

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Indrek Taal 176080IDCR

Security Risk Analysis of Wireless Mesh Network for Emergency Networks

Diploma thesis

Supervisor: Toomas Lepik
Master's degree

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Indrek Taal 176080IDCR

Turvaanalüüs hädaabi kommunikatsiooniks kasutatavale traadita silmusvõrgule

Diplomitöö

Juhendaja: Toomas Lepik
Magistrikraad

Tallinn 2020

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Indrek Taal

08.01.2021

Abstract

A survey on emergency communication systems during a disaster suggests the need for an “Always-On-Network” to ensure a reliable communication channel between the people in emergency zones and the outside world. Existing solutions for providing communication during such emergency’s present certain disadvantages. This thesis assesses and analyses possible technological solutions to implement secure mobile mesh network using commercially off the shelf hardware with little as possible modification, setup and implement it secure manner.

Latest research papers and analyses were the basis of selection of technologies and their implementation standards to follow. Proposed solution was evaluated utilizing the Open Web Application Security Project (OWASP) Top 10 IoT security concerns as a reference benchmark.

We assessed best technology for our use case is LoRa, removing the complexity, high operating cost and need for licensing. Selected hardware to provides flexibility of networks and redundancy of connection with multinetwork connectivity. Security risk analysis results reviled Thread as MAC protocol for most secure implementation. Security hardening methods were proposed for default configurated setup on basis of OWASP IoT Top 10 Security concerns.

The result is secure and flexible mesh network for emergency scenarios. By solving proposed problem security by design we uncovered using technology specific not mature protocol (LoRaWAN) can offer wider attack area than porting mature protocol over (Thread).

This thesis is written in English and is 68 pages long, including 7 chapters, 14 figures and 3 tables.

Annotatsioon

Turvaanalüüs hädaabi kommunikatsiooniks kasutatavale traadita silmusvõrgule

Uuring teemal hädaabi sidesüsteemidest hädaolukorra ajal viitab vajadusele „alati töös võrgule“, et tagada usaldusväärne suhtluskanal hädaolukorras olevate inimeste ja välismaailma vahel. Olemasolevad lahendused side pakkumiseks sellise hädaolukorra ajal toovad kaasa teatud puudused. Selles lõputöös hinnatakse ja analüüsitakse võimalikke tehnoloogilisi lahendusi turvalise mobiilsidevõrgu juurutamiseks, kasutades kaubanduslikult kättesaadavat riistvara, võimalikult vähe modifitseerides, seadistades ja juurutades seda turvaliselt.

Viimased uurimistööd ja analüüsid olid valiku aluseks tehnoloogiatele ja nende rakendamise standarditele. Pakutud lahendust hinnati, kasutades raamistikuna avatud veebirakenduste turbeprojekti (OWASP) kümmet levinumat värvõrgu turvaprobleemi.

Leidsime, et meie kasutusjuhtumi puhul on LoRa parim tehnoloogia, eemaldades keerukuse, litsentsi vajaduse ning kõrged tegevuskulud. Valitud riistvara pakub paindlikkust ja ühenduse mitmekesisust. Turberiskide analüüsi tulemusena valiti Thread MAC-protokoll rakendamiseks. OWASP IoT Top 10 turbeprobleemide põhjal pakuti välja turvalisuse karastamise meetodid.

Tulemuseks on turvaline ja paindlik silmusvõrk hädaolukordade jaoks. Turvalisust silmaspidades lahendamisel tuvastasime, et tehnoloogiaspetsiifiline mitte küps protokoll LoRaWAN, pakub laiemat rünnakuala kui üle teisaldatud küpse protokoll Thread.

Lõputöö on kirjutatud Inglise keeles ning sisaldab teksti 68 leheküljel, 7 peatükki, 14 joonist, 3 tabelit.

List of abbreviations and terms

COW	CELL ON WHEELS
WMN	WIRELESS MESH NETWORK
OWASP	OPEN WEB APPLICATION SECURITY PROJECT
LORA	LONG RANGE IS A LOW- POWER WIDE-AREA NETWORK
MANET	MOBILE AD HOC NETWORK
WI-FI	WIRELESS FIDELITY
PRNET	PACKET RADIO NETWORK
LPWAN	LOW-POWER WIDE-AREA NETWORK
ISM	INDUSTRIAL, SCIENTIFIC AND MEDICAL
IOT	INTERNET OF THINGS
IEEE	INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS
WPAN	WIRELESS PERSONAL AREA NETWORK
WLAN	WIRELESS LOCAL AREA NETWORK
WMAN	WIRELESS METROPOLITAN AREA NETWORKING
MBWA	MOBILE BROADBAND WIRELESS ACCESS
PHY	PHYSICAL LAYER PROTOCOL
MAC	MEDIUM ACCESS CONTROL
ESS	EXTENDED SERVICES SET

GHZ	GIGAHERTZ (THOUSANDS OF MHZ)
LR-WPANS	LOW-RATE WIRELESS PERSONAL AREA NETWORK
CSS	CYCLIC SHIFTED SEQUENCES
FSK	FREQUENCY SHIFT KEYING
3GPP	3RD GENERATION PARTNERSHIP PROJECT
TGAH	IEEE TASK GROUP AH
MBPS	MEGABITS PER SECOND
MW	MILLIWATT
BLE	BLUETOOTH LOW ENERGY
GFSK	GAUSSIAN FREQUENCY-SHIFT KEYING
CPFSK	CONTINUOUS PHASE FREQUENCY SHIFT KEYING
DQPSK	DIFFERENTIALLY ENCODED QUADRATURE PHASE-SHIFT KEYING
BPSK	BINARY PHASE-SHIFT KEYING
PAM	PULSE AMPLITUDE MODULATION
OOK	ON-OFF KEYING
PWM BPSK	PULSE WIDTH MODULATION BINARY PHASE-SHIFT KEYING
QPSK	QUADRATURE PHASE-SHIFT KEYING
O-QPSK	OFFSET QUADRATURE PHASE-SHIFT KEYING
OFDM	ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING

M-QAM	M-ARY QUADRATURE AMPLITUDE MODULATION
QAM	QUADRATURE AMPLITUDE MODULATION
OFDM CSS	ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING CYCLIC SHIFTED SEQUENCES
FHSS	FREQUENCY-HOPPING SPREAD SPECTRUM
DS-UWB	DIRECT SEQUENCE ULTRAWIDEBAND
MB	MEGABYTE
DSSS	DIRECT-SEQUENCE SPREAD SPECTRUM
CCK	COMPLEMENTARY CODE KEYING
OFDMA	ORTHOGONAL FREQUENCY-DIVISION MULTIPLE ACCESS
AES	ADVANCED ENCRYPTION STANDARD
CTR	COUNTER
WEP	WIRED EQUIVALENT PRIVACY
EAP	EXTENSIBLE AUTHENTICATION PROTOCOL
EAP-TLS	EXTENSIBLE AUTHENTICATION PROTOCOL TRANSPORT LAYER SECURITY
X.509	STANDARD DEFINING THE FORMAT OF PUBLIC KEY CERTIFICATES
MD5	MESSAGE DIGEST 5
HMAC	HASH-BASED MESSAGE AUTHENTICATION CODE
GPIO	GENERAL-PURPOSE INPUT/OUTPUT

PSRAM	PSEUDOSTATIC (RANDOM-ACCESS) MEMORY
GPS	GLOBAL POSITIONING SYSTEM
SDK	SOFTWARE DEVELOPMENT KIT
TCP/IP	TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL
ALOHA	ADDITIVE LINKS ON-LINE HAWAII AREA
MTU	MAXIMUM TRANSMISSION UNIT
TTL	TIME TO LIVE
IP	INTERNET PROTOCOL
IPV6	INTERNET PROTOCOL VERSION 6
6LOWPAN	IPV6 OVER LOW -POWER WIRELESS PERSONAL AREA NETWORKS
UDP	USER DATAGRAM PROTOCOL
DTLS	DATAGRAM TRANSPORT LAYER SECURITY
MLE	MESH LINK ESTABLISHMENT
SSID	SERVICE SET IDENTIFIER
OSI	OPEN SYSTEMS INTERCONNECTION
CSMA	CARRIER-SENSE MULTIPLE ACCESS
MAC	MESSAGE AUTHENTICATION CODE
COAP	CONSTRAINED APPLICATION PROTOCOL
HTTP	HYPertext TRANSFER PROTOCOL
RF	RADIO FREQUENCY
DOS	DENIAL OF SERVICE

CSMA/CA	CARRIER-SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE
ACK	ACKNOWLEDGEMENT
OTA	OVER-THE-AIR
FTP	FILE TRANSFER PROTOCOL
USB	UNIVERSAL SERIAL BUS
SD CARD	SECURE DIGITAL NON- VOLATILE MEMORY CARD FORMAT
HTTPS	HYPertext TRAnSFER PROTOCOL SECURE
CVE	COMMON VULNERABILITIES AND EXPOSURES
EMTC	ENHANCED MACHINE TYPE COMMUNICATION
LTE	LONG-TERM EVOLUTION

Table of contents

1 Introduction	15
1.1 Objective.....	15
1.2 Methodology.....	16
1.3 Thesis structure.....	16
2 Technologies.....	17
2.1 Mesh networking	18
2.1.1 Background.....	19
2.2 Mesh technologies	20
2.2.1 Technical Overview.....	21
2.2.2 LoRa	21
2.2.3 IEEE 802.15.4	22
2.2.4 IEEE 802.11	23
2.2.5 IEEE 802.15.1	23
3 Comparisons of technologies	24
3.1 Comparative performance analysis	25
3.2 Technology selection.....	26
3.2.1 Evaluation.....	27
3.3 LoRa hardware	29
3.3.1 Hardware selection	29
4 MAC Layer Protocols.....	30
4.1 LoRaWan.....	31
4.1.1 Limitations in LoRaWAN	31
4.2 Meshtastic.....	33
4.3 Thread.....	34
4.3.1 Overview of the Thread Protocol	34
4.3.2 Thread stack.....	35
4.4 Security considerations of protocols.....	36
4.4.1 Security Risk Analysis of LoRaWAN v1.1.....	36
4.4.2 Security Risk Analysis of Thread.....	37

4.4.3 Selection of protocol.....	38
5 Test Setup and Security Framework Selection.....	39
5.1 Setup architecture	40
5.2 Security Frameworks	41
5.2.1 Comparison of IoT Security Frameworks	41
5.2.2 Conclusion.....	44
6 Assessment of OWASP Top 10 Security Concerns	45
6.1 Weak, guessable, or hardcoded passwords.....	45
6.2 Insecure Network Services	48
6.3 Insecure ecosystem interfaces	49
6.4 Lack of secure update mechanism.....	49
6.5 Use of insecure or outdated components.....	51
6.6 Insufficient privacy protection.....	52
6.7 Insecure data transfer and storage	53
6.8 Lack of device management.....	55
6.9 Insecure default settings	56
6.10 Lack of physical hardening.....	56
7 Summary.....	58
References	59
Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis	65
Appendix 2 Proposed solutions	65

List of figures

Figure 1 Star topology	17
Figure 2 Mesh topology.....	18
Figure 3 Comparison between LAN, Cellular and LPWAN technologies.....	20
Figure 4 The IEEE 802.15.4/ZigBee protocol stack architecture.....	22
Figure 5 Comparison among the analysed communication protocols, based on data rate and transmission range. Minimum and maximum values (with average) are indicated for each performance metric.	24
Figure 6 Comparative performance analysis, with 9 relevant metrics, of 4 wireless mesh technologies.	25
Figure 7 LoRaWAN OSI layers	31
Figure 8 Basic Thread Network Topology and Devices	34
Figure 9 Thread OSI layers and subsequent standards implemented.....	35
Figure 10 PyMesh OSI layers.....	38
Figure 11 Test setup with named components	39
Figure 12 Implemented simplified architecture	40
Figure 13 Future expanded architecture	40
Figure 14 Pycom FiPy development board hardware architecture with named components.....	51

List of tables

Table 1 Technical parameters of technologies	26
Table 2 Performance metrics used to evaluate the analysed WMN protocols	28
Table 3 LoRa commercially off the shelf development hardware	29

1 Introduction

Number of emergency and disaster scenarios worldwide is been increasing creating the need for more reliable emergency networks [1]. Most emergency scenarios include damaged power systems and communication infrastructure thwarting the efforts of first responders and other people in affected area.

