Cybersecurity in LoRaWAN Networks: Vulnerability Analysis and Enhancing Security Measures for IoT Connectivity

Università di Genova

Junaid Qadir

Department of Electrical, Electronic and Telecommunication Engineering and Naval Architecture (DITEN)

University of Genoa

Supervisor

Prof. Daniele D. Caviglia,
Prof. Paolo Gastaldo & Dr. Ismail Butun

September 20, 2023

Acknowledgements

I would like to extend my heartfelt appreciation to my dedicated supervisors, **Prof. Daniele D. Caviglia**, **Prof. Paolo Gastaldo**, and Ph.D coordinator **Prof. Muarizio Valle** for their unwavering support, guidance, and boundless patience throughout this transformative journey.

Prof. Caviglia, your belief in my potential and unwavering confidence in my abilities have pushed me beyond my limits, and I am truly thankful for that. Your ability to spark creative thinking, encourage unconventional exploration, and foster a forward-looking perspective has had a lasting impact on my academic and personal growth. My accomplishments serve as proof of your excellent mentorship.

I would also like to extend a special note of appreciation to **Dr. Ismail Butun**. Your invitation to collaborate on Ph.D. research at the **KTH Royal Institute of Technology**, Stockholm, Sweden has greatly enhanced the depth and quality of my work.

Finally, I extend my sincere thanks to the **Università degli Studi di Genova** for graciously granting me the opportunity to conduct my research within its esteemed institution. The assistance and resources provided have been invaluable in shaping the course of this work.

Junaid Qadir

Genova, November, 2023

Dedicated to my parents, siblings, and spouse, whose continuous provision of opportunities has ensured an unimpeded trajectory throughout my academic journey, leaving no room for any semblance of deficiency.

Abstract

The realm of Low Power Wide Area Network (LPWAN) has a paramount influence on the way we work and live. For instance, real-time applications and rapid packet transiting for long-range have now come into practice that was previously considered mysterious. However, euphoria becomes a problem when it comes to security considerations, as low-power devices possess limited processing units that are unable to elucidate robust security algorithms. In this case, the Low Power Wide Area Network (LoRaWAN) stepped into a technological competition that filled the gap with adopting the end-to-end security feature. However, several problems have been pinpointed in the newer version such as one issue with key distribution in LoRaWAN 1.1 is that the keys are often pre-installed on the devices at the time of manufacturing. It can introduce security risks if the keys are not adequately protected or if the devices are compromised before they are deployed. In other words, the pre-installed keys may not be updated regularly, which can also introduce security risks. Thus, the keys need to be handled securely to maintain the security of the network and the over-the-air firmware updates feature could introduce new security challenges for the key distribution. This thesis presents a key generation and distribution (KGD) mechanism that securely exchanges the root key between the *ED* and the application server (*AS*). The KGD protocol provides authentication by integrating Advanced Encryption

Standard (AES-128) in addition to a secure hash function known as *Argon2*. The proposed protocol utilizes Elliptic-Curve Diffie-Hellman (ECDH) key exchange method that makes the protocol resilient to cyber threats. The ECDH algorithm exchanges the keys on the insecure channels and is, therefore, vulnerable to Man-in-the-Middle (MITM) attacks in the network. Therefore, to validate the key agreement and avoid adversaries, the KGD protocol considers the Elliptic Curve Digital Signature Algorithm (ECDSA) that authenticates and allows legitimate instances in the network. In last, a formal security analysis using the Scyther tool validates the security enhancement of the KGD protocol.

Contents

List of Symbols and Acronyms

The following list describes several symbols and acronyms that will be later used within the main text of this thesis

$\mathbb{I}$	Identity element
ED_{ith}	The number of *EDs*
ED_{pub}	Public key of the *ED*
ED_{sec}	Secrete/private key of the *ED*
ED_{sh}	Shared key of the *ED*
G	Generator point
h	Hashed message
JS_{pub}	Public key of the *JS*
JS_{sec}	Secrete/private key of the *JS*
JS_{sh}	Public key of the *JS*
k	Random bit generated
mod	Modulus
ABP	Activation by Personalization
ADR	Adaptive Data Rate

AES Advance Encryption Standard

AppKey Application Key

AppSKey Application session Key

AS Application Server

CBC Cipher Block Chaining

CR Coding Rate

CRC Cyclic Redundancy Check

CSS Chirp Spread Spectrum

CTR Counter mode

DevAddr Device Address

DevEUD Device Unique identifier

DH Diffie Hellman

DoS Denial of Service

EC Elliptic Curve

ECDH Elliptic Curve Diffie Hellman

ECDSA Elliptic Curve Digital Signature Algorithm

ED End Device

FCC Forward Error Correction

FNwkSIntKey Forwarding Network Session Integrity Key

FSK Frequency Shift Keying

HSM Hardware Security Module

IoT Internet of Things

ISM Idustrial, Scientific, and Medical

IV Initialization Vector

JS Join Server

JSEncKey Join Server Encryption Key

JSIntKey Join Server Integrity Key

KGD Key Generation and Distribution

LoRaWAN Long Range Wide Area Network

LPWAN Low Power Wide Area Network

MAC Medium Access Control

MACs Message Authentication Codes

MIC Message Integrity Code

MITM Man-in-the-Middle

NACK Negative Acknowledgment

NS Network Server

NwkKey Network Key

NwkSEncKey Network Session Encryption Key

OTAA Over-The-Air Activation

PDR Packet Delivery Ratio

SF Spread Factor

SNwkSIntKey Serving Network Session Integrity Key

TTN The Things Network

List of Figures

List of Tables

List of Publications

0.1 Publications during Ph.D.

1. J. Qadir, I. Butun, P. Gastaldo, O. Aiello and D. D. Caviglia, "Mitigating Cyber Attacks in LoRaWAN via Lightweight Secure Key Management Scheme," in IEEE Access, vol. 11, pp. 68301-68315, 2023, doi: 10.1109/ACCESS.2023.3291420.

2. J. Qadir, I. Butun, R. Lagerstrom, P. Gastaldo and D. D. Caviglia, "Towards Smart Sensing Systems: A New Approach to Environmental Monitoring Systems by Using LoRaWAN," 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2022, pp. 176-181, doi: 10.1109/ZINC55034.2022.98 40717.

3. Qadir, J., Butun, I., Gastaldo, P., Caviglia, D.D. (2023). Review of Security Vulnerabilities in LoRaWAN. In: Berta, R., De Gloria, A. (eds) Applications in Electronics Pervading Industry, Environment and Society. ApplePies 2022. Lecture Notes in Electrical Engineering, vol 1036. Springer, Cham. https://doi.org/10.1007/978-3-031-30333-3_33

4. Qadir, J., Cabus, J.E.U., Butun, I., Lagerström, R., Gastaldo, P., Caviglia, D.D. (2023). Analysis of LPWAN: Cyber-Security Vulnerabilities and Privacy Issues in LoRaWAN, Sigfox, and NB-IoT. In: Butun, I., Akyildiz, I.F. (eds) Low-Power Wide-Area Networks: Opportunities, Challenges, Risks and Threats. Springer, Cham. https://doi .org/10.1007/978-3-031-32935-7_5

5. Mohamed, A.; Wang, F.; Butun, I.; Qadir, J.; Lagerström, R.; Gastaldo, P.; Caviglia, D.D. Enhancing Cyber Security of LoRaWAN Gateways under Adversarial Attacks. Sensors 2022, 22, 3498. https://doi.org/10.3390/s22093498

Chapter 1

Introduction

1.1 Trend

Cyber attacks on connected devices and systems can potentially jeopardize the security and privacy of the low-power wide area network (LoRaWAN) communication protocol. As the LoRaWAN's proliferation continues to be widely increasing in the Internet of Things (IoT) applications [3], the risk of cyber attacks targeting the network's CIA triad (i.e., confidentiality, integrity, and availability) is also increasing. LoRaWAN elucidates the security in the network and application layers [4]. On the one hand, the application layer ensures the confidentiality of the end-to-end packet exchange between the end-device *ED* and itself. On the other hand, the network layer security defines the authenticity and integrity of the packet advancement from the *ED* and vice versa. Despite strong cryptographic encryption techniques, there are still void holes in the security of LoRaWAN. As such, a security threat has been pinpointed in [5] in which the authors stressed the conflict of interest between the network server (*NS*) and the *AS*. If not handled appropriately, it would trigger cyber attacks across the entire network.

