KARADENIZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

AUTHENTICATION IN IOT: 6TİSCH SECURE JOIN USING DISTRIBUTED STORAGE FOR THE KEY ESTABLISHMENT

MASTER THESIS

YOSR BOUMAIZA

MARCH 2019
TRABZON

KARADENİZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

AUTHENTICATION IN IoT: 6TiSCH SECURE JOIN USING DISTRIBUTED STORAGE FOR THE KEY ESTABLISHMENT

Yosr BOUMAIZA

This thesis is accepted to give the degree of

"MASTER OF SCIENCE"

By

The Graduate School of Natural and Applied Sciences at

Karadeniz Technical University

The Date of Submission : **04/01/2019**

The Date of Examination : **15/03/2019**

Supervisor : **Dr. Prof. Sedat GÖRMÜŞ**

Trabzon 2019

KARADENIZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

Yosr BOUMAIZA

AUTHENTICATION IN IoT: 6TiSCH SECURE JOIN USING DISTRIBUTED STORAGE FOR THE KEY ESTABLISHMENT

Has been accepted as the thesis of

MASTER OF SCIENCE

After the Examination by the Jury Assigned by the Administrative Board of the Graduate School of Natural and Applied Science with the Decision Number 1792 dated 19 / 02 / 2019

Approved By

Chairman	**: Prof. Dr. Abdulsamet HAŞILOĞLU**	.....
Member	**: Dr. Öğr. Üyesi Sedat GÖRMÜŞ**	.....
Member	**: Dr. Öğr. Üyesi İbrahim SAVRAN**	.....

Prof. Dr. Asim KADIOĞLU

Enstitü Müdürü

ACKNOWLEDGEMENT

This thesis is submitted to the Karadeniz Technical University, Institute of Science, as the concluding part of my Master Program of Computer Engineering and was carried out in the period between September 2016 and January 2019. Additionally, it accounts for 30 ECTs towards the completion of my MSc. in Engineering.

In this work, we presented the technology of IoT, such as its applications domain, the enabling technologies and its requirements. We worked more specifically on the security of IoT networks. Taking into consideration the architectural structure of the small devices, the protocols used to ensure secure communication are studied. Furthermore, the security mechanisms in the IoT layers and the difficulties encountered are mentioned. We also mentioned some related works. Authentication is one of the most significant security requirements. In this context, a new authentication mechanism for 6TiSCH protocol has been suggested. The implementation and performance results were discussed taking into account security requirements.

First of all, I would like to sincerely thank my respected thesis supervisor, Dr. Prof. Sedat GÖRMÜŞ, for valuable discussions and comments, as well as constructive feedback and suggestions throughout the realization of this thesis. Without his guidance and assistance, this thesis would never have come into being.

I would like to express my deep gratitude to Hakan AYDIN, PhD student and a good friend, for keeping me on the track when I got lost in the code programming and supporting me during this thesis.

I would like to give my most deep thanks to my family for all the sacrifices they made for me and for their endless support. Without forgetting the most precious person in my life, I would thank my husband, who gave me so much love, support, encouragement and confidence during my entire studies.

Thank You Very Much

Yosr BOUMAIZA

Trabzon 2019

DECLARATION

This is to declare that the research work in this thesis entitled “Authentication in IoT: 6TiSCH secure join using distributed storage for the key establishment” is a record of an authentic research work carried out by me, Yosr BOUMAIZA. Neither this thesis nor any part of it has been submitted to any other University of Institute for the award of any degree or diploma, except where due reference acknowledgement have been given in the text. 15.03.2019

Yosr BOUMAIZA

CONTENT

Page Number

Master Thesis

SUMMARY

Authentication in IoT: 6tisch secure join using distributed storage for the key establishment

Yosr BOUMAIZA

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Dr. Prof. Sedat GÖRMÜŞ
2019, 81 Pages

Nowadays, the Internet of Things (IoT) is among the most important topics of research. It is essentially an interconnection of a great number of small devices over the Internet. These small devices are generally equipped with low power radios and suffer from numerous constraints, notably limited memory, limited power, and low computational capacity. Developing security mechanisms for such Internet enabled low power networks pose a real challenge. Authentication and access control are significant and critical functionalities in the context of IoT to permit secure communication between devices. For the IoT, it is not possible to carry out directly the classic security countermeasures since most of the existing mechanisms are designed for devices with higher computational capacity as compared to IoT devices.

Security in IoT is often bootstrapped with the aid of cryptographic key distribution amongst devices. In this case, one of the greatest research challenges is to assure secure key storage via sensors taking their limited resources into account. This thesis introduces an enhancement of the existing secure bootstrapping mechanism of the standard IETF 6TiSCH protocol in which a distributed data storage is used for the management of the authentication keys. With this distributed approach, the objective is to minimize the communication overhead of the IETF 6TiSCH authentication mechanism, enable an efficient authentication process and enhance energy efficiency of the network through holding the authentication parameters at the edge of the IoT network.

Key Words: IoT, IETF 6TiSCH, Bootstrapping, Authentication, Key distribution, Distributed storage.

Yüksek Lisans Tezi

ÖZET

Authentication in IoT: 6tisch secure join using distributed storage for the key establishment

Yosr BOUMAIZA

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr.Öğr.Üyesi Sedat GÖRMÜŞ
2019, 81 Sayfa

Nesnelerin İnterneti (IoT) günümüzün en önemli araştırma konuları arasındadır. Temeli, İnternet üzerinden çok sayıda küçük cihazın bağlantısına dayanmaktadır.

Bu küçük cihazlar genellikle düşük güçlü radyo ile donatılmıştır ve sayısız kısıtlamadan, özellikle de sınırlı bellek, sınırlı güç ve düşük hesaplama kapasitesinden dolayı kısıtlı işlem gücüne sahiptir.

Bu küçük cihazlar genellikle düşük güçlü radyo ile donatılmıştır ve sayısız kısıtlamadan, özellikle de sınırlı bellek, sınırlı güç ve düşük hesaplama kapasitesinden dolayı kısıtlı işlem gücüne sahiptir.

IoT için doğrudan klasik güvenlik önlemlerini uygulamak mümkün değildir, çünkü mevcut mekanizmaların çoğu IoT cihazlarına kıyasla daha yüksek hesaplama kapasitesine sahip cihazlar için tasarlanmıştır.

IoT'deki güvenlik, çoğu zaman cihazlar arasında kriptografik anahtar dağıtımı yardımı ile gerçekleştirilir. Bu durumda, en büyük araştırma zorluklarından biri, sınırlı kaynaklara sahip olan sensörler aracılığıyla güvenli anahtar depolamanın sağlanmasıdır. Bu tezde, kimlik doğrulama anahtarlarının yönetimi için dağıtık bir anahtar depolamanın kullanıldığı IETF 6TiSCH protokolünün mevcut güvenli önyükleme mekanizmasına yeni bir yaklaşım sunmaktadır.

Bu yaklaşımda amaç, IETF 6TiSCH kimlik doğrulama mekanizmasının iletişim yükünü en aza indirgemek, verimli bir kimlik doğrulama işlemini mümkün kılmak ve IoT ağının kenarında kimlik doğrulama parametrelerini tutarak ağın enerji verimliliğini arttırmaktır.

Anahtar Kelimeler: IoT, IETF 6TiSCH, Önyükleme, Kimlik Doğrulama, Anahtar dağıtımı, Dağıtık depolama

LIST OF FIGURES

Page Number

LIST OF TABLES

Page Number

ABBREVIATIONS

IoT	Internet of Things
IoE	Internet of Everything
RFID	Radio Frequency Identification
ITU	International Telecom Union
WSN	Wireless Sensor Network
M2M	Machine To Machine
IPv6	Internet Protocol version 6
6LoWPAN	IPv6 Low Power Wireless Personal Area Networks
NFC	Near Field Communication
SOA	Service Oriented Architectures
GPS	Global Positioning Systems
GIS	Geographic Information Systems
IPv4	Internet Protocol version 4
RF	Radio Frequency
WoT	Web of Things
RPL	Routing Protocol for Low power and Lossy networks
IP	Internet Protocol
SDN	Software Defined Networking
XMPP	Extensible Messaging and Presence Protocol
AMQP	Advanced Message Queuing Protocol
MQTT	Message queue Telemetry Transport
CoAP	Constrained Application Protocol
HTTP	Hypertext Transfer Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IEEE	Institute of Electrical and Electronic Engineers
AES	Advanced Encryption Standard
CSMA-CA	Carrier Sense Multiple Access/Collision Avoidance
FFD	Full Function Device

RFD	Reduced Function Device
PAN	Personal Area Network
P2P	Peer to Peer
MAC	Media Access Control
LR-WPAN	Low-Rate Wireless Personal Area Network
MTU	Maximum Transmission Unit
IETF	Internet Engineering Task Force
IPsec	Internet Protocol Security
DTLS	Datagram Transport Layer Security
6BR	6LoWPAN Border Router
QoS	Quality of Service
TTL	Time To Line
EUI-64	Extended Unique Identifier - 64
6LoWPAN-GHC	Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks
ROLL	Routing Over Low-Power and Lossy Networks
DODAG	Destination Oriented Directed Acyclic Graph
DIO	DODAG Information Object
DIS	DODAG Information Sollicitation
DAO	DODAG Destination Avertissement Object
DAO-AKC	DAO Acknowledgment
URT	Uniform Resource Identifier
TLS	Transport Layer Security
API	Application Programming Interface
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4
DOS	Denial of Service
DDOS	Distributed Denial of Service
AKS	Asymmetric Key Schemes
PKC	Public Key Cryptography
DH	Diffie Hellman
RBAC	Role-Based Access Control
ABAC	Attribute Based Access Control

SSL	Secure Sockets Layer
CPU	Central Processing Unit
LLN	Low power Lossy Networks
OTP	One Time Password
KDC	Key Distribution Center
MAC	Message Authentication Code
EAP	Extensible Authentication Protocol
LAN	Local Area Network
EAPOL	EAP over LAN
ECC	Elliptic Curve Cryptography
ROM	Read Only Memory
RAM	Random Access Memory
BSD	Berkeley Source Distribution
ICMP	Internet Control Message Protocol
KB	Kilobyte
UDGM	Unit Disk Graph Medium
TX	Transit
RX	Reception
DGRM	Directed Graph Radio Medium
MRM	Multipath Ray-tracer Medium
JRC	Join Registrar/Coordinator
EB	Enhanced Beacon
P	Pledge
ASN	Absolute Slot Number
OSCOAP	Object Security for CoAP
MIC	Manufacturer Installed Certificate
JP	Join Proxy
JN	Joined Node
P-JRC	Proxy Join Registrar/Coordinator
OF	Objective Function
SHA	Secure Hash Algorithm

NSA	National Security Agency
REST	Representational State Transfer
RSVP	ReSerVation Protocol
NSIS	Next Steps in Signalling

1. INTRODUCTION

At the turn of century, a new concept appeared named the Internet of Things (IoT) [2], which is a technological innovation representing the future of communication and computing. This new concept of globally connected small devices relies on new developments in areas such as wireless sensors and nanotechnology.

This concept would enable anything to communicate to everything, helping to make everyday life easier for everybody, through remote monitoring and controlling industrial systems, our homes, our health, and so on [3]. It integrates a great number of end systems allowing mutual communication and open access to data among them.

The principal benefit of the Internet of Things is evident; it is effective in data gathering and exchange using minimal energy during this exchange. Additionally, IoT supplies cost-efficient methods for application areas such as energy distribution and environmental monitoring [14].

The interconnected objects, which forms the IoT network, are equipped with low power radios and suffer from numerous constraints, notably limited memory, limited power, and low computational capacity.

Two of the most challenging subjects in such interconnected system are security and privacy. In computer networks, the essential basics to resolve security and privacy problems are authentication and access control technologies. They are able to preclude unauthorized users from obtaining access to data, inhibit allowable users from accessing resources in an unauthorized sort, and allow legitimate users to get access to data in an authorized way. While constructing an Internet of Things system, it is very important to take into account the scalability, the energy usage, storage, and the quality of service [51].

Security in IoT is often bootstrapped with the aid of cryptographic key distribution amongst devices. This requires a significant number of communication overhead since the devices bootstrapping to the network has to authenticate themselves to a central entity. An alternative to this centralized bootstrapping approach is that the authentication process is handled within the network by the selected proxy authentication nodes. In this case, one of

the greatest research challenges is to assure secure key storage in these selected devices taking their limited resources into account. This work introduces an enhancement to the existing secure bootstrapping mechanism of the standard IETF 6TiSCH protocol in which a distributed data storage is used for the management of the authentication keys distributed within the network. With this distributed approach, the objective is to minimize the communication overhead of the IETF 6TiSCH authentication mechanism, enable an efficient authentication process and enhance energy efficiency of the network through holding the authentication parameters at the edge of the IoT network.

2. BACKGROUND

The IoT is an emerging paradigm in the world of computer networks. It can be defined as an evolution of the Internet for the inclusion of all objects and places in our surroundings (refrigerators, thermostat, houses, vehicles, roads, etc.). The promising concept of IoT will simplify our lives, make us saving time, discharge our brain from memorizing logistical data (routes, medication times, etc.). Thus, ubiquitous access to different types of information would enable lifestyle sophistication and significant improvement in the quality of services in different areas. IoT, which is a new wave of the Internet, is actually a nascent (growing) part of the Internet of the future, called the Internet of all objects or IoE (Internet of Everything). It aims to connect people, data and all objects, so that there is a fusion between the digital world (virtual) and the real world (physical). The objects of the physical world will be incorporated into the virtual world of the Internet. This calls for new trends and innovations in terms of communications architectures or the presentation and operation of services. [4]

This chapter is devoted to the presentation of the Internet of Things domain and its related aspects.

2.1. Background for the Internet of Things

2.1.1. Definition of IoT

The emergence of the Internet of Things is only a result of convergence between numerous technologies, namely the Internet, wireless communication, embedded systems, microelectronic systems, and nanotechnology [1].

The IoT is an expression that has been coined in the late 20th century by a man named Keving Asthon. He is regarded as the first person who used the term IoT in a presentation concerning the connecting RFID tags to the Internet at Procter and Gamble in 1999 [Ash 09] [2] [3].

A precise definition of Internet of Things cannot be established although many have attempted to put the concept of the Internet of Things into words. The IoT comprises different objects, each made for different purpose, manufactured by different vendors and

with different capabilities, complexities, and bit-rates, which have a shared way of communicating for authorizing transfer of data, where this data is understood by two or more objects in order to make a process more effective [5].

The IoT vision will introduce a new dimension to information and communication technologies: besides the two temporal and spatial dimensions that let people connect from anywhere at any time, we will have a new dimension "object" that will let them connect to any object [7]. An annual report of IoT is published by the International Telecom Union (ITU) in 2005 where they extended the concept posing the foreground “Anytime, Any Place and Any Thing Connection” [8]. Figure 1 shows this new dimension of connectivity.

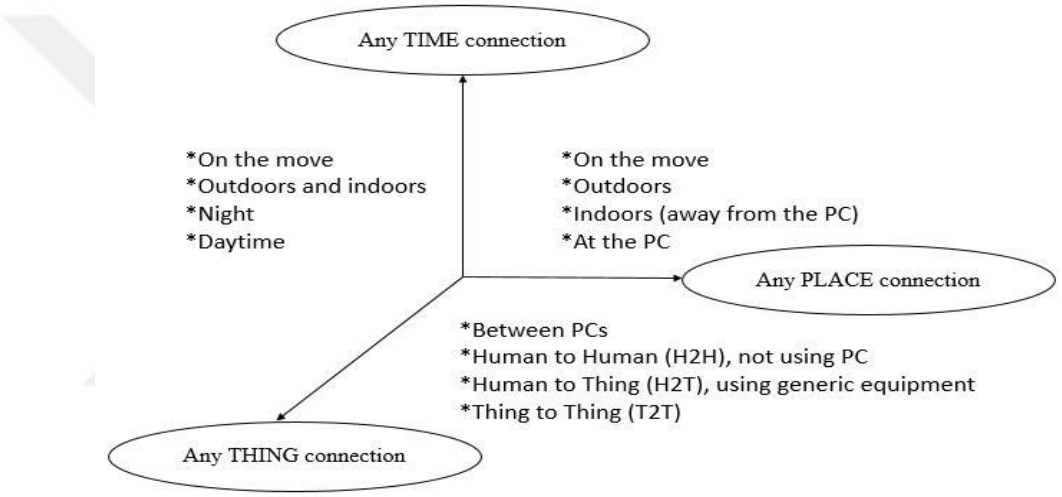

Figure 1. New dimension of connectivity

IoT technology has become feasible and friendly thanks to the recent development and acceptance of technologies like wireless sensor networks (WSN) and radio frequency identification (RFID) [5].

The IoT as it is considered as the intermediate between the physical and numerical world enables user to collect data from and monitor ordinary objects. We can say that a connected device is the end-point. Generally, the essential task of IoT are to collect, treat, and submit data in an independent way. By possessing end-nodes equipped with sensors and modules for communication, they can determine or compute location, temperature, and other features, and transfer data to an appropriate device or server. The evaluation of the data can then be executed, and worthy information can be offered to the user. The IoT also eases

actuation. Users, with actuation, can operate the end-points remotely by giving them instructions carried over the Internet [9].

In IoT, we often consider objects (sensors) that are unable to join the Internet without an intermediary, which is named as the gateway device. Indeed, the Internet is not dimensioned to manage the addressing of many connected objects. On the other hand, IPv6 is a protocol often too heavy to be exploited directly by small devices. Today IPv6 is used via the 6LowPAN protocol (IPv6 Low Power Wireless Personal Area Networks) which is operated over IEEE 802.15.4 PHY layer protocol [10]. Gateways also translate proprietary protocols to a standard internet protocol and some gateways can act as network aggregators. The networks that connect them must also meet these needs of lightness, simplicity and low cost. Thus, telecom operators must adapt to accommodate such low bitrate traffic since carrying a few kilobytes per sensor over the cellular network can be very resource intensive for cellular networks.

Everything in an Internet of Things environment will be connected, where each person and object in the real world would be addressable, locatable and legible through his/her virtual existence on the Internet. Consequently, security is substantial and without robust mechanisms of security, offensives and crashes in Internet of Things systems will assure that the risks surpass the potential advantages. Moreover, the ubiquitous and unobtrusive ways in which Internet of Things devices gather and treat sensitive data further enhance the importance of security [5].

2.1.2. Application Domain of the IoT

The Internet of Things is not just a huge ensemble of interconnected smart objects, but it is also, and more importantly, the applications that are actually the reason for this new wave of connectivity on the Internet [1]. The IoT will touch almost every aspect of our daily lives. This will allow the emergence of intelligent spaces around an omnipresent computing [6]. The existence of smart objects with new possibilities of automatic and intelligent communications will significantly improve the lifestyle of people as well as the quality of services in various fields through high degrees of autonomy and intelligence. In this section, we cite the featured applications of IoT [1].

Logistic:

The logistic is an important sector that can take advantage of the changes of the IoT. Due to RFID and NFC, real-time tracking is possible at any stage of the supply chain. This ensures that the business can react to any change in the supply change immediately. This enables businesses to plan more efficiently so that a full safety stock will be useless[11] [12].

The agriculture:

In this area, interconnected IoT sensor networks can be employed to control the environment of cultures. This will allow better decision support in agriculture, including optimizing irrigation water and agricultural work planning. These networks can also be used to fight against air, soil and water pollution and improve the quality of the environment in general [6].

Military applications:

The Internet of Things is a fertile area for both civilian applications and military applications. In the field of defense, sensors and nano-drones connected to the Internet make it possible to envisage sophisticated applications for the exploration, the monitoring of battlefields and borders as well as the pursuit of enemy forces. Military forces tend to use proprietary infrastructure for connectivity and communications. By transiting to the Internet, it will be possible to use cloud infrastructures, which may offer operational flexibility [1].

Transportation:

IoT, in this domain, may help existing efforts around smart vehicles. The inter-vehicle communication is expected to enable application such as congestion avoidance, collision detection and autonomous driving [12]. Thus, IoT can be thought as an extension of the so called "intelligent transport systems". IoT networks can be employed to increase the efficiency of traffic management, road safety, improve the journey comfort and reduce the vehicle energy consumption [6].

Industrial:

The industrial sector is another area that will be revolutionized by the IoT. The number of wireless sensor devices, RFID tags, and embedded controllers are growing significantly in industrial production systems, supply chains and even in products. This helps companies to enhance the goodness of their manufacturing process and provide a more competitive after-sales service. Thus, factories connected to the Internet are more productive, efficient and intelligent than those that are not.

Smart Cities:

IoT will allow a better control of city services such as water, gas and electricity by enabling real time monitoring and control of these services. Sensors can be used to improve the management of car parks and urban traffic resulting in significant CO2 emission reductions [6].

Smart health:

IoT will have many applications in the healthcare sector where the goal is to be able to prevent serious situations and remotely follow patients with chronic diseases and act quickly if necessary. The body sensors implanted in the patient's body gather information about medical parameters, such as temperature, blood sugar, heartbeat rhythm or even blood pressure [1].

