Project ANR McBIM

Deliverable 2.6

Low level security aspects for the wireless communications protocols

Clarisse Erpeldinger, Gael Loubet, Eric Alata, Daniela Dragomirescu

This report is a security analysis of the Wireless Sensor Network of the ANR McBIM project. It provides an overview of the security mechanisms of the communication protocols LoRa and BLE and their vulnerabilities. Eventually, LoRa and BLE security issues and their solutions applied to the project are studied.

Contents

Part I
LoRa technology

1 Introduction to LoRa

LoRa is a physical layer technology enabling long range communications for low-power, wide area networks (LPWANs). It is a radiofrequency modulation based on the Chirp Spread Spectrum technique (CSS), created by Cycleo in 2010 and acquired two years later by Semtech. LoRa features provide low data rates and low power consumption over several kilometers (up to 15 kilometers). LoRa operates in a fixed-bandwidth channel of 125 KHz) or 500 KHz for uplink channels, and 500 KHz for downlink channels [1]. Based on the OSI model (figure 1), a MAC layer named LoRaWAN (LoRa for Wide Area Networks) is added to extend the LoRa physical layer onto Internet networks. It is an open networking protocol standardized and maintained by LoRa Alliance.

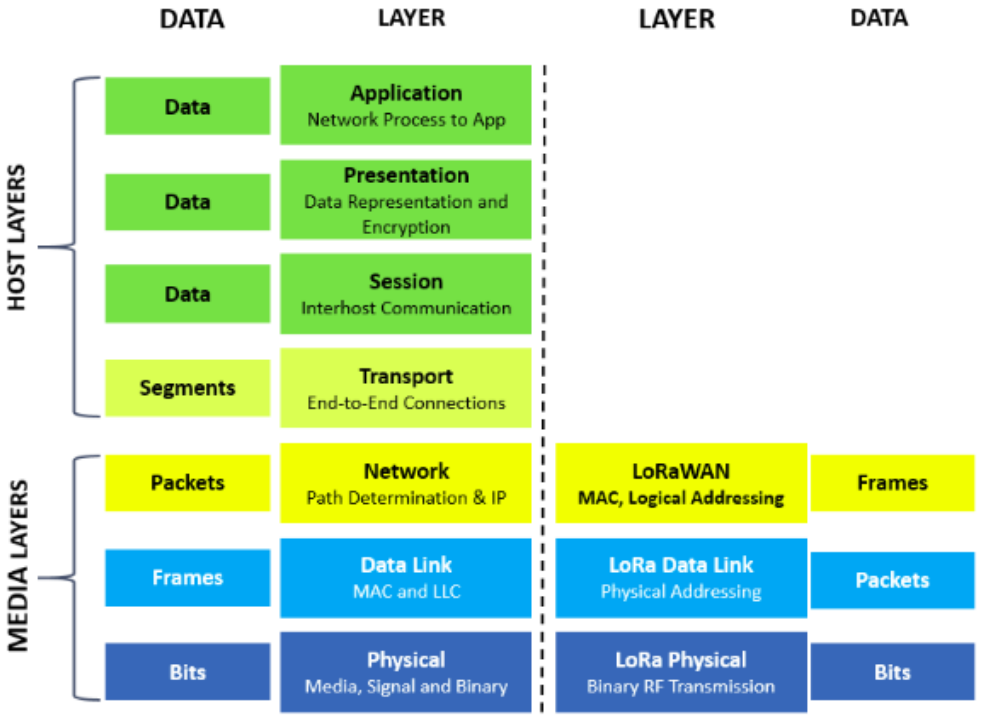


Figure 1: OSI seven-layer network model [1]

1.1 LoRaWAN Network architecture

LoRaWAN is a "star-of-stars" networking topology made of end-devices, Long Range relays named gateways, a Network Server (Long Range Controller) and applications servers. Communication is based on uplink and downlink transmissions. An uplink transmission is defined as transmission from an end-device to the Network Server or a gateway, while a downlink transmission is a transmission from the Network Server or a gateway to an end-device.

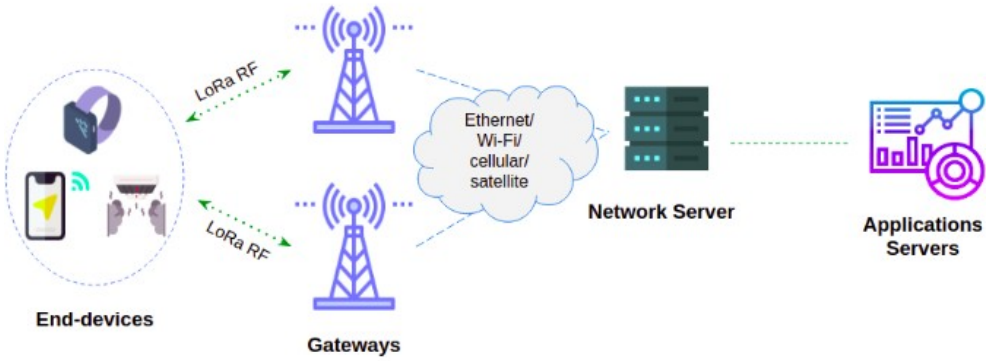


Figure 2: LoRaWAN network implementation

End device End-devices are nodes communicating with one or several gateways via LoRa modulation. In the ANR McBIM project, end-devices are Sensing Nodes (SN) measuring parameters such as humidity and temperature sending data to a LoRa gateway [2].

Gateway A gateway is a Communicating Node (CN) forwarding data it has received from end-devices to the Network Server. It relays messages in both directions between the "physical world" and "digital world" [2] via backhaul (Ethernet, cellular, Wi-Fi or satellite), using IP connection [3].

Network Server The Network Server manages the network and filters data exchanges. It is a centralized intelligence responsible for configuring packet exchange parameters and checking security [4]. This server is connected to applications servers and, in LoRaWAN v1.1, to a Join Server.

Application server Applications servers deploy IoT applications and securely handle, manage and interpret sensor application data. They generate the application-layer downlink payloads to the connected end devices [1].

Join Server The join server is only available in LoRaWAN version 1.1 and 1.0.3 [5]. It manages end-devices activation and join procedure to join the network.

1.2 Joining a LoRa network

In order to exchange data over the network, the end-device requires to join the network to be considered as active. Two activation methods are provided by LoRa: Activation By Personalization (ABP) and Over-The-Air Activation (OTAA). Activation provides several parameters to the end-device [4]:

- DevAddr composed of NwkID (network identifier) and NwkAddr (network address)
- DevEUI: end-device identifier
- AppEUI: application identifier, identifies the end-device
- NwkSKey: network session key, specific to the end-device and used both by the server and end-device to ensure data integrity
- AppSKey (AES-128 key): application session key

1.3 Activation By Personalization (ABP)

ABP uses a direct connection, sending messages directly to the server. The NwkSKey, AppSKey and DevAddr are pre-defined and stored in the end-device and the Network Server. This procedure is set by default.

1.4 Over-The-Air Activation (OTAA)

OTAA is a two-way handshake initiated by the end-device using join procedure. First, the end-device sends a Join-request packet with its information to the Network Server. Then, the Network Server responds with a Join-accept message and both session keys NwkSKey and AppSKey are calculated to encrypt data. In LoRaWAN v1.0.3 and v1.1 the Join Server manages the Over-The- Air Activation procedure and generates the session key for the Network Server and applications server [6]. This procedure is more secure than ABP.

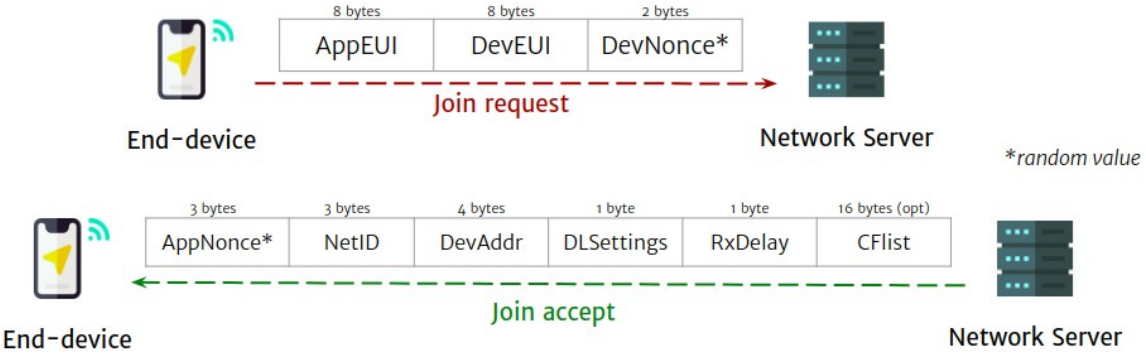


Figure 3: Join packets during OTAA

1.5 LoRaWAN operation modes

In order to fit best with the needs of IoT applications (low energy consumption, battery lifetime, etc.) LoRaWAN protocol has three operation modes (class A, B and C). In order to send messages from end-devices, a random sending channel is chosen per transmission. All classes use this method, however downlink message reception differs. In addition, there are two types of messages [4]:

- Unconfirmed messages: no acknowledgement is required from the Network Server.
- Confirmed messages: the end-device requires a response from the Network Server and opens two reception windows Rx1 and Rx2 for downlink transmissions. The windows are opened RxDelay1 and RxDelay2 respectively after the end of an uplink data transmission.

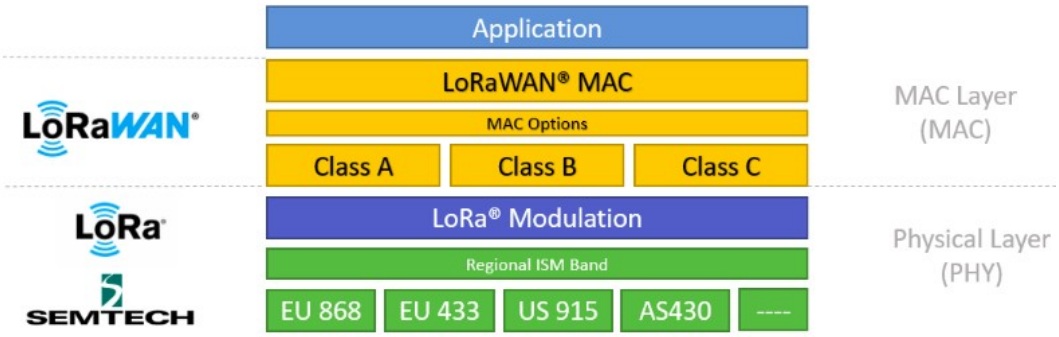


Figure 4: LoRaWAN technology stack

1.5.1 Class A

End-devices are in a deep sleep mode (i.e. idle state) until they send data. Once an uplink message is transmitted, two shorts downlink receiving windows RX1 and RX2 are opened to listen to Network Server transmissions. RX1 reuses the uplink channel. The window duration and the delay between RX1 and RX2 are configurable by using MAC commands. If no downlink response is received during RX1, the device stops listening until RX2. Still, if no message is received, the packet can be retransmitted until the receipt of an acknowledgement or the maximum attempt of retransmission is exceeded (also configurable).