Numerous research works in the literature revealed that LoRaWAN is susceptible to a number of attacks; as such, the reply attack and its prevention are discussed in [6] and [7]. The bit flipping attack and its countermeasure are presented in [8] in which some parts of the cipher text have been changed without decryption. The possible remedy is to shuffle the octets in the frame payload

could secure the communication between the *ED* and the *NS*. Furthermore, the LoRaWAN specification is also susceptible to denial of service (DoS) attacks. To put it into practice, the researchers in [9] identified three vulnerabilities that help in executing a DoS attack against the *ED* in the LoRaWAN network. In addition, another notorious issue that makes the specification security insufficient, is uncovered in [10] in which the authors analyzed the weakness in a nonce ("a number that used once"). Nonce is used in the join request message and may pose security threats in LoRaWAN's network if generated improperly. With disclosing such attacks, the LoRaWAN specification is therefore required to enhance the security consideration. Doing so, the LoRa Alliance® is persistently improving the security odyssey and released the latest version of the specifications. The newer version LoRaWAN v1.1.x [11] is featured with the advanced security considerations by adding several encryption keys (discussed in Chapter 3 Section 3.2) and a trust-based entity called the join server (*JS*). However, researchers have penetrated every sphere of the LoRaWAN specifications and unveiled that there are still loopholes that could precipitate possible cyber attacks [12; 13]. In view of this, researchers like Hofmann *et al.* from the Deutsche Telekom [14] have stressed, the new architecture is still vulnerable to several security breaches and the *ED* could be manipulated if the attacker's hands on the device's firmware.

Security is getting advanced as the LoRaWAN v1.1.x orchestrates the application key (AppKey) and the network key (NwkKey) to fill the risk gaps in the previous version v1.0.x. However, scientific research has disclosed other security breaches in v1.1. One of the security risks is the use of unrecommended cryptographic modes such as ECB operation in the join accept message [11]. It is important to remark that v.1.1 is prone to availability attacks, as an attacker can observe and interrupt the packet exchange via an RF jamming tool. The victim *ED* is therefore required to resend the join request message after a timeout, if prevented the response, the *ED* gets loose from the network. This attack creates a void hole in the network and thus diminishes the network's availability. Several problems in [15] are differentiated into three categories such as minor, major, and critical attacks in which the re-keying issue is highlighted as a critical risk.

Cyberattacks on the LoRaWAN end may not be challenging, because it does have a number of vulnerabilities that might be exploited if the network's encryp-

tion keys were compromised. In contrast to the previous version, the *ED* in the LoRaWAN v1.1.x holds two root keys which are further used to generate and delegate the session keys. However, it does not come into resistance in cyberattacks, as an attacker can use brute-force and dictionary attacks to crack the root key. In practice, the root key cracking is employed in [16] in which Cesar *et al.* used a pair of two messages such as the join request and join accept messages with the message integrity code (MIC). Making such activities cumbersome and protecting the network from attacks, this thesis proposes the key generation and distribution (KGD) algorithm that mitigates the root key attacks and securely delegates the session keys between the *ED* and the *JS*. The key summary of the proposed work is described below in detail.

1.2 Thesis Objective and Motivations

The root key identifiers (AppKey and NwkKey, as shown in Fig. 1.1) are the critical components of the security of a LoRaWAN network. It is used to derive the NwkSKey and AppSKey that are used to authenticate communication between the device, the network, and the *AS*. As such, it is important to ensure that the root keys are kept secure and are not compromised by an attacker. If one of them is compromised, it could potentially allow unauthorized devices to join the network, which could compromise the security and integrity of the network.

Several potential security vulnerabilities that could affect the root keys in the LoRaWAN network include;

- ***ED* physical attacks:** The device deployed in the deployment region may unveil its root keys if an attacker is able to physically approach the device. Most of the devices lack a hardware security module (HSM); therefore, an attacker with physical access can extract the root keys from the device.
- **Network attack:** A potential attack may occur if the message is being transferred between the *ED* and the *NS*. An attacker can intercept communication and may try to extract the root keys from the message.
- **Weak key generation:** If the root keys are generated with improper

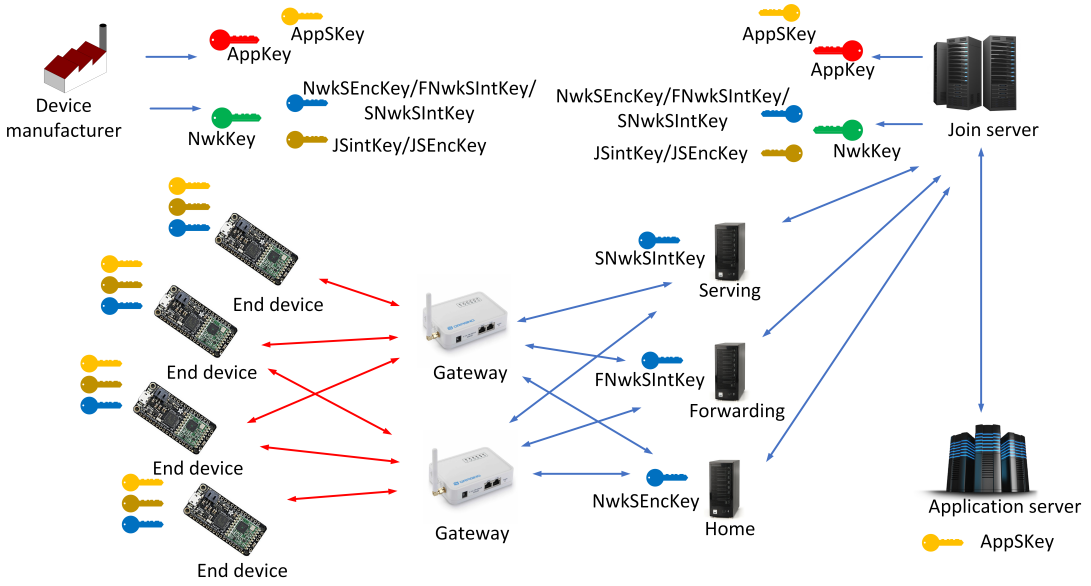


Figure 1.1: Network architecture of LoRaWAN.

technique or insufficient randomness, an attacker may consider the cryptoanalysis of the *ED* and extract the root keys.

- **Weak key updating:** If the root keys are not updated frequently enough, an attacker may be able to exploit vulnerabilities in the key to gain access to the network.

