Y. Yavuz, D. Ünal, E. Gül, Deep learning for detection of routing attacks in the internet of things, International Journal of Computational Intelligence Systems 12 (1) (2018) 39–58.

[185] G. J. Mailath, L. Samuelson, Repeated games and reputations: long-run relationships, Oxford university press, 2006.

[186] J. Ratliff, Game theory.
URL http://virtualperfection.com/gametheory

[187] R. Axelrod, et al., The evolution of strategies in the iterated prisoner's dilemma, The dynamics of norms (1987) 1–16.

[188] R. Axelrod, W. D. Hamilton, The evolution of cooperation, science 211 (4489) (1981) 1390–1396.

[189] R. Axelrod, The evolution of cooperation (1985).

[190] Iterated prisoner's dilemma simulation software.
URL http://www.lifl.fr/IPD

Abstract: The Internet of Things (IoT) concept has attracted significant attention from both industrial and research communities. Nevertheless, the IoT is facing many security issues such as authentication, availability, privacy, and trust management. Indeed, establishing trust relationships between nodes in IoT represents a primary security milestone to have a reliable system that excludes malicious nodes. However, trust management in an IoT constrained, and ubiquitous environment represents a challenge. In this thesis, we study the issue of designing secure trust management protocols taking into account the specificities of IoT. In the IoT Low-Power and Lossy Networks (LLNs), the sensing or perception layer is based on the IEEE 802.15.4 protocol. On the other hand, the network layer is based on the Routing Protocol for LLNs (RPL) that have been standardised to fulfil the routing requirements in such networks. Both protocols suffer from security issues and lack strong security solutions, especially in the field of trust management. In response to the above security issues, a new trust management scheme has been introduced to address the MAC unfairness attacks against the IEEE 802.15.4 MAC layer. Furthermore, a Metric-based RPL Trustworthiness Scheme (MRTS) has been proposed to enhance RPL security by introducing a new trust metric and a new trust-based objective function, thus, rendering RPL more efficient in compromised networks. Along this dissertation, MRTS went through three refinement versions, which were the subject of a succession of three contributions. The efficiency of MRTS has been validated through extensive simulation experiments under different scenarios using the open-source Contiki-Cooja simulator. The results demonstrate significant performance enhancements in terms of routing security, power consumption, packet delivery ratio, and throughput. Besides, mathematical analyses prove that MRTS meets the requirements of consistency, optimality, and loop-freeness and that the proposed trust-based routing metric has the isotonicity and monotonicity properties required for a routing protocol. Furthermore, game theory mathematical analyses and evolutionary simulation results show that MRTS, as a strategy, is an efficient approach in promoting the stability and the evolution of IoT networks.
Keywords: Internet of Things, IoT, Wireless Sensor Network, IoT security, Trust Management, Trust, RPL, Secure Routing.

Résumé: L'Internet des objets (IoT) a attiré une attention considérable de la part des communautés industrielles et de recherche. Cependant, l'IoT est confronté à de nombreux problèmes de sécurité tels que l'authentification, la disponibilité, la confidentialité et la gestion de la confiance. En effet, l'établissement de relations de confiance entre les nœuds dans l'IoT représente une étape de sécurité principale pour disposer d'un système fiable qui exclut les nœuds malveillants. Cependant, la gestion de la confiance dans un environnement IoT contraint et omniprésent représente un défi. Dans cette thèse, nous étudions la problématique de la conception de protocoles de gestion de confiance sécurisés prenant en compte les spécificités de l'IoT. Dans les réseaux IoT à faible consommation et avec perte (LLN), la couche de détection ou de perception est basée sur le protocole IEEE 802.15.4. D'autre part, la couche réseau est basée sur le protocole de routage pour les LLN (RPL) qui a été normalisé pour répondre aux exigences de routage dans de tels réseaux. Les deux protocoles souffrent de problèmes de sécurité et manquent de solutions de sécurité solides, en particulier dans le domaine de la gestion de la confiance. En réponse aux problèmes de sécurité ci-dessus, un nouveau schéma de gestion de la confiance a été introduit pour traiter les attaques d'injustice MAC contre la couche MAC IEEE 802.15.4. En outre, un schéma de fiabilité RPL basé sur des métriques (MRTS) a été proposé pour améliorer la sécurité RPL en introduisant une nouvelle métrique de confiance et une nouvelle fonction objectif basée sur la confiance, rendant ainsi RPL plus efficace dans les réseaux compromis. Au cours de cette thèse, MRTS est passé par trois versions, qui ont fait l'objet d'une succession de trois contributions. L'efficacité du MRTS a été validée par des expériences de simulation approfondies dans différents scénarios à l'aide du simulateur open source Contiki-Cooja. Les résultats démontrent des améliorations significatives des performances en termes de sécurité de routage, de consommation d'énergie, de taux de livraison de paquets et de débit. En outre, des analyses mathématiques prouvent que MRTS répond aux exigences de cohérence, d'optimalité et d'absence de boucle et que la métrique de routage basée sur la confiance proposée possède les propriétés d'isotonicité et de monotonie requises pour un protocole de routage. De plus, les analyses mathématiques de la théorie des jeux et les résultats des simulations évolutives montrent que le MRTS, en tant que stratégie, est une approche efficace pour promouvoir la stabilité et l'évolution des réseaux IoT.
Mots-clés : Internet des objets, IoT, réseau de capteurs sans fil, sécurité IoT, Trust Management, Trust, RPL, Secure Routing.