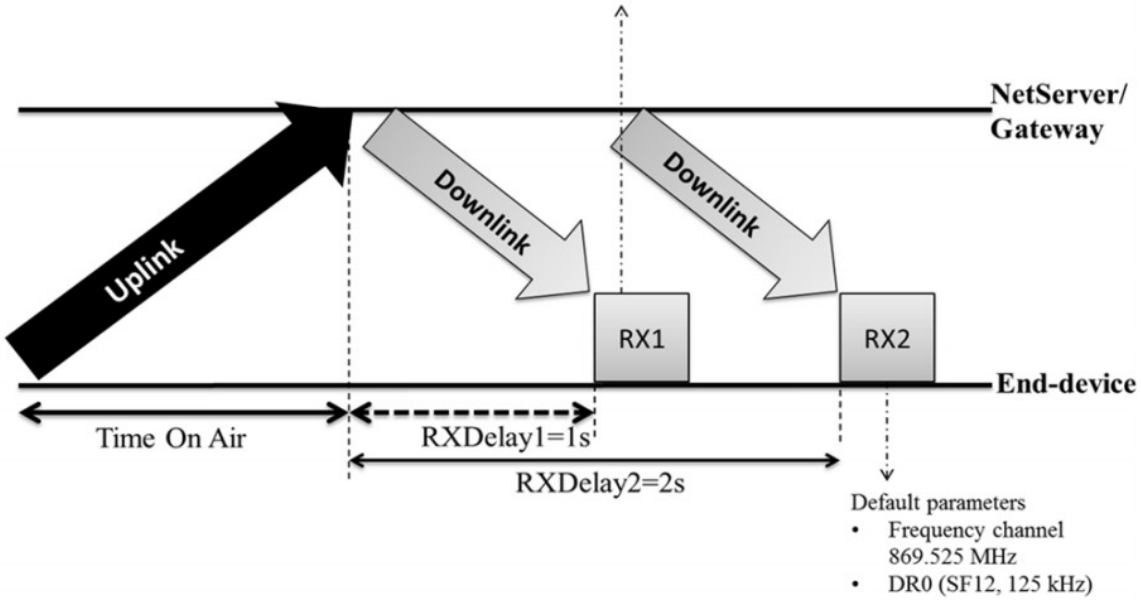


Figure 5: Data exchange procedure in Class A [4]

Class A operation mode enables unidirectional unicast and broadcast transmissions, which means applications cannot interrogate an end-device. This mode is implemented by default in every LoRa devices and ensures a low energy consumption.

1.5.2 Class B

Class B is an improvement of class A, based on a periodic listening for downlink transmissions. Besides the two reception windows, there are extra scheduled receive slots and the gateway will send

time synchronized beacons to the end-device to provide time reference (built-in GPS timing source are required in gateways) [4]. Moreover, transmissions are bidirectional and multicast messages are enabled.

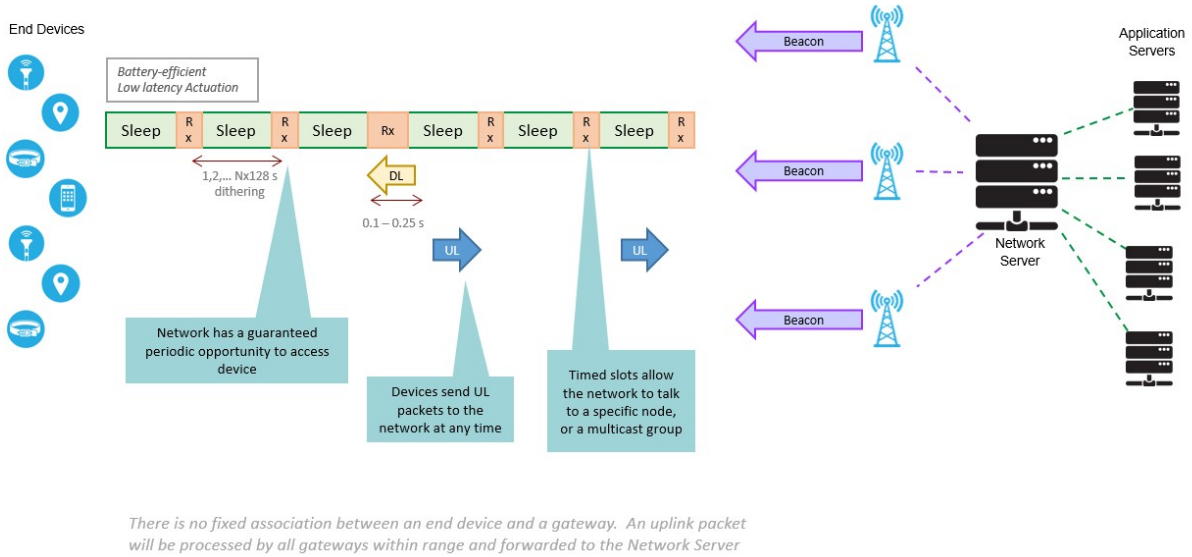


Figure 6: Class B beaconing operations [1]

1.5.3 Class C

In addition to class A receive slots, there is a continuous listening on the medium for downlink transmissions. This class offers a low latency for communications from the server to an end-device, however, class C devices require a battery as it is power consuming.

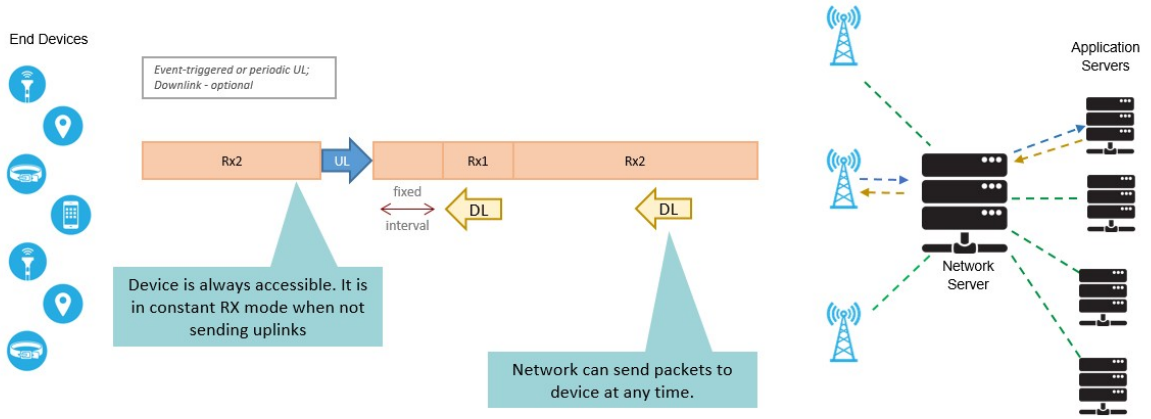


Figure 7: Class C operation [1]

Figure 8 summarizes the features of the three classes.

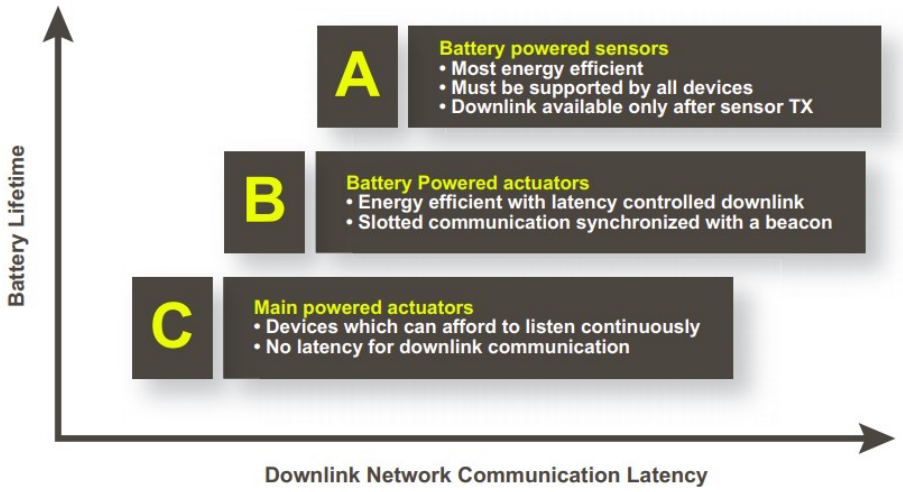


Figure 8: Comparison of LoRaWAN classes [3]

2 Security features

2.1 Authentication: join procedure

The join procedure is used in OTAA and is based on join and accept requests. Both end-device and the Network Server have a unique individual root key AppKey in LoRaWAN v1.0.2, assigned before communication [7]. There are two root keys in LoRaWAN v1.1 and v1.0.3, AppKey and NwkKey [5] [8]. The AppKey is an AES-128 key used to derive both session keys NwkSKey and AppSKey, which will be shared with the Networks Server and application servers [9].

Join request The end-device generates a random value called DevNonce to generate session keys. The join request message is not encrypted, however, the AppKey signs and generates the Message Integrity Code (MIC) to insure the integrity of the message.

MAC Header	Join Request or Join Accept or MAC payload	MIC

Table 1: LoRaWAN message physical payload structure [7]

Join accept The Network Server checks MIC, DevEUI and AppEUI values and generates the AppNonce if the device is accepted in the network. DevNonce and AppNonce values enable the generation of NwkSKey and AppSKey. The AppKey signs and encrypts the join accept message [7].

OTAA procedure ensures unicity for AppKey, DevEUI,AppEUI, AppNonce and DevNonce val-ues, and thus reduce the probability of compromising the whole network by compromising a node. There is also a buffer for DevNonce values to prevent replay attacks (cf. section 4.2). DevNonce values are stored and their unicity is verified. If the value has already been used, the device is not allowed to join thenetwork.

2.2 Key management

AppSKey and NwkSKey are unique AES-128 symmetric keys used for only one communication session.

NwkSKey This key is shared with the network and used for interactions between end-devices and the Network Server. It checks the integrity of messages (MIC check).

AppSKey The AppSKey is a private key used to encrypt and decrypt the payload between the end-device and application servers. Each application data is encrypted by a XOR operation [9].