- **Compromised server:** The *ED* shares the root keys with the *NS*. If the security confidentiality of the root keys at the server is breached, all information that is being exchanged between the *ED* and the *NS* is breached.

Addressing the root keys' breaches is one of the significant fields in LoRaWAN communications. The researcher in [15] and [17] urges that an attacker can potentially exploit communication in LoRaWAN, if the root key is not periodically updated. Hence, the security of the network could be violated if the root keys are unveiled to the attacker as it is used to generate the session keys. The research study in [18] [19] [20] [21] proposed distinct approaches to generate and update the root keys. However, several challenges are confronted as these approaches are noncompliant with the tiny *EDs*. Since, *EDs* are equipped with low computing resources and are unable to compute heavy cryptography algorithms. In addition,

such approaches select a non-standard randomness that is used to generate the root key. Weak randomness can jeopardize the root keys that are injected into the memory of the microcontroller in the *ED*.

This study is motivated by the aforementioned fact; therefore, we propose a new lightweight and authenticated secure root key management in LoRaWAN. Furthermore, this approach is put into practice on a testbed in the real world, and its effectiveness is evaluated.

1.3 Contributions

Contributing to the conventional key generation schemes [20] [22], [23] in LoRaWAN, this study proposes the KGD mechanism that updates and securely exchanges the root keys in LoRa possible. The KGD protocol makes use of Diffie-Hellman (DH) key exchange in addition to elliptic curve (EC) cryptography which makes the protocol resilient against cryptanalysis and other cyberattacks. In essence, this paper lists the following key points.

- **Key generate:** In a LoRaWAN network, the root keys are typically hard-coded into the *ED* while it is being fabricated in the industry. The root keys are used to generate the session keys, which in turn, encrypt and decrypt the payload being exchanged between the device and servers. However, security breaches of the root keys can potentially allow an attacker to gain unauthorized access to the *ED* and the *NS*. To cope with the root keys' breaches, there are a few different ways that the root keys can be generated, depending on the specific implementation of the network. This work proposes a novel root key-generating mechanism to make the network more resilient against cyberattacks. In the KGD mechanism, the master root key is used to generate the root keys that are used for authentication and encryption. The KGD considers a cryptographically secure random number generator to generate the initialization vector (IV) and AES-128 bit in CBC mode.

- **Key distribute:** LoRaWAN nodes are often deployed in large-scale, geographically dispersed networks such as Smart Cities, Industrial IoT, and

agricultural monitoring. In these scenarios, it can be challenging to manually distribute the root keys to each node. Manual distribution of the root keys involves physically exchanging the key, which can be costly, and if the key is compromised, it can be difficult to update the key on all the nodes. To update the shared keys e.g., NwkSKey and AppSKey in the node deployed in a scattered area, the KGD protocol uses the Elliptic Curve Diffie-Hellman (ECDH) key exchange method, as the RSA protocol is more computationally demanding. The ECDH algorithm uses a smaller key length size and provides high security at the cost of lower energy consumption.

- **Key authenticate:** After the secret keys exchange that generates the session keys, an authentication process takes place since the ECDH method is vulnerable to Man-in-the-Middle (MITM) attacks. The proposed KGD protocol considers an elliptic curve digital signature algorithm (ECDSA) to authenticate communication between the *ED* and the *JS*. By using ECDSA, only the authorized node is allowed to communicate with the relevant *NS*.

- **Formal security analysis:** Unlike the lightweight key exchange approach in [24], we analyze the security vulnerability of the proposed KGD algorithm and provide formal security analysis using the Scyther tool focusing on the key exchange technique. We analyze, explore, and inspect the security features of the KGD protocol. However, there is no vulnerability found against the KGD protocol; which, in turn, clarifies the integrity, secrecy, and authentication of the proposed implementation.

1.4 Thesis Outline

The document is organized as follows: Chapter 2 provides an introduction to LoRa as a physical layer, highlighting its unique modulation technique, and LoRaWAN as the MAC layer built on top of it. In Chapter 3, we offer an overview of relevant research related to LoRaWAN security, network performance, and recent changes to the specification. Chapter 4 delves into the cybersecurity vulnerabilities and privacy concerns within LoRaWAN. It covers potential attacks on the

network and presents corresponding countermeasures. Also, reviews related work from existing literature. Chapter 5 outlines our experimental design and analysis. Finally, the document concludes with Chapter 6, summarizing the overall work.

Chapter 2

Background Theory

Summary

This chapter presents an introduction to the core principles of LoRa and LoRaWAN. Its primary purpose is to establish a foundational understanding that will be crucial for comprehending the subsequent exploration of vulnerabilities and security issues within this technology. While we do provide a concise overview of the specification and physical layer, our emphasis lies on specific aspects that bear relevance to the security aspects of the protocol.

2.1 LoRa Technology

LoRa, short for "Long Range," is a variant of low power wide area network (LPWAN) designed for wireless devices that operate on battery power. It is intended to facilitate communication over regional, national, or even global networks. LoRa technology is developed and overseen by the LoRa Alliance, a collaborative and nonprofit organization comprised of members who work together to develop the LoRaWAN specification. The aim is to establish LoRaWAN as the standardized, open global protocol for secure and robust LPWAN connectivity. One of the key advantages of LoRa technology is its ability to transmit and receive data efficiently over long distances without incurring high power consumption. In Europe, LoRa operates within the 868MHz ISM band and can cover substantial

distances, reaching up to 30 kilometers, depending on the environmental conditions. Different frequency bands are utilized in various parts of the world, such as 902MHz in the United States and 915MHz in Australia. LoRa employs a spread spectrum approach, using wide bandwidth to enhance resistance to deliberate interference and environmental noise. This feature contributes to its reliability and effectiveness in providing long-range communication for Internet of Things (IoT) devices and other applications.

2.1.1 LoRa Modulation

LoRa modulation is a proprietary technology that was originally patented in 2012. Subsequently, Cycleo was acquired by Semtech, which now manufactures LoRa hardware and also licenses the technology to other entities. While Cycleo and now Semtech have been pioneers in introducing a Chirp Spread Spectrum (CSS) type modulation to the market, which they have highlighted for its notable advantages in terms of range and link budget, many of the specifics related to the physical layer and encoding of LoRa technology have been kept relatively confidential or concealed.

2.1.2 LoRa Physical Layer

LoRa is a proprietary chirp spread spectrum (CSS) modulation whose key properties are determined by the spreading factor (SF), bandwidth (BW), and coding rate (CR) [25]. The spreading factor is the ratio of symbolic and chip rate as in Eq. (1) that facilitates the signal with multiple grades starting from SF=7 to Sf=12. The SF method utilizes forward error correction (FEC) to provide long-range communication with the price of low speed.

$$SF = \ln \left[\frac{R_c}{R_s} \right] \tag{2.1}$$

LoRa modulation with minimal error is cumbersome in some situations because of the diverse effects in the channel. Therefore, it implies FEC implementation by encoding 4-bits data with the variant of redundant bits for instance

5-bits to 8-bits as shown in Table 2.1. This implementation significantly reduces interference in the channel. Selecting the CR value is adjusted in accordance with the channel effect. A higher value of CR is recommended for a high interference channel. However, the higher CR value results in higher latency in transmission. The BW also represents as chirp rate in LoRa modulation is the frequency range that is used for imposing the baseband data. LoRa transmits the packet by using the BW value of 125 kHz, 250 kHz, 500 kHz. It is impervious to interference because of all of these properties combined.