Smart Home:

The house of the future will be an object connected to the Internet, accessible remotely by its owners via smartphones, tablet or connected computers. The door, the television, the thermostat, the refrigerator, the umbrellas, the watches, etc. will be connected to the Internet. For example, a connected door may inform the parents when their children return home. Classically, television has been only a receiving terminal. With the connected television, viewers can send and receive e-mails, make phone calls over the Internet, or watch IP based TV content [1].

2.1.3. Advantages and Disadvantages of the IoT

A. Advantages

We give a brief list of advantages of IoT in this section.

- An omnipresent (ubiquitous) access to data generated by everyday devices for a smarter world.
- Improved quality of service and remote monitoring in different fields of application, namely the industrial, medical and other fields.
- Improve productivity and client experience: Connected items transmit reports to their builders pointing out purchaser favourites and customs supporting firms to act more proactively and responsibly which satisfies the demand and customer requirements.

- Time saving is another advantage of IoT. The unnecessary displacements are then replaced by a simple navigation on the web to order products, to control the state of the objects and/or connected places.
- The IoT, in certain applications, allows an improved energy efficiency by automatizing different aspects of daily life including smart electricity use, traffic management and smart city services.

B. Disadvantages:

While it attempts to optimize several aspects of life, certain drawbacks of IoT that should be taken into consideration:

- As the devices within the network are of various fabrications, there can be interoperability issues. Hence, international IoT standards shall be created to alleviate such compatibility challenges.
- IoT will be formed by many devices with differing capabilities. Hence, assuring the reliability of such a complex system can be a challenging task.
- Security and privacy are an important concern even with current internet enabled devices. Mechanisms that will guarantee the security and privacy of user data must be implemented.
- With the advent of IoT technology, improved efficiency in workplace may render some professions obsolete. This may lead to undesired sociological results.
- Having a greater dependency on technology for goods and services may result in unintended consequences when these services break down [1] [13].

2.2. Enabling Technology

Despite the fact that the Internet of Things is a relatively new concept, the technologies that enabled it exists. The evolutions observed by wireless technologies and the field of telecommunication networks and the Internet have opened up new perspectives for IoT technologies. Now we have mature telecommunications technologies that can be developed into IoT solutions [1]. In the next part of this section, we present three of the keys enabling technologies for Internet of Things. They are the Internet protocol version 6 (IPv6), wireless sensor network (WSN) and radio frequency identification (RFID) technologies. It is significant to mention that this is not a complete list of technologies that Internet of Things includes. The following are also a part of the IoT and will not be explored in detail:

intelligent sensing devices, near field communication (NFC), service-oriented architectures (SOA), cloud computing, global positioning systems (GPS), geographic information systems (GIS) and mobile cellular devices [5].

2.2.1. IPv6: Internet Protocol Version 6

It is extensively believed that the volume of devices connected to the Internet is developing promptly and coming into new fields, and the forecast for the future indicates no indications of slowing down. It is widely supposed that there will be 20 billion connected devices by 2020 [3]. IPv4 only supports an address range of 4.2billion, the industry identified the addressing problems with IPv4 before IPv6 addressing has been introduced. This fact also points out how IPv4 is inadequate for Internet of Things solutions with billions of connected devices. For enabling this vision, Internet of Things must use IPv6 addressing. IPv6 addressing can support an address space consisting more than 340 trillion unique addresses. Consequently, this offers a solution for IoT network where each device is expected to have a unique IPv6 address. Therefore, the development of IoT technologies almost universally focuses on IPv6 addressing. Additionally, IPv6 addressing has the following features such as:

- Flow labelling capability: marking of packets belonging to a specific traffic flows.
- Header format simplification: removing certain IPv4 headers.
- Authentication and privacy capabilities: specified extensions to support authentication, data integrity and data confidentiality.
- Advanced support for extensions and options: modify encoding, less strict limits on the length of options and larger flexibility in offering new alternatives.

2.2.2. RFID: Radio Frequency Identification

RFID is one of the key-enabling technologies of the IoT. Although its deployment in the commercial and private areas has been pretty recent, it was first employed to recognize friendly aircrafts during World War II; while it was not as portable and energy efficient back then.

An RFID system includes one or more readers and a set of labels (called tags, markers, identifiers or transponders) with micro-powers. Tags are tiny devices equipped with a chip

containing information and an antenna for radio communication. They are attached to the object we desire to track or gather information about. RFID reader is a device that can detect/recognize the existence of an RFID tag and is capable to read the information stored on them. Additionally, RFID technology enables information to be recovered from tagged objects wirelessly with radio waves. RFID tags may have sensors and actuators to gather information and modify the environment as needed. Furthermore, RFID tags are quite limited in terms of bandwidth, power, processing energy, and memory. RFID technology typically works whereby an RFID reader broadcasts a radio frequency (RF) signal, which a tag receives and converts into energy to power its chip. The tag then transmits its identity back to the reader. Although there are variations, this is how in general RFID technology works. For example, some tags are capable to encrypt the messages they send when interacting with a reader and some can even ignore readers that do not give the proper password. [1] [5]

2.2.3. WSNs: Wireless Sensor Networks

IoT networks is composed of many very small devices able to sense, compute and communicate data. Indeed, the sensors allow the illustration of the dynamic characteristics (temperature, humidity, pressure, movements…) of objects and locations of the physical world in the virtual world represented by the global Internet network. Thus, with the incorporation of sensor networks in the Internet, sensors become servers (service providers) in what is represented to as the WoT (Web of Things). Thereby, the services (applications) of the WSNs are added to all the services and applications of the future Internet that will bring together a diversity of highly heterogeneous networks (hardware and software), subject to different constraints and that are deployed for various applications, in order to have a very sophisticated real world [1]. The WSNs have three possible topologies, which are mesh, cluster tree and star [33]. One common point in these topologies that there is a gateway node called central point. This point connects the sensors to the rest of the network infrastructure [3].

2.3. IoT Specific Requirements

IoT is a concept that can provide effective solutions to the challenges of monitoring and remote surveillance in different areas. In return, IoT raises some challenging questions about its maturity and acceptability. Here we summarize some of the most important requirements specific to IoT solutions [1].

2.3.1. Scalability and Interoperability

Interoperability is one of the biggest challenges in realizing the IoT. It is, in fact, the coexistence of different devices belonging to different vendors communicating to each other reliably. There is a recent trend towards the standardization and unification of protocols in IoT. This is to facilitate the collaboration between connected objects, as well as the coupling with external entities on the Internet.

The number of smart objects that will populate the Internet of the future is expected to be in the order of billions. Hence, the adoption of new mechanisms that effectively support interoperability and scalability is highly recommended [1].

2.3.2. Bandwidth, low Processing and low Power

When it comes to IoT devices, we frequently discuss small-embedded devices designed to attend a single goal such as industrial sensors, wearables, and other applicable devices. Main limitations of these devices can be summarized as size, limited battery capacity and processing power. Thus, improvement and innovation for IoT should account for these limitations [3].

The IoT as an evolution of the existing Internet allows a considerable improvement in our way of life and the way in which intelligent objects in our environment interact with each other and with their users, so that our activities, our goods, our state of health, our expenses… can be controlled effectively and ubiquitously. In this chapter, we mainly discussed the core technologies as well as the featured applications of IoT. We have also highlighted the constraints related to the deployment of IoT and that should be carefully addressed to achieve the predefined objectives and achieve optimal returns.

3. IOT ARCHITECTURE AND PROTOCOLS

Referring to Mr. Pete Lewis where he said; “The IoT is the integration of people, processes and technology with connectable devices and sensors to enable remote monitoring, status, manipulation and evaluation of trends of such devices.” [14]. Nowadays computer systems and computing capabilities are incorporated into nearly each industrial product. Therefore, the Internet of Things reference model has been divided into three layers. On the other hand, to uphold the limited application layer protocols, four IoT standard protocols have been offered.

In this chapter, an architecture for IoT and an overview of these protocols are discussed.

3.1. Architecture of the IoT (protocol stack)

The IoT should not be considered as a utopian concept. In fact, it will be based on several enabling technologies such as NFC, wireless sensors and actuators, RFID, M2M, ultra-wideband or 3 / 4G, IPv6, 6LowPAN and RPL, etc. who should all perform a significant role in the improvement of IoT. The IoT has its roots going back to M2M technologies for remote production process control. This technology has evolved into the concept of the IoT since the advent of IP over mobile cellular networks in the 2000s [6].

We may say, from an architectural point of view, that the Internet of Things is organized into three main layers: the data perception layer, the network layer and thirdly the application layer [1].

3.1.1. The Data Perception Layer

The perception layer, at the lowest level in the hierarchy, is responsible for the capture of data, as well as their identification in their environment. Thus, this layer includes the hardware necessary to achieve the collection of contextual data of connected objects, namely sensors, RFID tags, cameras, GPS, etc. [1]. These sensors are employed to recognize the objects as well as delivery the gathered data to the following layer. Devices aggregate and

upload information to the network layer either immediately or indirectly. It is anticipated that all devices will be IPv6-capable in the future [14].

3.1.2. The Network Layer

The role of this layer is ensuring the reliability of the transmission of the data generated in the perception layer as well as the assurance of connected inter-object connectivity and among smart devices and the other hosts of the Internet. It is expected that the data from the perception layer will be increasing exponentially because the number of objects connected to the Internet is rising rapidly. Thereby, it has proved necessary to put in place mechanisms and equipment for mass storage and processing of these data on the Internet, at low cost. This is indeed guaranteed by cloud services that provide elastic management of storage and processing resources on the data centers residing on the Internet and that are able to effectively absorb the data load generated from the IoT.

3.1.3. The application Layer

This layer, on the other hand, defines intelligent service profiles and data management mechanisms of different types, from different sources. The application layer includes a diversity of practical applications of Internet of Things depend on the needs of the users [1]. The layer utilizes a number of different protocols, like extensible messaging and presence protocol (XMPP), the advanced message queuing protocol (AMQP), the message queue telemetry transport (MQTT) protocol, and the constrained application protocol (CoAP) [14].

The architecture can be extended to a fourth layer called the middleware layer between the application layer and the other two layers. The application layer serves as an intermediate between the applications and the hardware layer. It includes quite complicated functionalities for managing devices and treats the aggregation, data analysis, filtering, and control of access to services.

The middleware layer also hides the complexity of the network's operating mechanisms and makes it easier for designers to develop applications [1].

3.2. Protocols of IoT

Because of the presence of a diversity of networks, applications and devices in an IoT surrounding, some standards are utilized, and several communities are being implicated [3]. The idea of a lighter IPv6 for connected objects gave birth to the 6LoWPAN working group within the IETF in 2006 [18]. The goal is to simplify the header of IPv6 messages in order that low power connected objects can read them. The IPv6 stack in question consists particularly of the IEEE 802.15.4 protocol that aims at limiting the consumption at the level of the radio link and the 6LoWPAN for the adaptation of the protocol.

3.2.1. IEEE 802.15.4 [14] [3]

802.15.4 is a standard established by IEEE. It states the details for the media access control and the physical layer for the low-power wireless personal area networks. 802.15.4 guarantees simple and smooth set up, low-cost and moderate battery life, reliable data transmission; additionally, it provides a low-cost communication network for low powered/ low- velocity devices. The IEEE 802.15.4 can be only used with the 6LoWPAN in order to construct a wireless embedded network for the Internet of Things. The basic framework communication scale is 10 meters with a transfer rate of 250 Kbit/s. The upper layers are not specified within the standard. The figure 2 depicts the architecture of the basic IEEE 802.15.4.

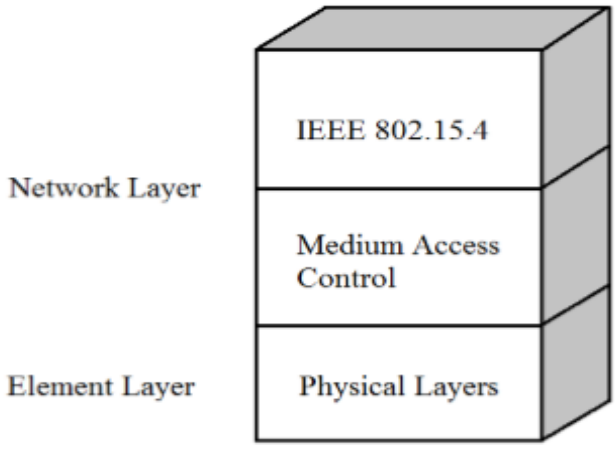

Figure 2. The Architecture of the IEEE 802.15.4. [14]

IEEE 802.15.4 performs the advanced encryption standard (AES) symmetric cryptography mechanism. Moreover, it supports different security modes. These security modes supply security services such as confidentiality, authentication, and integrity, as shown in Table 1.

Table 1. Security Services and the Security Modes of the IEEE 802.15.4 [14]

Security Service	Security Mode	Security Mechanism
Confidentiality	AES-CTR	Data is encrypted using AES in the counter mode with 128-bit keys
Authentication Integrity	AES-CBC-MAC/MIC-32	Data is encrypted using AES in the cypher block chaining mode with message authentication code and message integrity code in 32/64/128-bit keys
	AES-CBC-MAC/MIC-64	
	AES-CBC-MAC/MIC-128	
Confidentiality Authentication Integrity	AES-CCM-32	Data is encrypted using AES in the counter and cypher block chaining mode with message authentication code and message integrity code in 32/64/128-bit keys
	AES-CCM-64	
	AES-CCM-128	

3.2.1.1. Characteristics [18]

The characteristics of the IEEE 802.15.4 are:

- Three frequency bands: 2450 MHz (worldwide), 915 MHz (USA) and 868 MHz (Europe).
- Rates of 250 kb/s, 100 kb/s, 40 kb/s and 20 kb/s.
- Short addressing on 16 bits or extended on 64 bits (IEEE EUI64).
- Allocation of guaranteed time intervals.

- Access to the channel using CSMA-CA.
- Protocol with acknowledgment for transfers to ensure reliability.
- Low consumption.
- Indication on the quality of the connection.

3.2.1.2. Components [3]

There are two sorts of devices, which can contain a full function device (FFD), an 802.15.4 network, and a reduced function device (RFD). A FFD has the abilities to function as a PAN coordinator or coordinator in the network. A PAN coordinator is the primary coordinator that takes the responsibility for the entire network whereas regular coordinators establish communication with devices and pass on messages via the network. On the other side, RFDs are simple devices that have no necessity to transmit great amounts of data and might thus be established employing minimal resources and memory, an instance of one of these devices can be a light switch or simple sensors.

3.2.1.3. Topologies of 802.15.4 [3]

802.15.4 has two kinds of network topologies, which are peer-to-peer networks or stars networks. The star topology has only one PAN coordinator that manipulates the communication between all the devices on the network. A P2P topology enables for more complex network structures/formations, and not each device needs to be connected to the PAN coordinator. In this situation, regular coordinators (FFDs) can function as masters for RFDs and forward messages to the PAN coordinator.

3.2.1.4. 802.15.4 Functional Overview

A. Structure of the frame

In the MAC layer of 802.15.4, a MAC frame is divided into 4 frames. A beacon frame utilized by the coordinator to transfer beacons, a data frame for transmission of data, while an acknowledgment frame utilized for approving successful reception of a frame and a MAC command frame that handle all MAC peer entity control transfers.

B. 802.15.4 frame format [18]

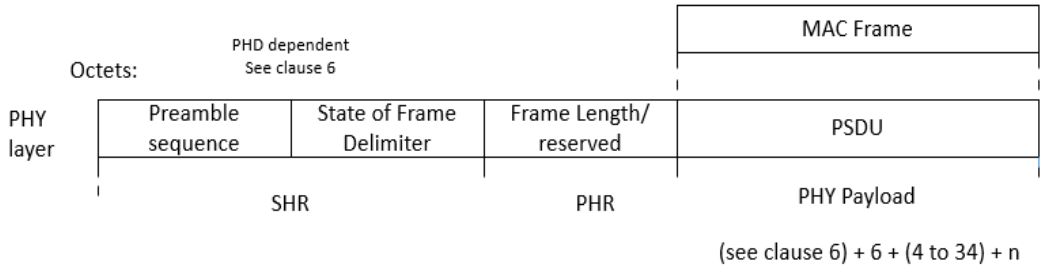

Figure 3. 802.15.4 frame format

C. Data Transfer

In 802.15.4, three different type of data transfer are defined. The first one is data transfer to a coordinator in which the device is conveying data; second one, data transfer from a coordinator where the device is receiving data and the last one between two peer devices. The third type of data transfer is exclusively used in a P2P topology, as data transfer in a star topology should comprise a coordinator. The three methods are depending on a beacon for network discovery. Data transfer may be executed without the beacon allowed likewise if there are no dependencies on synchronization or low latency devices.

D. Enhancing probability of successful data distribution

802.15.4 uses various mechanisms to enhance the probability of successful data distribution among devices. Certain of the significant ones are listed below.

- Slotted CSMA-CA in beacon-enabled networks, which use aligned back off periods on all devices in accordance with the PAN coordinator to verify if a channel is empty or busy.
- Unslotted CSMA-CA in non-beacon-enabled networks, a device desiring to transfer data waits for a random period earlier than it verifies the channel.
- ALOHA protocol in the light loaded network. Without verifying the status of the channels, a device transmits data.
- Frame acknowledgment; if no acknowledgment is acquired; the transfer of the frame is re-tried.
- Cyclic Redundancy Check to pick up faults.

- Energy consumption concerns via duty cycling, i.e. devices changing among sleeping and awake condition, periodically listening to the RF channel to decide if a message is pending.

E. Security of 802.15.4

Due to the low-cost and low powered devices that are generally existed in an LR-WPAN, there are restricted on the security overhead and are depending on upper layer implementation of numerous security architectural elements. The cryptographic mechanism in 802.15.4 is based on symmetric-key cryptography where upper layer procedures supply the keys. The upper layer mechanisms guarantee a reliable handling of cryptographic procedures and secure and authentic storage of the keys. With cryptographic mechanism from upper layers, 802.15.4 offers data authenticity, replay protection, and data confidentiality.

3.2.1.5. Addressing in an IEEE802.15.4 Network

Every autonomous PAN should select a completely unique identifier, which permits for short 16-bit addressing within a network and lets in transmissions between devices belonging to independent networks. The PAN coordinator is responsible for allocating this short address to a device that has sent an association request.

3.2.1.6. IPv6 Implementation Issues on 802.15.4

Several IPv6 implementation issues on 802.15.4 were related to conserving low power consumption:

- Strong constraints on the MTU: IPv6 requires an MTU of 1280 bytes and indicates that any link, which does not satisfy this constraint, must implement fragmentation at the link layer so that it is transparent from the point of view of IPv6.
- Cost of the IPv6 header: the protocol has a minimum header size of 40 bytes to which will be added the 8 bytes of UDP, thus for an 802.15.4 packet to which we will apply an encryption (native for 802.15.4), the IPv6 header and the UDP header.
- Link Level Fragmentation: the obligation of transparent fragmentation for the network layer will lead to a very high load of cutting / reconstruction for elements clearly having no potential.

- Definition of QoS: this is one of the primary goals of IPv6. Speed, network coverage, error rate, and service differentiation. All this has a cost!
- Routing: The routing protocols for mobile IPv6 are not suitable for different reasons: the massive use of packets, multicast etc.

3.2.2. 6LowPAN / RPL

The 6LowPAN that is proposed by the Internet Engineering Task Force (IETF) [14] is an adaptation of the IPv6 protocol to facilitate operation in low power wireless networks. Furthermore, it is the main network protocol employed by Internet of Things and simply put, it allows limited devices that cannot treat the conventional IP stack, utilized by the Internet, to run and connect to devices on the Internet [5]. It relies on the IEEE 802.15.4 standard that defines the operation on the media access and physical layers. The IEEE 802.15.4 has 127 bytes MTU [11]. On the other hand, IPv6 standard needs MTU to be at least 1280 octets. Indeed, the fact that applications, which use IEEE 802.15.4 link layer in most situations, possesses a highly constrained payload size, there still can be situations when the IP packet cannot be forwarded without fragmentation. 6LowPAN is to alleviate this problem by establishing the new layer above the IEEE 802.15.4 link layer and underneath IP layer. With a purpose to attain maximum performance, the 6LowPAN standard defines two important concepts.

(i) Header compression: it specifies the way to compress IP and UDP headers to reduce the payload.

(ii) Fragmentation: it defines an algorithm to fragment and gather IP packets that are the layer than IEEE 802.15.4 MTU. [15]

The purpose of this compression is to reduce the bandwidth usage over low power networks that are constrained in terms of bandwidth and time available for transmission. This compression technique works by splitting communications into separate “contexts” that share common knowledge of IPv6 addresses and other metadata [16]. Nevertheless, this protocol does not assure security in anyway; preferring to depend on other protocols for this, such as the IPsec and DTLS protocols [5]. In addition, routing at this layer can be achieved through the RPL protocol and security can be established through the IPsec protocol.

The following figure shows the location of the 6LoWPAN layer in the protocol stack of the sensors and the base station that is called 6BR for IPv6 Border Router [17].