Session keys generation differs when using OTAA or ABP. Both methods provides unique keys:

- OTAA: NwkSKey and AppSKey are generated from AppKey, DevNonce and AppNonce values. Session keys and nonces are regenerated at each reset or rejoin request [7].
- ABP: sessions keys are static keys assigned and stored directly in the end-device.

Key name	Key type	Length (bits)	Generated by	Usage
AppKey	Symmetric	128	application	• MIC for join request and accept • Encrypt/decrypt join accept • Generate session keys
AppSKey	Symmetric	128	AppKey	Encrypt/decrypt data messages
NwkSKey	Symmetric	128	AppKey	• MIC for messages • Encrypt/decrypt command-only messages

Figure 9: LoRaWAN key management [7]

To ensure a better security, nonces values and AppKey are transmitted, instead of sending keys over the air [7].

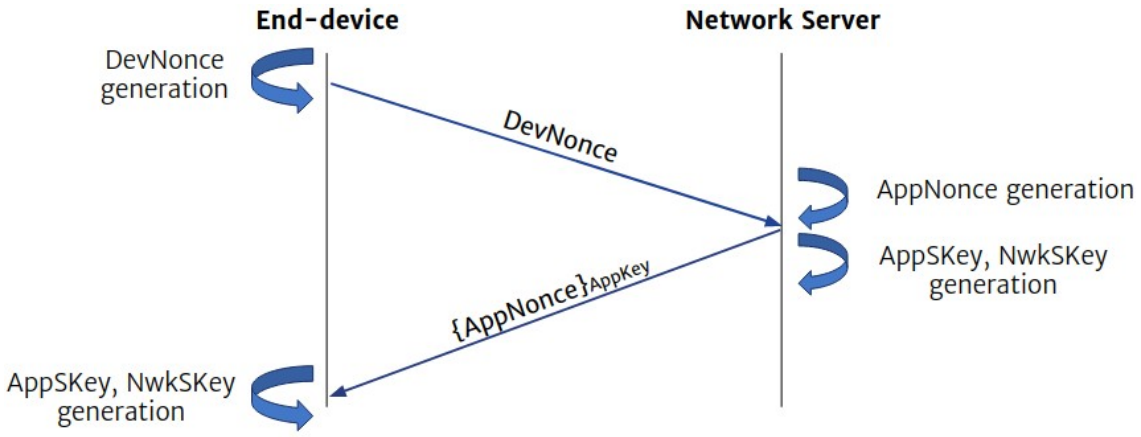


Figure 10: OTAA session keys exchange [7]

2.3 Encryption and message signing

2.3.1 Data message

Messages are encrypted using an Advanced Encryption Standard (AES) algorithm in counter mode (CTR), supporting 128 bits block length. First, the payload is encrypted with the NwkSKey

if it contains MAC commands only, otherwise AppSKey is used [7]. Then, the integrity of the message is checked with the Message Integrity Code (MIC).

2.3.2 Join message

Unlike the join request, the join accept is encrypted. Join accept is signed with Cipher-based Message Authentication Code (CMAC), encrypted with the block cipher method Electronic Code Book (ECB) and signed [7].

2.4 Counter management

In order to prevent replay attacks and packet loss, there are two frame counters to keep uplink and downlink messages synchronized [8]. FCntUp counts uplink messages in the end-device and FCntDown counts downlink messages in the Network Server. If the difference between FCntUp and FCntDown is greater than a limitation value MAX_FCNT_GAP, messages are dropped [7].

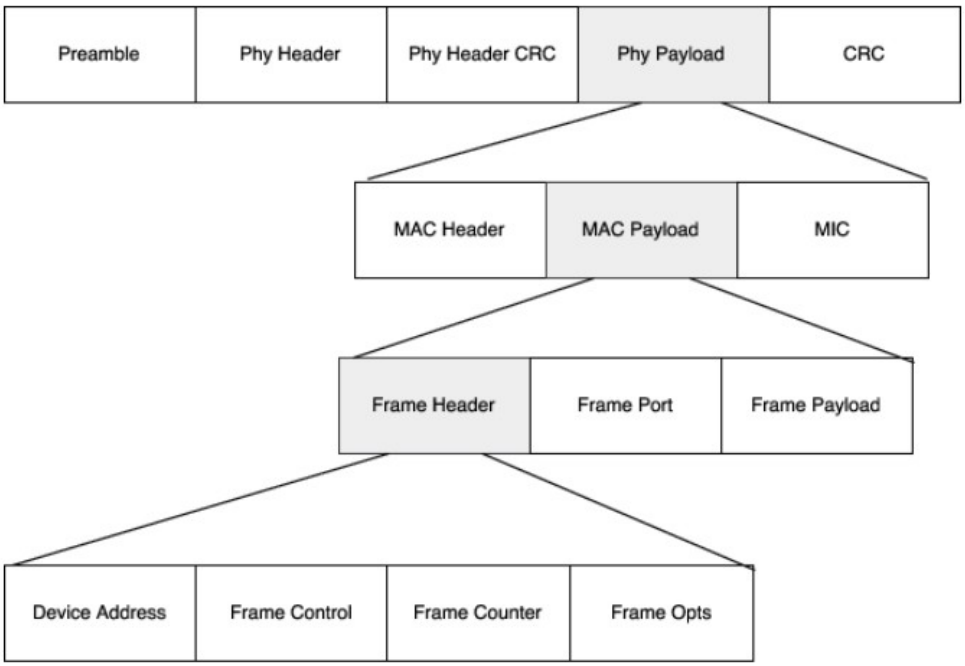


Figure 11: LoRaWAN packet structure [9]

2.5 Message acknowledgement

Acknowledgements are sent in response of uplink confirmed messages if they are acceptable [7]. The uplink message is retransmitted if ACK is not received during the end-device receive windows. Eventually, the message is lost or rejected if no ACK is received after several retransmissions.

2.6 Comparison of LoRaWAN versions

There are several versions of LoRaWAN to adapt depending on the use case. LoRaWAN version 1.0.2 is a more stable version, consuming less power.

Version 1.1 introduces a roaming architecture and three roles for the Network Server (home, forwarding, serving). There is a also a separation between network and application trust by using two separate keys (AppKey, NwkKey).

LoRaWAN 1.0.3 and 1.1 are more suitable for battery-powered class B end-devices. However, [10] specifies that no upgrade is needed to version 1.0.3 for class A and C operating devices.

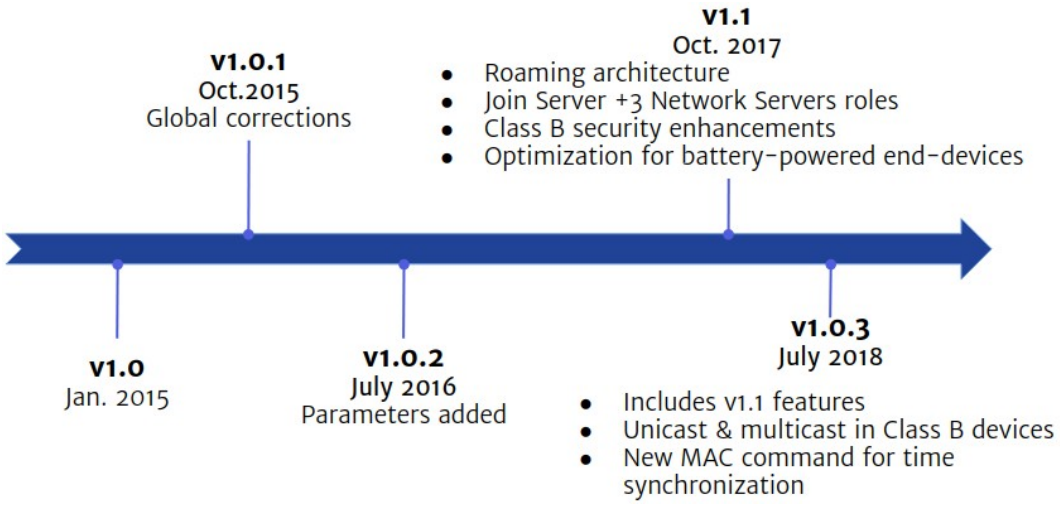


Figure 12: LoRaWAN technology versions evolution

3 Common LoRa vulnerabilities

3.1 Physical access to devices

Physical access to end-devices enable keys extraction (e.g. via reverse engineering by deriving key from public information from an end-device) [9], thus, end-devices and network keys might be compromised. Communications could be decrypted, the attacker could create a mock device with same credentials to impersonate a legitimate device, data payload can be manipulated, etc. Furthermore, additional hardware (e.g. radio modules) can be used near devices to intercept command and data exchanges [9]. To prevent this, exchanges of critical data should be avoided.

Note: in LoRaWAN it is not possible to decrypt the transmissions between end-devices and gateways without knowing AppKey (the payload is encrypted by AppKey) and NwkSKey (tampering with data makes the Message Integrity Code check fail).

3.2 Lack of association of messages

One of the most important vulnerability in LoRaWAN protocol is that there is no association between data messages and their acknowledgements, and between Join-accept messages and their requests, thus it promotes replay attacks and ACK spoofing [8] (cf. section 4.3). Two solutions are implemented for both issues in version 1.1:

- A ConfFCnt is included in MIC calculation of data messages and a ACK flag is set. Conf- FCnt needs to be set in the FCnt field of the acknowledged message to associate it with the acknowledgment.
- DevNonce value is included in MIC calculation for join-accept messages and the end-device expects a join-accept in response to a join-request.

3.3 Re-use of Nonce Values

Nonce values are pseudo-randomly generated and used only once. There is a risk of generating a value already used, especially in v1.0.2 where nonces values are not tracked. Thus, the network

is vulnerable to replay attack or eavesdropping (cf. section 4.5). LoRaWAN v1.1 implements a solution by turning nonces into counter nonces and preventing reuse of nonce values by storing and tracking only the last used values [8].

3.4 Frame counter management

Frame counter reboot ABP procedure sets FCntUp and FCntDown counters to 0 when the device is rebooted. According to [9] "if a malicious entity is able to reset the end-device, messages which were obtained before by sniffing the transmission between the end-device and gateway could be replayed back to the gateway". A solution to this could be storing the counter value in the server when the device is rebooting, and rejecting all messages while the device counter is less than or equal to the value of the server counter. However this would decrease the availability of the device [8].

Note: The Things Network has its own mechanism to block all the messages coming from the device until it frame counter reaches the frame counter stored in the gateway [9].

Frame counter overflow When then counter value reaches its maximum, it will be reset and set to 0. The vulnerability is the same as the frame counter reboot. It is common to ABP and OTAA [7]. A solution provided in LoRaWAN v1.1 is to rekey the end-device with rejoin-request [8].

These vulnerabilities enable replay attacks.