Table 2.1: Coding rate for LoRa packets.

Coding rate (CR)	CR = 4/(4+CR)
1	4/5
2	4/6
3	4/7
4	4/8

2.1.3 Key Features and Use Cases

LoRa technology is specifically designed to operate within the Industrial, Scientific, and Medical (ISM) bands. The specific frequencies in these bands may vary based on local regulations, but they share a common characteristic: they can be used by anyone without needing a license. This approach offers the advantage of not incurring additional operating costs, but it comes with trade-offs. Non-exclusive usage is allowed, but there are typically strict limitations on transmission power and how long a device can transmit.

By structure, the LoRa frame comprises three fundamental parts: a preamble designed for synchronization purposes, an optional physical header, and an application payload as shown in Fig. 2.1. It is essential to note that both the preamble and the physical header undergo processing within the transceiver, making them beyond the direct reach of the application developer. Following their successful processing, the payload is then transmitted to the application layer for subsequent handling. The presence of a physical header in the LoRa frame is subject

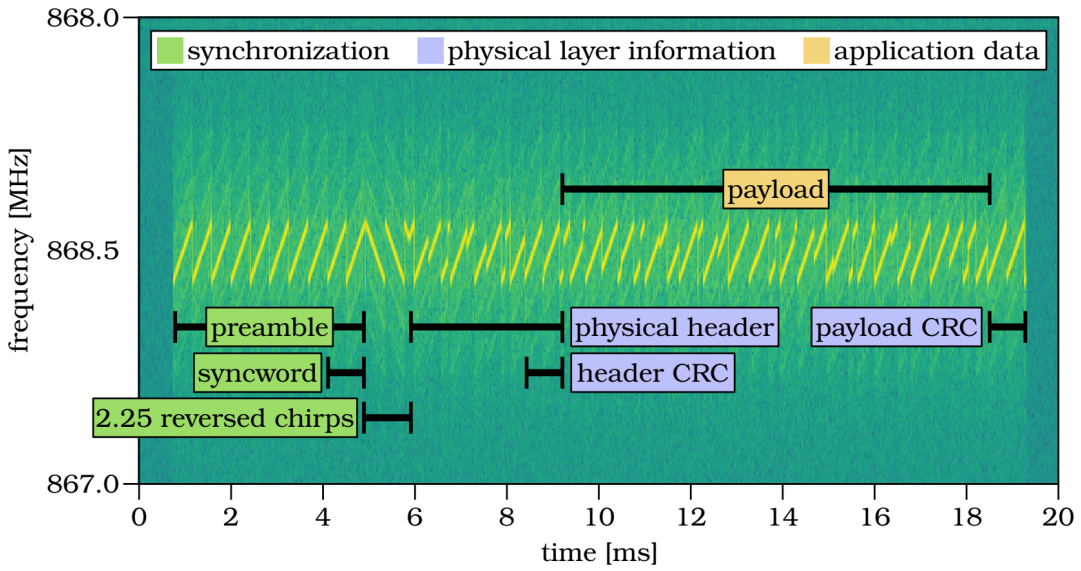


Figure 2.1: A spectrogram structural composition of the LoRa frame [1].

to discretion. In the specific configuration denoted as "explicit header mode," this header serves as a repository for vital frame structure information. This information encompasses details such as the employed coding rate, frame length, and whether the application payload benefits from integrity protection through a cyclic redundancy check (CRC) at the physical layer. Conversely, in the "implicit header mode," the physical header is entirely omitted. In such instances, the information typically conveyed by the header must be separately communicated to the receiver through out-of-band means. It is important to note that while the inclusion of a payload CRC remains optional, frames with CRC failures in this regard remain accessible to the application layer. However, it is crucial to underscore that a CRC failure occurring within the header results in the frame being discarded.

2.2 The LoRaWAN Standard

LoRaWAN, a communication protocol introduced by the LoRa Alliance, has gained notable adoption. According to data provided by the LoRa Alliance, the number of countries actively utilizing LoRaWAN deployments has expanded

to encompass 142 nations. Within this global network, there are currently 121 network operators spanning 58 different countries [16]. LoRaWAN is a standard that defines the Medium Access Control (MAC) layer and outlines the necessary backend infrastructure, serving as the comprehensive framework for Low-Power Wide Area Network (LPWAN) deployments. It re presents an open specification developed by the LoRa Alliance, initially released in January 2015 [26]. This specification provides comprehensive coverage of essential components, including key actors, network entities, message types, operational procedures, and the configuration specifics of the physical layer. The essence of the LoRaWAN specification is comprised of two integral components; the LoRaWAN Specification itself and a document for additional Regional Parameters. While the LoRaWAN Specification outlines the overarching structure and functioning of the protocol on a global scale, the Regional Parameters document assumes a crucial role in addressing shortcomings that are inherently tied to regional regulatory frameworks. A prominent illustration of such region-specific considerations involves the establishment of band plans tailored to specific geographical regions. In this thesis, an initial attempt was made to employ the EU433 region. However, owing to the deprecation of The Things Network (TTN), a decision was made to transition to the EU868 region, which now serves as the designated operational domain for this research. These documents are complemented by a set of Feature Specifications. These specifications define additional and optional functionalities, such as those related to the backend interface, multicast support, or data fragmentation. The collection is then finalized with the incorporation of Technical Recommendations, which provide valuable best practices and guidelines rather than specifying mandatory protocol details. The LoRaWAN specification, since its initial publication has undergone multiple revisions as shown in Fig. 2.2.

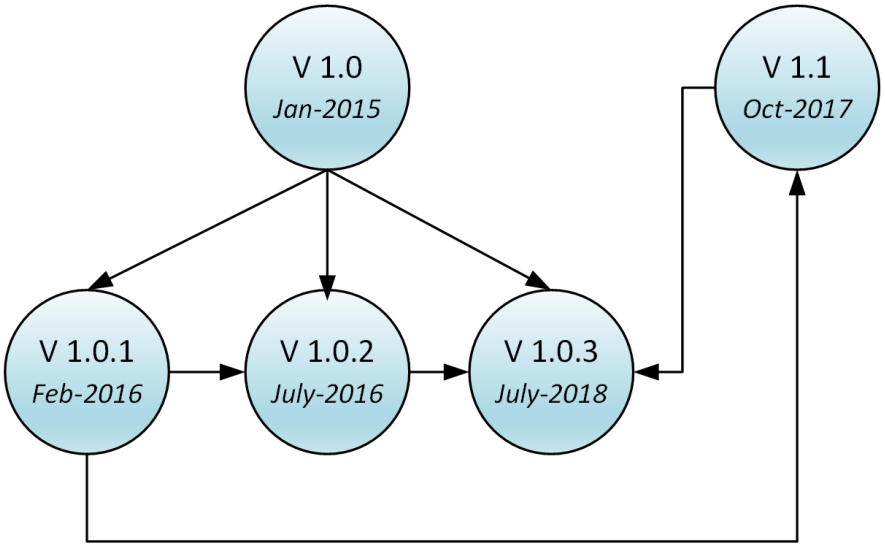


Figure 2.2: Multiple versions of LoRaWAN.