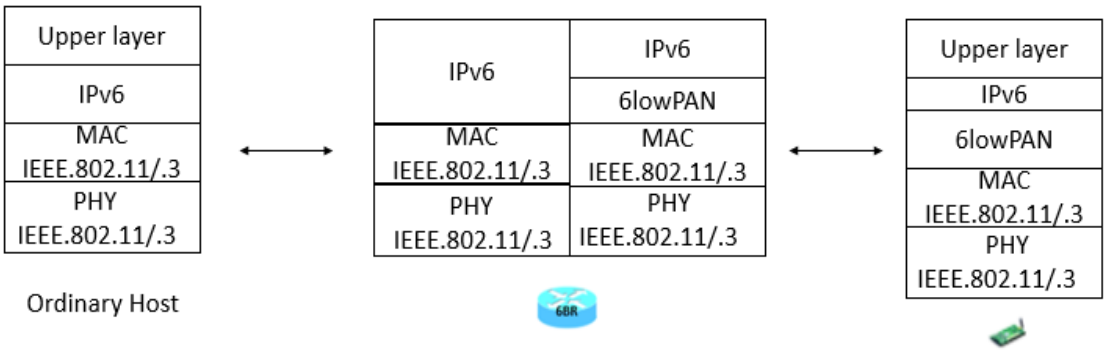

Figure 4. The position of the 6LoWPAN layer in the protocol stack of the sensor and the on-board router. [1]

3.2.2.1. Compression of the IPv6 header

Since the IPv6 header has a length of 40 bytes, it is a requirement for the low power and lossy networks to compress them.

- Version: This information is simply eliminated by 6LoWPAN assuming that all communications with the sensors will use the IPv6 protocol.
- Traffic class and stream identifier: these two fields, which are used for quality of service management, are maintained in the header if QoS management is enabled. Otherwise, both fields will be set to zero. In this case, they are to be revoked.
- Data length: this information is redundant because the size of the data can be deduced from the length of the frame at the MAC level, or from the 6LoWPAN fragmentation header.
- Next header: can be either kept or revoked if the following header is compressed by 6LoWPAN.
- TTL (Time to Live) or hop limit: This field is compressed only in the case of local and direct communications. Where applicable, the field is kept.
- The source address and the destination address: IPv6 source and destination addresses are the largest fields in the IPv6 (128-bit) header. An IPv6 address is composed of two parts: the first part (64 most important bits of the address) represents the network identifier (network address) while the second part (the 64 least important bits of the address) defines the identifier of the host. For reasons of

simplicity, the IEEE 802.15.4 MAC address of the host (EUI-64) is itself the host identifier in its IPv6 address as shown in the figure below.

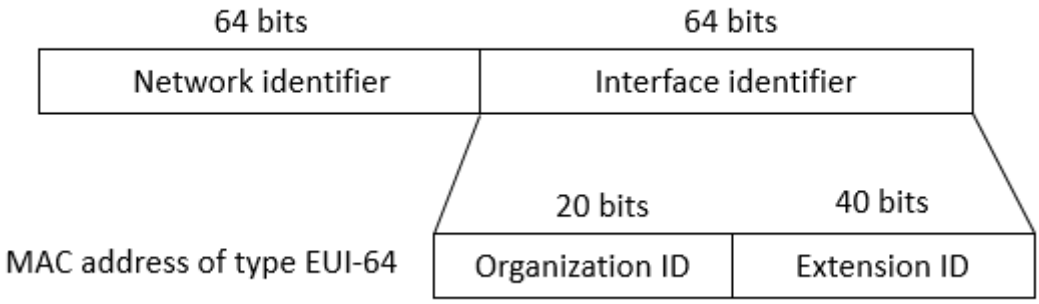

Figure 5. The general structure of an IPv6 address

If the source and the destination are in the same network of sensors and they are more directly related, the two IP addresses of the source and the destination are compressed, and the devices only address the MAC level by their MAC addresses. If the source and destination are always in the same network but one is distant from the other, IPv6 addresses can be replaced by 64 bits long or 16 bits short IEEE 802.15.4 addresses.

Note that 6LoWPAN compression requires that the compressed header be preceded by a dispatch byte that identifies the state of the IPv6 header (For example, if the dispatch byte is 01000001 then the next header is an uncompressed IPv6 header and if it is 01000010, the following header is a compressed IPv6 header) and one byte (or two) of encoding. In this byte, we identify the compressed header, and we indicate the compressed fields. The general format of the 6LoWPAN datagram (compressed IPv6 datagram) is as follows:

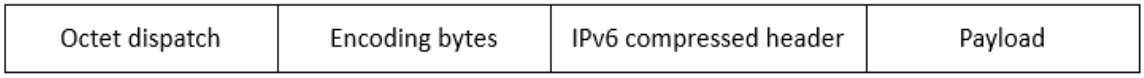

Figure 6. General format of the compressed 6LoWPAN datagram

The resulting compressed IPv6 header would include a lesser amount of information; only the necessary information is communicated within the WSN. Therefore, the size of the

header and datagrams in general, become small. As shown in the figure 7, with 6LoWPAN compression, the IPv6 header size can be minimized by up to 2 bytes (of encoding) in the case of local direct network communications. The size may be up to 7 bytes (2 bytes of encoding, 1 byte of the TTL field, 2 bytes of the source address, and 2 bytes of the destination address) in the case of local multi-hop communication. In the case where one of the communicating entities is located outside of the sensor network, the size of the compressed IPv6 header becomes 21 bytes (2 bytes of encoding, 1 byte of the TTL field, 2 bytes of the address source / destination, 16 bytes of destination / source address).

The 6LoWPAN compression mechanism does not just apply to the IPv6 protocol header, but it can also affect its extension protocols, like the header of the UDP protocol that may be compressed from 8 bytes to 5 bytes. Moreover, as long as the UDP header does not encompass a field to identify the header of the application protocol that follows it directly, it is necessary to compress the headers of these protocols as part of the payload of the UDP packet. This 6LoWPAN compression version is referred to as 6LoWPAN-HCg [19].

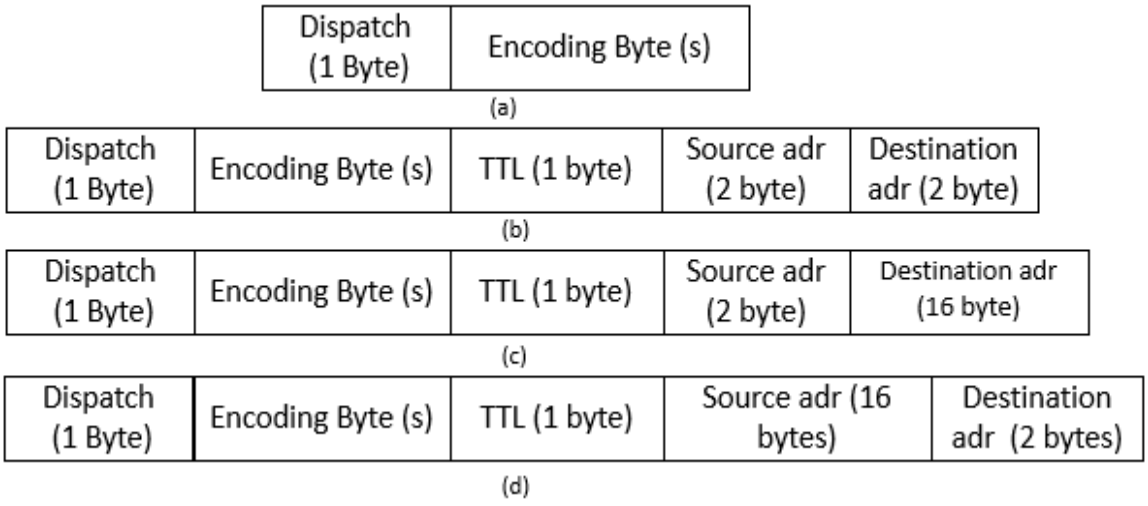

Figure 7. The IPv6 header compressed using the 6LoWPAN standard with (a) direct local communication, (b) local multi-hop communication, (c) the source belongs to the connected sensor network and the destination is outside the network (d) the source is outside the network (d) the source is outside the RCSFs and the destination belongs to it.

3.2.2.2. The Fragmentation of IPv6 Datagrams

6LoWPAN adaptation layer takes also the responsibility of the fragmentation of large data packages so that they can be transmitted over IEEE 802.15.4 PHY layer. The reason is that 802.15.4 PHY packet size is 127 bytes. Once the fragmented packets arrive at their final destination, they are reassembled by 6LoWPAN to create the original frame as shown in the following figure.

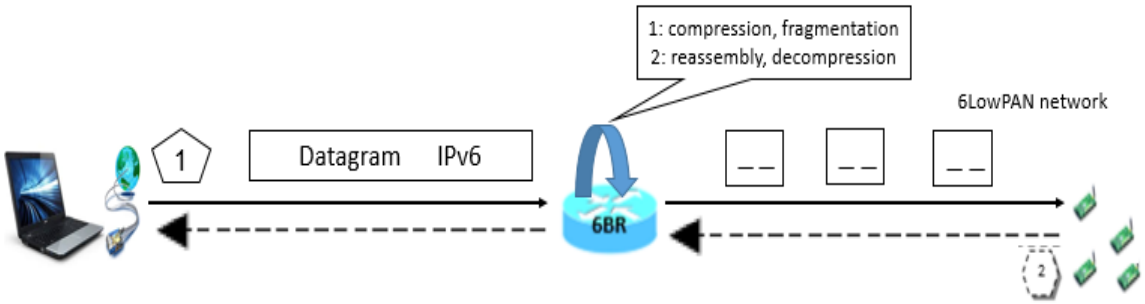

Figure 8. The fragmentation process [1]

Each 6LoWPAN fragment must contain a header specific to the fragmentation process and helping entities to properly fragment and reassemble 6LoWPAN datagrams. The format of the header of the first 6LoWPAN fragment is as follows:

1100	Size of the datagram (11 bits)	Datagram identifier (16 bits)

Figure 9. Fragmentation header of the first 6LoWPAN fragment

The first five bits associated to the header of the first fragment, the datagram size field gives the size of the initial 6LoWPAN datagram before fragmentation. The identification of the datagram is an information that helps the receiving station to locate the corresponding fragments in the same datagram. In the fragmentation header of the rest of the 6LoWPAN fragments, it is necessary to include additional information relating to the location of the fragment in the initial datagram (shift field). As shown in the figure below.

1100	Size of the datagram (11 bits)	Datagram identifier (16 bits)	Shift (8 bits)

Figure 10. Format of the fragmentation header for the next 6LoWPAN fragments

3.2.2.3. Routing in 6LoWPAN networks

Routing in 6LoWPANs is the vital feature that ensures the proper routing of 6LoWPAN packets between nodes in the same network or between the 6BR border router and the other end-points in the Internet. Two mechanisms are defined by the IETF to support routing in this type of networks: 6LoWPAN layer-level mesh routing and RoLL group Routing Protocol for Low power and lossy networks (RPL) routing.

A. Mesh routing

This mechanism exploits information from the MAC layer, specifically MAC addresses, to realize the routing of the compressed IPv6 frames at the 6LowPAN adaptation layer. Communication between source and destination is considered as a single IP hop whose intermediate nodes (routers) make the routing decision based on the analysis of the MAC address of the destination. If this one does not correspond to the MAC address of a relay node, this node realizes that the fragment (and the datagram) is not intended for it and thus consult its routing table at the link level to find the next node. In this case, the fragments pick routed independently of each other. This means that fragments of the same frame can take different paths to reach their final destination. The problem with this solution is that in case of loss of at least one fragment, the loss can only be detected at the final destination and in this case, all fragments (including the missing one) must be retransmitted again for recovery. The following figure illustrates the format of the mesh routing mechanism header in 6LoWPAN networks.

1 0	O	F	Hop nbr (4 bits)	Source address (16/64bits)	Destination address (16/64bits)

Figure 11. Mesh routing mechanism header defines in 6LoWPAN layer

With two first bits used to identify the header. In addition, since the IEEE 802.15.4 standard supports two types of addresses (short addresses and long addresses). The O

(originator) and F (Final) flags are one if the MAC addresses of the source and destination entities, respectively, are short (16-bit) and zero if the addresses are long (64-bit). The value of the number of hops field is limited to 15 and is decremented as the fragment progresses towards the destination. When the value of the field becomes zero, the entity that received it discards it if it is not its final destination.

B. The RPL protocol

RPL is an IPv6 routing protocol for 6LoWPAN networks in the IoT [20]. It constitutes an optimized and dynamic topology with loop avoidance and consideration of quality of service parameters for routing IPv6 frames from and to sensor nodes.

Each intermediate node behaves like an IP router, it first reassembles all the fragments to rebuild the initial IPv6 frame, and then it analyses the destination IPv6 address to decide if the packet will go to the transport layer or if it must be communicated to another node until it arrives at the correct final destination.

RPL builds an acyclic DODAG (Destination Oriented Directed Acyclic Graph) graph that routes information to or from a single destination called the DODAG root. In some cases, the same 6LoWPAN physical network should be optimized to support several applications each with its own topology. These multiple DODAGs for the same network is called RPL instances. RPL instances are distinguished by a well-defined routing metric. These metrics in some instances can be the residual energy of the node; they can be link quality between the communicating nodes in other instances. During the construction of the graph, the nodes use the objective function that defines the method of calculation of the routing metric, and exchange four types of messages: DIO (DODAG Information Object), DIS (DODAG Information Solicitation), DAO (DODAG Destination Advertisement Object) and DAO-ACK (DAO Acknowledgment).

- The DIO message is broadcasted first by the 6BR (the root) to trigger the process of building the topology. The neighbouring sensor nodes of the root receive the message and decide whether they can join the graph or not (the decision depends on several factors such as the objective function and the cost of the announced path). Once the node has joined the graph, it automatically has a path to the root. If the node is configured to be a router, it in turn broadcasts DIO messages to its neighbours.

- The DIS message is utilized by the nodes to request information about the graph from neighbouring nodes that will respond by sending a DIO message.
- DAO type messages are used for announcing the presence of the node to its parent in the graph. Each node on the routing path from the node to the root node registers the routing information embedded in the DAO and passes it to the next node until DAO message reaches the root (6BR).
- The DAO-ACK message is forwarded by the parent node to the child node, in response to its DAO message (to acknowledge receipt).

In addition, RPL supports two routing techniques: non-storing mode (stateless) and storing mode (with state). In non-storing mode of operation, the entire path to be taken is stored in the packet in the source routing header, and the intermediate nodes use this header to decide the next hop. By contrast, in the storing mode of operation, the packet only carries the address of the final destination, and the routing is decided at each intermediate node according to the information contained in a local routing table. The 6BR therefore maintains a complete list of all nodes in the topology. Note that for the avoidance of routing loops, each node must calculate its position (or rank) in the hierarchy relative to the root. When the storing mode of operation is employed, the local communications between the sensor nodes having a parent in common do not need to go through the root node. However, nodes that do not have a common ancestor must pass through the main root (the 6BR edge router), as shown in the figure below, where the red arrows represent local communications, and numbers 1 until 4 represent the ranks of the nodes in the graph.

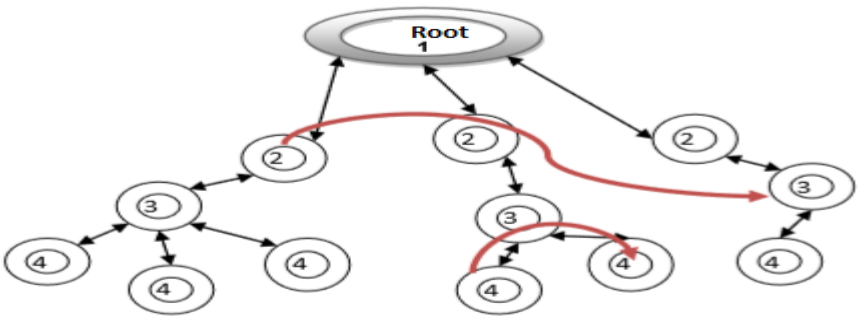

Figure 12. Working of RPL protocol [1]

3.2.3. CoAP (Constrained Application Protocol)

CoAP is an application-layer protocol established to enable web-based applications in devices with limited resources. CoAP is created to connect readily with constrained nodes and has support for HTTP like web application programming interface where the interaction model is analogous to the client/server model [22]. The most important benefit of the CoAP protocol is that it is allow overhead protocol using UDP transport layer instead of TCP [23] [24]. Moreover, it supports multicast and unicast communications as well as a built-in device discovery function. Additionally, it has mechanisms for assuring the QoS [25]. As CoAP focuses on the REST model, the CoAP server (a low power device) makes resources accessible to clients through Uniform Resource Identifiers (URIs) identifying resources, and well-defined methods that clients specify in their queries. There are four types of methods [1]:

- The GET method: The CoAP client uses this method to request a resource. It has the code 0.01.
- The PUT method: Having the code 0.02, it is used to update or create a resource in the CoAP server.
- The POST method: This method requires that the representation indicated in the message be applied to the resource. Its code is 0.03.
- The DELETE method: The resource identified in the query containing the DELETE method must be removed at the server level. The code corresponding to the method is 0.04.

These request messages are handled by the CoAP server and a response is returned with a response code including/excluding a resource representation. CoAP follows the statelessness constraint as its architecture is on REST framework. The protocol, however, utilizes datagram-oriented transport like UDP to treat these interactions asynchronously. While the UDP protocol is untrustworthy, this is accomplished logically using a message layer, which supports non-mandatory reliability. Since CoAP requests and responses are completed in independent messages, to guarantee reliability every message should have one message type. Each request has an extra bit to explain the message type. The message types are as below:

- Confirmable message (CON): it necessitates an acknowledgment. The client will maintain resending a CON message up to it gets an ACK message with same message ID.
- Non-confirmable message (NON): this message does not require an acknowledgment which includes GET request.
- Acknowledgment message (ACK): It approves that the previous message was received and treated.
- Reset message (RST): Similar to the acknowledgment message but it additionally confirms that the recipient could not treat it by cause of network loss or other motives.

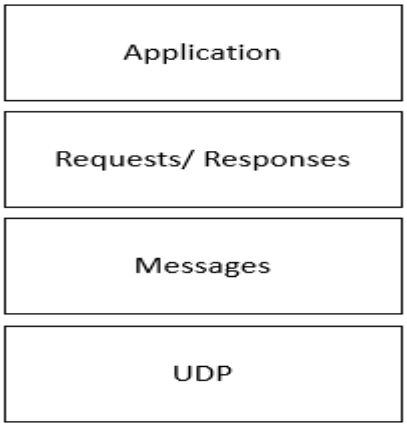

Figure 13. The Abstract Layer of CoAP [22]

From figure 13, logically, CoAP can be defined as a two-layer approach. One layer treats UDP and the asynchronous nature of the communication while the other one treats request/response interactions utilizing the method and response code. As CoAP is based on the exchange of compact messages, the messages are encoded in a simple binary format. The binary structure of CoAP message format is represented in Figure 14. The CoAP package format has a maximum length of 1400 bytes while the header has a length of 32 bits (4 bytes). First 2 bits for version control next 2 bits for message type and remaining 4 bits for token length.

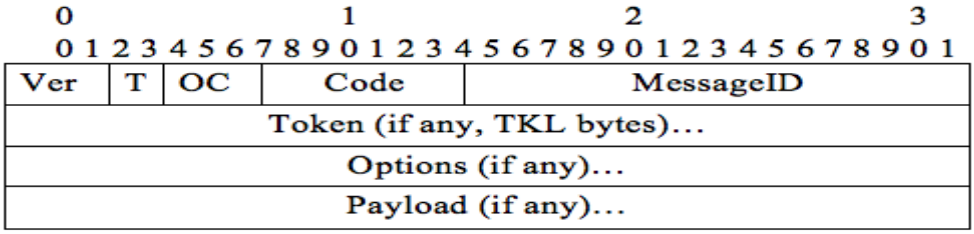

Figure 14. The Message Format of CoAP

In spite of the fact that it was specifically established for limited devices, the CoAP protocol does not have any integral security characteristics. Similar to the HTTP protocol that depends on the TLS protocol to assure security, the CoAP protocol depends on the DTLS protocol to handle security as suggested by the IETF. Another possible security protocol for CoAP is the IPsec protocol but contrary to DTLS, the IPsec protocol has not been approved for utilize in coupling with the CoAP protocol by the IETF [5].

Where clients are concerned about getting a speedy response to the representation of a resource during a period of time, the traditional request/response model is not appropriate. In these scenarios, MQTT runs well since it utilizes peer-to-peer communications. Nevertheless, CoAP supplies real-time solutions along together with retaining the properties of REST to push resource representation from servers to the concerned client. The clients of CoAP might be seated as Observers and register their interests using the GET request with a particular ‘observe’ option activated for one or multiple resources. In addition, CoAP is commonly used due to the integration of 6LowPAN, easy portability with HTTP, and UDP to support low connection overhead [22]. Thangavel et. al. compared the performance of MQTT and CoAP in terms of end-to-end transmission delay and bandwidth usage [26]. The test indicates that MQTT operates well when the data packet loss is low. Nonetheless, CoAP performs better than MQTT when messages are small and data loss rate is beneath 25%. In Smartphone environment, CoAP’s bandwidth utilization and round-trip time are smaller than MQTT [27].