3.5 Lack of end-to-end integrity protection

The application data integrity is unprotected between the Network Server and application servers. LoRaWAN v1.0.2 and v1.1 specifications acknowledge the vulnerability but it is left to the implementation of applications [8].

3.6 Packet and payload vulnerabilities

The structure of a LoRaWAN packet does not include any time based data or signature to validate the time of the message, which makes it vulnerable to replay attacks. In addition, the payload length is fixed: it is the same before and after the encryption. Therefore, an attacker could overflow counters to restore the key stream from the encrypted messages [9].

4 Attacks and detection

4.1 Radio jamming

Radio jamming consists in transmitting a powerful radio signal near application devices by a malicious entity to disrupt radio transmissions. It is possible to jam end-devices or gateways using commercial-off-the-shelf LoRa hardware (costing approximately 30 euros) because of CSS modulation coexistence issues [9]. All transmissions can be wiped out at the frequency used by devices to communicate. For instance, if there are several LoRa transmissions on the same frequency with the same Spreading Factor, they interfere. The attacker only needs an Arduino platform with a LoRa radio module to flood LoRa messages at a certain frequency to wipe out communications (99% of LoRa transmissions are affected by this attack). This attack can be detected by observing a sudden drop out from the network. When the attack is detected, it is recommended to change the network frequency.

4.2 Replay attack

During a replay attack, the attacker captures a valid data transmission to repeat it or delay it to fool the device with handshake messages or old data from the network. The attacker needs to know the communication frequencies and channels used to practice this attack. Replay attacks can be prevented with the use of the tracking frame counters, join procedure via OTAA or physical protection. Replay attacks could lead to Denial of Service (DoS) [8], which intends to disrupt services.

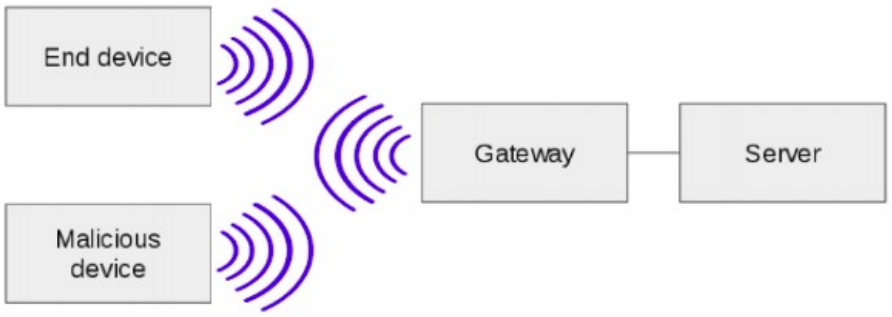


Figure 13: The LoRaWAN network setup for replay attack [7]

4.3 ACK spoofing

This attack results from the lack of association between acknowledgment and confirmed messages. The attacker prevents the reception of downlink frames (e.g. via jamming) and captures downlink ACK messages to acknowledge another confirmed uplink message from the same end-device. The goal of the ACK spoofing attack is mainly to take control of gateways, damage the network or provoke DoS. This attack is possible on uplink frames if the attacker can prevent recep- tion of uplink frames by gateways in the listening range.

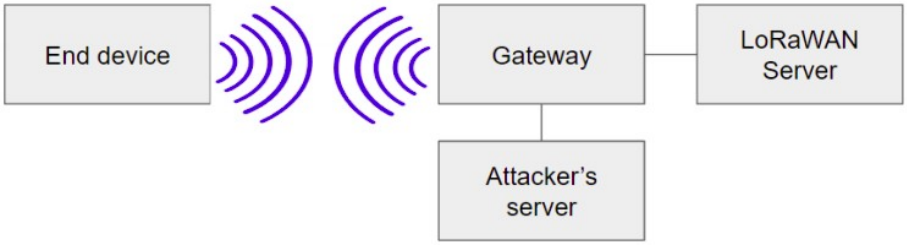


Figure 14: The LoRaWAN network setup for ACK spoofing [7]

4.4 Bit flipping

The missing end-to-end integrity protection of application data vulnerability enables bit flipping. If the transport layer security between the Network Server and the application server does not exist or is compromised, and the attacker is able to act on the channel, then the malicious entity could alter application data and compromise the confidentiality of the application.

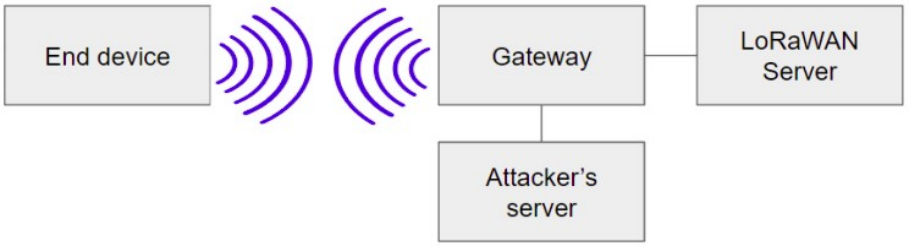


Figure 15: The LoRaWAN network setup for bit flipping [7]

4.5 Eavesdropping

Eavesdropping can be passive (e.g. sniffing) or active (e.g. relay attack, man-in-the middle). Sniffing is the most common passive eavesdropping vulnerability in LoRaWAN. During sniffing attack, the attacker captures packets transmitted over a network. For instance, an attacker might sniff the wireless traffic between end-devices and gateways to compromise the encryption method [8].

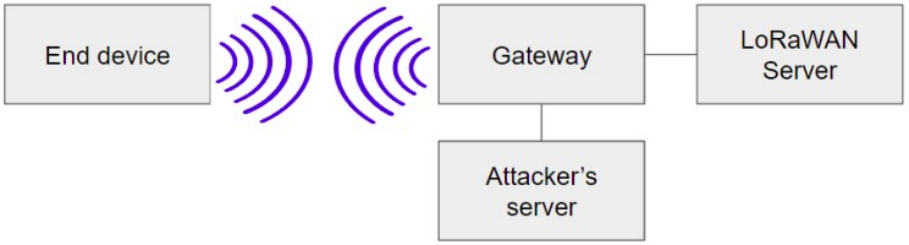


Figure 16: The LoRaWAN network setup for eavesdropping [7]

4.6 Other attacks

Relay attacks Relay attack occurs when a malicious entity creates a relay between the end-device and the Network Server and initiates a communication to relay the messages to another malicious entity.

Attacks against Class B networks Class B beacons are not encrypted nor signed and can be a source of data injection [7] [8].

Other various attacks exist, however they were not considered in this study due to the relevance in the ANR McBIM project.

Part II
Bluetooth Low Energy

5 Introduction to Bluetooth Low Energy

Bluetooth Low Energy (BLE) or Bluetooth Smart is a wireless technology developed in 2010 with the Bluetooth Core Specification version 4.0, by Bluetooth Special Interest Group (SIG). It is designed for low power and short range communications.

5.1 Stack overview

Bluetooth LE protocol stack is composed of:

- A controller managing radio channel access and emission (Rx) and transmission (Tx) of packets.
- A host in charge of high protocol layers, defining profiles.
- A Host Controller Interface (HCI) enabling communications between the host and controller.

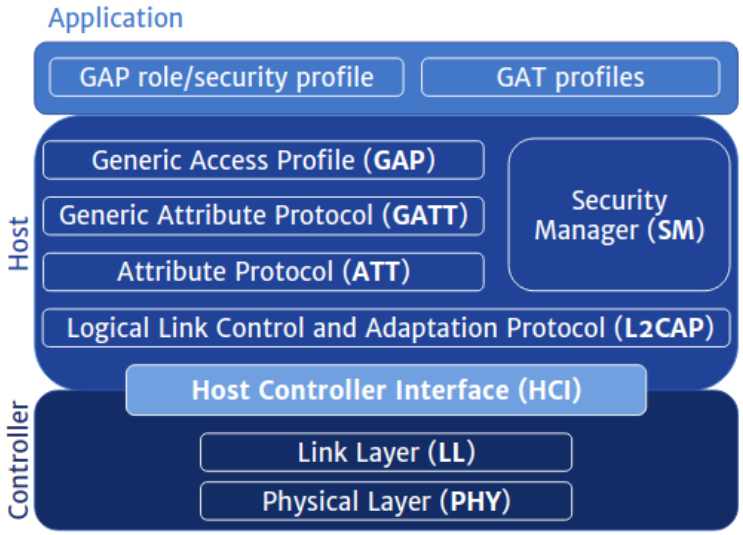


Figure 17: BLE stack

5.2 Controller

Physical Layer BLE is based on a Frequency Hopping Spread Spectrum (FHSS) and operates in the ISM 2.4 GHz free band using a Gaussian Frequency Shift Keying (GFSK) modulation. 40 channels are allocated, spaced 2MHz, including 3 advertising channels and 37 data channels.

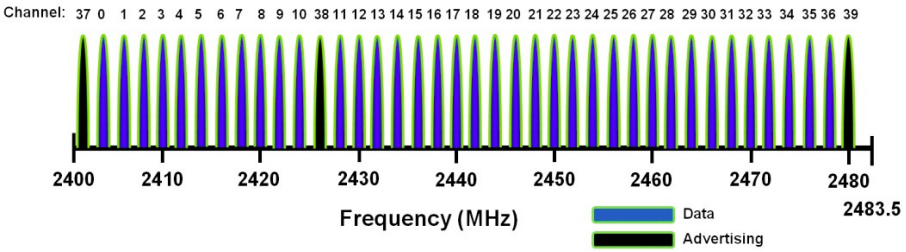
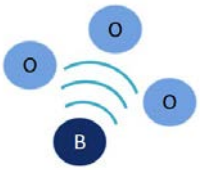


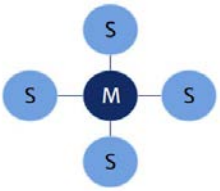
Figure 18: BLE channels [11]

Link Layer The link layer enables two communication modes:

- Advertising (broadcast/observer): broadcasts transmissions. This mode can be used to discover devices for connection establishment.
- Connected (master/slave): connects devices for bi-directional communications. A master can manage several connections while a slave is allowed to be connected to only one device. In order to save energy, slaves are in sleep mode and wake up periodically to listen to packets sent by the master, which coordinates the medium access with Time Division Multiple Access.



Avertising mode



Connected mode

5.3 Host

Logical Link - Control and Adaptation Protocol (L2CAP) This layers offers a best-effort approach to multiplex data and control signalling of ATT, SM and LL.

Attribute protocol (ATT) ATT layer defines communication between a server and a client and maintains a corresponding set of attributes on the server. Clients access the server's attributes via requests and server responds with either notifications (unconfirmed message) or indications (the client must acknowledge the message). Each attribute has a handler, UUID, value and set of permissions. The transmissions follow the stop-and-wait scheme.