A survey on emergency communication systems during a disaster suggests the need for an “Always-On-Network” to ensure a reliable communication channel between the people in emergency zones and the outside world [2]. Existing solutions for providing communication during such emergencies present certain disadvantages [3] [4]. According to Singhet al. [5], the disadvantages of using these technologies include traffic congestion, high operating costs, licensing, and complexity of usage. Hence, novel, and updated approaches to emergency Internet service are needed to sidestep these complexities and provide a highly available, dynamically deployable, and centrally managed network [2].

Wireless technology has seen playing a crucial role in search and rescue operations. The introduction of mesh networks has created a new class of self-configuring networks [6]. The wireless mesh network (WMN) can be of enormous help to first responders in rescue operations as nodes can join and leave the network at any time. Even though the research and development of WMNs have seen considerable improvement, a significant market has not yet developed for the use of WMNs in forming emergency networks.

1.1 Objective

This thesis will assess and analyse possible technological solutions to implement secure mobile mesh network using commercially off the shelf hardware with little as possible modification, setup and implement it secure manner.

1.2 Methodology

Latest research papers and analyses will be the basis of selection of technologies and their implementation standards to follow.

Final proposed solution will be evaluated utilizing the Open Web Application Security Project (OWASP) Top 10 IoT security concerns as a reference benchmark.

The results of this analysis will serve developers and security professionals in better understanding what risks selected solution addresses and what challenges remain. It will help future solution to better analyse how devices are implementing the protocol at the data link, network, and transport levels.

1.3 Thesis structure

The thesis is divided into six chapters.

First chapter states the problem, objective, methodology.

The second chapter introduces the wireless network and a background of the mesh technology. Further review of wireless mesh technologies by describing the most essential characteristics and features of LoRa, ZigBee, WiFi and Bluetooth. The network architecture, operation, and design of these technologies is also described

The third chapter comparison of technologies in different aspects including cost, energy efficiency, frequency bands, capacity, mobility, regulations, and coverage. Based on this theoretical comparison, advantages and disadvantages of all technologies are clearly explained.

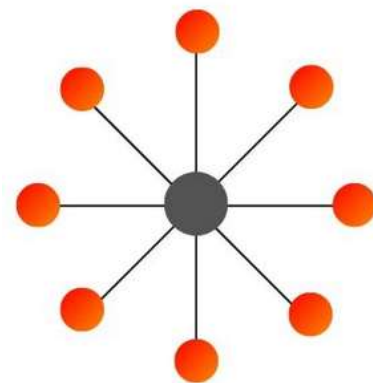
The fourth chapter gives a comparison and review of security issues of reviewed MAC layer protocols to be chosen to be implemented.

The fifth chapter describes the developed setup, components, and future development. Comparison of IoT specific security frameworks is given and framework is selected.

The sixth chapter assesses proposed solution against OWASP IoT Top 10 security concerns.

2 Technologies

Star (centralized) network is the simplest topology with a dedicated link between two nodes. This network performs better (faster), the sent signal reaches only the intended node, failure of one node does not affect other nodes (high availability), it has centralized management, and it is easy to troubleshoot and maintain. However, it is expensive and depends on centralized management failure, which affects the entire network [7]. Figure 1 [8].



Star Topology

Figure 1 Star topology

2.1 Mesh networking

Mesh (decentralized) networking is a type of network topology in which a node (device) transmits its own data as well as serves as a relay for other nodes. In other words, all nodes cooperate in the distribution of data in the network. The word 'mobile' adds mobility for nodes. There are different attempts to classify mobile mesh networks [9]. We will follow to the below-described classifications. Figure 2 [8].

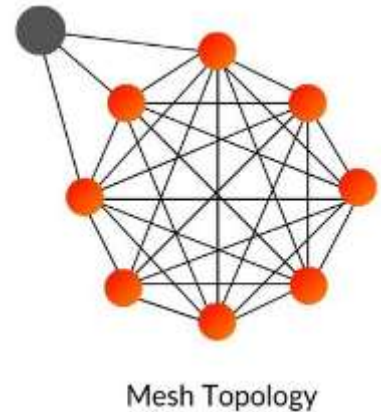


Figure 2 Mesh topology

Main classes are:

- Mobile Ad-hoc NETWORKS (MANET)
- Wireless Mesh Networks (WMN)

A mobile ad hoc network (MANET) is an infrastructure-less, multi-hop, continuously self-configuring network of mobile devices. In computer networking, a term 'ad hoc network' refers, in general, to a network connection established for a single session. The wireless standards (Bluetooth, Wi-Fi, etc.) allow direct communications among network devices within the transmission range of their wireless interfaces.

In general, MANET applications belong to the military areas. The typical applications include:

- Military battlefield: ad-hoc networking would allow the military to take advantage of commonplace network technology to maintain an information network between the soldiers, vehicles, and military information headquarters.
- Emergency: ad-hoc can be used in emergency/rescue operations for disaster relief efforts, e.g. in fire, flood, earthquake, large scale power outage.

Ad-hoc networks can create the infrastructure for emergency communications where the default communication infrastructure does not exist or damaged.

- Personal: ad-hoc networks could form a short-range network for closed communities, conferences, group meetings, etc.

2.1.1 Background

2.1.1.1 The History of Mesh Networking

A wireless mesh network (WMN) is a communications network made up of radio nodes organized in a mesh topology instead of star topology used in most of the networks, according to Akyildiz, X. Wang in the book of Wireless Mesh Networks [10]. It is not a new concept at all, as it had emerged from the Multiple Ad Hoc Networks in the 70s from Packet Radio NETWORK (PRNET) created by The Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense. [11]. Later in the 90s, many other civil solutions had also been proposed and created for different uses such as mesh routers form a mesh of self-configuring, self-healing links among themselves. With gateway functionality, mesh routers can be connected to the Internet. Infrastructure/Backbone WMNs are the most commonly used WMN as its simple and easy to integrate with the existing devices as only the routers need to be fitted to the mesh networks.

2.1.1.2 LPWAN

Low powered wide area network is wireless based WAN technology that enables Low power consumption, long range, lower bandwidth with low bit rates. Unlike 3G/4G/5G or WiFi, these systems do not focus on enabling high data rates per device or on minimizing latency [12]. Figure 3 [13].

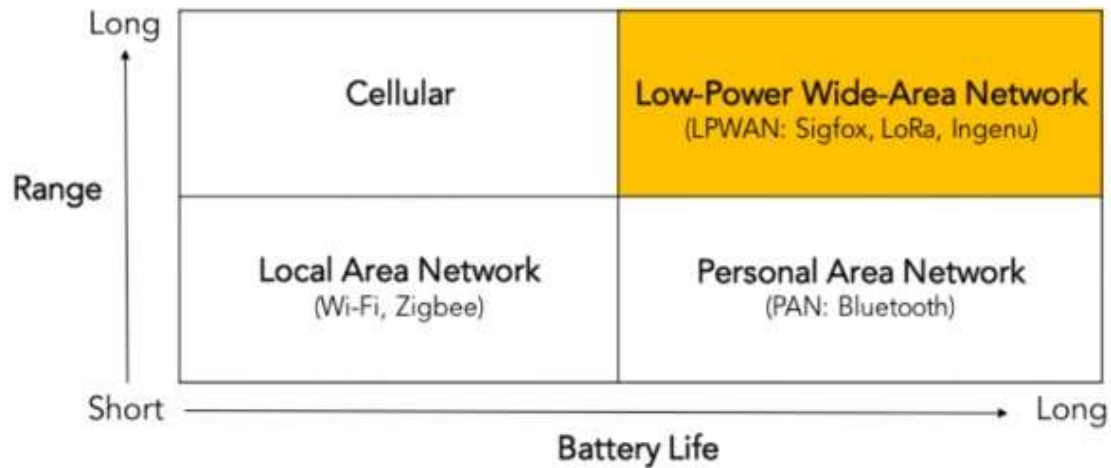


Figure 3 Comparison between LAN, Cellular and LPWAN technologies.

Sub-GHz unlicensed ISM bands (e.g., 868MHz in Europe, and 915MHz in the U.S.) are used to operate LPWAN. The communication range for LPWAN reaches up to 30km in rural areas [14], and up to 5km in urban areas [15]. This long range of LPWAN is possible with a new physical layer design that allows for significantly high receiver sensitivities, e.g., -130dBm. To support the long-range communication of LPWAN, its data rate is necessarily low as a few hundred to thousand bits/sec. Therefore, LPWAN is better suited for low-power IoT devices that transmit a small amount of data over a long distance, in contrast to short-range technologies such as Bluetooth and Zigbee [15].

2.2 Mesh technologies

Dedicated IEEE Task Groups (TGs) have been established defining the requirements for mesh networking in Wireless Personal Area Networks (WPAN), WLANs, Wireless Metropolitan Area Networks (WMANs) and Mobile Broadband Wireless Access (MBWA) [16]. The IEEE 802.15.5 TG was formed to determine the necessary mechanisms enabling mesh networking in WPANs PHY and MAC layers.

Facing the throughput degradation and unfairness in IEEE 802.11 multi-hop networks, the IEEE 802.11s TG addresses the needs for wireless mesh in WLANs and aims to extend 802.11 architectures and protocols to provide ESS (Extended Service Set) mesh functionalities. The implementation of this specification shall be directly reflected over the existing PHY layer of IEEE 802.11a/b/g/n operating in the unlicensed spectrum of 2.4 and 5 GHz [6].

Furthermore, the ZigBee Alliance has been working on the specifications of Low Rate WPANs (LR-WPANs) based on 802.15.4. The IETF Control and Provisioning of Wireless Access Points (CAPWAP) WG emerged with the objective to address architectures and operations of managing large scale WLANs deployments. Mesh networking is one of the architecture examples defined by this WG and is classified as distributed WLAN architectures.

2.2.1 Technical Overview

2.2.2 LoRa

The name LoRaWAN Stands for Long Range Wide Area Network first release came in 2015 by LoRa Alliance as a wireless standard. LoRa and LoRaWAN are not interchangeable and there is the difference between them. LoRa describes the modulation in physical layer and LoRaWAN is MAC protocol which supports low power, long range and high capacity in LPWA network. Generally, system architecture and communication standard determine the technical performance of the technology, like energy efficiency to save battery charge, network capacity and data rates for various applications [17].

2.2.2.1 Technology specifications

LoRa use chirp spread spectrum (CSS) modulation technique. LoRa modulation scheme has key features such as strong robustness against interference and losses compare to other modulation schemes in wireless systems for example frequency shift keying (FSK) [18]. A research conducted in North Jutland, Denmark where different kinds of technologies were tested in different areas. This research is based on simulation results, where they used Telenor's actual base station locations [18]. Base stations were equipped with omnidirectional antennas with maximum transmitted power which in case of LoRa is 14 dBm. Environment conditions are highly dependent on modelling the channel which impact signal propagation. The used propagation model was 3GPP macro non-line-of-sight model. From results, it shows that LoRa provides outdoor coverage better than 99%. In indoor environment, LoRa covered more than 95% users with 20 dB penetration loss [19].