A CoAP device can send multicast UDP message to its surroundings to verify who else is around. It is able to begin scouting and interact with other devices according to REST architecture and without any human interaction. All of the things can be considered as the

world of APIs. With CoAP everything is possible, home automation, controlling and remote tracking with the mobile device. [22]

3.2.4. 6TiSCH

The IEEE 802.15.4, as indicated in section 3.2.1, is a standard devoted to low power, low cost and low rate PANs, which defines the MAC and physical layers of the OSI reference model. Originally, the standard was designed for applications that do not need particular exigencies in terms of reliability, scalability, and latency [52]. To defeat these limitations and treat the new requirements of embedded industrial systems, the IEEE started a Working Group called 802.15e WG with the goal of improving and adding functionality to the IEEE 802.15.4 [54]. The IEEE 802.15.4e is an extension of the IEEE 802.15.4 standard, in which new MAC behaviours are inserted to support application areas requiring high reliability. One of the MAC behaviour modes is Time Slotted Channel Hopping (TSCH). It is targeted at application areas like process automation with strict timeliness and reliability demands. [53]

All sensor motes in IEEE 802.15.4e network are synchronized. A schedule is calculated and distributed between the nodes. In addition, under TSCH, time is divided into time slots, each typically, 10ms long, which are sufficiently long to transfer data packet and get an acknowledgement indicating successful reception. A group of timeslots forms a slot frame (typically 10-1000 timeslots long) that continually repeats over time. The schedule organizes all communication in the network. It instructs to each node the action that needs to do in each slot of the slot frame: listen, sleep, transmit (TX) or receive (RX). Moreover, the schedule specifies on which frequency for each slot in which the node communicates [55] [56] [57].

The channel diversity is accomplished by indicating a channel offset for each TX or RX slot. In successive slot frame cycles, the same slot offset renders into a different communication frequency, leading to “channel hopping”. Channel hopping decreases the effect of external interference and multi-path fading, thus enhancing the reliability of the network [58]. The TSCH schedule can be depicted by 2-D matrix of slots and frequency bands in a slot frame. Each element in the matrix is named a cell. A cell is scheduled when utilized to either transmit and/or receive. Each scheduled cell creates a communication

opportunity for a sensor mote to interact with its neighbour. A distributed or centralized scheduling entity manages the schedule. Furthermore, this entity assures that there are enough communication opportunities to fullfil the communication requirements of the applications such as latency, reliability, and amount of traffic load provided to the network. 6TiSCH protocol defines the mechanism of scheduling but does not define the scheduling policy (how the schedule is established and maintained). Building a schedule is application and implementation-specific and the scheduling function can be tweaked to meet the demands of the application in terms of delay, reliability and throughput. The scheduling entity needs a method to distribute and set up the schedule at each sensor mote.

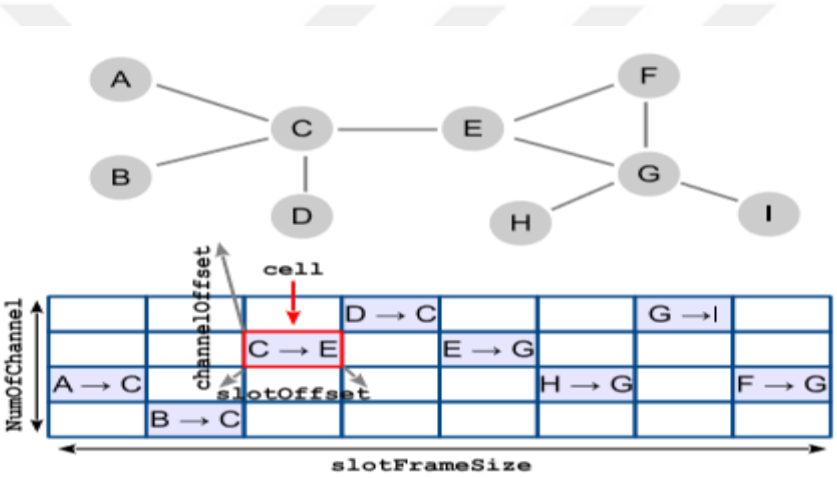

Figure 15. A simple TSCH schedule [55]

Each time slot is marked with Absolute Sequence Number (ASN). ASN is a variable that calculates the number of timeslots from the establishment of the network. Thereby, each node makes the decision when to send or receive a frame based on ASN and its scheduler. Furthermore, to overcome interference and noise, and consequently, to permit high reliability, TSCH suggests carrying out a channel hopping scheme. A cell, in TSCH, is transmission opportunity presented by a channel offset and time slot pair. The choice of the actual channel is derived from the channel offset and the ASN at the starting of each time slot. As shown in the figure 16, a cell may be either shared, transmit or receive. In the shared cells, many interfering nodes are authorized to transmit: they should execute a slotted CSMA-CA mechanism to evade collisions. For the cells allocated as receive and transmit, a

collection of non-interfering transmitters are the owners of the cells: In this case, communicating pairs of nodes transmit in contention-free manner in the allocated cells.

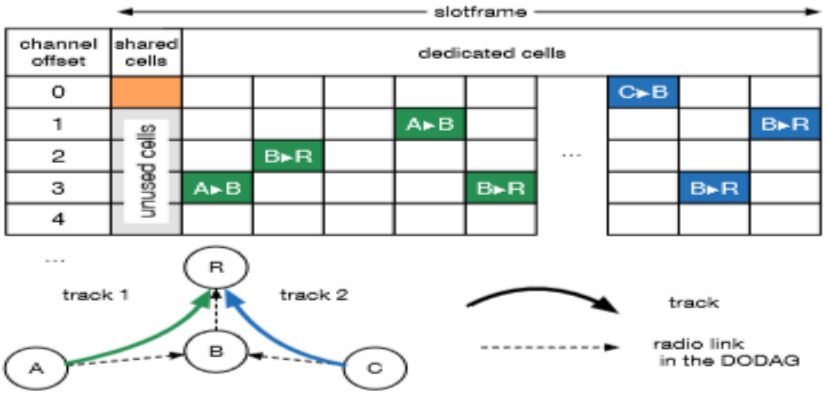

Figure 16. Schedule in a 6TiSCH network, using two different tracks for traffic isolation

The purpose of 6TiSCH is to establish multi-hop IPv6 connectivity over TSCH. The 6TiSCH protocol stack includes IEEE802.15.4e TSCH, IETF 6LoWPAN, RPL and CoAP. [59] [60]

6TiSCH protocol overview

As shown in figure 16, in 6TiSCH minimal [61], at the starting of the slot frame, one shared cell is reserved to exchange packets of control. For example, Enhanced Beacons (EBs) are transferred within the shared timeslot in order to associate the neighbours with the current network. Moreover, 6TiSCH protocol defines a protocol to negotiate the cells between the nodes in a distributed manner. This protocol layer is named as 6Top layer [62]. The algorithm determining the number of cells is named as the scheduling function [63]. 6Top layer can incorporate any scheduling algorithm; a new SF can be easily integrated to the protocol via the interfaces defined by the protocol. Thereby, either the protocol can run in a centralized fashion (e.g. a node solicits a Path Computation Element for new cells to employ) or in a distributed fashion, (e.g. SF0 decides the number of cells to allocate based on the local adjustments). 6TiSCH involves the notion of the track. A track conforms to dedicated radio resource, along with a multi-hop path. In other words, a group of cells (named bundle) is reserved for each hop. In this situation, a track-forwarding scheme is

carried out; when 6Top layer receives a frame to transfer, automatically finds the outgoing bundle attached to the incoming cells. Additionally, 6TiSCH places the idea of the chunk; each node is capable to divide the scheduling matrix in non-overlapping chunks. Then, the cell bundles are allotted to different nodes in a centralized or distributed fashion. For instance, as depicted in figure 16, the flow from A to the border router R, through node B, will be attributed to track 1. Moreover, the same node (node B here) can dispatch a further flow, for example from C, employing another track (i.e., 2). [56]

Protocol stack

To provide an open standards-based protocol stack for deterministic wireless mesh networks, the 6TiSCH attempts to fulfil the omission between the IETF upper layers and IEEE lower layers of the protocol stack. Figure 17 demonstrates a general overview of the incorporated protocol stack rooted at the IEEE802.15.4 physical layer. The IEEE802.15.4e TSCH MAC modifications enable determinism while guaranteeing very low power communication between the devices. As explained above, 6TiSCH protocol provides necessary functionalities to enable a deterministic and low power mesh protocol. The protocol works together with IETF 6LoWPAN and RPL to create a low power next generation IoT protocol stack

The deterministic feature of IEEE802.15.4e TSCH and its rigorous necessities in regard to schedule management make it essential to have a sublayer that permits management entities to run the network. Consequently, the 6Top sublayer is one of the essential parts of the 6TiSCH protocol. The 6TiSCH architecture decides how packets are labelled and routed or dispatched upon the mesh within jitter and latency budgets. The schedule management translates into routes and track management. In centralized fashions, as presented in figure 17, schedule management is treated by a PCE and its control messages are transferred protocols like PCEP/PCC [64] or COAP/DTLS on top of UDP. In decentralized techniques, the management entities take advantages from reservation protocols along with RSVP [65] or NSIS [66] in which quality of service necessities are carried and hooked up along a path, after which are implemented as cellular reservation inside the time table of every node on the path.

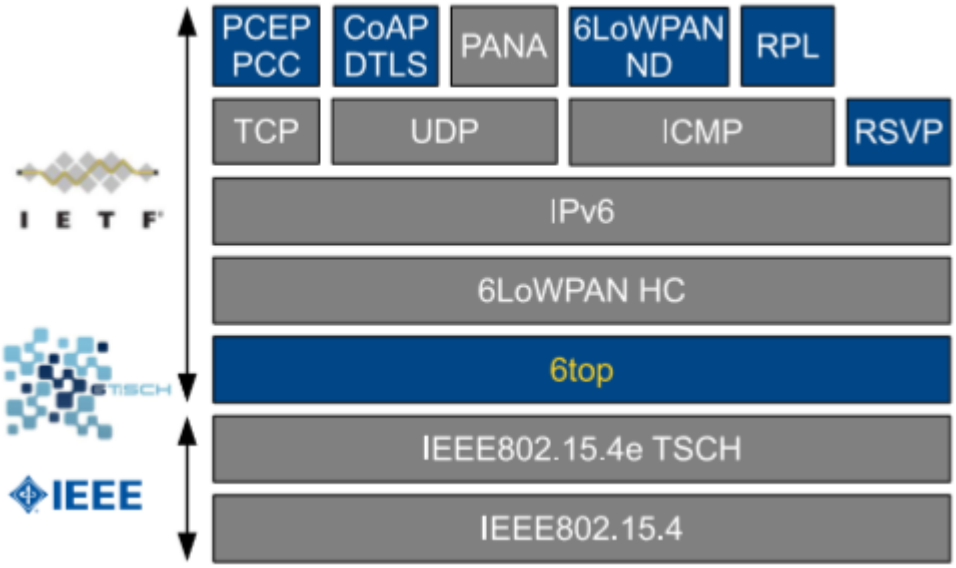

Figure 17. 6TiSCH IPv6-enabled protocol stack for LLNs [67]

6TiSCH also includes important security aspects. The security requirements are specified and the 6TiSCH WG have been working on a scalable and secure key management structure (and exigencies of related protocols) for 6TiSCH networks. The Protocol for transporting Authentication for Network Access (PANA) is a possible candidate for Key Management Protocol for the bootstrapping phase of the 6TiSCH networks. The protocol holds up mutual authentication, stateless authentication relay function and enciphered distribution of attributes [67].

4. SECURITY IN IOT AND AUTHENTICATION

4.1. Security in IoT

Security and privacy in the IoT can be tackled from three complementary angles that reflect its technological, human and systemic dimensions. The safeguard of technology is principally related to the security of data, communications, and network infrastructures. This safeguard is substantial to block traditional and future attacks on integrity, the authenticity, and confidentiality of the data, as well as attacks on network infrastructures and their functionalities. The security of individuals will take in charge the protection of the privacy of users that demands, in addition to technological solutions, a suitable regulation, which sets up the responsibilities in case of litigation [6].

The increase of the deployed devices puts the information systems in danger. In fact, a significant number of devices are liable to security attacks, such as Denial-of-service (DoS) and replay attacks, which are the result of their limited resources and the shortage of security methods. Nevertheless, the Internet of Things security has been one of the most debated and still unresolved problems, even after the introduction of IPSec and TLS protocols for TCP/IP and DTLS for UDP and CoAP [31] [32]. The main reasons are the difficulty of setting up (IPsec) for end users and the shortage of scalable certificate management for DTLS [34]. Security for the IoT is not widely extended and deployed. Therefore, a significant part of the Internet traffic goes on being transferred unencrypted. To ensure the security in IoT, it is necessary to come up with new mechanisms [15].

4.2. Security Requirements

4.2.1. Confidentiality

Confidentiality is assuring only the people or devices that should have access to the information, have access to that information. Providing confidentiality is extremely crucial for IoT devices since they unobtrusively and ubiquitously gather data, which may be very sensitive. Generally, this service is realized by using encryption and cryptographic

mechanisms and is notably substantial when IoT nodes communicate with each other. The imposition of confidentiality thwarts eavesdropping [5].

4.2.2. Integrity

Data integrity is very important for making sure that the data/information is correct and has not been corrupted or changed in any way by illegitimate entities. This is generally extremely important during the transfer of data between the devices since this is where attacks commonly occur. In addition, this requirement is achieved using collision resistant hash functions and digital signatures [5].

4.2.3. Authentication and Authorization

A. Authentication

Authentication is a process that allows you to specify if someone or something is really, who they pretend to be; and not a virulent user claiming to be someone they are not.

Authentication for IoT is crucial since most communications will occur without user interaction. In addition, it is important to assure that data, commands, and requests are obtained from the right devices. Different means are used for ensuring authentication such as the passwords, digital signatures, and Biometrics, which is too computationally intensive to be applied by the limited IoT devices [5].

B. Authorization

The usage of access control and authorization mechanisms is to restrict the privileges of a device and defines which activities a device is capable to execute. This favor can be in rapport with, but not restricted to, the access of resources and data. Consequently, authorization mechanisms define the operations each device is able of performing and the information it has access to.

In addition, the wide scale of IoT environments and their ubiquitous features makes such devices ideal targets for attackers. As such, authorization mechanisms guarantee a limitation on the operations an attacker is capable to carry out, in the case that the system is compromised [5].

4.2.4. Availability

This requirement ensures the users may get right of access to the information resources in everyday scenarios. IoT systems needs to have techniques to insure the data availability and reliability. DoS and distributed denial of services (DDoS) attacks induce the security challenges for data availability; router filtering might overcome the issue and warrant the data availability of the Internet of Things systems [14]. IoT devices availability may be affected by many factors such as implementing energy efficiency of the communication protocols for battery-powered devices, reliable and low complexity encryption mechanisms, incorporating energy harvesting and saving mechanisms and even performing DoS countermeasures [5].

4.3. Security Protocols Classification in IoT

The existing security suggests in the IoT are divided into two principal classes: security that depends on asymmetric key schemes and security that depends on pre-distribute symmetric keys.

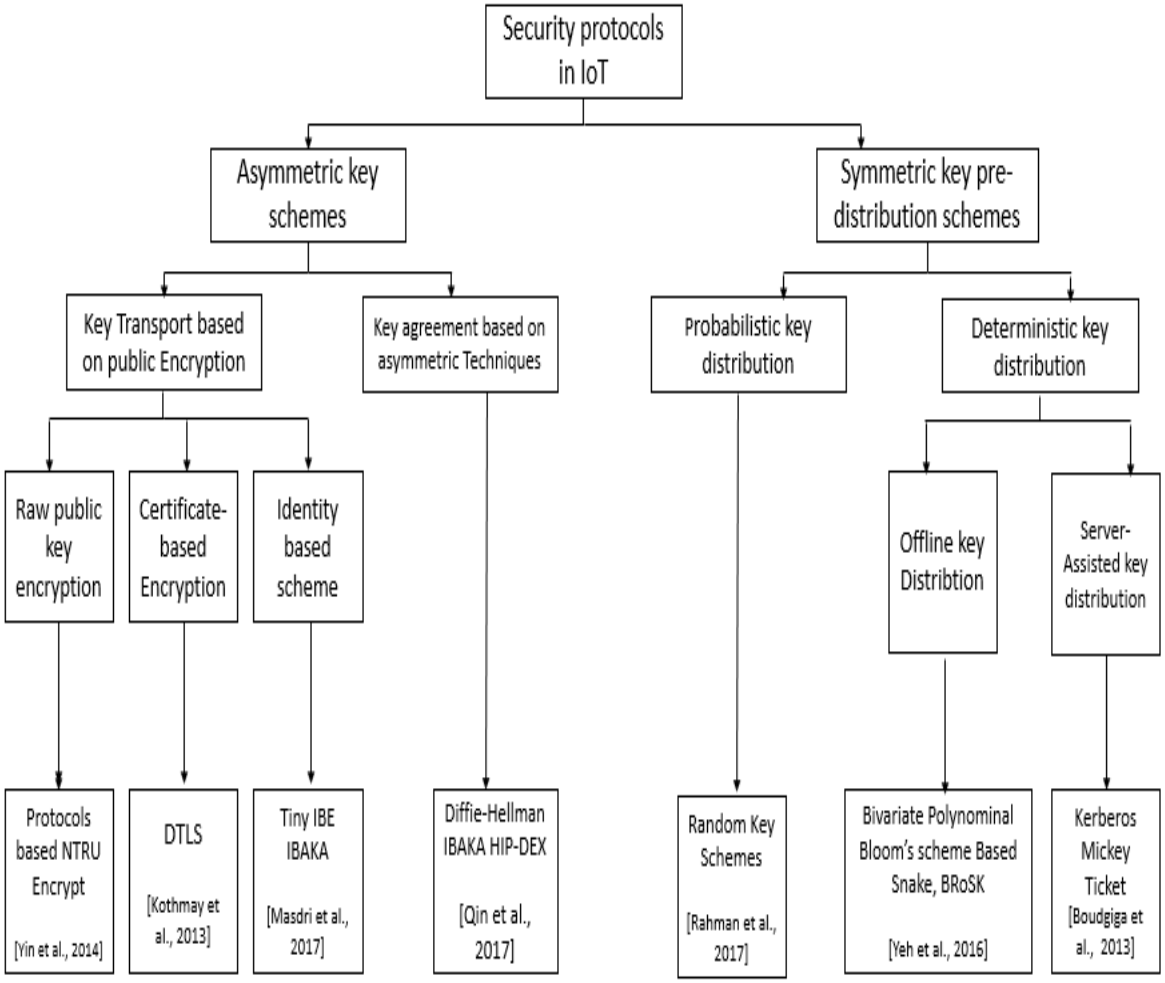

Figure 18. The Security Protocols Classification in IoT [10]

Asymmetric key schemes (AKS): The key schemes based on asymmetric cryptography, also named Public-Key Cryptography (PKC) are a very popular method for developing a reliable connection among two or more entities. In IoT, the usage of AKS has one significant drawback, which is the power consumption and computation cost. In spite of costly operations, many of researchers still look for carrying out AKS in the IoT systems. The suggested schemes can be classified into two families: the first is key transport based on public key encryption, which is relatively similar to the traditional key transport mechanism. Several key establishment methods have been offered for the IoT, going from raw public key usage to complex implementations in X.509 standard [35]. However, the second family is a Key agreement based on asymmetric techniques where a shared secret is

derived between two or more entities. In this class, the Diffie-Hellman (DH) key exchange protocol can be cited [36].

Symmetric Key Pre-distribution Schemes: Researchers, for asymmetric approaches, put forward also diverse techniques employing symmetric key establishment mechanisms in the IoT security. Symmetric methods frequently suppose that nodes taking part in the key establishment possesses common credentials. The pre-shared credentials can be a symmetric key or some random bytes burned into the sensor before its deployment. This class may be divided into two fundamental subgroups: the first class is the probabilistic key distribution. In this approach, the security credentials are (keys or random bytes) selected arbitrarily from a key pool [37]. In this case, every node pair, during their initial communication, can find a shared key with some probability to create a secure connection. The second approach is the deterministic key distribution. Here, to establish the key pool and diffuse evenly the keys, every pair of nodes share a common key, a deterministic design is utilized [38].

4.4. Bootstrapping in IoT Networks

The IoT has as goal of bringing Internet to a number of application areas involving small device communications such mobile-health (m-health) systems, Smart Homes, and Smart Cities. This kind of scenarios necessitates secure mechanisms to preclude loss of private data and dangerous actuating activities. The current IP-based IoT frameworks and primitives are not completely conceived with the restriction of resource-constrained IoT devices. Therefore, lightweight security solutions are essential to guarantee the security of IoT devices.

The restriction at Internet of Things end-nodes encompasses the aspects listed below: storage space, processing power, network capacity, (CPU (MCU) processor, RAM), display, energy consumption and lack of user interface.

In a network security, authentication is one of the most substantial elements warranting that the identity of users of the services are accurate. A diversity of authentication process and protocols are there, however, all serve the same goal. [28]

4.4.1. Motivation/ Need for secure authentication

Internet usage, online applications and ultimately the IoT are encountering an impressive growth. Many devices, which are now part of the IoT, they can be connected to the web and can be accessed via a smartphone. Intruders have the potential to join a wireless thermostat or overrun the smart TV and from there, get access to some other network putting the users at great security risks.