Generic Attribute Protocol (GATT) GATT layer creates frameworks to discover services and exchange profiles (characteristics of a device). A characteristic is a set of data composed of a value, properties (read, write, notify, etc.) and a descriptor (user description, enabling notifications, presentation format, unit), while a service is a set of characteristics. Each data is related to services and characteristics stored in attributes.

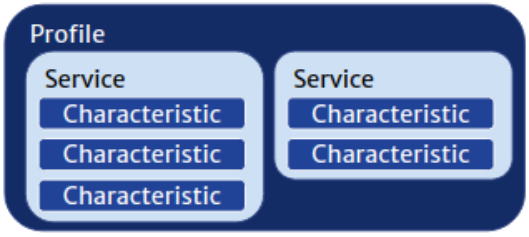


Figure 19: Structure of a BLE device profile

Generic Access Profile (GAP) This layer specifies the role of a device among:

- Broadcaster: broadcasts data via advertising channels (no connection required with other devices).
- Observer: receives data transmitted by the broadcaster.
- Central: initiates and manage multiple connections. Mostly, a central device is a client.
- Peripheral: single connection to a central device. Mostly, a peripheral device is a server.

Security Manager (SM) The Security Manager handles security protocols and defines the pairing mechanism and the method of key negotiation to use. Three phases are needed: exchange of physical characteristics, pairing process and key distribution.

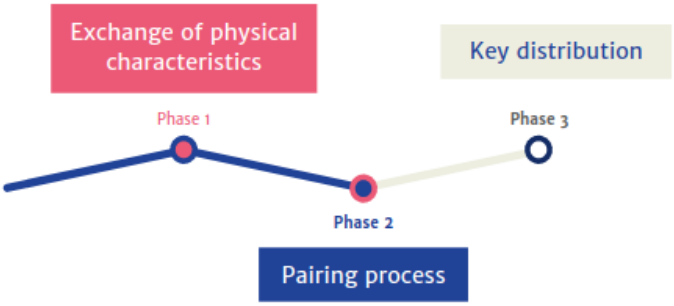


Figure 20: Phases in SM layer

6 BLE security mechanisms

6.1 Standards for protocols

Several security mechanisms are already implemented by default in BLE technology, such as the frequency hopping which avoids interferences with other wireless technologies using similar frequencies. Moreover, BLE fulfills the IEEE standard 802.15.4 for Bluetooth Wireless Technology. Security processes are recommended in NIST 800-121-R1.

6.2 Security modes

Two LE security modes with several levels of security are defined to encrypt and sign data.

LE security mode 1 (encryption)

- Level 1: no security
- Level 2: service level enforced security (authentication, confidentiality, authorization)
- Level 3: link level enforced security
- Level 4: service level enforced security with encrypted key exchange

Each security level shall include the lower level [12] (e.g. security mode 1 level 3 also satisfies level 2 of mode 1). Moreover, security mode 1 levels 3 and 4 shall satisfy security mode 2.

LE security mode 2 (data signing)

- Level 1: unauthenticated pairing with data signing
- Level 2: authenticated pairing with data signing

Security mode 2 is used for connection based data signing and cannot be combined with security mode 1 levels 2, 3 or 4. However, if both security modes are required, there are solutions depending on the security features needed (e.g "if there are requirements for both LE security mode 1 and LE security mode 2 level 2 for a given physical link then LE security mode 1 level 3 shall be used." [12]).

Secure Connections Only Mode LE security mode 1 level 4 might be used for Secure Connections Only, so that "the device shall only accept new outgoing and incoming service level connections for services that require Security Mode 1, Level 4 when the remote device supports LE Secure Connections and authenticated pairing is used" [12]).

6.3 Security manager

6.3.1 Pairing process and bonding

The security manager defines the pairing process during which devices exchange device information to establish a secure link to communicate. BLE versions 4.0 and 4.1 uses Legacy Pairing while version 4.2 and older versions use Secure Connection. In this report, we will only consider BLE version 4.2 and later as lower versions suffer from Temporary Key brute-force attack [13].

The pairing process has three main phases [14] [15]:

1. Pairing feature exchange: both devices exchange (with no encryption) their input and output capabilities in order to select the most suitable Short Term generation key method used in phase 2 [16].
2. Short Term Key (STK) generation and authentication: a Temporary Key (TK) is generated and exchanged using a pairing method to generate the STK. The STK encrypts the connec- tion and the authentication aims to protect against Man-in-The Middle (MITM) attacks (cf. section 8.3).
3. Link encryption, Long Term Key (LTK): after authentication, devices compute the LTK to encrypt the communication link.

An optional phase consists in exchanging transport keys parameters regarding bonding. Bonding procedure can be used to store security keys and information exchanged during pairing process for later connections between a central and peripheral devices. It avoids repeating the entire pairing process every time the device needs to be connected.

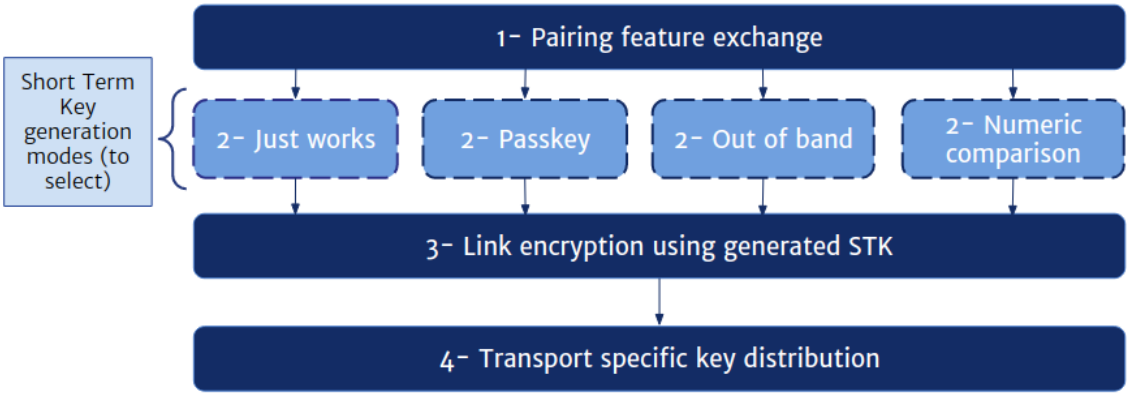


Figure 21: BLE pairing process

There are four pairing process for Bluetooth LE Secure Connections.

Just works It is implemented by default, used with devices unable to enter or display 6 digits.

Passkey One device must have a 6-digits output capability to display the key and the other requires a 6-digits input to enter the displayed key, randomly generated [17]. It provides protection against passive attacks and MITM attacks [18].

Out Of Bands (OOB) This method is dedicated to devices using interfaces other than Bluetooth. It provides protection against passive attacks and MITM attacks.

Numeric comparison Same as just works method but requires two buttons to validate the key. It provides protection against passive attacks and MITM attacks.

The following table details the use of the different modes according to the type of device.

Responder	Initiator					
	I/O Capabilites	DisplayOnly	Display YesNo	Keyboard	NoInput NoOutput	Keyboard Display
	Display Only	**Just Works** **Unauthenticated**	**Just Works** **Unauthenticated**	**Passkey Display** (responder displays, initiator inputs) **Authenticated**	**Just Works** **Unauthenticated**	**Passkey Display** (responder displays, initiator inputs) **Authenticated**
	Display YesNo	**Just Works** **Unauthenticated**	**Just Works** [For LE Legacy Pairing] **Unauthenticated** **Numeric Comparison** (For LE Secure Connections Pairing) **Authenticated**	**Passkey Display** (responder displays, initiator inputs) **Authenticated**	**Just Works** **Unauthenticated**	**Passkey Display** (responder displays, initiator inputs) [For LE Legacy Pairing] **Authenticated** **Numeric Comparison** (For LE Secure Connections Pairing) **Authenticated**
	Keyboard Only	**Passkey Display** (initiator displays, responder inputs) **Authenticated**	**Passkey Display** (initiator displays, responder inputs) **Authenticated**	**Passkey Display** (initiator displays, responder inputs) **Authenticated**	**Just Works** **Unauthenticated**	**Passkey Display** (initiator displays, responder inputs) **Authenticated**
	NoInput No Output	**Just Works** **Unauthenticated**	**Just Works** **Unauthenticated**	**Just Works** **Unauthenticated**	**Just Works** **Unauthenticated**	**Just Works** **Unauthenticated**
	Keyboard Display	**Passkey Display** (initiator displays, responder inputs) **Authenticated**	**Passkey Display** [For LE Legacy Pairing] (initiator displays, responder inputs) **Authenticated** **Numeric Comparison** (For LE Secure Connections Pairing) **Authenticated**	**Passkey Display** (responder displays, initiator inputs) **Authenticated**	**Just Works** **Unauthenticated**	**Passkey Display** [For LE Legacy Pairing] (initiator displays, responder inputs) **Authenticated** **Numeric Comparison** (For LE Secure Connections Pairing) **Authenticated**

Figure 22: Pairing process methods based on BLE devices I/O capabilities and roles [19]

6.3.2 Encryption

BLE communications are encrypted using either the E0 encryption (4.0, 4.1 and 4.2 devices) or a AES-128 Cipher Block Chaining-Message Authentication Code (CCM) algorithm (4.2 devices and above). AES-CMM uses a 128 bit key length, generated with the Elliptic Curve Diffie Hellman (ECDH) method [16]. Connections using encryption and authentication use a Message Integrity Protection (MIC), that is appended to the payload of the data packet and a Cyclic Redundancy Check (CRC) mechanism protects it all.

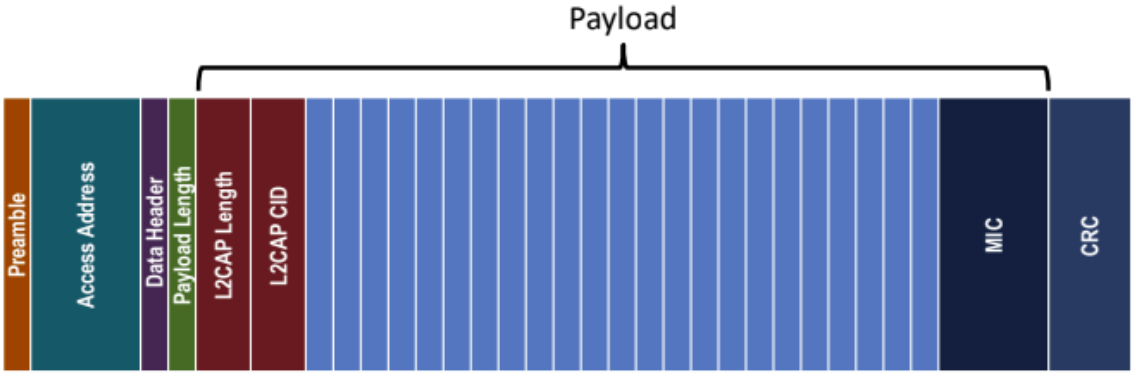


Figure 23: BLE payload encryption [20]

Transmissions of authenticated data over an unencrypted Link Layer connection use a 12-byte signature, placed after the data payload at the ATT layer and computed with an 128-bit AES algorithm [16]. A counter is required as input to prevent from replay attacks.