2.2.3 IEEE 802.15.4

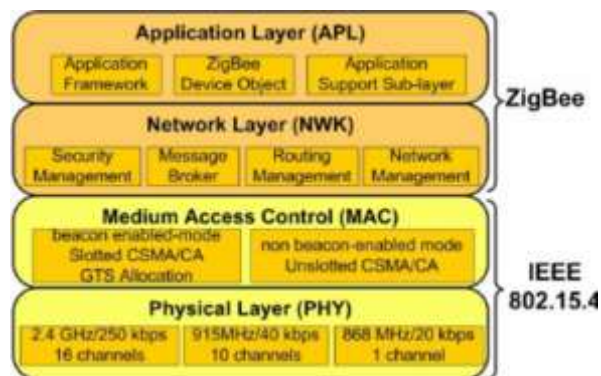


Figure 4 The IEEE 802.15.4/ZigBee protocol stack architecture

IEEE 802.15.4 [20] is a standard specifying the physical layer and data link layer for Low-Rate Wireless Personal Area Networks (LR-WPANs). Supporting three un-licensed frequency bands (868 MHz, Europe; 928 MHz, North America; 2.4 GHz, worldwide), IEEE 802.15.4 can offer data rates up to 250 kbit/s at a transmission range largely

dependent on the environment; while for a clear line-of-sight, up to 1000 m is possible; alas in most cases, the transmission range is measured in tenths of meters. Built on top of the IEEE 802.15.4 physical and data link layers, ZigBee [21] offers application-facing communications profiles and a network layer. Figure 4 [22].

2.2.4 IEEE 802.11

IEEE [23], [24] provides a wireless LAN standard that operates at sub-1-GHz license-exempt bands. The work is conducted by the IEEE 802.11 ah Task Group (TGah). Compared to IEEE 802.11 (operating at 2.4 GHz and 5 GHz), 802.11 ah supports a longer transmission range up to 1 km at the default transmission power of 200 mW. Depending on the bandwidth assigned, 802.11 ah can operate at 4 Mbps or 7.8 Mbps. If the channel condition is good enough, 802.11 ah can provide a hundreds of Mbps data rate, thanks to the novel modulation and coding schemes brought from 802.11 ac.

2.2.5 IEEE 802.15.1

Released in 1999 by a consortium led by Ericsson, Nokia and Intel, Bluetooth v1.0 was initially designed to, wirelessly, replace cables to connect devices typically used together, such as cell phones, laptops, headsets, keyboards, etc., offering a lower data rate (1-Mbps raw data rate, max) and a relatively short range (in theory, officially up to 100 m, at maximum transmission power, realistically, 5–10 m) while also a low power consumption. Several revisions of Bluetooth later, Bluetooth 4.0 was completed in 2010. Fully compatible with Bluetooth 1.0, this revision supports a higher data rate (24-Mbps raw data rate, based on WiFi) and includes a “low energy” extension (called Bluetooth/LE or “Smart”). As compared with the “non-LE version”, Bluetooth/LE provides rapid link establishment functions (simpler pairing) and further trades off the data rate

(approximately 200 kbps) for lower energy consumption, with the target to run a wireless sensor for at least one year on a single coin cell (approximately 200mAh) [25].

3 Comparisons of technologies

Comparing different mesh capable technologies allows to review performance under different conditions. For that a research papers results are dissected and presented to make the selection for our use case.

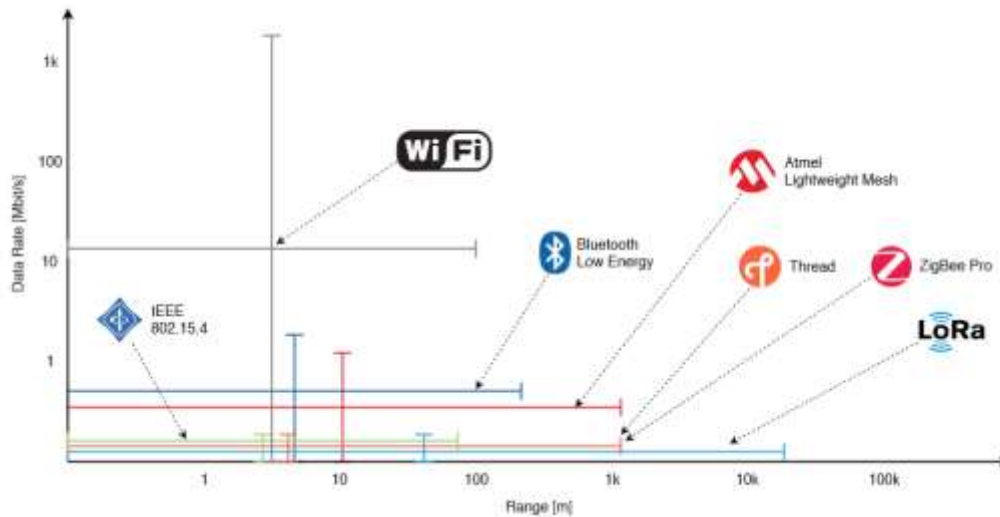


Figure 5 Comparison among the analysed communication protocols, based on data rate and transmission range. Minimum and maximum values (with average) are indicated for each performance metric.

Figure 5 [26].

3.1 Comparative performance analysis

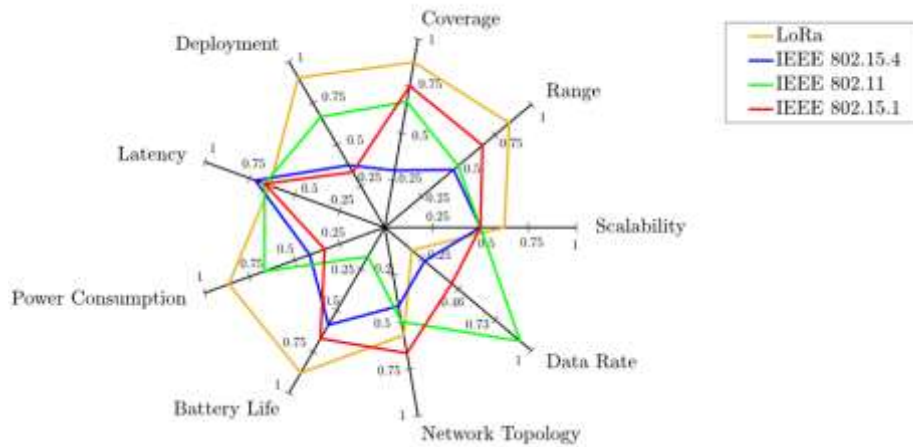


Figure 6 Comparative performance analysis, with 9 relevant metrics, of 4 wireless mesh technologies.

In Figure 6 [27], is presented a comparative overview of the considered wireless technologies. More in detail, the following relevant metrics were considered.

1. Coverage (dimension: [m²]): intended as the area which can be covered using a mesh network based on the analysed technology.
2. Range (dimension: [m]): intended as the transmission range of a single node in the mesh network.
3. Scalability: intended as the capability of a mesh network, based on used wireless technology, to scale.
4. Data Rate (dimension: [b/s]): measured on single nodes.
5. Network topology: intended as the degree of complexity reachable building different network topologies.
6. Battery Life (dimension: [days]): measured on single nodes.
7. Power Consumption (dimension: [W]): measured on single nodes.
8. Latency (dimension: [s]): intended as the capability of a technology to obtain low latencies among nodes communications.
9. Deployment: intended as the complexity to deploy a mesh network based on the specific wireless technology. [28]

The values shown in Figure 6 have been obtained analysing relevant works appeared in the literature namely [29]–[37] and normalizing each data in order to directly compare (through a dimensional values) the communication technologies presented in the paper [28].

3.2 Technology selection

For better understanding of analysed values an overview of technical parameters are compiled and presented table below.

Table 1 Technical parameters of technologies

Protocols	Bluetooth [38], [39], [40]	ZigBee/IP [38],[39], [40]–[45]	Wi-Fi [46], [38], [39]	Wi-Max [40], [47], [48]	LoRa [49]
Frequency band	2.4 GHz	868/915 MHz; 2.4 GHz	2.4; 5 GHz	5.1- 66 GHz 2.4;	433;868;915 MHz
Max signal rate	720 Kb/s	250 Kb/s	54 Mb/s	35-70 Mb/s	50Kb/s
Nominal range	10 m	10 - 1000 m	10-100 m	0.3-49 Km	<10km
Nominal TX power	0 - 10 dBm	-25 - 0 dBm	15 - 20 dBm	23 dBm	+14dbm
Number of RF channels	79	1/10; 16	14 (2.4 GHz) 64 (5 GHz)	10;4;8;20	10
Channel bandwidth	1 MHz	0.3/0.6 MHz; 2 MHz	25-20 MHz	20;10 MHz	125/250kHz
Modulation type	GFSK, CPFSK, 8-DQPSK	BPSK, QPSK, O- QPSK	BPSK, QPSK, OFDM, M-QAM	QAM16/64, QPSK, BPSK, OFDM	CSS
Spreading	FHSS	DSSS	MC-DOSSS FDM, CCK,	OFDM, OFDMA	OFDM, OFDMA
Basic cell	Piconet	Star	BSS	Single-cell	Star-on-Star
Extension of the basic cell	Scatter net	Cluster tree, Mesh	ESS	PTMMP,ePs ThCM,	Cluster tree, Mesh
Max number of cell nodes	8	> 65000	2007	1600	~300

Encryption	E0 stream cipher	AES block cipher (CTR)	RC4 stream cipher (WEP), AES block cipher	AES-CCM cipher	AES block cipher (CTR)
Authentication	Shared secret	CBC-MAC (ext. of CCM)	WPA2 (802.11i)	EAP-SIM, EAP AKA, EAP-TLS or X.509	AES-CMAC
Data protection	16-bit CRC	16-bit CRC	32-bit CRC	AES based CMAC, MD5-based HMAC, 32-bit CRC	AES based
Success metrics	Cost, convenience	Reliability, power, cost	Speed, Flexibility	Throughput, Speed, Range	Range, power, cost
Application focus	Cable replacement	Monitoring, control	Data network, Internet, Monitoring,	Internet, Monitoring, Network Service,	Monitoring

3.2.1 Evaluation

In the following, was present the findings of previous research and data in Figure 6. These values, as previously highlighted, have been obtained through a literature analysis on several research works [29]–[37]). In each of these papers, at least one technology and one performance metric were considered.

In Table 1, for each considered metric indicate the reference where relevant information is reported. Then, assigned a value representative of the performance for the corresponding metric. For each technology, we consider the arithmetic average among the collected performance values and, finally, we divide this average by the maximum among all values. In this way, the final result is normalized between 0 and 1 [28].

Table 2 Performance metrics used to evaluate the analysed WMN protocols

	LoRa				IEEE 802.15.4				IEEE 802.11	IEEE 802.15.1
	[29]	[30]	[31]	[32]	[33]	[32]	[34]	[35]	[36]	[37]
Scalability	4	3.5	—	—	—	3	3	3	3	3
Range	5.5	5	4.5	8	3.5	—	—	3	3	4
Coverage	5.5	5	—	7	1	3	3	4	4	4.5
Deployment	5.5	5.5	—	7	2	—	—	4	4	2
Latency	4	4	5	4	5.5	—	—	4	4	4
Power Consumption	—	—	6	7	4	—	—	4	4	2
Battery Life	5	5.5	—	—	—	3.5	—	1	1	4
Network Topology	—	—	4	—	—	2.5	—	3	3	4
Data Rate	—	—	—	1	1.5	—	—	5.5	5.5	2.5
Max Scale ($V_{maxi,k}$)	6	6	7	8	8	6	6	6	6	6

For example, in terms of scalability, from the values in Table 1 one can conclude that: for LoRa the value will be $((4/6)+(3.5/6))/2=0.625$; for IEEE 802.15.4 the value will be $((3/6)+(3/6)+(3/6))/3=0.5$; for IEEE 802.11 the value will be $(3/6)=0.5$; for IEEE 802.15.1 the value will be $(3/6)=0.5$.

For our use case LoRa performs the best, complying with set requirements such as deployability, low power usage and long range.

3.3 LoRa hardware

In selection of hardware for this implementation wide availability, documentation, active development support from manufacturer, multiple networks for redundancy were criteria.