The existing threats in IoT landscape include hackers or intruders crashing airplanes by hacking into the networks, remotely disabling cars, remote murder through hacked medical devices, accessing home automation through hacked thermostats and hacked financial accounts [29]. Online fraud and hacking have become a significant source of income for criminals all around the world. The motivation can be for monetary assets, business gains or out of simple curiosity. This has produced the detection and prevention of such activities a top priority.

Today, all the functions that can be executed offline have an online equivalent. This expands on email access, social networking, etc. to manage funds and transactions online. Where a user's internal network necessitates being protected from unauthorized access, when users need to connect from distant locations, when an employee is accessing responsive business data or when a user wants to implement a secure physical access solution, robust authentication is required. There will always be a trade off between security and convenience. However, a credible balance between security and facility of access needs to be implemented in all functional areas of the IoT. All the IoT applications highlight the importance of authentication as the principal gateway to a secure connection and communication of data. [30]

4.4.2. Difference Between Traditional and the IoT Authentication

The IoT is oftentimes deployed other than the standard internet. Consumers are getting comfortable sharing, storing and accessing their confidential data online. Traditional methods such as usernames, passwords, and fact-based questions are inconvenient and inadequate to provide security for these applications. Password-based authentication connection is the most popular but also one of the most insecure methods and is not adapted

for computer networks. Password transmitted over a network may be intercepted by attackers and can be eventually used to impersonate the user.

Traditional authentication technologies in wireless networks contain the public key or private key based authentication, certificate based authentication, random key distribution or hash function-based authentication methodologies. However, most traditional authentications are based on human credentials such as biometrics, username password combinations or some personal information. Human involvement is not included in the authentication process of IoT networks and the identity information depends on the machine/device that is concerned.

Traditional authentication component was based on traditional role-based access control (RBAC), but in the Internet of Things world, the attributes of a node or IoT object make more sense which makes attribute-based access control (ABAC) more suitable. RBAC monitors access based on the users' roles into the system and on rules stating what access is permitted for what users in given roles. ABAC provides access to users based on who they are rather than what they do. IoT identity has, however, various contexts so a centralized solution will not be feasible. The necessity of the hour is a multi-context aware mechanism [31]. At every step during this authentication, it should be clear who is asking the access, who is granting the access, what access is requested, if requesting party has the necessary privileges, the location, etc. Traditional authentication never had to obtain all these security details when access requests are issued.

The authentication methodology that will be implemented for IoT solutions has to take into consideration low memory capacity of the IoT as the highest priority. The processing power or CPU power in these devices/machines are much lower than that of personal computers. Many IoT networks are deployed on low-power lossy networks (LLN) whereas others have very dynamic topologies according to the application, like medical devices, home automation or vehicular networks. In accordance with Cisco, LLNs are a category of the network wherein both the routers and their interconnects are limited [30]. LLN requires nodes be autonomous and conserve energy to enable nodes to operate over long periods using small batteries.

Besides offering convenience, the IoT authentication systems require providing enhanced security features, flexibility, scalability and privacy to their consumers.

4.4.3. Classification of Authentication Techniques

Authentication techniques are divided in two classes. The first class of the classification is done depending on how the authentication mechanism is executed centralized/distributed and hierarchical/flat. The second class is depending on various attributes that reflect the common features of the authentication mechanism.

The first class:

- Centralized: users/ devices connect to a centralized server to supply the needed credentials such as an authentication server or an ID-provider or any trusted third party.
- Distributed: components communicate and coordinate with the communicating peers separately to attain a common goal.
- Hierarchical: in a hierarchical authentication system, users must be granted access to the available information depending on the user's access rights.
- Flat: authentication process is achieved without hierarchy being taken into consideration.

The second class:

- Two-way authentication: This depends upon on if mutual authentication or one-way authentication is carried out.
- Extra hardware: needed if additional hardware is required for the authentication process to be accomplished.
- Multiple credentials: They are intended to be checked at numerous levels as a way to validate the user's or device's identity.
- Multiple authentication levels: Multiple levels of authentication are required, each with different credentials.
- Registration: This phase is required to register the user's, the devices' or the server's information.
- Offline/ pre-deployment phase: This phase is needed to configure the network or for updates.

4.4.4. Existing Authentication Schemes: Related Work

Shivraj et al. believes in the two-factor One Time Password (OTP) techniques, which is developed employing a lightweight identity-based Elliptic Curve Cryptography (ECC) scheme [39]. Additionally, it was proved to be more effective and reliable than other current processes since the Key Distribution Center (KDC) neither needs any key storage nor stores private and public keys of devices, it only stores their IDs. According to this protocol, fewer resources were required without compromising security itself. The proposed methodology does not address the case where a device wants to control another device from another gateway and one using a different security scheme.

Crossman and Liu [40] have proposed another model for the user or device verification by achieving the process user-centric. In place of a verification code, in this new verification scheme named Smart Two Factor Authentication, after the user enters the password rightly, he has to provide a security token like a Smartcard as the second authentication factor to recover a dynamic encryption key. The user has complete control of all their resources, which protects them from disturbing if the manufacturer's database is hacked into and saves the manufacturer and national security concerns.

As password authentication usually depends on users being successful to remember complicated information, emerging digital memories was suggested by [41], which implicates the establishing of a repository of memories specific to individuals. This scheme includes user identity assurance into authentication scheme and one factor in this uses challenges from users' own digital memories. Here, the possibility of attackers is decreased since all the factors cannot be hacked or known by anyone other except the user. This methodology makes difficult phishing attacks, social engineering, and shoulder surfing for attackers and the complexity can be variable based on the type of service being used.

Yao et al. designed a message authentication code (MAC), after modifying Nyberges fast one-way accumulator, a based multicast authentication mechanism for small-scale IoT applications, which would be very efficient for resources-constrained devices [42]. Multicast provides efficient communication in network access layer and data sensing layer of IoT.

In [43], the author proposed a mechanism, which optimizes the interoperability among the isolated IoT domains and tackles security and privacy problems in the IoT model. The proposed lightweight authentication and authorization mechanisms ensure certain assets of

smart grid environments. The process is based on EAP over LAN (EAPOL), which surcharges the constrained devices since they have to carry out and execute EAPOL in addition to DTLS.

The authors in [44] realized a low-cost symmetric payload embedded key based strong authentication and key management on CoAP. They introduced a lightweight security verification scheme with key protection management to set up a secure channel for Vehicle Tracking System. This lightweight protocol removes expensive handshaking for reliability and is based on information about the vehicle circumstances.

Chu et al. proposed an authentication scheme based on ECC where the elliptic curve public parameters are computed by the base station (BS) in the initialization phase [45]. While parameters are computed at the BS, extensive communication of values is necessary after calculation. Furthermore, it demands an offline/pre-deployment phase to install the network. Authors, in [46], depicted how the components implicated can interact with each other to implement an authentication mechanism in various IoT cloud scenarios. Here, merely secure Internet of Things devices may be recognized for joining an IoT Cloud provider. After the joining process, several categories of authentication are considered for Manufacturer, Advanced user access, and Basic user. Industrialists might regularly access the IoT device for configuration setup, firmware updates, bug fixing, and other administrative missions. Basic users get right of access to IoT devices via the Cloud IoT Platform, while Advanced Users get direct admission to Internet of Things devices. Nevertheless, for implementing such authentications, users should authenticate themselves through the IoT Cloud Provider. The proposers of an OAuth based Authentication Mechanism for IoT Networks; perform an authorization management for the RESTful Web Service API access based on the OAuth2.0 authorization framework [47]. When IoT devices should be accessed by users, to avert communicating by means of cloud infrastructure for every request, security infrastructure is carried outside the IoT device. This makes it less prone to physical attacks. In this situation, access to device resources is achieved immediately, in a distant access scenario via the cloud infrastructure [48].

5. CONTIKI OS

5.1. Background

Contiki is an operating system written in C language, portable and open source for miniature sensors and the Internet of Things. Contiki is specially designed for respect the constraints of the WSNs, in particular, those related to limitations of memory space (it occupies about 32 kilobytes of ROM and 4 kilobytes of RAM) [1]. Contiki is an open-source software published under a BSD license. In 2002, Adam Dunkels created Contiki and now other developers from Oxford University, Texas Instruments, SAP, and Cisco then develop it. The origin of the word Contiki comes from Thor Heyer dahl's famous k -Tiki raft [49].

The OS has upheld for many different low powered devices, with a purpose to permit IoT capability by proposing IPv4, IPv6, 6LoWPAN, CoAP, etc. Nowadays, Contiki is utilized in both commercial and non-commercial applications to provide services as industrial tracking, smart city devices, and much more. [3]

Contiki supplies multitasking and it disposes of a built-in internet protocol (TCP/IP). It needs 30KB (ROM) and 10KB (RAM). Whether a system encompasses graphical user interface then it needs 30KB (RAM). [49]

Contiki includes an event-driven kernel, above which application programs can be dynamically loaded and unloaded at runtime. Contiki processes utilize proto threading; a programming style that offers a good compromise between event programming and multithreading programming. In addition to proto threading, Contiki also supports pre-emption between threads.

For communication, Contiki implements two mechanisms: Rhyme and uIP. The first mechanism is a layer just below the applications. Such layer provides a set of communication instructions. For its part, the second mechanism (uIP: micro IP) is a suitable implementation of an IP-based protocol stack (Protocols: TCP, UDP, IP, ICMP (Internet Control Message Protocol)). The adoption of such a communication mechanism makes possible the direct communication between a sensor and any IP host [1].

5.2. System Architecture

In applications with constrained memory, a multithreaded model uses a lot of this memory. Each thread necessitates a devoted stack, and it is not possible to anticipate the size of the stack previously. Thereby, the stack is overloaded, as this memory allocation should be accomplished at the time of thread set up. Similarly, when we have common resources, we have resource locking mechanisms so that existing operations do not modify resources. Event driven systems deals with processes as event handlers that monopolize the CPU until completion. Nevertheless, this also becomes the issue of event-driven systems, the CPU is incapable of answering the external stimulus if the event being run is long (e.g. cryptographic algorithms). Contiki resolves this issue with the aid of a hybrid model which is based on an event-driven kernel where pre-emptive multi-threading is carried out as an application library that is optionally connected with programs that demand it.

A running Contiki system includes libraries, a program loader, kernel, and a group of processes (could be either an application program or service). A process is determined by means of its event handler function and an optional poll handler function. The state of the process is upheld in its own memory whereas the kernel only contains a pointer to the process state. All processes have the same address space; they do not function in various protection domains. Inter-process communication is performed by displaying events.

As depicted below, figure 19, the Contiki system is divided into core and loaded programs. This division is done at compile time. The core includes the program loader, the Contiki kernel, the most popularly employed parts of language runtime and support libraries.

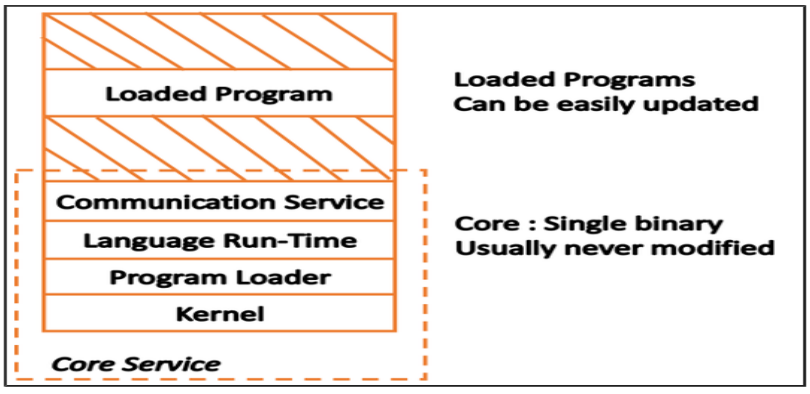

Figure 19. Contiki OS: Partitioning into core and loaded programs [50]

The kernel is composed of a schedule of lightweight event, which affects events to executing processes and regularly verifies the polling handlers of these processes. Program execution is based on events or by the polling mechanism. It does not acquire an event handler once it has been scheduled. Thus, event handlers should perform to completion. The kernel carries asynchronous (queuing of target process), synchronous (direct scheduling of target process) and polling. High precedence events, in polling, are directly transmitted to the processor in between another event. The kernel utilizes a single common stack for the processes to optimize memory usage. As referred to above, Contiki kernel supports loadable programs. Contiki includes a power save mode preservation that conserves when the network is in disabled state.

All functionalities of Contiki OS (communication, device drivers, sensor data handling, etc.) are supplied in the form of services. Every service has its proper interface and implementation, applications, which use an individual service, have only know the service interface not the execution of it. Figure 20 point out the block chart of the Contiki OS architecture. [7]

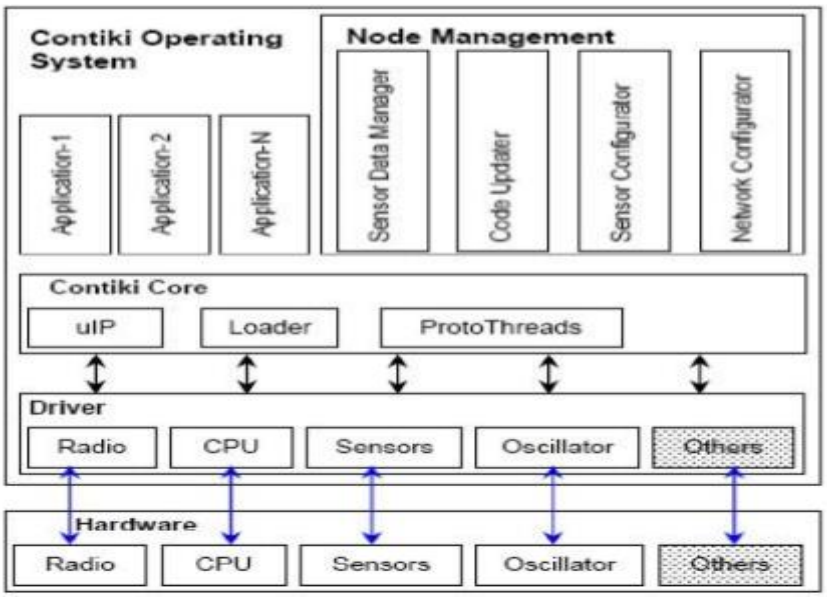

Figure 20. Contiki OS Architecture [7]

5.3. Contiki Characteristics

The Contiki OS has many advantages characteristics; below four of them are listed with short explication.

5.3.1. Management and Memory Allocation [7]

Contiki is created for minuscule systems that are able to work only some kilobytes of memory available. A standard design needs 2 KB of RAM and 40 KB of ROM; so, it is extremely memory efficient and supplies an ensemble of memory allocation process (memory block allocation-memb, a managed memory allocator-mmem and standard C memory allocator-malloc). The OS takes in charge dynamic linking of the programs and dynamic memory management, involving managed memory allocator (mmem) with the main mission of maintaining the allocated memory free from fragmentation through clustering the memory when blocks are free.

5.3.2. Power Awareness [7]

The OS is conceived for highly low-power frameworks that can require running for long time on a couple of AA batteries. It does no longer deliver any power economy capabilities; setting the devices into rest mode or another sort of power economy action shall be manipulated by the applications. Nevertheless, Contiki gives a system power consumption estimation mechanism to verify where the power was consumed.

5.3.3. Cooja Simulator [50]

COOJA is a network simulator that enables the emulation of physical equipment systems. Cooja plays the role of the application of Contiki OS focusing on network conduct. With this application, the user can simulate WSNs without none special mote. Cooja sustained an ensemble of standards, which are Contiki-RPL, TR 1100, IEEE 802.15.4, TI CC2420, uIPv4 stack and uIPv6 stack. It exists four spread models in the COOJA simulator, which should be chosen before beginning a further simulation. The prime model is constant loss Unit Disk Graph Medium (UDGM), which takes the ideal transmission range disk where the motes inside the transmission disk get data packages and motes outside the disk

do not receive anything. The second model is distance loss UDGM, which is the extending of the first model. It treats the radio interferences. Packages are transferred with “success ratio TX” probability and packets are recognized with a probability of “success ratio RX”. The third one is Directed Graph Radio Medium (DGRM) and it exposes the spread delays for the radio binds. The fourth model is Multipath Ray-tracer Medium (MRM) that utilizes the process of ray tracing (Friis formula, as an example) to compute the receiver power. MRM is as well able of calculating reflections, and refractions along the radio links, and diffractions.

The interface of the COOJA network simulator includes five windows. The physical disposition of the motes is shown in the network window. One ought to modify the physical position of the motes in order to build a topology. All the different motes have separate colours according to their task, for example, the sink mote has a green colour while the sender mote, and it is colour is yellow. The network windows allow the visibility of the mote attributes, mote type, radio environment of each mote and radio traffic between the motes. The window of simulation control displays the rate of the simulation and to stop, launch and reload the actual running simulation. For writing the theory and essential points of the simulation and save them, a note window exists. In the running simulation, for each mote, the network simulator points out a timeline. The usage of the timeline is to display the power consumption and the network traffic in the WSNs.

5.3.4. Full IP Networking [7]

Contiki supplies a full IP network stack; every application may employ both IPv4 and IPv6 stack. It consists of an implementation of µIP, a TCP/IP stack for 8-bit micro-controllers written in C, with a minimum set of characteristics to uphold TCP, UDP, ICMP and IP protocols, and the new low-power standards such as 6lowpan and RPL.

6. THE PROPOSED SCHEME

Because of the limited nature and scale of devices involved in the IoT, the security of the network is a fundamental challenge in such networks. Besides the security and privacy, several aspects require being taken into account for the IoT, together with context-recognition, scalability, and ease of deployment. The quantity of the data that the IoT generates is growing promptly, and requisition for real-time treatment is surging for cyber-physical systems. To fulfil these requirements, Cisco Systems introduced the Fog computing concept for the Internet of Things. [73]

Fog computing uses edge computing devices, such as smart gateways, laptop computers, and mobile phones, which can perform as gateways to the Internet. Edge computing is distributed, open IT architecture that represents decentralized treatment power, allowing mobile computing and IoT technologies. In edge computing, instead of transferring data to a data center, it is treated by the device itself or a local computer/server [74]. Processing or analysing data at the edge instead of in a centralized server would be beneficial for numerous reasons such as reducing Internet traffic and energy efficiency. Additionally, these solutions will enhance the scalability of wireless IoT Mesh networks in which the data packages will require going through a number of hops to arrive at the concentration entity as described by IETF 6LowPAN solution using IPv6 RPL routing protocol [75] [76].

Due to reasons given above, centralized protocols are not adequately suitable for multi-hop sensor networks. Hence, distributed algorithms must be established to deal with scalability challenges posed by multi-hop IoT mesh networks. The nodes, in the distributed algorithms, are able to make local decisions based on cached data obtained from a centralized entity of information acquired by listening to the local network. It is also recognized that the localized decision-making has its own challenges, as being locally optimum and freshness of the cached information in the case where a central entity forwards the information to local nodes to accelerate some process. The communication system among a great number of constrained resource devices has a significant role in the security and privacy of the engaged objects. The authentication in one of the fundamental requirements of a successful IoT network security architecture to prevent malicious users/devices from joining the network.

6.1. The Proposed Work

This thesis introduces an enhancement of the existing secure bootstrapping mechanism of the standard IETF 6TiSCH protocol in which a distributed data storage is used for the management of the authentication keys. With this distributed approach, the objective is to minimize the communication overhead of the IETF 6TiSCH authentication mechanism, enable an efficient authentication process and enhance energy efficiency of the network through holding the authentication parameters at the edge of the IoT network.

A zero-touch authentication approach for bootstrapping of IoT units is proposed by IETF 6TiSCH [77] working group. The contemporary inspiration consists of a centralized node for allowing authentication referred to as Join Registrar/Coordinator (JRC) wherein the nodes through a join-proxy/helper authenticate themselves to this node, as shown in figure 21. This mechanism has the plain disadvantage of a centralized method where scalability becomes a real challenge. Certainly, one may additionally claim that the procedure of authentication is a unique case wherein the node is bootstrapped to the network inside the starting and left for alone. However, it might be possible that the authentication credentials will require refreshing to enhance the security of the network that could place an extra communication overhead on the network with a restricted throughput.