6.3.3 Privacy feature

A privacy feature is supported to limit the identity tracking of a device: its address is private and changes frequently [12] [16]. Private addresses are generated via encryption of the public address of the device.

6.3.4 Trust mode

Trust modes define the trust level of a relationship. A device "trusted" allows a fixed relationship and unrestricted access to all its services, while an "untrusted" device restricts access to a set of services. This feature limits automatic access to services when a device is untrusted although authentication succeed. Nevertheless, distrusting a device removes its bonding information [18].

7 Vulnerabilities in BLE protocol

7.1 Pairing process

Although the Temporary Key is not transmitted through packet, it is a 16 bytes input value which is predictable. Just works method pre-defines its value to 0x00. Passkey generation mode transmits STK generation parameters in packets which could enable an attacker to calculate the STK and decrypt data. Also, just works is vulnerable to MITM attacks because the user cannot verify the authenticity of the connection [21].

7.2 Discoverability

BLE has a discoverability mode, used before pairing devices. It is advised that "devices should be set to undiscoverable by default except as needed for pairing, to prevent visibility to other Bluetooth devices" [22]. A discoverable device is vulnerable because it allows other devices to access information in a 10 meter range, such as its name, class and services. Turning off the discoverability mode prevents devices to responding to scanning attacks [18].

8 Common BLE attacks and security analysis

8.1 Classification of attacks

Bluetooth technology is vulnerable to many attacks, which can be classified by penetration method or by the impact of the attack (figure 24) [23].

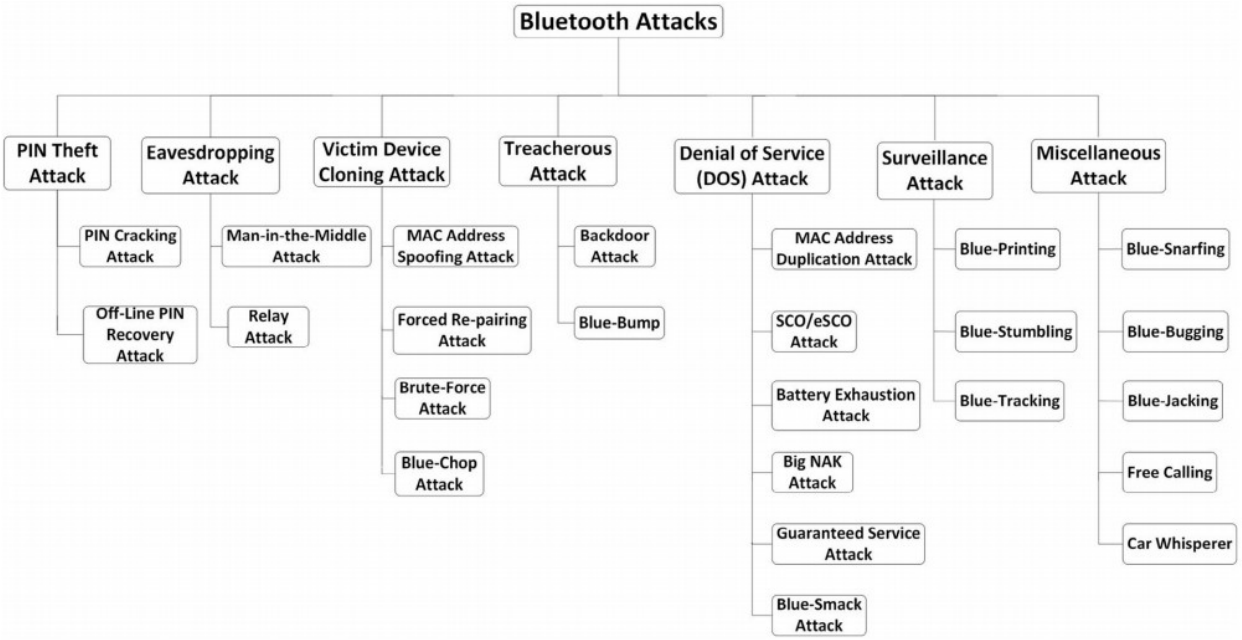


Figure 24: Classification of Bluetooth attacks [23]

In this report we will focus on Eavesdropping attack and Denial of Services (DoS) attack categories, which are the most common attacks of Bluetooth Low Energy.

8.2 Passive approaches

Passive approaches are mostly passive eavesdropping such as sniffing. An attacker can sniff BLE communications during different stages: new connections, active connections and negotiation phase. However, sniffing BLE communication is complex as there are 40 channels used and the change of frequency is fast, though it is an expensive attack.

New connections The goal is to sniff communications from the establishment of the connection, to capture the CONNECT_REQ packet. This packet contains several parameters to set the fre- quency hopping algorithm (AA, Interval, ChM, Hop) and CRC calculation (CRCInit). AA is the Access Address, Interval specifies the time spent on each channel, ChM is the channel map and Hop is the increment value used for channel hopping [24].

LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Figure 25: BLE CONNECT_REQ packet (Link Layer data)

Knowing these parameters, the attacker uses them to set up its algorithm to listen on one advertising channel. The success rate of this attack is $\frac{1}{3}$ as there are three advertising channels and it is expensive to sniff all of them at once [11].

Active connections The attacker deduces connection parameters (CRCInit, Interval, etc.) from an existing communication [25]. This is an exhaustive approach because it assumes all channels are systematically used and it is not effective on short term communications as it requires time [11].

Negociation phase this attack occurs when the attacker sniffs packets transmitted during the SM negotiation phase at the beginning of communications. The attacker gets the encryption key to decrypt communications [11].

8.3 Actives approaches

BLE is vulnerable to active attacks such as radio jamming, active eavesdropping and DoS.

Radio jamming The use of a strong radio signal near the BLE devices might cause interferences to jam communications [26]. Attackers can jam connection and advertising transmissions by saturing the radio spectrum, interrupting a master-slave connection or hijacking a connection by forcing the master to disconnect itself. Preventing radio jamming is hard as it requires physical protection from interference.

MITM MITM attacks occur when an attacker intercepts communications between two devices and modifies it. Some attacks consists in cloning the GATT server to simulate an identical device to which the master will be connected. It allows the fake device to connect to the legitimate device to capture the traffic, impersonate a device, inject data, modify or redirect packets and provoke DoS. Theses attacks are easy to implement as they only require two BLE adaptors and there are no encryption issues because the attacker negotiates encryption parameters.

BLE is also vulnerable to replay attack, relay attack and spoofing like LoRa protocol (cf. section 4).

8.4 Audit tools

Many audit tools exist to assess the resistance to attacks of installations [11]:

- BLEAH: information collection, exhaustive scan of GATT server services and characteristics
- Crackle: exploits flaws in pairing process, guess and brute force TK, STK, LTK to decrypt data
- BTLEJuice: man in the middle attacks
- BTLEJack: sniff, jam, hijack
- GATTacker: man in the middle, DoS, spoofing
- Mirage: collection of data, hijacking, sniff, jam, man in themiddle

Part III
Security aspects of LoRa and BLE specific to the ANR McBIM project

In this part, we will study the security issues of the LoRa and BLE protocols specific to the ANR McBIM project. This project is a communicating reinforced concrete project [27], supported by a consortium of three French laboratories and a company. The LAAS-CNRS mission's is to provide a Wireless Sensor Network (WSN) devoted to be embedded in reinforced concrete, in order to assure structural health monitoring tasks [2] [28].

9 Architecture of a wall made of a communicating reinforced concrete

9.1 Deployment infrastructure for connected walls

A McBIM ANR wall is composed of concrete beams with:

- several Sensing Nodes (SN) embedded in the beam;
- at least one Communicating Node (CN) affixed on the external surface of the beam.

The reinforced concrete communicates with the Internet to send the data collected to applications to monitor the structure.

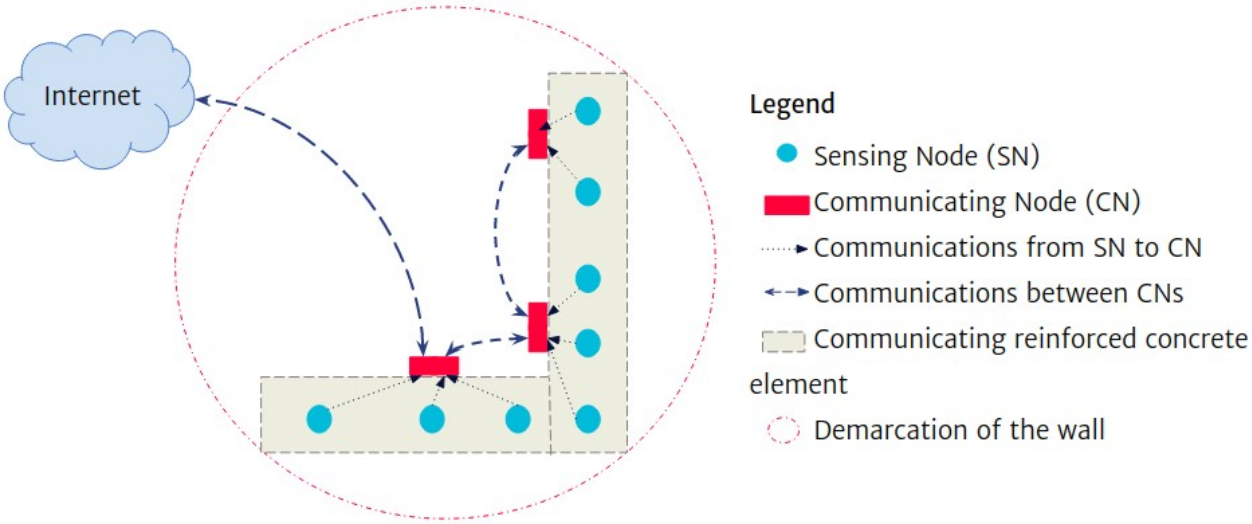


Figure 26: Composition of a wall made of communicating reinforced concrete elements

9.2 Behavior of communicating reinforced concrete elements

Sensing Node (SN) The SN is embedded in reinforced concrete. It is a low cost micro-controller dedicated to measure parameters such as humidity, temperature or mechanical stress. Sensing Nodes are designed to be battery-free using energy harvesting through a Wireless Power Transmission from a Communicating Node. Once the SN has enough energy to send its measures, it transmits the data to a CN. The frequency of data transmission depends on the RF power received. Data transmissions are unidirectional and two communicating protocols are considered: LoRa and BLE.