3.3.1 Hardware selection

Product G was selected as hardware solution for this use case as it has best documentation, supporting manufacturer and support of alternative networks for redundancy of connectivity.

Table 3 LoRa commercially off the shelf development hardware

Product	A	B	C	D	E	F	G
Features	WIFI	Arduino support	ARM Mbed	GPS with Easy Mode	Arduino support	Arduino support	WiFi, Sigfox*
	Bluetooth	Serial AT		NB-IoT			BLE
	8MB PSRAM	16kb	64KB	4kb	32kB		4MB
	4MB Flash	128kb	256kB	32kb	256kB		8MB
	3D Antenna	Onboard solar energy management system					LTE-CAT M1/NB1*
GPIO	14	6	26	28	20	20	22
LORA							
Operating frequency:	868/915	863-870	863-870	863-870 433-434	868/915	868/915	868

Transmit power:	+20dBm	22 ± 1dB	18dB ± 2dB	14dB	+20dBm	+14dBm	+14dBm
Sleep current	0.2uA sleep 1.5uA IDLE	3.5uA	4uA	7uA 20mA	~300uA	0.2uA 1.5uA	25uA
GPS	+	-	-	+	-	-	+
Power							
Power Supply Input	USB 5V/1A	PIN 3.3-5V/150mA-500mA	PIN 2.7-6V/150mA-500mA		PIN 3.3/150mA	1.8-3.7V	3.3V/400mA
Battery Input	3.7-4.2V	2.7-6V	1.8-6V	2.4-5.5V	3.3V	3.3V	3.3V - 5.5V
Documentation	5/10	7/10	4/10	4/10	6/10	3/10	9/10
SDK	-	-	-	-	-	-	+
Security features	-	-	-	-	-	-	+
Price	22\$	12\$	52\$	132€	30€	28€	30-60*€

4 MAC Layer Protocols

Medium Access Control is crucial in wireless communications, which defines the way how wireless nodes contend and share the scarce radio resources. Generally, it is impossible for a wireless node to transmit and receive at the same time over the same bandwidth, and hence collisions hard to detect during transmission. Simultaneous transmissions of hidden terminals can cause a collision at the common receiver. In addition, the exposed terminal problem can reduce the system utilization. Therefore, a primary goal of MAC protocols for WMNs is to avoid collisions and allow simultaneous transmissions whenever possible. LPWAN specific MAC protocols which are also compatible with selected hardware, are reviewed and analysed by literature review.

4.1 LoRaWan

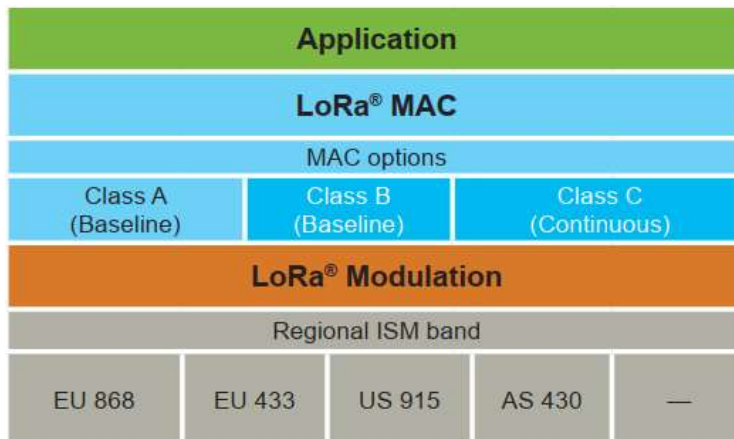


Figure 7 LoRaWAN OSI layers

LoRaWAN is a well-known LPWAN media access control protocol that is designed for networks using LoRa or frequency-shift keying communication specifically. LoRaWAN networks use star topologies, where each node device only connects to one or many gateways through a single-hop communication [15] Figure 7 [50]. A gateway is defined as a sink point in a LoRaWAN network where messages are forwarded from the LoRaWAN network to a central server through a TCP/IP connection. Roughly, up to 10,000 end devices can connect to a single gateway simultaneously [51]. Even so, due to the limited number of available channels, the overall throughput will decrease as the number of connected node devices increases [52].

4.1.1 Limitations in LoRaWAN

As LoRa and LoRaWAN are still in an early stage of development, their application scenarios have not been explored very much. Therefore, the LoRaWAN protocol also has the following shortcomings:

- Limited MTU and No Transport Protocol.

According to the Semtech data sheet [53], the maximum payload size of a LoRa packet is between 51 bytes and 255 bytes, depending on the SF setting. Such payload size is only sufficient for small sensor data values, such as temperature and humidity, motion detection, and ambient light level, etc. For large data units, many packets have to be used for transmission. However, there is currently no transport protocol designed for LoRa wireless networks that efficiently supports transmission of messages requiring many packets.

- Random Channel Access.

In LoRaWAN, pure ALOHA is utilized for media access control [52]. Pure ALOHA works very simply: it transmits whenever there is data to send, without sensing whether another transmission occurs on the same channel and time [54]. When a collision happens, the packets will not be received correctly, and retransmissions will be required. Typically, as the number of end devices grows, the maximum throughput decreases as the chance of packet collisions increases. Thus, the random nature of pure ALOHA is not efficient and optimal in IoT systems [52]. In random channel access, each node device can send a data packet without sensing the status of the channel. In other words, there is no busy period detection before transmitting, and this can lead to a high probability of message collision during the busy period. When a collision occurs, a packet that is being transmitted must be discarded and needs to be retransmitted again later. As a result, electricity is wasted. Also, the throughput cannot be guaranteed, and latency is unbounded when random channel access is used, as node devices may need a very long time to send data successfully when the network is busy.

- Single Hop Communication.

Actual maximum transmission distance is usually less than the theoretical distance. This is often due to the uneven terrain and obstacles, which weaken the signal. In such cases, single hop communication is no longer sufficient, no matter what the theoretical signal range is, and multi-hop communication is needed for bypassing the obstacles. Multi-hop

communication is also helpful for extending the network to an even broader range. In other wireless communication protocols, such as Bluetooth, ZigBee, and Wi-Fi, multi-hop communication is feasible and helpful in many aspects. In contrast, LoRa lacks support for it.

4.2 Meshtastic

As an alternative to industry specifications, author explored possibility to implement open-source protocol on top of LoRa PHY.

As described on website, “Meshtastic™ is a project that lets you use inexpensive GPS mesh radios as an extensible, super long battery life mesh GPS communicator. These radios are great for hiking, skiing, paragliding - essentially any hobby where you don't have reliable internet access. Each member of your private mesh can always see the location and distance of all other members and any text messages sent to your group chat.

The radios automatically create a mesh to forward packets as needed, so everyone in the group can receive messages from even the furthest member. The radios will optionally work with your phone, but no phone is required.” [55]

Meshtastic acts as a broadcasting LoRa Star-on-Star with broadcasting TTL of 3 (by default). Meshtastic relies on ESP32 chipset and community-built protocol of multi-hop flooding (specified in configuration) communication. Due to being open-source and community development, no specific standards are being implemented. The protocol security mechanisms are still in development and untested. Therefore, usage in critical infrastructure is not considered.

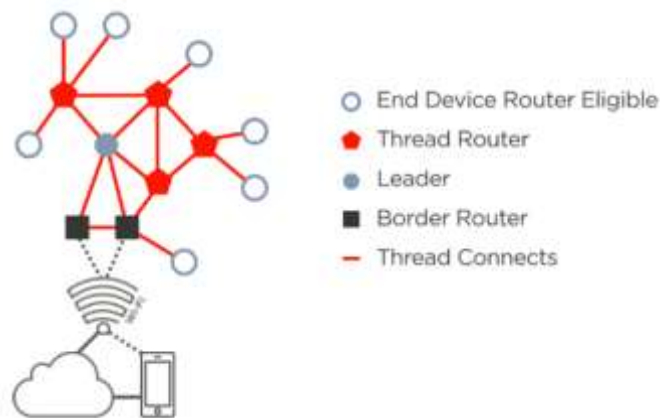


Figure 8 Basic Thread Network Topology and Devices

4.3 Thread

As a possibility author explored mature and robust protocol for LoRa PHY.

4.3.1 Overview of the Thread Protocol

The Thread Group released the latest Thread 1.2 Specification on September, 2019. The specification provides extensive detail on the Thread protocol and claims to provide everything necessary to implement a Thread networking stack. The Thread standard is best referred to as a “network stack” in that it combines existing standards and protocols with specific implementation guidance to define the desired networking architecture. Various protocols were selected to meet the goals of Thread, to include support for IP-based addressing, use of existing hardware technology, scalability, low latency and power requirements, and simplified security. As shown in Figure 9, the Thread networking stack primarily addresses the transport and network layers of the interconnect model, utilizing existing IEEE 802.15.4 radio components at the physical layer. Thread provides flexibility at the application layer, allowing a variety of market applications. According to the Thread technical overview, “Thread defines how data is sent in the network but not how to interpret it”. [56].

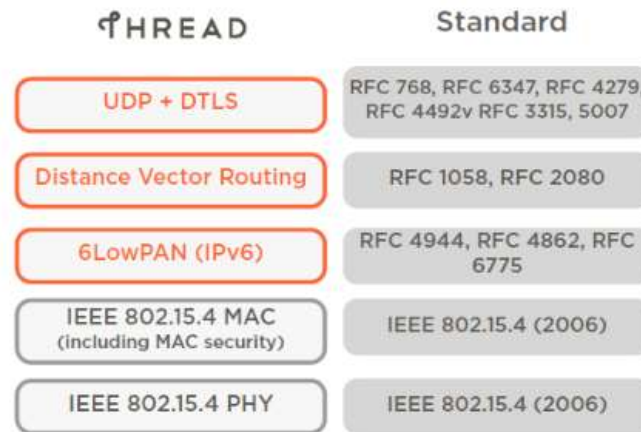


Figure 9 Thread OSI layers and subsequent standards implemented

4.3.2 Thread stack

IEEE 802.15.4, which Thread is built on is a technical standard which defines the operation of low-rate wireless personal area networks (LR-WPAN). The emphasis lies in extremely low manufacturing cost and technological simplicity. As mentioned earlier, Thread operates below the application layer and implements the Transport and Network layer of the OSI model [57]. IEEE 802.15.4 MAC (Medium Access Control) is used for message handling and congestion control. This layer implements CSMA (Carrier-Sense Multiple Access) to verify the absence of network packets on a channel before transmitting to that channel, frame retries and acknowledgement frames to ensure reliable communication. MAC (Message Authentication Code) security functionality is used to provide integrity and confidentiality to messages at higher layers of the software stack. ICMP is supported for error messages and UDP is used for delivering IP packets between devices. DTLS (TLS over UDP) is used for authenticating a joining (untrusted) device. Lastly the Constrained Application Protocol (CoAP) is used as a replacement for HTTP for communication with the application which is built on Thread.

4.4 Security considerations of protocols

New technologies bring new and never seen issues that need fixing and maturing over the time. To get an overview of risks that can occur implementing and securing a protocol for our purpose, a risk analysis has been done using accessible research and literature on protocol security.

4.4.1 Security Risk Analysis of LoRaWAN v1.1

LoRaWAN v1.1 possesses Minor risk at security attacks of:

- Bit-flipping or Message Forgery Attack
- Destroy, Remove, or Steal End-device
- False Join Packets
- Frame Payload Attack
- Network Flooding Attack
- Network Traffic Analysis
- RF Jamming Attack
- Selective Forwarding Attack
- Sinkhole or Blackhole Attack [58], [59]

LoRaWAN v1.1 possesses Major risk at security attacks of:

- Beacon Synchronization DoS Attack
- Impersonation Attack
- Plaintext Key Capture
- Security Parameter Extraction [58], [59]

LoRaWAN v1.1 possesses Critical risk at security attacks of:

- Device Cloning or Firmware Replacement
- Self-Replay Attack
- Rogue End-Device Attack [58], [59]

LoRaWAN v1.1 is more susceptible to physical attacks such as capture, rogue end-device and rogue gateway attacks, rather than attacks towards higher layers of the communications stack, such as network layer attacks (Sinkhole, Blackhole, etc.) [59].