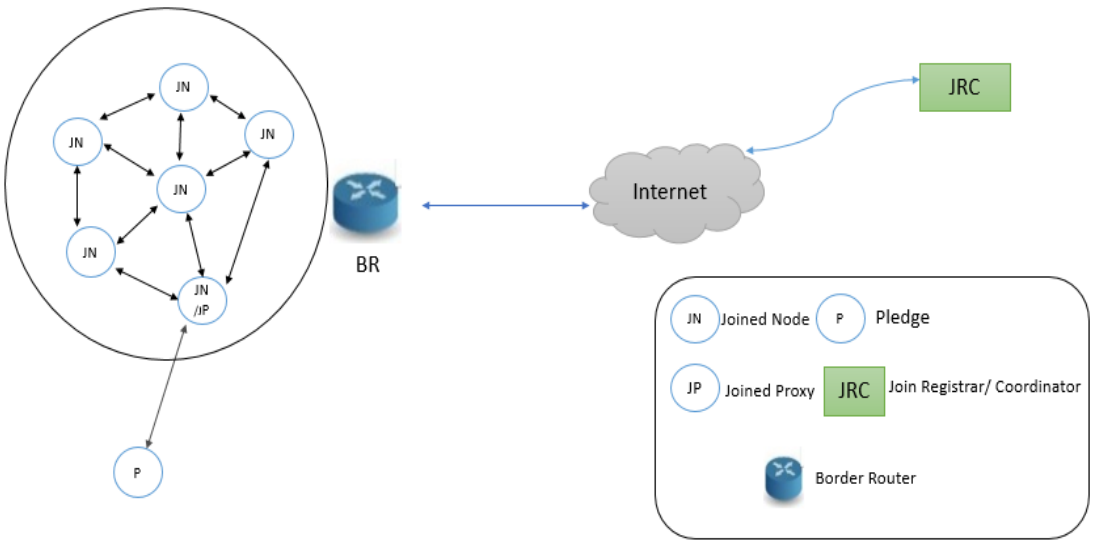

Figure 21. Overview of the join process

The suggested approach defines the minimum mechanism needed to accomplish a secure boot configuration of a device, which wants to join the 6TiSCH network [78]. The intent of the configuration is to establish a secure session between the device wanting to join the network and the node, which aids to integrate the device into the network with the use of a central authorization service. To achieve the defined steps, the device should be synchronized to the network by installing the link layer keys. At this level, the devices will communicate with each other securely. In addition to the events that occurs in the link layer, further security mechanisms may be included to the top layers.

The standard security mechanisms determined in IEEE 802.15.4 are used when link-layer frames are secured. Authentication on the link layer should be applied to the whole frame even the 802.15.4 heading. Encryption can be applied to the data load of the 802.15.4 standard. Using two different encryption keys (K1, K2), the standard performs the verification process. The K1 key is used to verify the tokens (it is assumed to be attributed to all devices). The K2 is utilized to verify and encrypt the data and confirmation frames. These keys can be preloaded on the devices or reported to the devices during the key distribution phase. The node, which wants to join the network, varies depending on whether the keys are preconfigured. If the K1 and K2 are preconfigured, the node that is involved in the system does not need a key distribution mechanism to learn these keys. Although K1 is known and K2 is unknown, the node confirms the tokens coming from neighbouring devices using K1 and depends on the key distribution mechanism to obtain K2. This key value and some other parameters are sent to the device in response to requests received by the central/distributed authentication entity. Once the node has validated the key value, it is included in the system and is now able to communicate with its neighbours.

All frames on the 6TiSCH network should use link layer security. Security options encompass data authentication and encryption. The candidate node validates these frames using AES-CCM to verify that the EB message is coming from 6TiSCH network. Link layer frames consists of 16 bytes key and 13 bytes nonce values, which have a great importance for encryption. This unique value used during encryption consists of the address of the device sending EB (8 bytes) and ASN (5 bytes).

Address (8 bytes)	ASN (5 bytes)

Figure 22. Link layer nonce structure

If the K1 and K2 are not preconfigured, the node make use of the OSCOAP protocol to be able to use the tokens. In the key distribution phase, the link layer key values are sent to the devices using this protocol. With these key values obtained by the device, they encrypt and validate data and confirm frames.

The IETF 6TiSCH group provides the authentication mechanism shown in figure 21 for booting IoT devices. The current proposal is a central structure to enable authentication entity called the JRC and auxiliary nodes having common properties with these nodes that help nodes to be involved in the network.

To be a part of a 6TiSCH network, the device must have a synchronization upon the network to get parameters like slot time duration, slot timing, channel-hopping sequence, etc. After that, the join process can start whether the wireless device holds the relevant authentication data for the network. At this step, it is expected to interact with the network to set up the link layer keys. After that, the node can initiate a secure end-to-end session with devices employing DTLS [79] or OSCOAP [80]. If application needs are known, the device communicates with its partners to ask for extra resources as required, or to reconfigure as network changes. As a result of joining the network, the candidate device waits for one or more link layer keys and, optionally, a temporary network identifier.

In the joining process, some preconditions must exist; for instance, a pre-shared key or certificate for authenticating the Enhanced Beacons (EB), another pre-shared key for the joining node to authenticate to the JRC, and a shared key or certificate for authenticating the device to the JRC response. The node, which wants to be a part of the network, is called Pledge in a 6TiSCH setting. After being authenticated to the network, the node is named Joined Node. While the Join Proxy is the node that transmits the join request to the JRC. The JRC in the 6TiSCH network takes the responsibility of authenticating and authorizing of the joining nodes. As depicted in [15], for authorizing the bootstrapping to occur, the 6TiSCH join operation specifies an Audit Token, also an Ownership Voucher pointing out that a definite range is the possessor of the joining node and a MIC (Manufacturer Installed

Certificate) for the bootstrapping process [81]. The actors in the proposed 6TiSCH bootstrapping process are shown in figure 21.

The joining node, in a 6TiSCH bootstrapping process, is supposed to hold a pre-shared key or a proper certificate for the network, which it chooses to be a part of. For synchronizing with the 6TiSCH network and finding out the parameters mentioned above, the Pledge first has to listen for EB messages. After getting a Beacon, the Pledge authenticates it with a pre-known key. The role of that step is checking that the Beacon is a 6TiSCH beacon and validate it as a 6TiSCH message. Subsequently, Pledge is synchronized upon the network but not authenticated. In the following operation, Pledge sends a Join Request to the node that it got the EB with the use of CoAP protocol. The request is transmitted to the JRC by the Join Proxy. Since the JRC address is known by the JP, join requests are always forwarded to the host server via the OSCOAP protocol. During the routing process, there is no need to know the nodes between pledge, JP and JRC. Network join requests are sent via multi-hops on the network layer without going to the application layer.

Utilizing each of two the pre-shared key or the pre-installed certificate, the JRC node authenticates the demand and forwards the Join Response back to the Join Proxy. Join Proxy sends the reply message to the pledge, which authenticates it using its authentication tokens for the JRC to ensure it got a Join Response from a legitimate JRC. As a result of the verification process, the candidate node may or may not be included in the network.

This work extends the 6TiSCH bootstrapping to encompass a Proxy JRC (P-JRC) that can contain authentication credentials for JRC and may authenticate devices to the network in the name of JRC by establishing a distributed bootstrapping model. The selection of the P-JRC device can be from the JNs or might be a pre-configured device with additional equipment. The aim of this study is to decrease the communication overhead of the bootstrapping, and authentication credential refresh procedure in the network. The implementation of the P-JRC can be in a unique trusted device into the network or in a distributed fashion where many nodes may mutually stock up the authentication data for the centralized JRC entity. In our suggested approach, as indicated in figure 23, a distributed P-JRC is realized within the network. We installed a duplicate P-JRC, which will cooperate to authenticate the Pledge. Each P-JRC holds partial credentials of the authentication for the pledge. As the instance for the standard 6TiSCH bootstrapping node, the process, in the proposed model, gets started when the Pledge sends a join request to the JP.

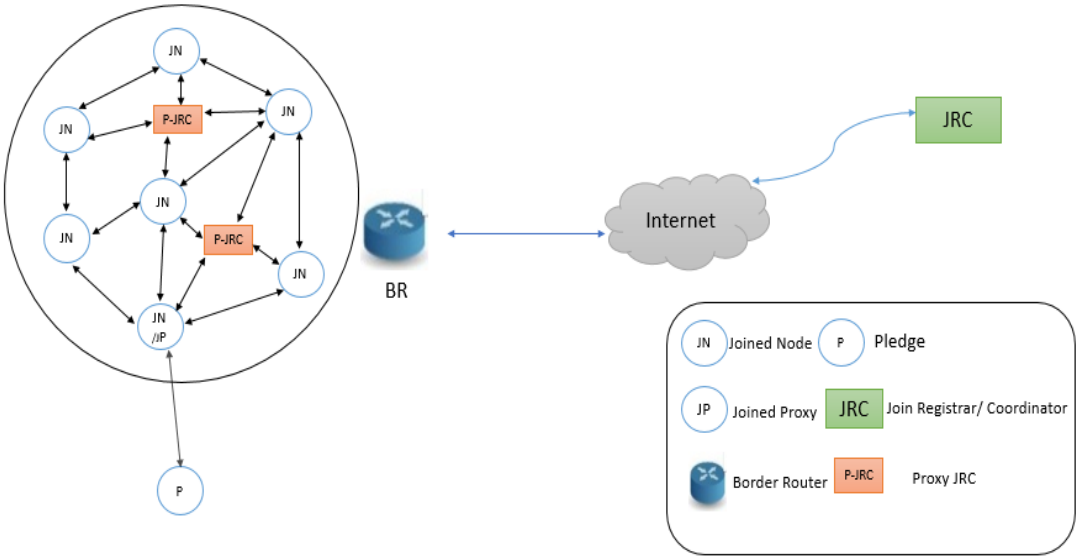

Figure 23. Proxy-based authentication infrastructure

With the target IPv6 address of the centralized JRC device, the Join Proxy transmits the request to the network. The request might reach the PJRC as it traverses the network. In this situation, the single P-JRC cannot authenticate the joining node alone. The PJRC contacts its duplicator, which is its distributed storage partner, to recover the full authentication key. After the Pledge is authenticated to the network, the authentication message is sent by the P-JRC to the Join Proxy to finalize the operation. To update the authentication tokens, the JRC device transmits new authentication tokens to the P-JRC device at predefined time-periods. Therefore, a lower communication overhead is enabled regarding to authenticating and refreshing the authentication credentials of the nodes centrally.

The key management has an important role in securing data and in the design of the authentication mechanisms. The current techniques of keys storage founded on a single trusted node (e.g. Base Station) deployment lead to communication bottlenecks and energy consumption imbalance [82]. Depending on the limited characteristics of the IoT devices, an appropriate data storage mechanism needs to be implemented. Therefore, storage of the Keys ought to be executed in a manner that it is possible to recover them on request reliably. To this end, a solution of a distributed storage is employed, wherein information belonging to a node is saved through some of the nodes and recovered with the aid of connecting a

subset of those nodes [83]. In this study, we used a distributed storage system to ensure a secure storage for the cryptographic keys.

RPL allows mutual traffic from the low power device to the Root node in the network and vice versa with the use of a DODAG. In our proposed mechanism, a network of limited energy devices is established employing RPL as the routing protocol. With RPL, the nodes may authenticate themselves to the JRC by means of the Root node of the network, which is designed as a border router between the emulated network and the Local Area Network (LAN) [75].

The proposed key management scheme is constructed on top of the above network model to offer powerful authentication and dynamic key establishment. The key material is generated at the JRC. To execute the scheme, some assumptions are made:

(i) The JRC is trusted entity where the symmetric keys and the dynamic IDs for the devices are stored.

(ii) Groups of two nodes are pre-selected.

(iii) The P-JRC nodes are trusted and secure entities.

The scheme includes two phases (1) System initialization and (2) Authentication of the Pledge.

Table 2. The notations used for the scheme

IDu	Device identity (can be the EUI64 of the device)
$DIDu$	Dynamic identity of the device
$PWDu$	Password of the device
$H\ (PWD)\ u$	Hashed version of the device password
$GKJRC{-}u$	Global symmetric key between the device and JRC
$PKJRC{-}u$	The symmetric private key between the JRC and device following authentication
TS	Timestamp for data freshness

The rest of table 2.

H (.):	One-way hash function (SHA-1)
$\|\|$	String concatenation operation

Message integrity is the feature by which data has not been changed in an unauthorized way since the time it was formed, transferred, or stored by means of an authorized source. The hash functions are the best way to protect the integrity of data blocks (e.g. text, messages, and files).

Objective functions are functions that produce output at fixed length, subjecting the data at any length to a particular extract algorithm. The advantages of this approach are often the guarantee of the integrity of the data, their speed, their output on a fixed length, the absence of extension of the file size and a high-performance communication.

The Secure Hash Algorithm (SHA), associated with some sort of message authentication method, ensures the integrity of programs and files when applied. In this technique, it is easy to calculate the value of an entry at a fixed length from an input value, but it is difficult to produce an input value that gives the same output value. The change of a bit in the input value changes the half of the bits in the value of the extract on average. It is mathematically very difficult to find a different input value that gives the same value as a given hashed value.

In this thesis, the most widely used SHA1 algorithm is utilized in the encryption algorithms designed by NSA [84]. With this algorithm, only encryption is done; decryption cannot be performed. The text used for encryption goes through certain operations and creates 160-bits. SHA1 consists of two stages, pre-processing and hashing. In the pre-processing phase, the messages are divided into 512-bit blocks and, if necessary, the length of the last block is reduced to 512 bits. The hash calculation uses this message to generate a series of functions, constants, and word operations, as well as a summary value.

The pre-processing must take place before the start of the extraction calculation and has three steps: fill the message (M), assign the filled message to the blocks, and set the initial summary value.

SHA-1 uses several logical functions such as $f0$, $f1$, ..., $f79$. $0 \leq t \leq 79$, each $f\ t$ function operates on three 32-bit words x, y, and z and produces a 32-bit output. The function f (t; x, y, z) is defined as follows.

$$f(t;x,y,z) = (x \wedge y) \vee \big((\neg x) \wedge z\big), \qquad 0 \leq t \leq 19 \qquad (1)$$

$$f(t;x,y,z) = x \oplus y \oplus z, \qquad 20 \leq t \leq 39 \qquad (2)$$

$$f(t;x,y,z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z), \qquad 40 \leq t \leq 59 \qquad (3)$$

$$f(t;x,y,z) = x \oplus y \oplus z, \qquad 60 \leq t \leq 79 \qquad (4)$$

In SHA-1, fixed words use Kt: K_0, K_1, ..., K_{79}, which consists of eighty 32-bit words.

$$K(t) = 5a827999, \qquad 0 \leq t \leq 19$$

$$K(t) = 6ed9eba1, \qquad 20 \leq t \leq 39$$

$$K(t) = 8f1bbcdc, \qquad 40 \leq t \leq 59$$

$$K(t) = ca62c1d6, \qquad 60 \leq t \leq 79$$

The purpose of filling with additional bits is to make sure that the message is 512 bits solid, depending on the algorithm. Once the message is added, it must be divided into 16 blocks of 32 bits so that the calculation of the hash can begin. Since the input block can be represented by 16 words of 32-bit, the first 32 bits of the message block are expressed as $M0^{(i)}$, the next 32 bits are $M1^{(i)}$ and the last $M15^{(i)}$. Before starting the calculation of the extract, the initial extraction value (H (0)) must be prepared for the algorithm to be applied. The size of H (0) and the number of words it contains relies on the size of the message digest. For SHA-1, the initial H (0) extract value contains the following five 32-bit words.

$$H0\ (0) = 0x67452301$$

$$H1\ (0) = 0xefcdab89$$

$$H2\ (0) = 0x98badcfe$$

$$H3\ (0) = 0x10325476$$

$$H4\ (0) = 0xc3d2e1f0$$

The computation is expressed utilizing two buffers, each consisting of five 32-bit variable and eight 32-bit message sequences. The words of the hash value are labelled as H (0), W (1), ..., W (79), while the words of H0, H1, H2, H3, H4 and 80 messages are labelled. The main algorithm uses to change the state of each 512-bit message block. The treatment

of a message block consists of a non-linear function, modular acquisition, and 80 similar operations on the left shift. As a result of these operations, the 160-bit summary is calculated. This value is stored for future use.

Conflicts in SHA1 are not easy to occur. SHA-256 requires much more processing than SHA-1 but has similar structure. Therefore, SHA1 was chosen in the study to reduce the cost of transaction. Powerful devices, which are the registration center in the verification process, have direct access to the actual identification information of the users. This feature allows malicious people to access identity information of users by attacking from the inside. To close this gap, the encrypted functions of the credentials were stored in databases using the hash functions. Even if an attacker physically sees the authentication server, one-way hash functions are used, so a reverse action cannot be obtained, and the user cannot obtain the actual identity values. For this reason, the authentication elements used within the scope of the thesis use the identity values created by the simplicity provided by the unidirectional hash functions of the users (nodes). With this feature, even if the elements are captured, the attacker cannot obtain real identity parameters. [85]

6.1.1. System Initialization Phase

At this phase, 8 bytes for ID number and 20 byte for password parameters are produced by the device. Generated credentials are created from the link address that is specific to each device. To establish the *DIDu* of the devices, the hashed version of the (*PWD*) and are concatenated and hashed. A secure symmetric key for the JRC device is also generated. However, before storing it in the registered device, the key will be divided into two parts. Each entity of the pre-selected groups stores its part of the key. This allows the authenticating node to confirm the JRC response.

$$(\mathrm{DIDu}) = (\mathrm{H}(\mathrm{IDu}||\mathrm{H}(\mathrm{PWDu}))) \quad (5)$$

The $DIDu$ and $H\ (PWD)\ u$ parameters are securely stored in the JRC device. Since both sides are supposed to trust each other at this stage, no attack situation is taken into consideration. Since the hashed versions of the credentials are stored in the JRC device, an attacker having access to the transmitted information of the device cannot procure the original identity of the devices from this hashed information. At this stage, JRC provides the

device a symmetric key value so that it can use it at other stages. On the other hand, the JRC can authenticate the incoming authentication message by comparing the incoming $DIDu$ to the DID generated from the stored credentials for the pledge.

6.1.2. Authentication Phase

As we explained in the last section, 6TiSCH demands the EBs to be authenticated employing a key, which can be pre-configured or obtained within the join process. After getting the EB from the network, the node becomes synchronized to it and can pick up network parameters (slot timing, duration, and slot frame) from the Enhanced Beacon. Next, the Pledge sets up its link-local IPv6 address and declares it to the node it is synchronized to, such as Join Proxy. Then, the Pledge starts the join authentication process as depicted in figure 24. The join process that is proposed and described in this thesis implements a slightly modified version of the proposal defined in [68].

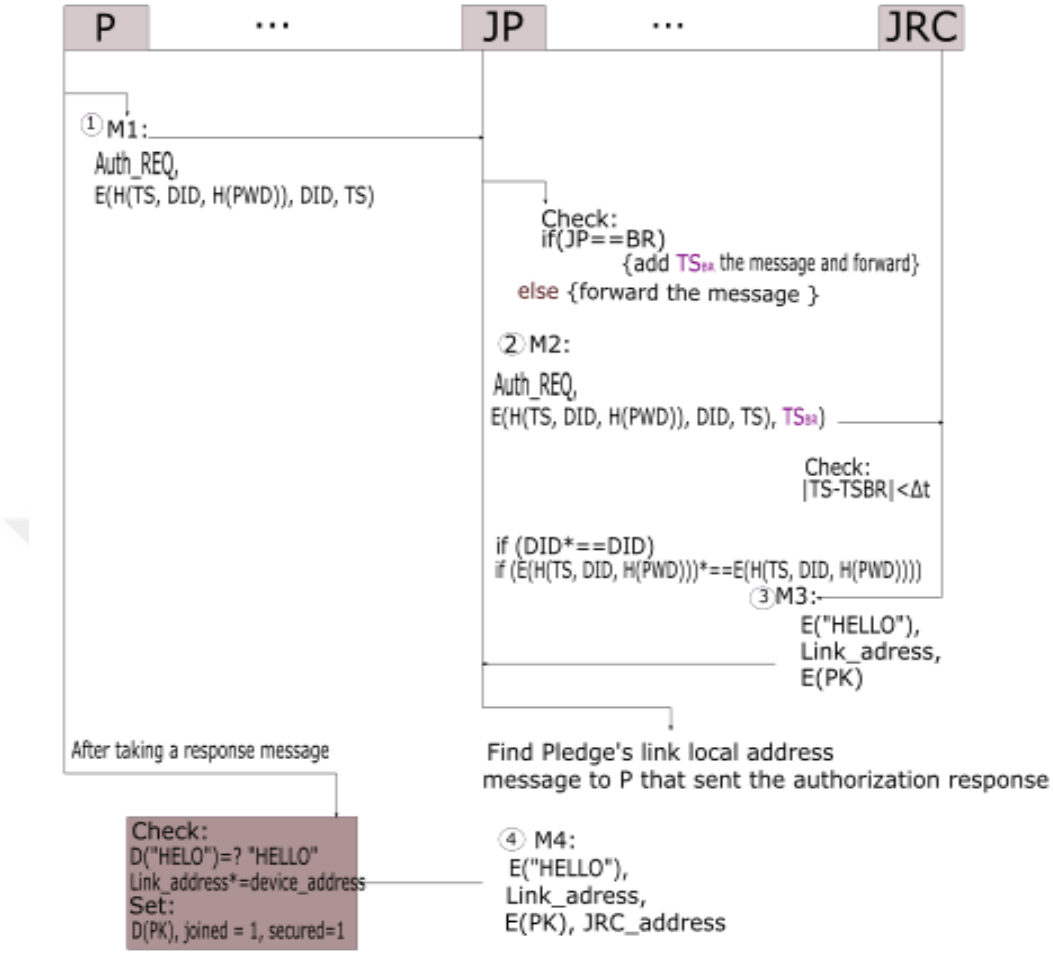

Figure 24. Implemented 6TiSCH bootstrapping model

For joining a 6TiSCH network, the node transmits an end-to-end encrypted message to JRC through the Join Proxy. Encryption of the Join Request message may be completed either using a pre-shared key or certificates installed on the Pledge. The authentication procedure, which employs pre-shared keys, needs a small number of messages to be transmitted between the Pledge and the JRC. For this reason, the pre-shared key-based authentication process is less communication intensive as compared to certificate-based authentication. Furthermore, in the case where the pre-shared key of the node is compromised, the intruder can get the keying material, which passed from the JRC to the Pledge. With this attack, the network can become unreliable. As well, it is the same case for a physically compromised node with pre-installed certificates. Where pre-shared keys are used, the JRC must keep separate keys for each node joining the network.