Communicating Node (CN) Communicating Nodes require to be on the external surface of the beam to be accessible for maintenance. They are connected to a RF power source (either a wired electric grid or a high capacitance battery) and wirelessly supply Sensing Nodes through far-field RF WPT. A CN collects, processes and stores data sent by the SN. Communicating Nodes are able to communicate with each other to send data to a particular CN (gateway) which will transmit all all data through the Internet. CN communications are bi-directional and supported either by LoRa or BLE.

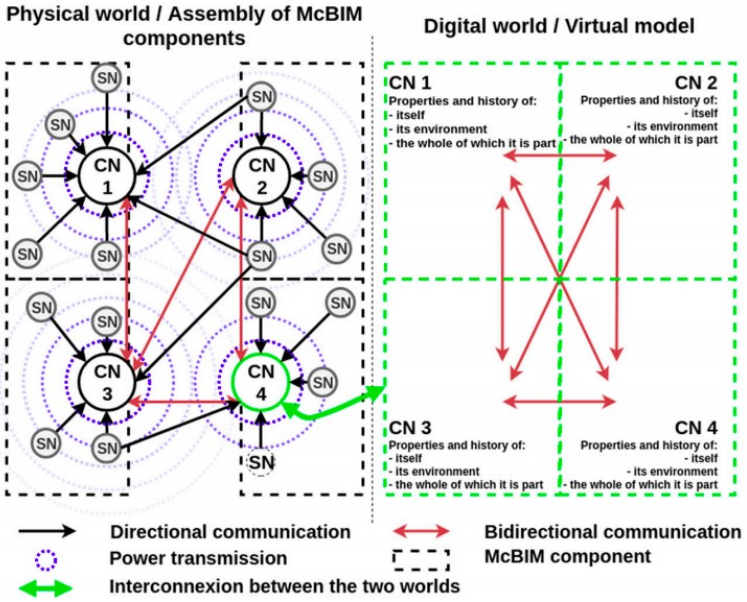


Figure 27: Schematic diagram of the architecture of the wireless smart-nodes mesh network [2]

The communications between SN-CN, CN-CN and CN-Internet raise security issues. The importance of this issue depends on the type of data transmitted, their criticality, etc. LAAS-CNRS is in charge of studying the security of communications of the Wireless Sensor Network at the physical/hardware level (some securities could be studied for the software level).

10 Malevolent goals

Three types of malevolent goals have been identified: the invasion of privacy, service alteration and service interruption.

10.1 Invasion of privacy

Invasion of privacy consists in gathering information on building activities (e.g. passive eavesdropping). This could be realized by the infrastructure owner to acknowledge, for instance, the movements or activities of the employees in the building. An outsider of the structure might also gather information on activities to identify the best moment to rob the building or to collect clas- sified information (such as the type of product, vibration, etc.).

10.2 Service alteration

Services delivered by the structure can be altered by falsifying the parameters to measure (man-in-the middle, relay attack, replay attack) or by modifying packet transmissions. As an example, an attacker could causes vibrations to make people believe to a deformation of the structure.

10.3 Service interruption

Services can be interrupted by stopping communications via Denial of Services or by wiping out communications through radio jamming or through CN battery exhaustion (thus, attack on SN autonomy).

11 Threat model

The threat model is based on two ranges attack.

11.1 Short range attack

It provides physical access by either being inside the building or outside the building but near enough to place objects (malicious nodes, gateway, etc.).

11.2 Long range attack

The maximum range depends on the communication technology, transmission power and the type of communication (continuous wave, chirp, pulse). The attacker is able to communicate with nodes or emit a powerful radio signal to jam communications.

12 Risks

12.1 Risk scale

The following probabilities and impacts of the risks relies on a personal estimate, based on scientific articles [18] [29] [30]. The impact of attacks depends on the potential harm the attack can inflict [6] and the human impact (benign injuries to death, due to failure of failure detection).

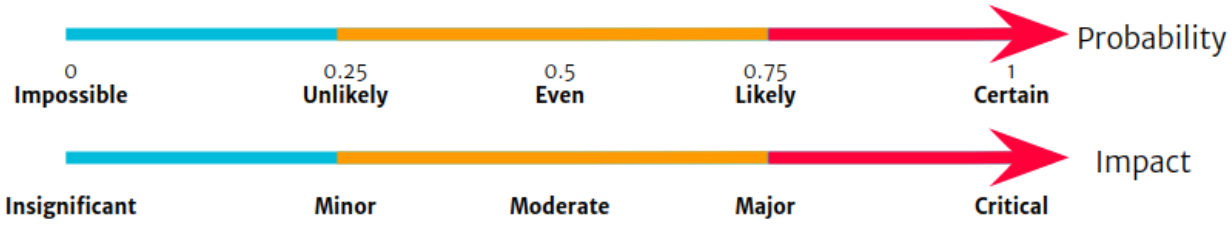


Figure 28: Risk scale

12.2 Invasion of privacy

Invasion of privacy implies several risks such as surveillance, the insertion of a malicious node into the network, the insertion of fake data by a malicious node and the compromise of nodes (see figure 29).

Risk	Surveillance	Insertion of a malicious node into the network	Insertion of fake data by a malicious node	Compromise of nodes
Probability	**Likely**	▪ SN embedded in concrete: **unlikely** (SN inaccessible) ▪ SN not embedded : **likely** ▪ CN: **even**	**Likely**	▪ SN: **unlikely** (less interesting than compromising a CN) ▪ CN: **likely**
Impact	**Insignificant** to **critical**: depends on the activity	▪ SN embedded in concrete: **insignificant** ▪ SN not embedded: **minor** ▪ CN*: **major** to **critical**	**Moderate** to **major**	▪ SN: **insignificant** ▪ CN*: **major** to **critical**

** Greater impact if the CN is a gateway*

Figure 29: Risks implied by invasion of privacy

12.3 Service alteration

Service alteration represents a waste of time and loss of money for mobilizing employees to check the structure. Risks are either deduction of the building activity or the alteration of data (see figure 30).

Risk	Building activity deduction	Alteration of data
Probability	**Likely**: depends on the security mechanism implemented (traffic is often encrypted)	**Even:** depends on the security mechanism implemented (bypassing authentication)
Impact	**Minor** to **critical**: depends on the activity (e.g. critical in a nuclear environment)	**Moderate** to **critical** (e.g. triggering of a false bridge collapse alert by modifying packets, falsifying mechanical stress by causing vibration of the wall)

Figure 30: Risks implied by service alteration

12.4 Service interruption

Services interruption also represents a loss of time and money as services provided by the structure are no longer available. The main risks in the McBIM project are radio jamming, attacks on node autonomy, creation of relay or cycles, damaging the rectenna, communication with nodes to get information and altering data (see tables 31 and 32).

Risk	Jamming communications [9]	Attack on node autonomy	Damaging the rectenna
Probability	▪ SN–CN: **likely** ▪ CN–CN: **likely** ▪ CN–gateway: **likely**	▪ SN energy autonomy: **impossible** (WPT) ▪ Destruction of SN components (by antenna, mechanical break, intense electromagnetic wave, wear): **unlikely** ▪ CN*: **even**	**Unlikely**
Impact	▪ SN–CN: depends of the number of SNs affected (**insignificant** to **critical**) ▪ CN–CN: **moderate** to **major** ▪ CN–gateway: **critical**	▪ SN: **insignificant** to **critical** (depends on the number of SNs affected and the total quantity of SNs) ▪ CN*: **major** to **critical** (depends on the number of CNs affected and the total quantity of CNs)	**Critical**

** Greater impact if the CN is a gateway*

Figure 31: Risks implied by service interruption (1)

Risk	Creation of a relay	Communication with nodes to get information	Making believe to a deformation / dysfunction of the structure (physical data alteration)	Creation of cycles
Probability	▪ Embedded SN: **unlikely** ▪ SN not embedded: **likely** ▪ CN: **likely**	**Even**	**Unlikely**	**Likely**
Impact	▪ Embedded SN: **unlikely** ▪ SN not embedded: **likely** ▪ CN*: **major** to **critical** (if a node fails, the attacker might take his place or create a loop)	**Minor** to **critical**: depends on the type of data stored	**Critical**	**Minor** to **Critical**: depends on the activity/type of data

Figure 32: Risks implied by service interruption (2)

13 Technical solutions

13.1 Cryptography

Communications can be secured with cryptography for authentication and data encryption by using the options offered by BLE (AES-128 CCM mode, Elliptic Curve Diffie-Hellman [25]) and LoRa (AES CTR mode [5] [7]). It is also possible to add another level of cryptography. Note that it is strongly advised not to use your own cryptography algorithm.

Adding cryptography is a flexible solution, easy to implement, but it is computationally expensive (especially public key cryptography) and it requires secrets to be stored in a safe and non-volatile manner. Moreover, it is power consuming, generating low latency (increase of operating time) and poorly suited to low energy devices (reduces SN autonomy).

13.2 Secure Elements (SE)

Secure Elements [31] might be an alternative to secure the Wireless Sensor Network. A SE is a tamper-resistant hardware platform (embedded chip) used to secure the storage of confidential and cryptographic data, host securely applications and implement end-to-end security. Resistive Random Access Memory Physical Unclonable Functions (RRAM PUF) might be implemented to manage authentication, key generation and storage. A Secure Element is cheap (around 5 euros a chip) and it consumes less energy than using cryptography (2mA to 18mA [32] [33]). However, it requires device driver software and integration of the capabilities of the secure element into existing software applications. In addition, bus link protection recommended [34].

13.3 Intrusion Detection System (IDS)

And Intrusion Detection System could be combined with Secure Elements to secure the network. It is based on detection methods: signature-based or anomaly-based. Signature-based is not adapted for IoT devices yet. Anomaly-based IDS uses learning systems to identify legitimates behaviours and detect suspicious behaviours. This could be used during two phases of ANR McBIM project:

- manufacturing and storage;
- construction of the complete structure.

IDS provides visibility on the network and adds a layer of defense. Nevertheless, it requires maintenance and is sensitive to false positives and negatives.

13.4 LoRa and BLE security features

13.4.1 LoRaWAN v1.0.2

We chose to study LoRaWAN v1.0.2 security mechanisms. Though, more security mechanisms are provided in LoRawan versions 1.1 and 1.0.3 but these versions do not fit best with the ANR McBIM project [35]. Figure 33 gathers all optional securities provided by LoRaWAN v1.0.2, the attacks prevented and the disadvantages of the use of the solution.