4.4.2 Security Risk Analysis of Thread

Thread protocol is vulnerable to the following attacks.

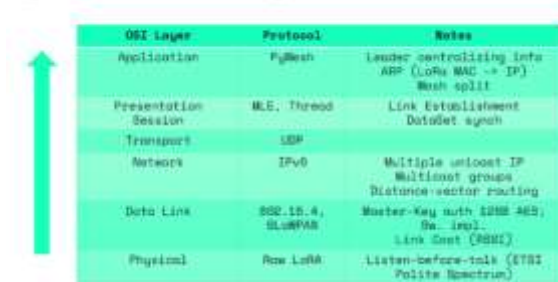
- Radio jamming [60]: Thread is inevitably prone to physical layer radio jamming, which is a form of a DoS attack.
- Link layer jamming and node-specific flooding [61]: Unlike radio jamming, link layer jamming creates a DoS attack by crafting link layer frames in a Thread network, to reduce network performance and throughput; node-specific flooding sends frames towards a particular node, either to drain its battery or to affect its functionality.
- Back-off manipulation and Clear Channel Assessment (CCA) manipulation [60]: An adversary could deviate from the CSMA/CA channel access mechanism used by IEEE 802.15.4, by either using a shorter back-off time or even skipping CCA. Doing so would deteriorate the throughput of the legitimate nodes in the Thread network.
- Acknowledgment (ACK) attack [62]: IEEE 802.15.4 does not mandate integrity or confidentiality protection for acknowledgment frames. Once a message is received by a Thread node, the node responds with an ACK frame that includes the sequence number of the received frame. Since the frame is sent in clear text, an adversary can forge an ACK message if it knows the corresponding sequence number.

While the above attacks can be used to degrade the performance of a Thread network, they do not exploit design flaws in the Thread protocol but are inherent to wireless communication. Furthermore, none of the vulnerabilities enables an attacker to violate message integrity or confidentiality.

4.4.3 Selection of protocol

In conclusion LoRa specific MAC protocol implementation LoRaWAN includes more security related risk than Thread. Thread protocol also brings full capability of mesh with IP networking capability. For these reasons Thread specifically Open Thread was selected as it was already included in selected hardware for implementation.

PyMesh (OSI) Layers



OSI layer	Protocol	Notes
Application	PyMesh	Leader controlling info APP (LoRa MAC -> IP) Mesh split
Presentation Session	M.E. Thread	Link Establishment Dataset sync
Transport	SDP	
Network	IPv6	Multiple unicast IP Multicast groups Distance-sector routing
Data Link	802.15.4, 6LoWPAN	Master-Key with 128bit AES Sec. Inval. Link Cost (RSSI)
Physical	Raw LoRa	Listen-before-talk (LBT) Folite Spectrum

Figure 10 PyMesh OSI layers

Chosen platform includes PyMesh protocol which is implementation of Open Thread on ESP32 platform for LoRa PHY. Figure 10 [63].

5 Test Setup and Security Framework Selection

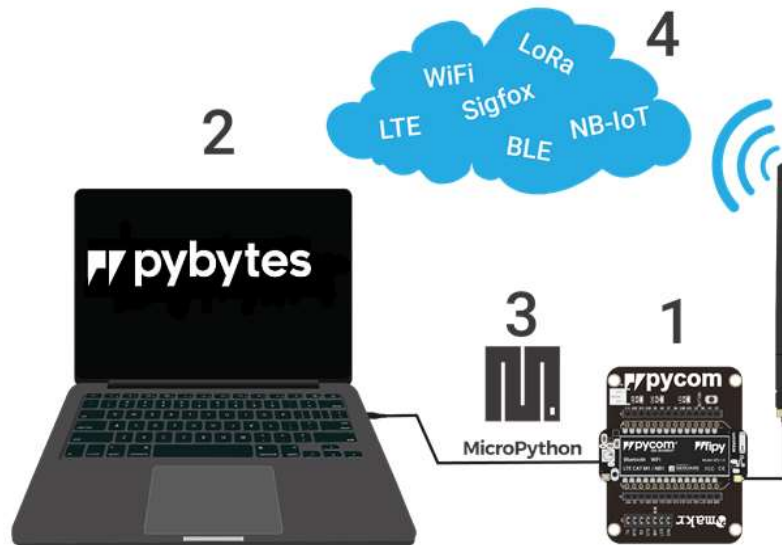


Figure 11 Test setup with named components

Basis of reviewed and analysed literature following implementation is compiled:

1. FiPy: Pycom development board with extension board
2. Pybytes: Cloud services for monitoring, provisioning, and OTA updates
3. PyMesh protocol: Implementation of Open Thread protocol library in MicroPython
4. Radio: LoRa + WiFi (Border router for OTA updates and provisioning) Figure 11 [64].

5.1 Setup architecture

For testing purposes simplistic network scheme is used as availability of development boards at the time of writing was limited. Setup was used as risk assessment environment against chosen framework.

- Cloud: Connection over WiFi to Pybytes
- Border router: Connection to Pybytes, shares Border routing address
- Leader: LoRa mesh network routing tabel holder

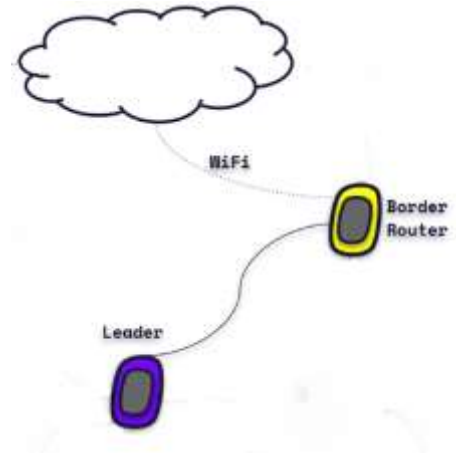


Figure 12 Implemented simplified architecture

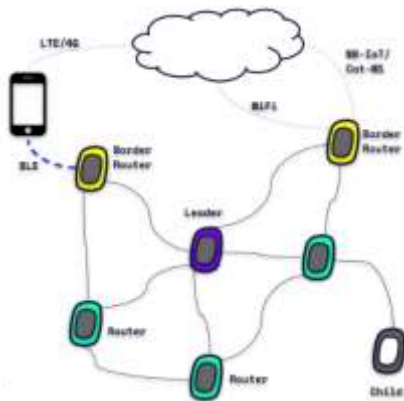


Figure 13 Future expanded architecture

Future development will include BLE capability implemented for application-level communication for end user device. Alternative networking (NB-IoT, SigFox) will be implemented for Border Router for cloud connection. Figure 13 [65].

5.2 Security Frameworks

This goal is to implement secure example network for future development. Considering all the previous risk, additional measures of configuration must be taken and validated to a standard. To have good coverage of most common security concerns brief comparison of IoT security frameworks is provided.

5.2.1 Comparison of IoT Security Frameworks

Several IoT frameworks have been devised that can help vendors in developing secure devices. These frameworks contain security measures to follow during development, helping vendors to create a secure device. This brief overview describes the differences between these security frameworks [66].

The following security frameworks are covered:

- ETSI TS 303 645 V2.1, provisions for the security of consumer devices that are connected to a network [67];
- IoT Security Compliance Framework, from the IoT Security Foundation [68];
- OWASP Internet of Things Security Verification Standard (ISVS) provides security requirements for IoT applications [69];
- ENISA Baseline security recommendations for IoT in the context of Critical Information Infrastructures [70].

5.2.1.1 ETSI EN 303 645

The European Telecommunications Standards Institute (ETSI) specifies 65 security provisions for consumer IoT devices that are connected to a network. The standard is meant for organizations involved in the development and manufacturing of consumer IoT devices, i.e. vendors. As such, it aims to provide a relatively complete set of requirements [67].

The requirements are less useful for testing a finished product; in a black box test it is difficult to observe whether some provisions have been implemented or not. Even so, it is a complete and usable set of provisions, and it supports most provisions, with examples and rationales provided [66].

5.2.1.2 IoT Security Compliance Framework

The IoT Security Foundation released the IoT Security Compliance Framework, which comprises a set of 233 requirements [68].

Requirements are either mandatory or advisory, and are applicable to certain device classes, which depend on the impact of a compromised device. Devices where a hack would cause minor inconvenience is denoted Class 0 and less security measures apply to such devices. Devices that handle sensitive data are denoted Class 3, and for these most security measures apply. As many devices handle sensitive data in some form, the security requirements this framework imposes are pretty strict [66].

The framework has a wide scope, and includes security requirements for mobile applications, cloud services, the supply chain and the production process. This causes several very similar requirements; passwords should be secure for the IoT device, for the mobile application, for the web interface, etc [66].

5.2.1.3 OWASP ISVS

The OWASP Internet of Things Security Verification Standard (ISVS) provides security requirements for Internet of Things (IoT) applications. It is modelled after the Application Security Verification Standard (ASVS), a standard that is growing in popularity for the verification of security controls for web-applications and web services [69].

It consists of a list of 90+ verification requirements that are predominantly targeted at the technical security aspects of an IoT application [69].

In its current form, as part of the ASVS, the ISVS defines three assurance levels with increasing depth. This essentially means that an IoT application is verified against more requirements when a higher security level is selected. Level 1 requirements can be considered as the bare minimum. The requirements at this level are typically easy to verify. Level 2 introduces requirements that defend against the majority of today's security risks. Level 3 is reserved for applications that need a high level of assurance and require significant security verification. Examples of such applications are in the area of military, health, financial or critical infrastructures [66].

5.2.1.4 ENISA Baseline Security Recommendations for IoT

The ENISA (European Union Agency for Cybersecurity) Baseline Security Recommendations for IoT provides measures on three main categories:

- Policies;
- Organizational, People and Process measures;
- Technical measures [70].

The measures regarding policies target the development process at the vendor. The Organizational, People and Process measures target the interaction between the vendor and the consumer, and cover vulnerability disclosure. Finally, the technical measures provide the most concrete measures of how the IoT device should behave.

It is self-evident that for a device to be secure, all its subcomponents need to be secure. However, for vendors that are unaware of how to develop secure components, indicating that something must be secure may be insufficient. For testers, it may even be unclear what level of security is demanded, or against what kind of attack the system should be secure. Most of these measures have been discarded as insufficiently specific [66].

5.2.2 Conclusion

All frameworks offer value and tools to better security, main differences are:

- ETSI 303 645 provides well explained and specific instructions on how to achieve basic security in IoT devices.
- IoT Security Compliance Framework provides additional security, not only in the device but in the business process and the surrounding systems.
- OWASP IoT Security Verification Standard provides a checklist to verify after development whether a product is secure.
- ENISA guidelines provide less formal process but are good recommendations on how to secure your devices.

For hackers and testers, the OWASP ISVS has potential to be the best match. It is specifically meant to provide a checklist of things to verify when testing. As our use case is for emergency network a critical infrastructure, Level 3 verification is needed to achieve.

Due to time and skill limitations of author an alternative was proposed for initial assessment. OWASP a Top 10 IoT Security concerns is used, as the most critical parts of verification will be covered and easily put in check list for future Level 3 testing.

6 Assessment of OWASP Top 10 Security Concerns

The Open Web Application Security Project (OWASP) is most commonly known for its Top 10 list of common web application vulnerabilities. The foundation describes the OWASP IoT effort as being “designed to help manufacturers, developers, and consumers better understand the security issues associated with the Internet of Things, and to enable users in any context to make better security decisions when building, deploying, or assessing IoT technologies” [71]. The OWASP approach is to take a holistic approach to IoT security to include hardware interfaces, software configurations, network communications, and applications. While no single technology can be expected to resolve all the concerns across the various surfaces of an IoT device, the OWASP Top 10 serves as a useful framework to view the technology’s contributions systematically and holistically.