The Pledge forwards a Join Request packet to the Join Proxy after getting its link-local IPv6 address and setting up a registration in its local neighbourhood table for the Join Proxy. The transmission of the request package has been realized in the shared slot mentioned by the means of the EB as per 6TiSCH specification [69]. Statelessly, the packet is transmitted by the Join Proxy to the JRC. As depicted in figure 24, the received request is authenticated by the JRC by defining the Pledge credentials such as $DIDu$, $DIDu*$ and time stamp in its key database. After verification, if this holds, JRC forwards a "HELLO" message with authentication tokens assigned to the Join Proxy. After verification, if this holds, JRC forwards a "HELLO" message with authentication tokens assigned to the Join Proxy. The JP pull out the Pledge's address from the response message. After that, the Join Proxy sends a request to it is neighbour node, which is sharing the keys with it, to get the correspondent part of the key, as shown in the figure 25.

The JP, upon getting the completed key, sends the encrypted message with new authentication credentials to the Pledge. At the same time of receiving the incoming response, the pledge starts the authentication of the packet with credentials, which is already stored for the JRC. In this manner, the authentication of the response is validated.

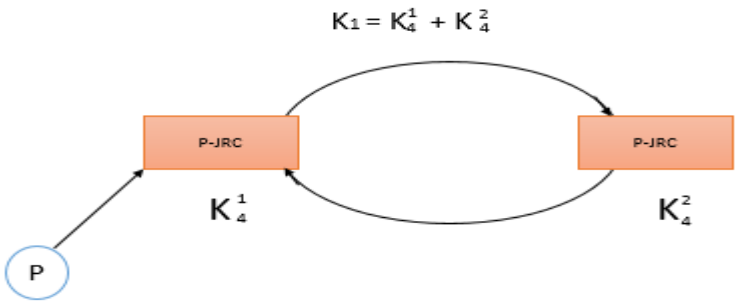

Figure 25. Key generation

In order to decrease the communication overhead brought through the centralized authentication process, in our approach, a duplicate JRC is located inside the 6TiSCH network. Certainly, it is predicted that the authentication tokens are not suitable for a whole network with low energy device. However, if it is anticipated that some of the devices in the network store a selected subset of authentication data for the part of the network (an RPL network), it is achievable to enhance the performance of the network through leveraging of

shorter authentication routes. The focus of our proposed scheme, it is only the effect of a distributed authentication system on the flow and energy consumption of the network. The picking out of the nodes that perform as duplicate JRC is out of the scope of this thesis and will be approached in the future work. In our process, we supposed that the nodes picked as P-JRC are preferably positioned inside the network and they have sufficient memory storage to store the authentication credentials for the network. Of course, the proposed approach aims to reduce the memory storage requirement by keeping partial keying material in neighbouring nodes. We should mention that in our proposed distributed mechanism that one P-JRC cannot authenticate the Pledge alone. Because, each P-JRC has partial credentials of the joining node. The P-JRC, which will be the authenticator of the Pledge, requires contacting its P-JRC partner to recover the completed key, as it shown in figure 25.

In the figure 26, which presented our proposed distributed authentication process, it is visible that if the P-JRC gets an authentication demand intended for JRC of its network, it cannot authenticate the Pledge on its own. The P-JRC sends a request to its P-JRC collaborator and gets the correspondent partial of credential. After recovering the whole key, P-JRC authenticates the Pledge and forwards a reply message to the Pledge along with authentication tokens for the Pledge. Otherwise, when the P-JRC is not over the routing path of the Pledge, as shown in figure 26 authentication mechanism occurs at the central JRC.

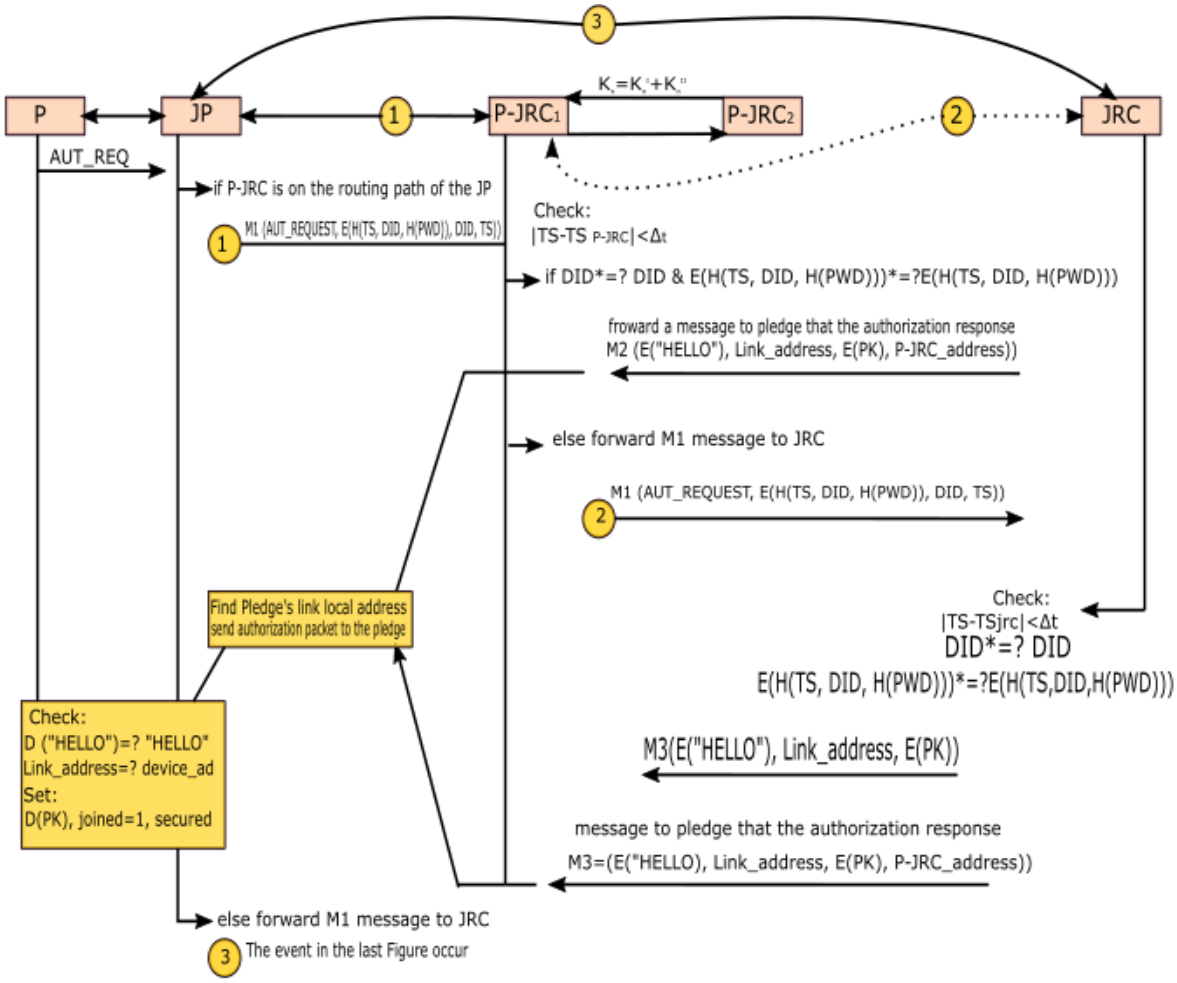

Figure 26. Proxy JRC based authentication model

In this thesis, the Contiki OS is used for the implementation of the suggested bootstrapping protocol for 6TiSCH and Cooja emulator exp5438 embedded platform [70] to evaluate the scheme. Table 3 presents the testing parameters. Every 20 and 25 designs are performed 5 times with distinctive random seeds and the medium authentication delay is estimated utilizing the average data from the simulations. For both designs, two perfectly placed P-JRC are employed. Forwarding its synchronization to the 6TiSCH network, every node, as mentioned in Table 3, transmits an authentication demand after a random rest time between 30 and 60 seconds.

Table 3. Testing parameters

Parameters	Value
Number of nodes	20-25
Start-up delay (minute)	30
Authentication request delay (second)	30-60
Rx (%)	70-80-90-100
Propagation node	Cooja UDGM

6.2. Experimental Results

Since the sensor devices have constrained resources, it is significant to measure the security mechanisms applied (the encryption speed and the energy consumption). This section presents the methods used to measure parameters such as workload, ROM, RAM and energy consumption of IoT applications. Then, these methods were used for evaluation. The most accurate method is to use a high precision oscilloscope to control the output pin of the data. This method seems to be more complex and many external factors must be taken into consideration for the measurement case. The second method is to use real-time timers in Contiki. By using real-time structure, the results with greater accuracy are seen in the execution time. The Contiki operating system supports real-time and on-time schedulers.

Since the WSN is equipped with a constrained power source, energy is a very limited resource. While normal applications consume energy in the micro joule space, security mechanisms with open key encryption complexity cause energy consumption in the mill-joule dimensions. For this reason, symmetric encryption technique with low energy consumption is used in the proposed authentication mechanism. Contiki offers a tool called ENERGEST for energy estimates. Energest is a software-based mechanism, which estimates the energy consumption of the sensor node. It is used by the sensor node to provide energy estimates for all components, such as the radio receiver and CPU in the Contiki. This process is based on the use of base clocks (timers). It holds a table with all components, such as CPU, radio receiver and LEDs. When a component is active, a counter-estimate starts measuring the energy consumption. When the component is turned off, the current value is added to the registration table for this component. The difference between the two values indicates the energy consumed during use by multiplying by the resulting power. To

normalize table values, the characteristics of the selected material must be known. Table 3 shows the parameters used for normalization. These values were obtained from the MSP-EXP430F5438 and CC2538 datasets [71, 72].

Table 4. Currents Driven by Components

Component	**Instantaneous consumption (mA)**
CPU	1.99
LPM	0.0545
Tx	20
Rx	17.7
Vcc	3.3V

Once each component has been registered, it must be multiplied by the voltage of the sensor node to obtain energy from non-standard values. The values obtained are divided into RTIMIER_SECOND of the Contiki operating system and used for real-time energy consumption. This variable refers to the real time clock. Vcc is used at 3.3V because the sensor node is in full power mode. The following formula shows the code format for the energy used in the study.

avg_power = ((1.9 * cpu + 0.545 * lpm + 20 * listen + 17.7 * transmit) * 3.3/ RTIMER_SECOND)); (6)

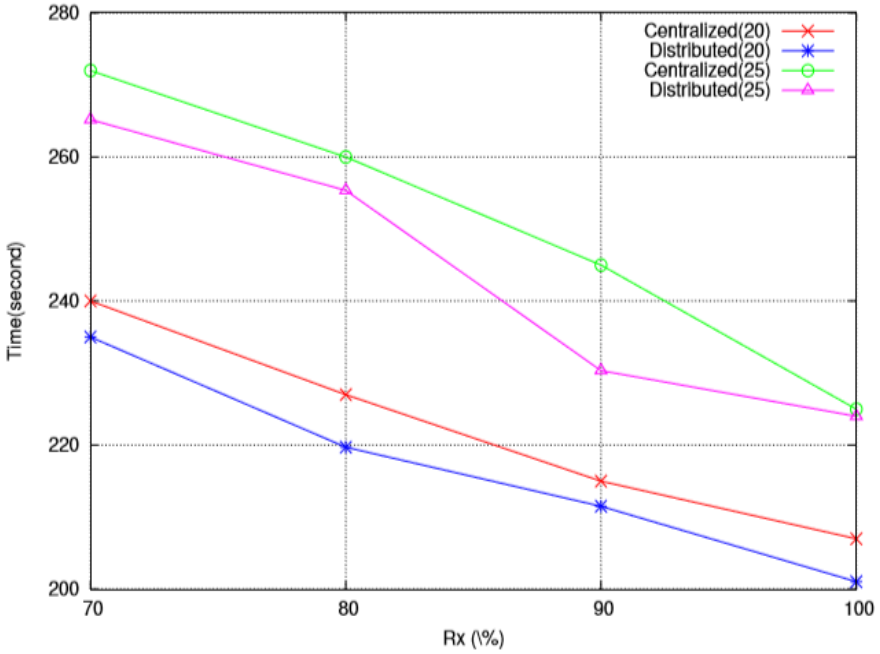

Figure 27. Authentication time for the evaluated networks

The performance of the both executed authentication process for 20 and 25 nodes is shown correspondingly in the figure 27. It is recognized, for the scenario of 20 nodes network that the authentication duration for the Pledge changes in accordance with the mean link quality of the network for the centralized and distributed approaches. However, the distributed scenario results are better in terms of authentication delay than the centralized approach. In the instance of the distributed authentication using two P-JRCs, authentication delay is less for all the link success probabilities comparing to the centralized authentication. Otherwise, in context of authentication time for the 20 and 25 nodes instances, the distributed approach surpasses the centralized approach process by approximately 10%.

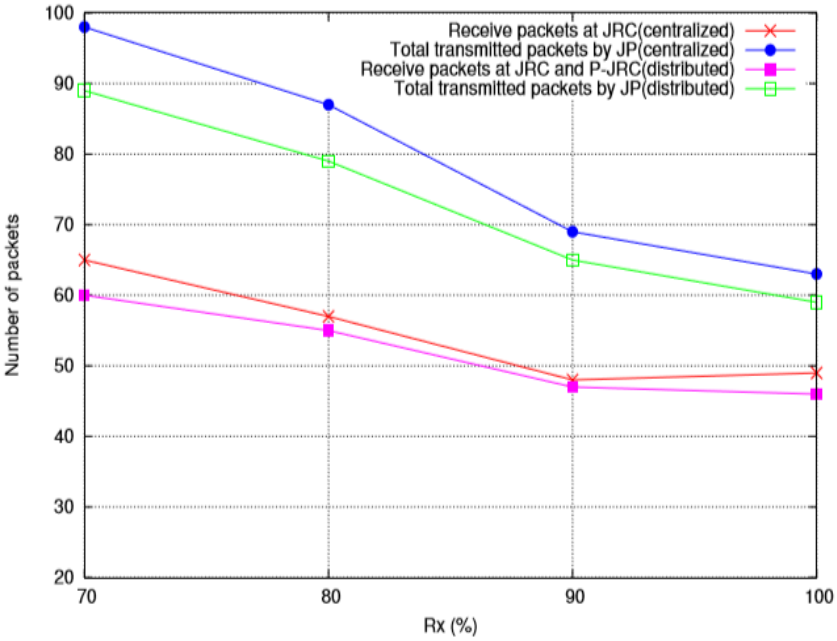

Figure 28. Number of transmitted and received packets for 25 nodes

The amount of the transmitted and received authentication packages for the centralized and distributed authentication process of 25 nodes network is represented in figure 28. The figure depicts the overall number of the transmitted and received packages in terms of link success probabilities. It is observed that in the case of a centralized authentication process, the overall number of packages, which should pass through the whole network for the authentication mechanism, is higher than the proposed distributed authentication approach. Certainly, we have supposed that the two P-JRC hold all the authentication keys before the evaluation. The effect of getting to update the keys inside the duplicated JRCs in the network has to be studied as future work. Despite that, the comparison of the results of the packets transmission indicates that the traffic generated by the distributed authentication mechanism is about 11% less than that of the centralized authentication for IETF 6TiSCH networks.

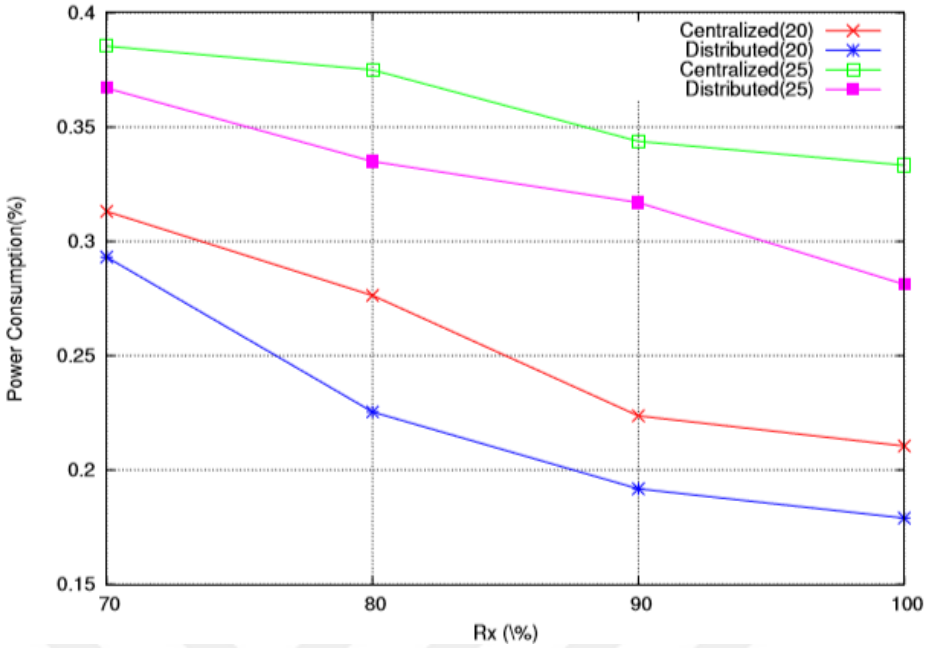

Figure 29. The average amount of energy consumed during the boot process

The last figure figure 29 shows the average amount of energy consumption all along the bootstrapping of the nodes in the network with regards to the battery power where a 3.3V100 mAh battery is supposed. The graph points out that the energy economy of the distributed authentication process gets to be more significant with the growing amount of the nodes inside the network. Moreover, the link quality of the network influences considerably the energy consumption of the bootstrapping process as expected. It is shown through the initial results that driving the authentication mechanism to the edge of the network may notably minimize the rate of the energy consumption of a 6TiSCH based IoT network.

However, because of the variable size of the stack, it is difficult to measure the use of RAM and the stack is used for dynamic memory allocation. Table.4 gives details of the executable areas for the exp5438-embedded platform. An executable program is divided into sections called .text, .bss, and .data.

- The **.text** field points to static memory, which contains code and static variables.

- The **.data** field creates a dataset that does not occupy any space on the ROM that is set to zero when the program starts, but that is stored in RAM during program execution.
- The **.bss** field is the value initialized at the start of the program, so it must be stored in the ROM as well as in the RAM.

The amount of memory used in the RAM for the authentication mechanisms used in the client application is almost the same. The amount of ROM used in the distributed authentication method is 5 kB more malice than the central structure. A larger network share of the limit router increases the memory consumption in RAM / ROM areas based on the client application. Similar to the Client application, the border router adds an additional cost of 5 kB to the ROM memory space when using a distributed authentication mechanism compared to central authentication. The memory consumption rates in the RAM area are very close to each other for the verification mechanisms used.

Table 5. Memory Usage of Centralized and Distributed Authentication Protocols (Bytes)

	Centralized	Distributed
User		
.text	111085	116986
.bss	13438	13468
.data	630	638
RAM (data + bss)	14068	14106
ROM(text + data)	111715	117624
Border router		
.text	112780	118680
.bss	14454	14484
.data	662	670
RAM	15116	15154
ROM	113442	119350

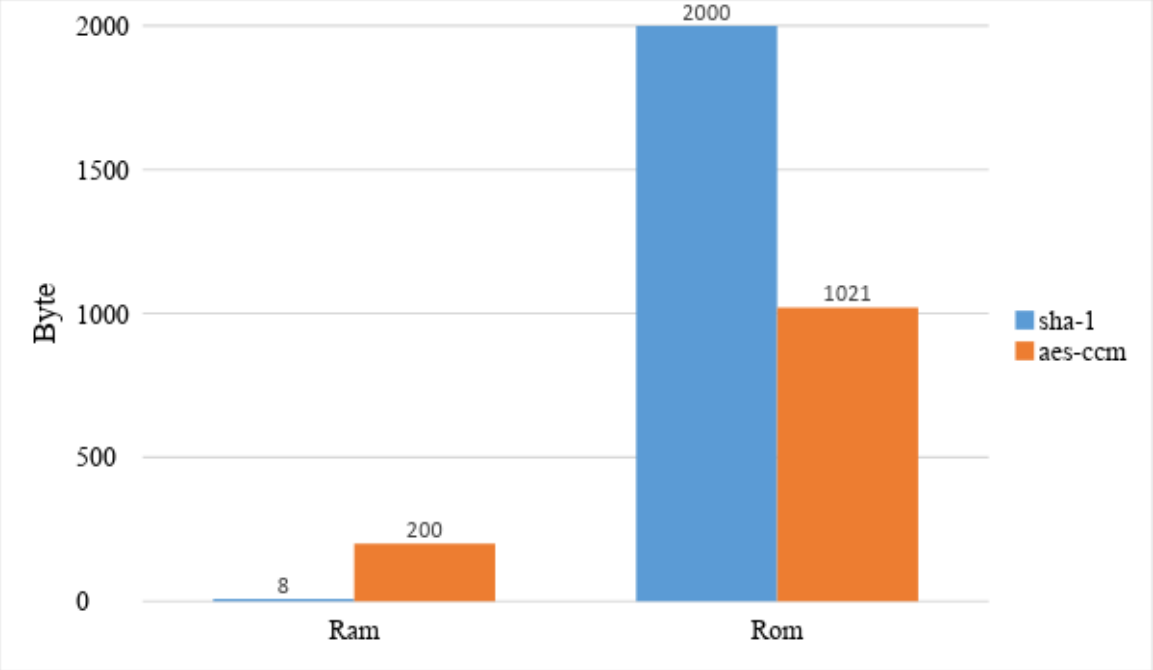

Figure 30. Memory Consumption (Bytes) for AES-128 and Sha-1

Figure 30 indicates the areas covered by the methods used for authentication mechanisms (AES-128, SHA-1). Most of these methods are stored in ROM memory. In the RAM memory, the variables created in the run state are recorded. The proposed method does not cause excessive to nodes because it includes a simple mechanism that performs symmetric encryption and integrity checking.