LoRaWAN versions 1.0.1 and 1.0.2 primarily serve the purpose of providing clarifications and rectifications to the initial version of the specification. Additionally, they introduced new frequency plans to facilitate the inclusion of more regions, which, in part, necessitated the incorporation of new functionality at the MAC layer. It's worth noting that as the specification pertaining to regional parameters expanded in scope, it was extracted into a separate document, as mentioned earlier. From a security perspective, a notable change is the shift from optional payload encryption on the MAC layer in the initial specification to its mandatory implementation in subsequent versions. The newer version LoRaWAN v1.1, is a significant revision of the LoRaWAN specification, introducing substantial changes to both the network architecture and the security model. These revisions, which include the introduction of a new key hierarchy, adjustments to checksum calculations, and refinements to the join procedure, are a direct response to the identification of vulnerabilities in earlier versions. It is important to note that these changes are not backward-compatible with LoRaWAN v1.0.

2.2.1 The Network Architecture

The LoRaWAN network is a comprehensive ecosystem composed of several integral elements as illustrated in Chapter. 1, Fig. 1.1. The end devices shown on the left side of the diagram typically consist of a microcontroller, a LoRa modem (for point-to-point connection), or a LoRaWAN modem, along with specialized hardware tailored to their specific functions. These end devices often include sensor nodes, which either periodically transmit data readings or respond to specific events, serving as prominent examples within this category. When end devices detect a packet, they initiate a broadcast to the gateways. Up-link transmission, in this context, signifies the action of sending packets from the end devices to the gateways. Conversely, down-link communication refers to the process of receiving packets from the gateways on the end devices. Essentially, up-link is about data transmission from the end devices to the network, while down-link involves data reception from the network by the end devices.

End devices establish direct communication with gateways utilizing LoRa or, if preferred, frequency-shift keying (FSK). Gateways are powered by the mains and are equipped with a LoRa concentrator, allowing them to cover multiple channels within their designated coverage area. These gateways are further equipped with a backend network connection, typically via Ethernet or cellular mobile networks like LTE. Gateways fulfill the crucial role of forwarding all received data frames to their designated network server. Gateways possess the capability to receive packets from all end devices encompassed within the network's coverage area. They function as relays, forwarding all messages originating from the end devices to the network server. This pivotal role ensures that data generated by the end devices is efficiently transmitted to the network server for further processing and management.

The network server, often referred to as the core component in the LoRaWAN ecosystem, is responsible for overseeing all functions related to end devices and gateways. End devices and gateways communicate via radio frequencies, whereas gateways and network servers rely on the Transmission Control Protocol/Internet Protocol (TCP/IP) for data exchange. Unlike traditional wireless networks, LoRaWAN adopts a star topology in the network. This star topology not only

enhances network capacity but also simplifies network architecture and minimizes energy consumption from device batteries.

The application server carries out tasks customized for the specific use case, which may include actions like storing measured values in a database. In its operational capacity, it can also deliver downlink data to the network server, which then facilitates the transmission of this data back through the network, ultimately reaching the end device. The join server is employed in LoRaWAN v1.1, when an end device establishes or reestablishes a connection with a network. It uses a specific extended unique identifier (EUI), known as JoinEUI, to designate a join server. The join server serves two primary functions: the separation of session key derivation for network and application keys, and support for roaming by enabling devices to locate their home network server.

2.2.2 Technical Analysis

LoRaWAN employs chirp spread spectrum (CSS) modulation technology for communication, enabling the network to achieve extensive communication ranges, reaching up to 20 kilometers. This extended range is made possible by the utilization of spreading factors (SF). Essentially, the higher the SF, the greater the communication distance but at the cost of increased energy consumption and reduced data throughput. It's important to emphasize that LoRa operates within unlicensed frequency bands, imposing a duty cycle limitation of 1% in Europe to manage spectrum usage. The maximum payload size supported is 243 bytes, with transmission rates ranging from 300 bytes per second to 50 kbps, contingent upon the chosen bandwidth and SF. LoRaWAN introduces an adaptive data rate (ADR) scheme that facilitates dynamic adjustment of communication rates between gateways and end devices. This ADR scheme optimizes network efficiency by selecting robust channels during data transmission from the end device to the gateway, thereby enhancing the overall network's lifespan. LoRaWAN devices are categorized into three classes: Class A, Class B, and Class C, as elaborated in Section 2.2.3. All end devices inherently support Class A, which is often referred to as the default class. In Class A operation, end devices utilize an ALOHA-style protocol, where two downlink windows are available concurrently with an

uplink transmission. This methodology facilitates bidirectional communication for end devices requiring it. Notably, Class A operation stands out as the most power-efficient method, as end devices can enter a sleep mode once they have defined their application. Downlink communication from the server is only necessary when the end device initiates an uplink transmission. Class B introduces the capability for additional receiving windows at scheduled times, supplementing the initiated windows of Class A. Class B devices receive periodic beacons to synchronize with the gateway. The programmable latency, which can be set to a maximum of 128 seconds, offers flexibility that can be advantageous in various applications, but it comes with trade-offs related to power consumption. It's important to note that the additional power consumption in Class B remains a relevant consideration, particularly in battery-dependent applications. Class C devices prioritize achieving the lowest latency, but this comes at the cost of higher energy consumption. In the Class C operating mode, devices keep their receiving windows open continuously, allowing the network server to communicate with them without any noticeable latency, as the receiving windows are always accessible. However, it's essential to be aware that Class C devices can have a relatively high power consumption, potentially reaching up to 50 milliwatts (mW). Consequently, it is recommended that devices in this class be powered continuously rather than relying on batteries in various application scenarios. In essence, the technical superiority of LoRaWAN over other Low-Power Wide-Area Network (LPWAN) technologies is depicted in the table 2.2.

Table 2.2: Comparison of LoRaWAN with other LPWAN Technologies.

Feature/LPWAN Technology	**LoRaWAN**	**SigFox**	**NB-IoT**
Adaptive data rate	Yes	No	No
Battery lifetime-2000 mAh	120 months	120 months	<120 months

Continued on next page

Table 2.2: Comparison of LoRaWAN with other LPWAN Technologies. (Continued)

Feature/LPWAN Technology	**LoRaWAN**	**SigFox**	**NB-IoT**
Coexistence	Yes	No	No
Data Rate	300-50K bps	100-600 bps	20K-200K bps
Frequency	868/915/433 MHz ISM	862/928 MHz ISM	LTE
Additional gateway requirement	Yes	Yes	No
Scalability (per cell)[a]	50K	50K	100K
Interference immunity	Very High	Low	Low
Link Budget	154 dB	154 dB	150 dB
Licensed[b]	No	No	Yes
Max. msgs/day (down-/up-link)	696/10	140/4	Unlimited
Maximum output power	20 dBm	20 dBm	20 dBm
Modulation	CSS	BPSK	OFDMA
Mobility/localization	Yes	Limited Mobility	Limited Mobility
Power efficiency	Very High	Very High	Very High
Packet payload length	243 Bytes	12/8 Bytes	1600 Bytes
Rx bandwidth	125-500 KHz	100 Hz	200 KHz
Standardization	LoRa-Alliance	SigFox and ETSI	3GPP

Continued on next page

Table 2.2: Comparison of LoRaWAN with other LPWAN Technologies. (Continued)

Feature/LPWAN Technology	**LoRaWAN**	**SigFox**	**NB-IoT**
Time latency[c]	2-10 Sec	1-30 Sec	1.6-10 Sec
Security aspects/Technology	**LoRaWAN**	**SigFox**	**NB-IoT**
Data confidentiality	Yes	Optional	Yes
Authentication and encryption	AES 128 b	Optional	LTE Encryption[d]
Security	Yes	Optional	Yes
Integrity Protection	Yes[e]	Yes[f]	Yes[g]
Availability[h]	Medium-scale	Small-scale	Large-scale

[a] "NB-IoT offers the advantage of very high scalability than Sigfox and LoRa. NB-IoT allows connectivity of up to 100 K end devices per cell compared to 50 K per cell for Sigfox and LoRa."[27]

[b] LoRaWAN and SigFox operate on the unlicensed band of the frequency spectrum (called ISM-band), whereas NB-IoT requires license (subscription) for operation.