6.1 Weak, guessable, or hardcoded passwords

For IoT, authentication and authorization primarily involve weak or insufficiently protected passwords or credentials, or faulty authentication schemes. Default passwords provide easy access, while lack of mandatory password complexity can result in quick brute-force attacks [72]. Some protocols, such as HTTP and FTP are notorious for passing credentials “in the clear” and can be easily sniffed and captured. These issues are all common in the implementation of IoT because developers often assume that interfaces will only be exposed on internal networks with minimal threat access [71].

The OWASP security concern goes beyond credentials for web interfaces and addresses key management and network service authorizations. With poor key management or authentication, loss of a single node can compromise the entire system or break the confidentiality and integrity of messages from other nodes [62]. Several credentials come

into play in the ordinary operation of Pycom devices, as well as in the joining and credentialing process.

List of default passwords or services enabled:

- FTP & Telnet Server default credentials
 - url: (ftp://)192.168.4.1
 - username: **micro**
 - password: **python**
- class network.LoRa.Mesh(*, key=masterkey)
 - This constructor network.LoRa.Mesh() creates and configures the Pymesh object.
 - By default, the key is **0134C0DE1AB51234C0DE1AB5CA1A110F**.
- The WLAN (WiFi) is enabled by default.
 - By default, **no password** is set for WiFi network created at boot.

All other possible services that FiPy enables need to be defined independently. Most of them require usage of passwords by default and are not hardcoded, but no minimum requirements are implemented to ensure the complexity. [83]

As discussed above, the web interface on commissioning devices, border routers, or the edge devices are not controlled by the Thread standard and may often be lacking appropriate security controls. However, the Thread standard does provide specific guidance on the implementation of transport and media access layer authentication and encryption. The standard claims that “Devices do not join the Thread Network unless authorized and all communications are encrypted and secure” [73].

In order to achieve this, Thread utilizes a network-wide key at the Media Access Layer (MAC) to implement standard IEEE 802.15.4 authentication and encryption. The Thread standard describes the MAC layer encryption key as being “an elementary form of

security used to prevent casual eavesdropping and targeted disruption of the Thread Network from outsiders without knowledge of the network-wide key” [74]. However, the network-wide key is pre-shared and stored in non-volatile memory in the edge device.

Any compromise of a Thread device could reveal the key and allow compromise of the network [74]. Also, distribution of the network-wide key to new devices on an IoT network is problematic. Asking consumers to enter authentication credentials into IoT devices that lack robust user interfaces adds complexity to the user experience, and the passing of credentials over unsecured connections would also be unacceptable.

The Pycom resolves this by enabling encrypting device via Secure Boot and Flash Encryption [64].

The Thread protocol commissioning process aids resolving this challenge as well.

During Thread network formation, the border router generates a random network master key. According to the Thread technical overview, the Thread software stack does not provide any mechanism for retrieving the key once created. If a Thread device is not yet a member of a Thread network and seeks to join, the thread protocol demands that the device first establish a secure Datagram Transport Layer Security (DTLS) connection with a Thread Border Router.

Meanwhile, the commissioning device (an off network smart phone, for example) establishes a secure DTLS session with the border router using a pre-determined commissioning passphrase. This passphrase is used to derive an enhanced key using key stretching [74]. A human operator then authenticates and authorizes the new joining device through the commissioning device (Pybytes app). Once authorized, the border router provides the device the necessary security material to attach to the network over the secure DTLS connection that attackers cannot intercept. At no point does the commissioning device ever receive or hold the network security credentials, protecting

from off-network exploitation [76]. Once joiner and border router exchange the network-wide key, the nodes utilize MLE messages “to establish and configure secure links, detect neighbouring devices, and maintain routing costs between devices as the network changes” [73].

Thread commissioning provides a secure means for distribution of key materials and simplicity in authorizing new devices to the network. The Thread border router commissioning process allows an autonomous self-configuring mesh protocol to implement MAC link-level security [75] in a simplified, user-friendly manner and significantly contributes in addressing the OWASP IoT concern for authentication and authorization.

6.2 Insecure Network Services

Weak network services in IoT devices can result in denial-of-service or facilitate attacks on other devices. Devices may contain open ports that are unnecessary for their intended functionality. Developers often overlook these ports on IoT devices, assuming the network interfaces will not be exposed to external networks. Besides providing an access vector with weak credentials, these services can also often be exploited via buffer overflow or fuzzing [71]. The Thread 1.1.1 Specification provides flexibility for implementation of various communication and commissioning topologies that may include border routers and off-network commissioning devices [76]. Thread does not mandate specific hardware, software, or operating systems for such componentry, allowing configuration and deployment to support vendor-specific features while mandating consistency for the Thread specific functions [76].

Network services are defined by end-user implementation as well by procedures implemented by default in the firmware. By default, at boot, open WiFi network is created

port 21 FTP as well, port 23 Telnet is opened for debugging, after determined time port 23 closes, leaving port 21 open with default credentials.

For production environment a profile with changed credentials as well considerations of opened ports need to be defined.

6.3 Insecure ecosystem interfaces

For most IoT devices, cloud-based data storage and access are integral to the required functionality. Off-premises storage of data leads to significant concerns for data protection. Insecure cloud interfaces often have weak credentials or allow account enumeration and manipulation of password reset mechanisms. The specific vulnerabilities are the same as the previous web interface concern which include default or weak passwords, lack of failed login lockouts, faulty password recovery mechanisms, or standard web-based vulnerabilities [71].

Implemented ecosystem compromises 3 interfaces:

- software.pycom.io OTA firmware update interface
- mqtt.pybytes.pycom.io telemetry interface
- pybytes.pycom.io cloud management interface

At the time of the writing no major risks of insecure interfaces were discovered.

6.4 Lack of secure update mechanism

Unauthorized software and firmware updates are a major threat vector for IoT cyber-attack. With the lack of any secure update mechanism in place, there's no guarantee that the security of the IoT device is as projected to end-users or as intended by developers.

There are four critical security requirements for delivering updates securely to IoT devices:

- Securing access to the updates
- Verifying the source of the updates
- Verifying the integrity of the updates
- Anti-rollback mechanisms

Pycom offers firmware updates for devices over:

- OTA
 - Cloud (Pybytes, Activation by key over HTTPS)
 - Mobile app (WiFi authentication + FTP credential authentication)
- USB
 - Firmware updater (FTP credential authentication)
 - Encrypted flash (Encryption keys)
- SD card
 - Telnet (credential authentication) to execute upgrade

Pycom's solution for updates includes secure access to update over HTTPS, source of updates with certificate pinning, verification of integrity with files hashes within manifest. Only missing is rollback feature which is left end user to implement as written in their documentation [65].

6.5 Use of insecure or outdated components

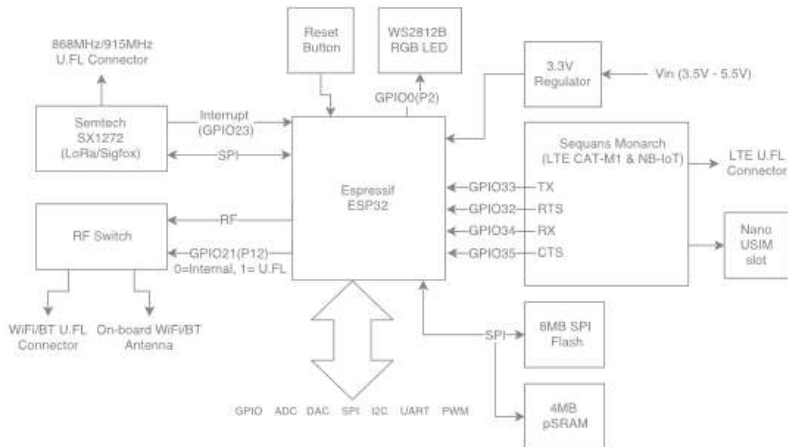


Figure 14 Pycom FiPy development board hardware architecture with named components

ESP32

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process [77]. Figure 14 [78].

CVEE overview:

- CVE-2020-15048 [79]
- CVE-2020-13629 [80], [81]
- CVE-2019-17391 [82]
- CVE-2019-15894 [83]

Reviewed disclosures of vulnerabilities have been patched and rolled out by the manufacturer regularly. Manufacturer has provided additional guidance for hardening the devices.

Sequans monarch

The Monarch GM01Q module is an all-in-one, single-mode LTE-M (eMTC) and NB-IoT module. The Monarch GM01Q module comprises Sequans' Monarch LTE Platform and all other elements necessary for a complete LTE modem system. The Monarch GM01Q module is compatible with any host running Linux, Windows and a wide range of embedded and real-time operating systems [84].

No vulnerabilities have been disclosed on module.

MicroPython

MicroPython is a full Python compiler and runtime that runs on the bare metal [85]. MicroPython implements the entire Python 3.4 syntax. The standard Python libraries have been "micro-ified" to fit in with the philosophy of MicroPython. They provide the core functionality of that module and are intended to be a drop-in replacement for the standard Python library [86].

As of MicroPython libraries are basic as possible and no external dependencies are carried along. Probability of a supply chain attack is minimal as this will require 0-day vulnerability to occur.

Pycom has provided a guide on how to update components, firmware, as well as libraries in case of this happening [87].

In conclusion, the selected solution does contribute to managing the risk of OWASP IoT Top 10 use of insecure or outdated components.

6.6 Insufficient privacy protection

Privacy concerns for IoT devices include both the collection and protection of personal data [71]. Given the emerging, ubiquitous nature of IoT devices, personal data can go

beyond financial and health records. IoT devices can provide insight into personal activities, preferences, and patterns allowing exploitation for nefarious purposes. Although the collection of personal data is an operational or functional concern, IoT privacy concerns magnify if a device has insufficient authentication, lack of transport encryption, or insecure storage of information [71].

For this, Pymesh supports several levels of encryption:

Mesh Masterkey

Each node (Lopy/Fipy) initializes Pymesh with a 128 bits Masterkey. This is used in: Authentication. Node which does not have the Masterkey of the peer, can't connect to peer's Pymesh further, it will create its own Pymesh, using its Masterkey, so it will become the Leader of a new Mesh network.

Encryption

All traffic inside Pymesh is encrypted with Masterkey, encryption is AES-128bits [63].

In conclusion, Pymesh in depth covers the IoT privacy concern.

6.7 Insecure data transfer and storage

Transport encryption prevents data from being viewed as it travels across networks. Local networks are usually unencrypted and visible to anyone on the network. Wireless networks can often be misconfigured resulting in unauthorized access. IoT devices may utilize proprietary or weak encryption protocols. Lack of encryption can lead to exposure of data, but more importantly, it can provide critical information necessary to further compromise an IoT device or network [71]. The use of encryption on IoT devices has

been a constant challenge given the significant power drain associated with advanced features.

According to a Thread overview briefing, “Host devices can typically operate for several years on AA type batteries using suitable duty cycles” [76]. To extend operations, Thread allows devices to sleep with adjacent nodes monitoring activities. The protocol mandates neighbour information exchange to include information on sleepy end devices and their sleep cycles [73]. These power management features allow the implementation of AES-128 link-layer security provided by the 802.15.4 MLE protocol. Additionally, since Thread utilizes 6LowPAN to encapsulate the 802.15.4 messages in IPv6, Thread allows the application to use any additional internet security protocol for end-to-end communication.

Pycom also provides possibility to implement end to end encryption and different encryption algorithms for key distribution.

Algorithms as:

- Symmetric encryption (crypto.AES class)
- Asymmetric encryption (crypto.rsa_encrypt() method) [63]

End to end encryption is used when Node A wants to communicate securely/secretly with Node B. The data packets will be routed by other nodes, but the actual message can't be decrypted by any middle Node.