Note that security mechanisms prevent attacks but there are still risks.

Optional security mechanisms[1]	OTAA (join procedure) [4] [7]	Frame counter [4] [7]	Message ACK [4] [7]
Attacks prevented	Replay attack [8] [9]	Replay attack [8] [9]	Replay attack, ACK spoofing [8]
Consequences of a successful attack	▪ Connection of a malicious device to the Network Server ▪ Exchange of data, etc.	(Re)use of valid messages to connect a device to the server, exchange data, etc.	(Re)use of valid messages to connect a device to the server, exchange data, etc. due to lack of association between request messages and acknowledgments
Disadvantages of the security	▪ Risk of replay attacks reduced but still possible due to reset of frame counter value to 0 (though, an implementable solution exists) ▪ Increase latency and energy consumption	▪ Decreases the availability of the device ▪ Frame counter reboot or overflow enable replay attack[3]	▪ No association between data messages and acknowledgements* ▪ Increase latency and energy consumption

Issue resolved in later versions of LoRaWAN

Figure 33: LoRaWAN v1.0.2 security issues and protection mechanisms

13.4.2 BLE

The use of BLE versions 4.2 and above is recommended as BLE 4.0 and 4.1 suffer from TK brute-force attack. Figures 34, 35, 36 and 37 summarize BLE security mechanisms, the attacks they prevent from and their disadvantages.

Optional security mechanisms	Security Mode 1 (encryption) [9]			
	Level 1	Level 2 (unauthenticated pairing with encryption)	Level 3 (authenticated pairing with encryption)	Level 4 (authenticated low energy Secure Connections pairing with encryption)
Attacks prevented	None	Limited eavesdropping protection	Eavesdropping, replay attack	Eavesdropping, replay attack, man-in-the middle
Consequences of a successful attack	▪ Decryption of data ▪ Traffic observation/injection ▪ DoS (stopping communications)			
Disadvantages of the security	N/A	Encrypted link required	MITM protected encrypted link required	LE Secure Connections MITM protected encrypted link required

Figure 34: BLE security issues and protection mechanisms (1)

Optional security mechanisms	Security Mode 2 (data signing)* [3]	
	Level 1 (unauthenticated pairing with data signing)	Level 2 (authenticated pairing with data signing)
Attacks prevented	N/A	Eavesdropping, replay attack
Consequences of a successful attack	▪ Traffic observation (no encryption) ▪ Traffic injection	
Disadvantages of the security	▪ Cannot be combined with Security Mode 1 level 2,3,4** [12] ▪ Connection based data signing ▪ Signing or encryption required	▪ Cannot be combined with Security Mode 1 level 2, 3, 4** [12] ▪ Connection based data signing ▪ MITM protected signing required, unless link is MITM protected encrypted

Data signing shall not be used when a connection is operating in LE security mode 1 level 2, 3 or 4
***More information about mixed modes in Bluetooth core specifications 4.2, p373*

Figure 35: BLE security issues and protection mechanisms (2)

Optional security mechanisms	Pairing process – STK generation mode [5] [20] [21] [22]			
	Just works	Passkey	Out Of Band	Numeric comparison
Attacks prevented	Passive attack if ECHD is used	Passive attack if ECHD is used, active man-in-the middle	Passive attacks, man-in-the middle	Passive attacks, man-in-the middle
Consequences of a successful attack	▪ Impersonate devices ▪ Decryption of data ▪ Traffic observation/injection ▪ DoS (stopping communications)			
Disadvantages of the security	Less secure, used when one device cannot display/enter 6 digits	One device must have the input capability, the other needs the output capability	Both devices use interfaces other than Bluetooth, they must be compatible	Two buttons required (yes/no)

Figure 36: BLE security issues and protection mechanisms (3)

Optional security mechanisms	Discoverable mode off [18]	Trust mode "device untrusted" [18]	Privacy feature (regular change of private address) [12]
Attacks prevented	Prevents from accessing information such as name, class, services, etc.	Limits automatic access to services	Identity tracking
Consequences of a successful attack	Steal of sensitive data	Trust mode on: a trusted node that has just been compromised enables access to all services to an attacker	Steal of sensitive data
Disadvantages of the security	No data transmission allowed	Removes bonding information	▪ Available only with connected mode ▪ Only a trusted device can be connected

Figure 37: BLE security issues and protection mechanisms (4)

References

[1] Semtech. (2020) What are lora and lorawan? [Online]. Available: https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/

[2] G. Loubet, A. Takacs, E. Gardner, A. De Luca, F. Udrea, and D. Dragomirescu, "Lorawan battery-free wireless sensors network designed for structural health monitoring in the construc- tion domain," *Sensors*, vol. 19, p. 1510, 032019.

[3] LoRa Alliance. (2015) A technical overview of lora and lorawan. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf

[4] C. E. Fehri, M. Kassab, S. Abdellatif, P. Berthou, and A. Belghith, " LoRa technology MAC layer operations and research issues," vol. 130, pp. 1096–1101. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050918305283

[5] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of LoRaWAN," vol. 148, pp. 328–339. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618306145

[6] K. Tsai, F. Leu, L. Hung, and C. Ko, "Secure session key generation method for lorawan servers," *IEEE Access*, vol. 8, pp. 54631–54640, 2020.

[7] X. Yang, " LoRaWAN: Vulnerability analysis and practical expoi- tation." [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3A87730790-6166-4424-9d82-8fe815733f1e

[8] T. C. M. Dönmez and E. Nigussie, "Security of LoRaWAN v1.1 in backward compatibility scenarios," vol. 134, pp. 51–58. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918311062

[9] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the security vulnerabilities of LoRa," in *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, pp. 1–6.

[10] LoRa Alliance. (2020) Lorawan specification v1.0.3. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v103

[11] R. Cayre, V. Nicomette, G. Auriol, E. Alata, M. Kaaniche, and G. Marconato, "Mirage: To- wards a metasploit-like framework for iot," 10 2019, pp. 261–270.

[12] Bluetooth SIG. (2020) Ble core specifications. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/

[13] Y. Zhang, J. Weng, R. Dey, and X. Fu, *Bluetooth Low Energy (BLE) Security and Privacy*. Cham: Springer International Publishing, 2019, pp. 1–12. [Online]. Available: https://doi.org/10.1007/978-3-319-32903-1_298-1

[14] G. S. Pallavi and A. Narayanan V, "An overview of practical attacks on ble based iot devices and their security," 2019, pp. 694–698.

[15] K. Ren. (2017, 01) Bluetooth pairing part 4: Bluetooth low energy secure connections – numeric comparison. [Online]. Available: https://www.bluetooth.com/blog/bluetooth-pairing-part-4/

[16] C. Gomez, J. Oller Bosch, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors (Basel, Switzerland)*, vol. 12, pp. 11734– 53, 12 2012.

[17] F. J. Dian, A. Yousefi, and S. Lim, "A practical study on bluetooth low energy (ble) through- put," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communica- tion Conference (IEMCON)*, 2018, pp. 768–771.

[18] A. Lonzetta, P. Cope, J. Campbell, B. Mohd, and T. Hayajneh, "Security vulnerabilities in bluetooth technology as used in iot," *Journal of Sensor and Actuator Networks*, vol. 7, p. 28, 07 2018.

[19] Microchip. (2020) Bluetooth low energy security modes and procedures. [Online]. Available: https://microchipdeveloper.com/wireless:ble-gap-security

[20] R. Heydon, "Bluetooth mesh." [Online]. Available: https://www.cl.cam.ac.uk/teaching/1819/MobSensSys/mobile-11.pdf

[21] G. Kwon, J. Kim, J. Noh, and S. Cho, "Bluetooth low energy security vulnerability and improvement method," in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2016, pp. 1–4.

[22] P. Cope, J. Campbell, and T. Hayajneh, "An investigation of bluetooth security vulnerabilities," 01 2017, pp. 1–7.

[23] S. Hassan, S. Bibon, M. S. Hossain, and M. Atiquzzaman, "Security threats in bluetooth technology," *Computers Security*, vol. 74, 03 2017.

[24] D. Cauquil, "Bluetooth low energy attacks, a crash course into bluetooth low energy attacks and associated counter-measures." [Online]. Available: https://nis-summer-school.enisa.europa.eu/2018/cources/IOT/nis-summer-school-damien-cauquil-BLE-workshop.pdf

[25] M. Ryan, "Bluetooth: With low energy comes low security," in *7th USENIX Workshop on Offensive Technologies (WOOT 13)*. Washington, D.C.: USENIX Association, Aug. 2013. [Online]. Available: https://www.usenix.org/conference/woot13/workshop-program/presentation/ryan

[26] S. Bräuer, A. Zubow, M. Roshandel, S. Mashhadi Sohi, and S. Zehl, "On practical selective jamming of bluetooth low energy advertising," 10 2016.

[27] G. Loubet, A. Takacs, and D. Dragomirescu, "Towards the design of wireless communicating reinforced concrete," *IEEE Access*, vol. 6, pp. 75 002–75 014, 2018.

[28] G. Loubet, A. Takacs, and D. Dragomirescu, "Implementation of a battery-free wireless sen- sor for cyber-physical systems dedicated to structural health monitoring applications," *IEEE Access*, vol. PP, pp. 1–1, 02 2019.

[29] S. Sandhya and K. A. S. Devi, "Analysis of bluetooth threats and v4.0 security features," in *2012 International Conference on Computing, Communication and Applications*, 2012, pp. 1–4.

[30] J. P. Dunning, "Taming the blue beast: A survey of bluetooth based threats," *IEEE Security Privacy*, vol. 8, pp. 20–27, 2010.

[31] T. Schläpfer and A. Rüst, "Security on iot devices with secure elements," in *Embedded World Conference 2019 - Proceedings*, Nuremberg, Germany, February 2019.

[32] Infineon. (2020) Optiga trust x sls 32aia. [Online]. Available: https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/optiga-trust/optiga-trust-x-sls-32aia/

[33] AVNET SILICA. (2020) Trusted objects to136. [Online]. Available: https://www.avnet.com/wps/portal/silica/products/product-highlights/2016/ot-morpho-trusted-objects-to136/

[34] Cerberus Security Laboratories. (2019) What is an iot hardware secure element? [Online]. Available: https://cerberus-laboratories.com/blog/iot_hsms/

[35] I. Butun, N. Pereira, and M. Gidlund, "Analysis of LoRaWAN v1.1 security: research paper," in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects - SMARTOBJECTS '18*. ACM Press, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3213299.3213304