[c] Regarding *time latency*, LoRaWAN finds itself in between SigFox and NB-IoT, yet closer to the NB-IoT side. In a recent study, Rydell *et al.* reported *time delays* from 250 ms to 2 secs for LoRaWAN transmission, depending on the deployment density (100 towards 500) and also the eagerness of the used reliability methodology for communications [28].

[d] "3G networks use the KASUMI block cipher with the UEA1 confidentiality and UIA1 integrity algorithms. The 4G LTE successor is the SNOW 3G stream cipher and the UEA2 confidentiality and UIA2 integrity algorithms." [29]

[e] "Integrity protection is provided in a hop-by-hop nature: one hop over the air through the integrity protection provided by LoRaWAN L2 and the other hop

between the Network Server and the Application Server by using secure transport solutions such as HTTPS and VPNs." [30]

[f] "Each message to be sent or received by the device contains a cryptographic token that is computed based on this authentication key. Verification of the token ensures the integrity of the message" [31].

[g] "Data over NAS (DoNAS) is a control plane cellular IoT optimization that allows the network to transport user data or SMS messages via the MME (mobility management entity) by encapsulating them in NAS (non-access stratum) signaling. DoNAS can be used to transport both IP and non-IP traffic. One key security benefit of this feature is that the customer/user data is encrypted and its integrity protected using the same mechanism reserved for network signaling, thus ensuring similar levels of protection." [32]

[h] It is strictly dependent on the number of the GWs and the base stations [33].

2.2.3 Device Classes

As described in Section. 2.2.2, the LoRaWAN end devices are categorized into three distinct classes. Class A represents the default implementation that all devices support, with Classes B and C serving as extensions of Class A. The characteristics of each class are outlined below:

2.2.3.1 Class A(ll)

Class A, short for "All," represents the obligatory implementation for all LoRaWAN end devices. It serves as the default class, wherein the end device opens two short windows after completing an uplink transmission. This class enables bidirectional communication and then returns to sleep mode until it needs to interact with its designated application. While Class A end devices are highly energy-efficient, they do introduce higher latency because of their sleep mode patterns.

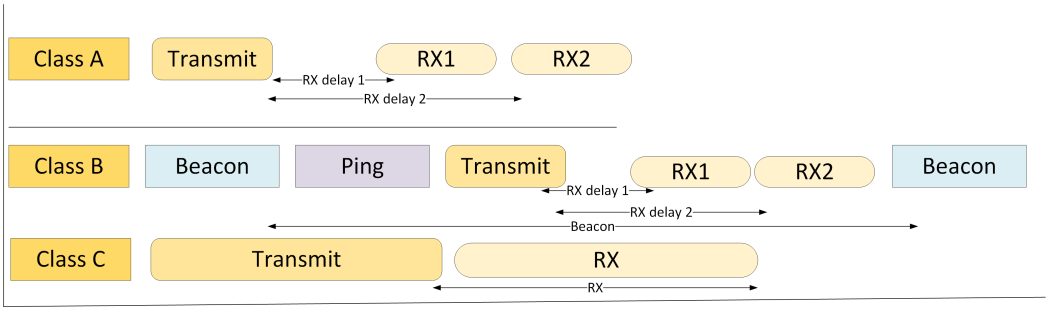


Figure 2.3: LoRaWAN end devices transaction of Class A, B, and C.

2.2.3.2 Class B(eacon)

Class B serves as an extension of Class A and incorporates additional receiving windows in addition to those of Class A. Class B devices introduce downlink ping slots by synchronizing with the network server through periodic beacons. This synchronization enables the network to exchange downlink transmissions with the trade-off of increased latency and power consumption by the end device. It's important to note that while Class B devices do consume more power, their energy consumption is still suitable for applications powered by batteries.

2.2.3.3 Class C(ontinues)

Class C devices are recognized for their superior latency efficiency as they keep all receiver windows open simultaneously. In this class, the network server engages in communication with devices via downlink transmissions, assuming that the end device's receiver windows are constantly accessible. Consequently, this class enables communication with no noticeable latency. However, it's important to acknowledge that Class C end devices consume more energy compared to Class A and Class B devices. Therefore, Class C devices are best suited for applications where continuous power is supplied. The interaction and characteristics of Class A, B, and C are illustrated in Figure 2.3.

2.2.4 LoRaWAN MAC Layer

LoRaWAN recently commercialized by LoRa Alliance is a network protocol and together with LoRa (physical layer), that enables a long-range communication

link. It has a significant influence on determining battery lifetime, security, quality of service, and network scaling. The device purposely installed in the network can last for up to several years powered by a battery source. It provides packet exchanging convenience with a range of 50 km in rural areas by ensuring high-security encryption such as 128-bit AES cryptographic technique. The adaptive data rate (ADR) allows the node located near the gateway to send the packet with a high data rate and use a lower data rate several kilometers from the gateway.

2.2.5 Data Layer

The deployment of LoRaWAN architecture can be accomplished by using star topology. In this deployment fashion, a node can establish a connection with one gateway which is referred to as a standard start topology. However, it is important to mention that a node can be connected to other gateways if available in the communication range. Therefore, it advances the packet with another topology called star-of-starts network topology. The node can start communication with the gateway without prior synchronization.

2.2.6 Adaptive Data Rate

LoRaWAN provides the network server with the capability to dynamically modify parameters for end devices. These parameters, which encompass SF, frequency, and transmitting power, collectively contribute to managing and enhancing the network's quality of service.

Opting for a lower SF results in transmitting packets with minimal latency, making it suitable for short-range communication. On the other hand, selecting a higher SF extends the packet's reach, allowing for long-range communication, but this does introduce higher latency. It's important to note that using a higher SF is generally not recommended due to the increased likelihood of packet collisions.

The Adaptive Data Rate (ADR) feature is a key aspect of LoRaWAN, enabling nodes situated in close proximity to the gateway to transmit data at higher data rates, thereby optimizing communication for short distances. Conversely, when nodes are positioned several kilometers away from the gateway, ADR adjusts the

data rate downward to ensure reliable and efficient communication.

2.2.7 Frame Payload

In LoRaWAN, the payload's careful arrangement within the physical LoRa frames is of paramount importance to facilitate the transmission of various message types. This payload is designed to be adaptive, featuring a type-dependent section enveloped by an initial MAC layer header. This MAC header, positioned at the beginning of the frame, plays a crucial role in identifying the specific message type through the MType field. Furthermore, at the completion of the payload, a Message Integrity Code (MIC) is incorporated to ensure data integrity and security. This well-structured format, as depicted in Figure 2.4, serves as the foundation for all typical LoRaWAN frames. Each frame initiation includes a physical header followed by the MAC header, where the MType field offers precise classification of the message type, enabling accurate communication and interpretation.