This encryption can be used even for communicating between Nodes that are not in the same mesh, as message is encrypted until destination [63].

6.8 Lack of device management

The ability to configure security options is essential in providing granular permissions for the access of data or controls for IoT devices. Broad access to certain data or functions on the IoT device may be a desirable feature for some applications, with the necessity of limiting access to administrative features such as the connection to new devices and password setting. To maintain high levels of security and privileged access, IoT devices require the ability to separate administrative users from ordinary users, and a means for monitoring and logging various security events [71].

Presented implementation makes use of service provided by the development board manufacturer.

Pybytes is cloud-based device management platform available for all Pycom development boards and modules. It works from smartphone or desktop to provision devices.

Pybytes features:

Mobile App. Pybytes app lets you provision your devices and gives you access to monitor them.

- Firmware Over the Air
- Device Management. Track and control rollouts on your devices.
- Integrations. Integrate with AWS, Microsoft Azure, Webhooks and Google Cloud.
- Network Management
- Code Build and Push. Pymakr allows to build code and easily push it to devices.
- Device Monitoring. Monitor all your devices to navigate layout. Set your alerts in application.
- Devices Tracking. Google Maps integration tracks and visualise your device location.

[88]

6.9 Insecure default settings

OWASP includes software and firmware security as a major IoT concern. According to OWASP, “the lack of ability for a device to be updated presents a security weakness on its own” [71]. First and foremost, devices must have mechanisms to allow easy updates as vulnerabilities are discovered and resolved. Additionally, software and firmware can be insecure if they contain hard-coded sensitive data or credentials. Depending on how systems distribute software and firmware updates, it is possible to intercept and compromise updates, unless mechanisms are in place to deny malicious software configurations, such as signing and verification of code [71].

Pycom’s FiPy provides possibilities to update configuration of connection to different networks (WiFi, LTE, Sigfox, LoraWAN) over the air. These networks can all be used to carry out an upgrade of firmware. Upgrades can be done over local WiFi server, by network owners locally hosted LoraWAN server or Pybytes cloud services. All these methods use authentication and integrity checks in the process [65].

6.10 Lack of physical hardening

The last of the OWASP IoT Top 10 security concerns addresses poor physical security. If an attacker can easily disassemble a device or otherwise exploit the provided external ports, the installed operating system, and stored data become exposed. Attackers can modify devices for use in other purposes than those originally intended. One must review how easily device software can be accessed if any ports are present that are not necessary for normal operation, or if any administrative functions are limited or protected from physical tampering. Encryption of data at rest can further protect data on physically compromised IoT devices. [71].

ESP32 Platform provides possibility as well good documentation for:

- Secure Boot. The secure boot support ensures that when the ESP32 executes any software from flash, that software is trusted and signed by a known entity. If even a single bit in the software bootloader and application firmware is modified, the firmware is not trusted, and the device will refuse to execute this untrusted code.
- Encrypted Flash. The flash encryption support ensures that any application firmware, that is stored in the flash of the ESP32, stays encrypted. This allows manufacturers to ship encrypted firmware in their devices [64], [89].

7 Summary

In this thesis we introduced the needs and limitations of emergency scenario communication solutions of today. We assessed best technology for our use case is LoRa. Our solution removes the complexity, high operating cost and need for licensing. Hardware was selected to provide flexibility of networks and redundancy of connection with multinet network connectivity. Security risk analysis was used to select Thread as MAC protocol for implementation. Importance of management system and update mechanisms were brought foreground and addressed. An evaluation of proposed default configured setup was conducted against OWASP IoT Top 10 Security concerns. Finally, risk managing solutions with a checklist of measures to implement for production use were presented which can be found in I Indrek Taal

1. grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis, "Security Risk Analysis of Wireless Mesh Network for Emergency Networks" supervised by Toomas Lepik

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

_____ (date)

Appendix 2 Proposed solutions.

In future assessed and analysed technological solution will be fully implemented in basis of this thesis.

References

- [1] W. Cai *et al.*, “Increasing frequency of extreme El Niño events due to greenhouse warming,” *Nat. Clim. Chang.*, vol. 4, no. 2, pp. 111–116, 2014.
- [2] V. Y. Kishorbhai and N. N. Vasantbhai, “AON: A survey on emergency communication systems during a catastrophic disaster,” *Procedia Comput. Sci.*, vol. 115, pp. 838–845, 2017.
- [3] L. Rabieekenari, K. Sayrafian, and J. S. Baras, “Autonomous relocation strategies for cells on wheels in public safety networks,” in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2017, pp. 41–44.
- [4] P. Karn, H. Price, and R. Diersing, “Packet radio in the amateur service,” *IEEE J. Sel. Areas Commun.*, vol. 3, no. 3, pp. 431–439, 1985.
- [5] R. Singh, M. Thompson, S. A. Mathews, O. Agbogidi, K. Bhadane, and K. Namuduri, “Aerial base stations for enabling cellular communications during emergency situation,” in *2017 International Conference on Vision, Image and Signal Processing (ICVISIP)*, 2017, pp. 103–108.
- [6] R. Bruno, M. Conti, and E. Gregori, “Mesh networks: commodity multihop ad hoc networks,” *IEEE Commun. Mag.*, vol. 43, no. 3, pp. 123–131, 2005.
- [7] K. L. S. Sharma, *Overview of industrial process automation*. Elsevier, 2016.
- [8] BoostHigh Sp. z o.o., “How Does Mesh Network Allow IoT Devices To Communicate?” <https://boosthigh.com/wp-content/uploads/2019/07/1-kopia.jpg> (accessed Nov. 04, 2020).
- [9] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Comput. networks*, vol. 47, no. 4, pp. 445–487, 2005.
- [10] X. W. Ian Akyildiz, “Wireless Mesh Networks/Ian Akyildiz.” London: John Wiley & Sons, (November 2004, Georgia Institute of Technology~..., 2004.
- [11] J. R. Okin, *The Internet revolution: The not-for-dummies guide to the history, technology, and use of the Internet*. Ironbound Press, 2005.
- [12] J.-P. Bardyn, T. Melly, O. Seller, and N. Sornin, “IoT: The era of LPWAN is starting now,” in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, 2016, pp. 25–30.
- [13] KotahiNet, “Network.” <https://kotahi.net/network/> (accessed Nov. 04, 2020).
- [14] J. Petäjälä, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, “Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring,” in *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2016, pp. 1–5.
- [15] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range

- communications in unlicensed bands: The rising stars in the IoT and smart city scenarios,” *IEEE Wirel. Commun.*, vol. 23, no. 5, pp. 60–67, 2016.
- [16] M. J. Lee, J. Zheng, Y.-B. Ko, and D. M. Shrestha, “Emerging standards for wireless mesh technology,” *IEEE Wirel. Commun.*, vol. 13, no. 2, pp. 56–63, 2006.
- [17] R. S. Sinha, Y. Wei, and S.-H. Hwang, “A survey on LPWA technology: LoRa and NB-IoT,” *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017.
- [18] B. Reynders and S. Pollin, “Chirp spread spectrum as a modulation technique for long range communication,” in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, 2016, pp. 1–5.
- [19] B. Vejlgard, M. Lauridsen, H. Nguyen, I. Z. Kovács, P. Mogensen, and M. Sorensen, “Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot,” in *2017 IEEE 85th vehicular technology conference (VTC Spring)*, 2017, pp. 1–5.
- [20] L. Alliance, “White Paper: A Technical Overview of Lora and LoraWan (The Lora Alliance, San Ramon, CA, USA, 2015). IEEE 802 Working Group and Others. IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),” *IEEE Std*, vol. 802, pp. 4–2011.
- [21] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, “Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards,” *Comput. Commun.*, vol. 30, no. 7, pp. 1655–1695, 2007.
- [22] Open-ZB, “ZigBee protocol stack architecture.”
https://www.researchgate.net/figure/The-IEEE-802154-ZigBee-protocol-stack-architecture-The-IEEE-802154-ZigBee-protocols_fig1_220747134 (accessed Nov. 04, 2020).
- [23] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, “A survey on IEEE 802.11 ah: An enabling networking technology for smart cities,” *Comput. Commun.*, vol. 58, pp. 53–69, 2015.
- [24] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, “IEEE 802.11 ah: the WiFi approach for M2M communications,” *IEEE Wirel. Commun.*, vol. 21, no. 6, pp. 144–152, 2014.
- [25] R. Want, B. Schilit, and D. Laskowski, “Bluetooth le finds its niche,” *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 12–16, 2013.
- [26] “Future Internet 2019, 11(4), 99;”
https://www.mdpi.com/futureinternet/futureinternet-11-00099/article_deploy/html/images/futureinternet-11-00099-g016.png (accessed Nov. 04, 2020).
- [27] “Future Internet 2019, 11(4), 99;”
https://www.mdpi.com/futureinternet/futureinternet-11-00099/article_deploy/html/images/futureinternet-11-00099-g017.png (accessed Nov. 04, 2020).
- [28] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, “Wireless mesh networking: An IoT-oriented perspective survey on relevant technologies,” *Futur. Internet*, vol. 11, no. 4, p. 99, 2019.

- [29] M. Hernandez, "Connectivity Now and Beyond; Exploring Cat-M1, NB-IoT, and LPWAN Connections," *Blog, Ubidots, July*, vol. 5, 2018.
- [30] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.
- [31] Gareth Halfacree, "Leti Boasts of LoRa, NB-IoT-Beating LPWA-CB Tests," 2018. <https://abopen.com/news/leti-boasts-of-lora-nb-iot-beating-lpwan-tests/> (accessed Nov. 04, 2020).
- [32] E. Pietrosevoli, "Wireless Standards for IoT: WiFi, BLE, SigFox, NB-IoT and LoRa," 2017. http://wireless.ictp.it/school_2017/Slides/IoTWirelessStandards.pdf (accessed Nov. 04, 2020).
- [33] M. Diez, "Secure Position Data Transmission for Object Tracking using LoRaWAN." Master's thesis, University of Zurich, Department of Informatics, Zurich~..., 2017.
- [34] E. E. Petrosky, A. J. Michaels, and D. B. Ridge, "Network scalability comparison of IEEE 802.15. 4 and receiver-assigned CDMA," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6060–6069, 2018.
- [35] E. Casilari, A. Flórez-Lara, and J. M. Cano-García, "Analysis of the scalability of hierarchical IEEE 802.15. 4/Zigbee networks.," in *Infoscale*, 2008, p. 3.
- [36] WBA PMO, "The Role of Wi-Fi & Unlicensed Technologies in 5G." <https://wballiance.com/the-role-of-wi-fi-unlicensed-technologies-in-5g/> (accessed Nov. 04, 2020).
- [37] K. Pothuganti and A. Chitneni, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *Adv. Electron. Electr. Eng.*, vol. 4, no. 6, pp. 655–662, 2014.
- [38] H. Fornazier, A. Martin, and S. Messner, "Wireless Communication: Wi-Fi, Bluetooth, IEEE 802.15. 4, DASH7," *ROSE 2012 ELECINF344/ELE CINF381, Télécom ParisTech, web site http://rose. eu. org/2012/category/admin*, 2012.
- [39] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society*, 2007, pp. 46–51.
- [40] K. Gravogl, J. Haase, and C. Grimm, "Choosing the best wireless protocol for typical applications," 2011.
- [41] G. Heidari, *WiMedia UWB: technology of choice for wireless USB and Bluetooth*. John Wiley & Sons, 2008.
- [42] M. D. D. Vishwakarma, "IEEE 802.15. 4 and ZigBee: A conceptual study," *Channels*, vol. 868, pp. 866–868, 2012.
- [43] V. P. Rao, "The simulative investigation of Zigbee/IEEE 802.15. 4," *Dresden Univ. Technol.*, 2005.
- [44] J. M. Varghese, K. V Nibi, V. T. Varghese, and S. Rao, "A survey of the state of the art in Zigbee," *Int. J. Cybern. Informatics*, vol. 4, no. 2, pp. 145–155, 2015.
- [45] R.-C. Wang, R.-S. Chang, and H.-C. Chao, "Internetworking between ZigBee/802.15. 4 and IPv6/802.3 network," *SIGCOMM Data Commun. Festiv.*, 2007.