الملخص: اجتذب مفهوم إنترنت الأشياء اهتمامًا كبيرًا من المجتمعات الصناعية والبحثية. لكن، تواجه إنترنت الأشياء العديد من مشكلات الأمان مثل المصادقة والتوافر والخصوصية وإدارة الثقة. في الواقع، يمثل إنشاء علاقات ثقة بين عقد شبكة إنترنت الأشياء معلمًا أمنيًا أساسيًا للحصول على نظام موثوق به يستبعد العقد الضارة. ومع ذلك، فإن إدارة الثقة في بيئات إنترنت الأشياء المقيدة والمنتشرة في كل مكان تمثل تحديًا حقيقيًا. في هذه الرسالة، ندرس مسألة تصميم بروتوكولات إدارة الثقة الآمنة مع مراعاة خصوصيات إنترنت الأشياء. في شبكات إنترنت الأشياء منخفضة الطاقة والمفقودة، تعتمد طبقة الاستشعار أو الإدراك على بروتوكول IEEE 802.15.4. من ناحية أخرى، تعتمد طبقة الشبكة على بروتوكول التوجيه للشبكات منخفضة الطاقة والمفقودة التي تم توحيدها للوفاء بمتطلبات التوجيه في مثل هذه الشبكات. كلا البروتوكولين يعانيان من مشاكل أمنية ويفتقران إلى حلول أمنية قوية، خاصة في مجال إدارة الثقة. استجابةً لقضايا الأمان المذكورة أعلاه، تم تقديم مخطط إدارة ثقة جديد لمعالجة هجمات الظلم ضد طبقة التحكم بالوصول إلى الوسط (IEEE 802.15.4 MAC). علاوة على ذلك، تم اقتراح نظام الجدارة بالثقة لبروتوكول التوجيه القائم على المقاييس لتعزيز أمان بروتوكول التوجيه من خلال تقديم مقياس ثقة جديد ووظيفة موضوعية جديدة قائمة على الثقة، وبالتالي، جعل بروتوكول التوجيه أكثر كفاءة في الشبكات المعرضة للخطر. في هذه الرسالة، مر نظام الجدارة بالثقة لبروتوكول التوجيه القائم على المقاييس بثلاثة إصدارات، والتي كانت موضوعًا لسلسلة من ثلاث مساهمات. تم التحقق من كفاءة نظام الجدارة بالثقة لبروتوكول التوجيه القائم على المقاييس من خلال تجارب محاكاة مكثفة في سيناريوهات مختلفة باستخدام محاكي Contiki-Cooja مفتوح المصدر. توضح النتائج تحسينات كبيرة في الأداء من حيث أمان التوجيه واستهلاك الطاقة ونسبة تسليم الحزمة والإنتاجية. إلى جانب ذلك، تثبت التحليلات الرياضية أن نظام الجدارة بالثقة لبروتوكول التوجيه القائم على المقاييس تفي بمتطلبات الاتساق والأمثل وخالية من الحلقات وأن مقياس التوجيه المقترح المستند إلى الثقة له خصائص تساوي التوتر والرتابة المطلوبة لبروتوكول التوجيه. علاوة على ذلك، تُظهر التحليلات الرياضية لنظرية الألعاب ونتائج المحاكاة التطورية أن نظام الجدارة بالثقة لبروتوكول التوجيه القائم على المقاييس، كاستراتيجية، هي نهج فعال في تعزيز استقرار وتطور شبكات إنترنت الأشياء.
الكلمات الرئيسية: إنترنت الأشياء، شبكة الاستشعار اللاسلكية، أمان إنترنت الأشياء، إدارة الثقة، الثقة، بروتوكول التوجيه، التوجيه الآمن.