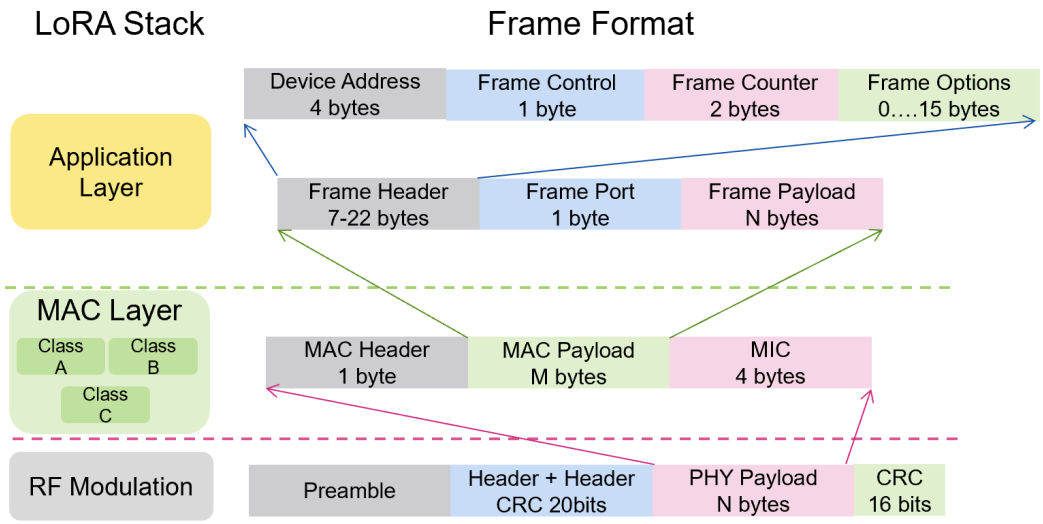


Figure 2.4: LoRaWAN message frame format.

Chapter 3

Security Characteristics of The LoRaWAN Protocol

Summary

This section presents the security consideration of LoRaWAN protocol, as security has been an integral part of the LoRaWAN specification from its initial version, and the specification provides security across various aspects.

3.1 LoRaWAN end-device activation Method

In LoRaWAN communication, the end-device is required to undergo an activation process and successfully complete the join procedure prior to enabling communication with the network server. This mechanism is essential to control the access from unrecognized end-devices to a LoRaWAN network server and prevent these devices from participating in communications. From the LoRaWAN specification, there are two activation methods for end devices: Activation by Personalisation (ABP) and Over-the-Air Activation (OTAA), which are described in the following subsections.

3.1.1 Activation by personalization (ABP)

In the ABP activation phase, the end device requires only session keys and the device address (DevAddr), all of which are hardcoded into the device. These session keys encompass the network session key (NwkSKey) and the application session key (AppSKey). The NwkSKey is employed to ensure message integrity, protecting data from unauthorized alterations during transmission. In contrast, the AppSKey is utilized for encrypting or decrypting the payload, which is essential for secure exchanges between the end device and the application server. The ABP activation method establishes a direct connection between an end-device and a specific LoRa network, bypassing the standard Join-request and Join-accept procedure.

3.1.2 Over-the-Air activation (OTAA)

The Over-the-Air Activation (OTAA) method is considered the most secure and preferred activation approach for end devices. In the OTAA process, the end device initiates the activation by sending a join request message to the network server. This join request message contains three essential components: the globally unique device identifier (DevEUI), a global application ID in IEEE EUI64 (APPEUI) in LoRaWAN v1.0.x or JoinEUI in LoRaWAN v1.1.x, and DevNonce, which starts as a random number set to zero. The network server then evaluates the join request and, if it verifies the correct keys, responds with a message called the join accept message. This join accept message consists of three critical values: AppNonce (a random number generated by the network server), DevAddr, and network identifier (NetID). Unlike the ABP activation method, the OTAA activation process is highly regarded for its robust security, mainly because it periodically changes the keys, thereby enhancing the overall security of the network. The structure of the OTAA activation process is illustrated in Fig. 3.1.

3.1.2.1 Join request message

The initiation of the join procedure always originates from the end-device, which sends the join-request message to the network server/join server. The join request

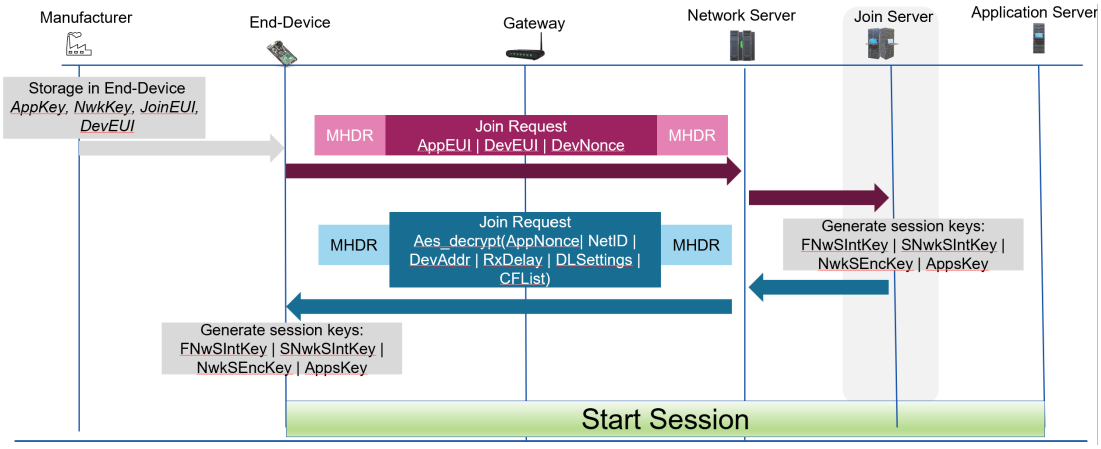


Figure 3.1: The OTAA activation process between the device and the server.

message comprises the JoinEUI and DevEUI of the end-device, followed by a 2-octet nonce referred to as DevNonce, as depicted in Fig. 3.2. The DevNonce is a 2-byte value that functions as a counter. It starts at 0 when the device is powered up and is incremented each time the device initiates a join request to the join server. This sequential incrementation of the DevNonce is a crucial security measure within the LoRaWAN protocol. It ensures that each join request is unique and not susceptible to replay attacks, thereby enhancing the overall security of the join procedure. The join-request message does not possess encryption; however, the message integrity code (MIC) is used to prevent it from tempering. It has the flexibility to be transmitted at any data rate and follows a randomized frequency hopping sequence across the designated join channels. The MIC of the join-request message is calculated below which is a security feature used to ensure the integrity of the received message, meaning it helps confirm that the message has not been tampered with during transmission.

$$cmac = aes128\_cmac(NwkKey, MHDR|JoinEUI|DevEUI|DevNonce)$$
$$MIC = cmac[0..3]$$

Join request	JoinEUI	DevEUI	DevNonce
Size (bytes)	8	8	2

Figure 3.2: Join request message format.