- [46] M. G. Jean-Paul, "Linmartz's Wireless Communication, vol. 1 (1)," *Baltzer Sci. Publ. PO Box*, vol. 37208, p. 1030, 1996.
- [47] B. Sidhu, H. Singh, and A. Chhabra, "Emerging wireless standards-wifi, zigbee and wimax," *World Acad. Sci. Eng. Technol.*, vol. 25, no. 2007, pp. 308–313, 2007.
- [48] M. Ibrahim, M. M. Zahara, S. Noaman, and A. S. Ragab, "Performance investigation of wimax 802.16 m in mobile high altitude platforms," 2013.
- [49] LoRa Alliance, "What are LoRa® and LoRaWAN®?" <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/> (accessed Nov. 04, 2020).
- [50] LoRa Alliance, "A technical overview of LoRa® and LoRaWAN™." <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf> (accessed Nov. 04, 2020).
- [51] F. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2186–2198, 2017.
- [52] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.
- [53] SEMTECH, "SEMTECH SX1276/77/78/79 DATASHEET." https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoKFfvaF_Fkpgp5kzjiNyiAbqcpqh9qSjE (accessed Nov. 04, 2020).
- [54] A. S. Tanenbaum and D. J. Wetherall, "Computer Networks: Pearson New International Edition," *Univ. Hertfordshire. Pearson High. Ed*, p. 54, 2013.
- [55] Geeksville, "Meshtastic: Mesh broadcast algorithm." <https://github.com/meshtastic/Meshtastic-device/blob/master/docs/software/mesh-alg.md> (accessed Nov. 04, 2020).
- [56] Thread Group Inc, "Introducing Thread: A New Wireless Networking Protocol for the Home." <http://threadgroup.org/news-events/press-releases/ID/20/Introducing-Thread-A-New-Wireless-Networking-Protocol-for-the-Home> (accessed Nov. 04, 2020).
- [57] H. Zimmermann, "OSI reference model-the ISO model of architecture for open systems interconnection," *IEEE Trans. Commun.*, vol. 28, no. 4, pp. 425–432, 1980.
- [58] S. Antipolis and P. Girard, "Low Power Wide Area Networks security. white paper by Gemalto Inc," 2015.
- [59] X. Yang, "LoRaWan: vulnerability analysis and practical exploitation," *Delft Univ. Technol. Master Sci.*, 2017.
- [60] Y. M. Amin and A. T. Abdel-Hamid, "Classification and analysis of IEEE 802.15. 4 PHY layer attacks," in *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, 2016, pp. 1–8.
- [61] Y. M. Amin and A. T. Abdel-Hamid, "Classification and analysis of IEEE 802.15. 4 MAC layer attacks," in *2015 11th International Conference on Innovations in Information Technology (IIT)*, 2015, pp. 74–79.

- [62] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 162–175.
- [63] Pycom, "Pymesh Security." <https://docs.pycom.io/pymesh/security/> (accessed Nov. 04, 2020).
- [64] Pycom, "SecureBoot and Encryption." <https://docs.pycom.io/advance/encryption/> (accessed Nov. 04, 2020).
- [65] Pycom, "Over The Air." Over The Air (accessed Nov. 04, 2020).
- [66] S. Langkemper, "Comparison of IoT Security Frameworks." <https://www.euofins-cybersecurity.com/news/comparison-iot-security-frameworks/>.
- [67] ETSI, "ETSI EN 303 645 V2.1.0," 2020. https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.00_30/en_303645v020100v.pdf (accessed Jan. 06, 2021).
- [68] IoT Security Foundation, "IoT Security Compliance Framework Release 2.1," 2020. <https://www.iotsecurityfoundation.org/wp-content/uploads/2020/05/IoTSF-IoT-Security-Compliance-Framework-Questionnaire-Release-2.1.zip>.
- [69] The OWASP Foundation, "The OWASP Internet of Things Security Verification Standard (ISVS)," 2021. <https://owasp-isvs.gitbook.io/owasp-isvs-pr/>.
- [70] European Union Agency for Network and Information Security (ENISA), "Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures," 2017. https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot/at_download/fullReport.
- [71] OWASP Foundation, "OWASP IoT Top 10 PDF," 2018. <https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf> (accessed Nov. 04, 2020).
- [72] B. Herzberg, D. Bekerman, and I. Zeifman, "Breaking down mirai: An IoT DDoS botnet analysis," *Incapsula Blog, Bots DDoS, Secur.*, 2016.
- [73] Thread Group Inc, "Thread Stack Fundamentals." https://www.threadgroup.org/Portals/0/documents/support/ThreadOverview_633_2.pdf (accessed Nov. 04, 2020).
- [74] Thread Group Inc, "Thread Commissioning." https://www.threadgroup.org/Portals/0/documents/support/CommissioningWhitePaper_658_2.pdf (accessed Nov. 04, 2020).
- [75] I. Silicon Laboratories, "Application Development Fundamentals: Thread." <https://www.silabs.com/documents/public/user-guides/ug103-11-fundamentals-thread.pdf> (accessed Nov. 04, 2020).
- [76] Thread Group Inc, "Thread 1.1.1 Spec." .
- [77] Espressif Systems, "ESP32 Overview." <https://www.espressif.com/en/products/socs/esp32> (accessed Nov. 04, 2020).
- [78] Pycom, "Pycom FiPy Specsheat." https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_FiPy_v2.pdf (accessed Nov. 04, 2020).

- [79] Raelize B.V., “Espressif ESP32: Bypassing Flash Encryption (CVE-2020-15048),” 2020. <https://raelize.com/posts/espressif-systems-esp32-bypassing-flash-encryption/>.
- [80] Raelize B.V., “Espressif ESP32: Bypassing Encrypted Secure Boot (CVE-2020-13629),” 2020. <https://raelize.com/posts/espressif-esp32-bypassing-encrypted-secure-boot-cve-2020-13629/> (accessed Nov. 04, 2020).
- [81] Espressif Systems, “Security Advisory concerning fault injection and ESP32 Flash Encryption & Secure Boot V1,” 2020. https://www.espressif.com/sites/default/files/advisory_downloads/Security_Advisory_CVE-2020-15048%2C_13629_EN%26CN.pdf (accessed Nov. 04, 2020).
- [82] Espressif Systems, “Security Advisory concerning fault injection and eFuse protections (CVE-2019-17391),” 2020. https://www.espressif.com/en/news/Security_Advisory_Concerning_Fault_Injection_and_eFuse_Protections (accessed Nov. 04, 2020).
- [83] Espressif Systems, “Espressif Security Advisory Concerning Fault Injection and Secure Boot (CVE-2019-15894),” 2020. https://www.espressif.com/en/news/Espressif_Security_Advisory_Concerning_Fault_Injection_and_Secure_Boot (accessed Nov. 04, 2020).
- [84] Sequans, “Monarch GM01Q Module,” 2020. https://www.sequans.com/wp-content/uploads/2019/10/PI-GM01Q-3-20191001_WEB.pdf (accessed Nov. 04, 2020).
- [85] “MicroPython,” 2020. <https://micropython.org/> (accessed Nov. 04, 2020).
- [86] “The MicroPython project,” 2020. <https://github.com/micropython/micropython> (accessed Nov. 04, 2020).
- [87] Pycom, “Modem Firmware Update.” <https://docs.pycom.io/updatefirmware/ltemodem/> (accessed Nov. 04, 2020).
- [88] Pycom, “What is Pybytes?” <https://docs.pycom.io/pybytes/> (accessed Nov. 04, 2020).
- [89] Espressif Systems, “Security Considerations.” <https://docs.espressif.com/projects/esp-jumpstart/en/latest/security.html#encrypted-flash> (accessed Nov. 04, 2020).
- [90] K. Pelechrinis, G. Yan, S. Eidenbenz, and S. V Krishnamurthy, “Detection of selfish manipulation of carrier sensing in 802.11 networks,” *IEEE Trans. Mob. Comput.*, vol. 11, no. 7, pp. 1086–1101, 2012.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis¹

I Indrek Taal

4. grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis, “Security Risk Analysis of Wireless Mesh Network for Emergency Networks” supervised by Toomas Lepik

1.3 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.4 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

5. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

6. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

_____ (date)

¹ The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 Proposed solutions

OWASP IoT Top 10 2018	Description	Solution to implement
I1 Weak, Guessable, or Hardcoded Passwords	Use of easily brute forced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.	<ul style="list-style-type: none"> • Change default Telnet, FTP credentials • Rename WiFi and create password • Using PyMesh create new network key
I2 Insecure Network Services	Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control.	<ul style="list-style-type: none"> • Limit Routers on boot launched services • Port or integrate micropython ssh library with sftp as default authentication
I3 Insecure Ecosystem Interfaces	Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.	<ul style="list-style-type: none"> • Create account with strong password and 2FA enabled • Consider complete OWASP ASVS test
I4 Lack of Secure Update Mechanism	Lack of ability to securely update the device. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.	<ul style="list-style-type: none"> • Implemented by default by Pycom firmware <p>ToDo</p> <ul style="list-style-type: none"> • Implement rollback features

I5 Use of Insecure or Outdated Components	Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain	<ul style="list-style-type: none"> • Components are up to date ToDo <ul style="list-style-type: none"> • Develop contingency plan for necessary update
I6 Insufficient Privacy Protection	User's personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.	<ul style="list-style-type: none"> • Implement certificate pinning
I7 Insecure Data Transfer and Storage	Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing	<ul style="list-style-type: none"> • Encryption implemented by default ToDo <ul style="list-style-type: none"> • Consider selfhosting PyBytes management
I8 Lack of Device Management	Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.	<ul style="list-style-type: none"> • Supported by device vendor's service ToDo <ul style="list-style-type: none"> • Develop contingency plan for cloudless management
I9 Insecure Default Settings	Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.	<ul style="list-style-type: none"> • All configuration open for modifying ToDo <ul style="list-style-type: none"> • Change all default settings • Create different role profiles
I10 Lack of Physical Hardening	Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.	<ul style="list-style-type: none"> • Implement Secure boot as well Flash encryption

Thread protocol security risk	Description	Solution to detect
Radio jamming [60]	Thread is inevitably prone to physical layer radio jamming, which is a form of a DoS attack.	<ul style="list-style-type: none"> Monitors the volume of uplink traffic from each and every client. A node that is able to send much larger volume of traffic is identified as a potential miscreant. [90] (PyBytes)
Link layer jamming and node-specific flooding [61]	Unlike radio jamming, link layer jamming creates a DoS attack by crafting link layer frames in a Thread network, to reduce network performance and throughput; node-specific flooding sends frames towards a particular node, either to drain its battery or to affect its functionality.	<ul style="list-style-type: none"> Ping device by nearest 3 to determine if node is isolated by attack (Device)
Back-off manipulation and Clear Channel Assessment (CCA) manipulation [60]	An adversary could deviate from the CSMA/CA channel access mechanism used by IEEE 802.15.4, by either using a shorter back-off time or even skipping CCA. Doing so would deteriorate the throughput of the legitimate nodes in the Thread network.	<ul style="list-style-type: none"> Visualise ping results of nodes (PyBytes)
Acknowledgment (ACK) attack [62]	IEEE 802.15.4 does not mandate integrity or confidentiality protection for acknowledgment frames. Once a message is received by a Thread node, the node responds with an ACK frame that includes the sequence number of the received frame. Since the frame is sent in clear text, an adversary can forge an ACK message if it knows the corresponding sequence number.	<ul style="list-style-type: none"> Anomaly detection view with ACK and frame numbers (PyBytes)