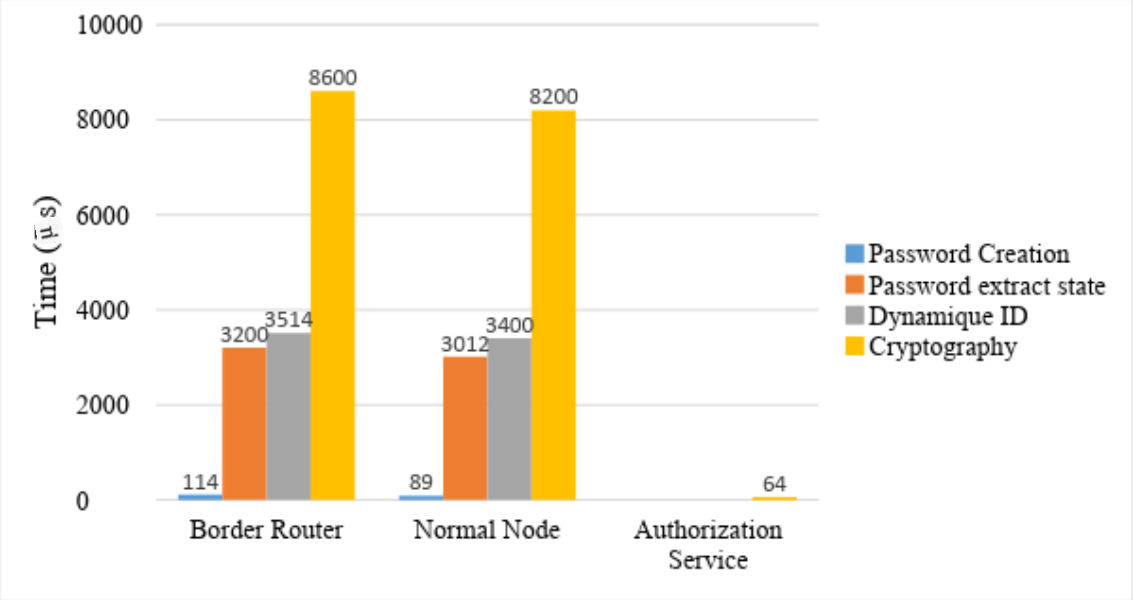

Figure 31. The average time of the operations performed for authentication

Figure 31 shows the average time of transmission for which the authentication has been performed by location devices. Here, the border router and the normal node are evaluated together because they have limited resources. The authorization service is considered a device that does not have resource constraints when it is handled centrally. For the password creation, it generates a value of 20 bytes for each specific device. This value is known as a new form with the help of hash function. This value is then saved for use in the JRC database. With the hashed state, the authorization service cannot directly access the actual identity information of the device. This helps to hide basic credentials. The creation of dynamic identity information has a common share with the benefit that the summary of the above-mentioned ciphers brings. With this value, the authentication process is in progress or terminates. As the AES-128 symmetric encryption technique is used in the encryption step, the system does not need additional cost. Since the system automatically switches to safe mode after the verification steps, all message traffic is encrypted. This feature ensures that devices included in the network can communicate securely with each other.

7. CONCLUSION

IoT is a very powerful puzzle word, which is able to change our ordinary lives into latest movies of science fiction when it is performed and incorporated with success. It is not something impossible to imagine an everyday life enclosed by smart objects, notably after the popularity that acquired the smartphones in the last few years.

This new concept has allowed us to discover a new dimension, brought us to encounter new technologies such as Contiki, and enhanced elderly technologies like ZigBee. Furthermore, it is a project in progress, which has a lot to offer and a path to pass.

More precisely, the IoT is an evolution of today's Internet that is born of the convergence of several types of networks and technologies, particularly IPv6 Internet, wireless sensor networks and RFID technology. Indeed, sensor networks represent the most interesting part of all the founding technologies of IoT. They have already achieved remarkable success in different areas of urban, rural, civil and military applications. In addition, with the integration to the Internet, their benefits and application efficiencies are expected to take a much wider space with new perspectives.

With the emergence of IoT, security risks are becoming enormous with a high degree of diversification and severity. On the other hand, the constraints imposed on the sensor networks (mainly the resource limitations) prevent the installation of highly robust security mechanisms because they require powerful devices, which is not the case for the sensors required.

Authentication, which is one of the most significant security requirements, is the main obstacles in front of the IoT vision. For authenticating, a network with tens of millions of nodes is not always an easy assignment notably when the network exigencies and the application areas can vary a lot. Nevertheless, in nearly every node, the behaviour of the resource- constrained is common. It gives us the chance to generalize and establish a security system. Supplying robust security, which is split into various levels or options to take into account different requirements of different applications and provide flexibility, is the intended system for IoT networks.

To establish a full security solution, data on nodes should be secured when stored and transferred. It is fundamental to find effective solutions for the resource-limited embedded systems. In this study, the proposed scheme combining security aspects of storage and communication is able to attain this purpose.

The proposed mechanism can be applied in many practical IoT applications where the authentication and key management are highly required.

8. REFERENCES

1. Sahraoui, P. S., Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things), Thesis, November 2016.

2. Oracevic, A., Dilek, S. and Ozdemir, S., Security in Internet of Things, *Ieee Isncc*, no. p. 100, June 2015.

3. Andersen, Ø. L., Security of Internet of Things Protocol Stacks, June 2016.

4. Bude, C. and Bergstrand, A. K., Internet of Things Exploring and Securing a Future Concept, June 2015.

5. Marshall, P., Re-engineering security for the Internet of Things, Telecom Asia, June 2015.

6. Challal, Y., Sécurité de l'Internet des Objets : vers une approche cognitive et systémique Objet intelligent, September 2013.

7. Kalyoncu, S., Wireless Solutions and Authentication Mechanisms for Contiki Based Internet of Things Networks, September 2013.

8. Bieller, S., Cybersecurity in the Industrial Internet of Things, June 2003.

9. Boé, A., Sécurité de l'internet des objets, Februry 2017.

10. Raheem, A. H., An integrated security protocol communication scheme for internet of things using the locator/id separation protocol network, PQDT - UK Irel., January 2017.

11. IEEE Computer Society, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), Institute of Electrical and Electronics Engineers, New York, NY, IEEE Standard 0-7381-4997-7, Sep. 2006 [Online]. Available: https://standards.ieee.org/getieee802/download/802.15.4-2011.pdf

12. Samaila, M. G., Neto, M., Fernandes, D. A. B., Freire, M. M. and Inácio, P. R. M., Security challenges of the Internet of Things, Internet of Things, no. 9783319507569, (53–82), May 2017.

13. http://www.techsparks.co.in/thesis-topics-in-internet-of-things/, Internet of Things.

14. Chen, L., Security Management for The Internet of Things, p. 71, April 2017.

15. Sitenkov, D., Seitz, S.-L., Raza, S. and Selander, G, Access Control in the Internet of Things.

16. Höglund, R., Lightweight Message Authentication for the Internet of Things, November 2014.

17. Montenegro, G., Kushalnagar, N., Hui, J. and Culler, D., Transmission of IPv6 packets over IEEE 802.15.4 networks, Request for Comments 4944, September 2007.

18. Abou, U. and Belkaid-Tlemcen, B., Developpement d'une application deployée sur un réseau de capteurs sans fil supportant 6LowPAN, June 2015.

19. Bormann, C., 6LoWPAN Generic Compression of Headers and Header-Like Payloads, Internet- draft-bormann-6lowpan-ghc-04, 2012.

20. Winter, T., Thubert, P., Randt A. B., Hui, J., Kelseky, R., Levis, P., Pister, K., Struik, R., Vasseur, J.P. and Alexander, R., RPL: IPv6 routing protocol for low-power and lossy networks, Request for Comments 6550, 2012.

21. Raza, S. and Avhandling, A., Lightweight security solutions for the internet of things, no. 139, June 2013.

22. Monir, S., A Lightweight Attribute-Based Access Control System for IoT, June 2017.

23. Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F. and AlonsoZarate, J., A survey on application layer protocols for the internet of things, Transaction on IoT and Cloud Computing, vol. 1, no. 1, 2015.

24. Keoh, S. L., Kumar, S. and Tschofenig, H., Securing the internet of things: A standardization perspective, Internet of Things Journal, IEEE, vol. 1, no. 3, pp. 265275, June 2014.

25. Lakkundi, V. and Singh, K., Lightweight DTLS implementation in COAP based internet of things, International Conference on Advanced Computing and Communications ADCOM 2014, 2014.

26. Thangavel, D., Xiaoping, Ma., Valera, A., Tan, H-X. and Colin Keng-Yan, T., Performance evaluation of MQTT and CoAP via a common middleware, IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), (1–6) IEEE, April 2014.

27. De Caro, N., Colitti, W., Steenhaut, K., Mangino, G. and Reali, G., Comparison of two lightweight protocols for smartphone-based sensing, IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT), (1–6).IEEE, November 2013.

28. Serbanati, A., Segura, A. S., Olivereau, A., Saied, Y. B., Gruschka, N., Gessner, D. and Gomez-Marmol, F., Project Deliverable D4.2 – Concepts and Solutions for Privacy and Security in the Resolution Infrastructure, Internet-of-Things Architecture IoT-A, 2012.

29. https://motherboard.vice.com/en_us/article/mg79v4/hacking-glossary,The Motherboard e-Glossary of Cyber Terms and Hacking Lingo, July 2016.

30. Wilson, P. and Tech, B., Inter-Device Authentication Protocol for the Internet of Things, 2012.

31. Betzler, A., Isern, J., Garles, C., Demirkol, L. and Paradells, J., Experimental evaluation of congestion control for CoAP communications without end-to-end reliability, Ad Hoc Networks, vol, 52, no 1, (183-194), December 2016.

32. Ngoepe, P., Meyer, W., Auret, W. and Diale, E., DLTS characterization of defects in GaN induced by electron beam exposure, Materials Science in Semiconductor Processing, vol 64, no 1, (29-31), June 2017.

33. Karlof, C., Sastry, N. and Wagner, D., TinySec: a Link Layer Security Architecture for Wireless Sensor Networks, Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, (162—175) Baltimore, November 2004.

34. Kothmay, T., Schmitt, C., Hu, W., Bruning, M. and Carle, G., DTLS based security and two authentication for the Internet of Things, Ad Hoc Networks, vol 11, no 8, (2710-2723), November 2013.

35. Uahhabi, Z. and Bakkali, H., An approach for evaluating trust in X.509 Certificates, IEEE International Conference for Internet Technology and Secured Transactions, vol, no 166674920, (196-203), 2016.

36. Qin, B., Liu, Sun, S., Deng, R. and Gu, D., Related-key secure key encapsulation from extended computational bilinear Diffie-Hellman, Information Sciences, vol 406-407, no 1, (1-11), September 2017.

37. Saikia, M. and Hussain, A., Improving the performance of Key pre-distribution scheme in sensor network using clustering of combinatorics, IEEE International Conference on Computing Communication and Automation, vol, no 16598900, (682-686), 2016.

38. Sharma, V., and Hussasin, Node authentication in WSN using key mechanism, IEEE International Conference on ICT in Business Industry & Government, vol. no 7892691, pp 17, 2016.

39. Shivraj, V. L., Rajan, M. A., Singh, M. and Balamuralidhar, P., One time password authentication scheme based on elliptic curves for internet of things (iot), (1–6), February 2015.

40. Crossman, M. A. and Liu, H., Study of authentication with iot testbed, Technologies for Homeland Security (HST), 2015 IEEE International Symposium on, (1–7), April 2015.

41. Shone, N., Dobbins, C., Hurst, W. and Shi, Q., Digital memories based mobile user authentication for IoT, (1796–1802), October 2015.

42. Yao, X., Han, X., Du, X. and Zhou, X., A lightweight multicast authentication mechanism for small scale IoT applications, IEEE Sensors Journal, vol. 13, (3693–3701), October 2013.

43. Hernndez-Ramos, J. L., Pawlowski, M. P., Jara, A. J., Skarmeta, A. F. and Ladid, L., Toward a lightweight authentication and authorization framework for smart objects, IEEE Journal on Selected Areas in Communications, vol. 33, (690– 702), April 2015.

44. Ukil, A., Bandyopadhyay, S., Bhattacharyya, A. and Pal, A., Lightweight security scheme for vehicle tracking system using coap, Proceedings of the International Workshop on Adaptive Security, ASPI ’13, (New York, NY, USA), (3:1–3:8, ACM), September 2013.

45. Chu, F., Zhang, R., Ni, R. and Dai, W., An improved identity authentication scheme for internet of things in heterogeneous networking environments, 16th International Conference on Network-Based Information Systems, (589– 593), September 2013.

46. Barreto, L., Celesti, A., Villari, M., Fazio, M. and Puliafito, A., An authentication model for iot clouds, Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, ASONAM ’15, (New York, NY, USA), (1032–1035), ACM, August 2015.

47. Emerson, S., Choi, Y. K., Hwang, D. Y., Kim, K. S. and Kim, K. H., An oauth based authentication mechanism for iot networks, Information and Communication Technology Convergence (ICTC), (1072–1074), October 2015.

48. Tschofenig, H., Fixing user authentication for the internet of things (IoT), Datenschutz und Datensicherheit - DuD, vol. 40, no. 4, (222–224), April 2016.

49. https://komalbattula.wordpress.com/, Contiki OS the open source operating system.

50. Mehmood, T., COOJA Network Simulator: Exploring the Infinite Possible Ways to Compute the Performance Metrics of IOT Based Smart Devices to Understand the Working of IoT Based Compression & Routing Protocols, December 2017.

51. Ndibanje, B., Lee, H-J. and Lee, S-G., Security Analysis and Improvements of Authentication and Access Control in the Internet of Things, August 2014.

52. Gaglio, S. and Lo Re, G., From ieee 802.15.4 to ieee 802.15.4e : A step towards the internet of things, Advances onto the Internet of Things, (135–152), January 2014.

53. Sudhakar, D.V.R., Effectiveness of time-slotted channel hopping in wireless in-vehicle networks, 2015.

54. IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer. IEEE Std 802.15.4e2012 (Amendment to IEEE Std 802.15.4-2011), (1–225), April 2012.

55. Palattella, M-R., Watteyne, T., Wang, Q., Muraoka, K., Accettura, N., Dujovne, D., Grieco, L. A. and Engel, T., On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks, January 2016.

56. Theoleyre, F., Georgios, Z. and Papadopoulos, Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks, November 2016.

57. Schindler, C. B., Watteyne, T., Vilajosana, X. and S. J. Pister, K., Implementation and Characterization of a Multi-hop 6TiSCH Network for Experimental Feedback Control of an Inverted Pendulum, May 2017.

58. Watteyne, T., Mehta, A. and Pister, K., Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense, Symposium on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks (PE-WASUN). (116–123) ACM, October 2009.

59. http://www.mavialp.com/en/product/detail/3/, Alp Protocol Stack A protocol stack for next generation IoT applications.

60. Watteyne, T., Tuset-Peir´o, P., Vilajosana, X., Pollin, S. and Krishnamachari, B., Teaching Communication Technologies and Standards for the Industrial IoT? Use 6TiSCH!, May 2017.

61. Vilajosana, X. and et, al., Minimal 6TiSCH Configuration, draft-vilajosana-6tisch-minimal-16, June 2016.

62. Wang, Q. and Vilajosana, X., 6top protocol (6p), draft-ietf-6tisch-6top-protocol-02, July 2016.

63. Dujovne, D., Grieco, LA., Palattella, MR. and Accettura, N., 6TiSCH 6top Scheduling Function Zero (SF0), draft-dujovne-6tisch-6top-sf0-00, March 2016.

64. Vasseur, J. P. and Le Roux, J. L., Path Computation Element (PCE) Communication Protocol (PCEP) RFC 5440, Internet Engineering Task Force, 2009.

65. Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S., Resource ReSerVation Protocol (RSVP): Version 1 Functional Specification, RFC 2205, Internet Engineering Task Force, 1997.

66. Hancock, R., Karagiannis, G., Loughney, J. and Van den Bosch, S., Next Steps in Signaling (NSIS): Framework, RFC 4080, Internet Engineering Task Force, 2005.

67. Palattellal, M. R., Thubert, P., Vilajosana, X., Watteyne, T., Wang, Q. and Engel, T., 6TiSCH Wireless Industrial Networks: Determinism Meets IPv6, January 2014.

68. Minimal security framework for 6tisch, https://tools.ietf.org/html/draft-ietf-6tischminimal-security-02, November 2017.

69. IETF. Ipv6 over the tsch mode of ieee 802.15.4e (6tisch). Available: https://datatracker.ietf.org/wg/6tisch/charter/, [Online]: October 2017

70. Msp-exp430f5438 experimenter board user's guide, texas instruments inc., da, texas, 2013. Available: http://www.ti.com/lit/ug/slau263i/slau263i.pdf, November 2017.

71. "Msp-exp430f5438 experimenter board userâ€™s guide, texas instruments inc., da, texas, 2013.." Available: http://www.ti.com/lit/ug/slau263i/slau263i.pdf, November 2017.

72. Lajara, R., Pelegr-Sebastiá, J. and Solano, J. J. P., Power consumption analysis of operating systems for wireless sensor networks, Sensors, vol. 10, no. 6, pp. 5809–5826, 2010.

73. Kim, H. and Lee, E. A., Authentication and Authorization for the Internet of Things, October 2017.

74. https://www.hpe.com/us/en/what-is/edge-computing.html, What is Edge Computing?

75. Winter, T., Rpl: Ipv6 routing protocol for low-power and lossy networks, 2012.

76. Internet Engineering Task Force (IETF) IPv6 over Low Power Wireless Personal Area Networks (6lowpan) Working Group. Available: http://datatracker.ietf.org/wg/6lowpan/, December 2010.

77. IETF. Ipv6 over the tsch mode of ieee 802.15.4e (6tisch). Available: https://datatracker.ietf.org/wg/6tisch/charter/, [Online]: October 2017.

78. 'Minimal security framework for 6tisch.' https://tools.ietf.org/html/draft-ietf-6tischminimal-security-02, EriÅŸim: November 2017.

79. Rescorla, E. and Modadugu, N., Datagram transport layer security version 1.2, 2012.

80. Selander, G., Mattsson, J., Palombini, F. and Seitz, L., Object security of coap (oscoap), IETF, Internet-Draft, 2015.

81. 6tisch zero-touch secure join protocol. Available: https://tools.ietf.org/html/draft-ietf-6tisch-dtsecurityzerotouch-join-01, [Online]: November 2017.

82. Weimin, G. and Lingzhi, Z., Distributed data storage in wireless sensor networks, International Journal of Database Theory and Application, 8(4): (179–182), 2015.

83. Gormus, S., Koc, A. G., Jin, Y. and Sooriyabandara, M., A storage centric approach to scalable sensor networks, Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st, (1–7). IEEE, 2015.

84. Eastlake, D. and Jones, P., Use secure hash algorithm 1 (sha1), tech. rep., 2001.

85. Aydin, H., IETF 6TiSCH Protokolü için Dağıtık Kullanıcı Kimlik Doğrulama Mekanizması, May 2018.

CURRICULUM VITAE

Yosr BOUMAIZA was born on the 2nd of May 1991 in Nabel, Tunisia. After finishing her primary and secondary education, BOUMAIZA enrolled in a Bachelor of Computer Science degree at the Faculty of Economic and Management Sciences of Nabel (FSEGN) of the University of Carthage. In 2014, she received a scholarship to continue her Master studies in Computer Engineering at Karadeniz Technical University in Turkey. BOUMAIZA speaks French, English and Turkish as well as her mother language, which is Arabic.