3.1.2.2 Join accept message

Upon the arrival of the join request, the network server initiates an automated verification process, simplifying the procedure for determining the end-device's eligibility to access the network. This process includes checking out elements such as the MIC, DevEUI, and JoinEUI. Its primary objectives are two-fold: firstly, to make a decision regarding whether the end-device should be granted access to the network. And secondly, to identify the specific application to which it should be connected. If the end-device fails to meet the established criteria, it will not receive any response. In contrast, when the criteria are satisfied, the network server promptly issues a "Join accept" message to the end-device, marking its secure authorization to establish a connection with the network. This verification process ensures both security and efficient network management. The join-accept message is transmitted like a regular downlink message, but it employs either *JOIN_ACCEPT_DELAY1* or *JOIN_ACCEPT_DELAY2* as specified delays. The join-accept message is comprised of various components, including a 3-octet server nonce (JoinNonce), a network identifier (NetID), an end-device address (DevAddr), a field detailing some of the downlink parameters (DLSettings), the delay between transmission and reception (RxDelay), and, optionally, a region-specific list of network parameters known as CFList, as outlined in [PHY]. The JoinNonce is a device-specific counter value, provided by the Join Server, which is used by the end-device to derive session keys (FNwkSIntKey, SNwkSIntKey, NwkSEncKey, and AppSKey). With each Join-accept message, the JoinNonce is incremented to ensure its uniqueness. The device retains the JoinNonce value utilized in the most recently successfully processed Join-accept message, which corresponds to the last key derivation process that was completed successfully.

If the device is prone to power issues, the JoinNonce resumes from where it left off, as it is stored in non-volatile memory. The MIC of the Join-accept message is calculated below:

$$cmac = aes128\_cmac(NwkKey, MHDR|JoinNonce|NetID$$
$$|DevAddr|DLSettings|RxDelay|CFList)$$
$$MIC = cmac[0..3]$$

Join accept	JoinNonce	NetID	DevAddr	DLSettings	RxDelay	CFList
Size (bytes)	3	3	4	1	1	(16) Optional

Figure 3.3: Join accept message format.

3.2 Cryptographic theory and practice in LoRaWAN

LoRaWAN has gained significant popularity in the field of IoT communication due to its remarkable characteristics, such as low power consumption and the capability to transmit data across extensive distances. As the number of users continues to rise, there is a corresponding increase in security breaches, which could potentially jeopardize the network and cause significant damage. Thus, researchers, scientists, and engineers have put their efforts into practice and investigated the new version of LoRaWAN which is in line with the existing standard. As mentioned above, the LoRaWAN V1.0.x is augmented by concatenating the new server called "*JS*" which orchestrates the join procedure in a more secure fashion. Contrary to this, the newer version is equipped with three different NS_s i.e., home, serving, and forward that potentially tackle inevitable distance problems. As can be seen in Chapter 1, Fig. 1.1, the addition of the NS_s enables packet roaming from local, wide, and worldwide contexts. The *ED* in LoRaWAN adapts two methods either OTAA or ABP to participate in the network. To

establish the OTAA process, the *ED* needs to store prior data such as; the Join Extended Unique Identifier (JoinEUI) is a unique 64-bit identifier assigned to each *NS*. The Device Extended Unique Identifier (DevEUI) is a unique 64-bit identifier assigned to each end device. The Device Nonce (DevNonce) is a random 16-bit value generated by the end device and included in the join request packet sent to the *NS* during the join process. Device root keys i.e., AppKey and NwkKey both are hard-coded in the *ED* during fabrication. The AppKey is a unique AES-128 bit key shared between the *ED* and the *AS*. It is used to encrypt and decrypt the payload of application data messages, and to derive the AppSKey, which is used to secure the communication between the *ED* and the *AS*. The NwkKey is a unique AES-128 bit key that is shared between all *ED* and the *NS*. As, several NS_s are involved in v1.1; therefore, NwkKey is used to generate the session key for each server, and specific lifetime keys for the *JS*. The detail of each session key is mentioned below:

Session keys: In LoRaWAN, the *NS* annexes the MIC to all uplink and downlink messages. Each *ED* has a specific session key which is generated from NwkKey, JoinNonce, JoinEUI, and DevNonce which are further explained below.

Forwarding Network Session Integrity Key *(FNwkSIntKey)*: Equation 3.1 shows the *FNwkSIntKey* which is a network session key assigned to each *ED* that is used to calculate the MIC of all uplink transmission. This key can be shared with a roaming forwarding *NS* but it should not be made public because it could cause the network to experience issues [47].

$$\begin{aligned} FNwkSIntKey = aes128\_enc(NwkKey, 0x01 \\ |JoinNonce|JoinEUI|DevNonce|pad16) \end{aligned} \tag{3.1}$$

Serving Network Session Integrity Key *(SNwkSIntKey)*: The *SNwkSIntKey* is given below in equation 3.2 which is used to calculate all downlink data messages in the network. Unlike the *FNwkSIntKey*, this key is considered private and should not be disclosed to the forwarding *NS*. The *SNwkSIntKey* is also used to compute MIC for half of uplink data messages and should be kept safe to avoid malicious activities.

$$SNwkSIntKey = aes128\_enc(NwkKey, 0x03 \\ |JoinNonce|JoinEUI|DevNonce|pad16) \tag{3.2}$$

Network Session Encryption Key *(NwkSEncKey)*: In the same fashion as the previous keys, the *NwkSEncKey* is a network session key and is specific to each *ED*. It is utilized to encrypt/decrypt uplink and downlink MAC commands that use port 0 or FOpt field as the way to exchange payload. In case the *ED* needs to connect to LoRaWAN v1.0.x, the *NwkSEncKey* is used to encrypt the MAC payload as well as compute MIC. Equation 3.3 shows the *NwkSEncKey*, that should not be disclosed to outsiders.

$$NwkSEncKey = aes128\_enc(NwkKey, 0x04 \\ |JoinNonce|JoinEUI|DevNonce|pad16) \tag{3.3}$$

Lifetime Join Session Keys: As the inclusion of the *JS* in the new version, there are also two additional lifetime keys that are specifically used for rejoining the network when lost the connection. The *JS* integrity key *(JSIntKey)* is used for computing the MIC of rejoin request type 1 and related join accept responses. In addition, the *JS* encryption key *(JSEncKey)* is used to encrypt the response released by the rejoin request message.

$$JSIntKey = aes128\_enc(NwkKey, 0x06|DevEUI|pad16) \tag{3.4}$$

$$JSEncKey = aes128\_enc(NwkKey, 0x05|DevEUI|pad16) \tag{3.5}$$

Equation 3.4 and 3.5 show both keys that are only required in connecting the *ED* via the OTAA method. The ABP activation process is not obliged to store these keys in the device before activation.

3.2.1 Encryption and decryption

Cryptographic encryption and deception are key concepts in LoRaWAN communication as the frame payload is encrypted during communication started from the end device towards the application server. During communications, the frame payload is encrypted in a specific way in LoRaWAN. If the frame payload contains only MAC (Media Access Control) commands, the NwkSKey is used for encryption. However, if the frame payload contains application data, the AppSKey is used for encryption. The encryption process involves the following steps:

- The frame payload is divided into blocks, denoted as Ai, where i ranges from 1 to k. The number of blocks, k, is determined by dividing the length of the frame payload by 16 and rounding up as:

$$Ai, i = 1...k \tag{3.6}$$

$$k = ceil(length(frame\_payload)/16) \tag{3.7}$$

- The block, Ai, is encrypted using the AES-128 encryption algorithm with the corresponding session key, either NwkSKey or AppSKey.

$$k = AES128\_encrypt(K, Ai) \tag{3.8}$$

$$fori = 1...k \tag{3.9}$$

$$k = NwkSKey/AppSKey \tag{3.10}$$

This process ensures that the frame payload is securely encrypted based on its content and the appropriate session key. It's a crucial step in protecting the integrity and confidentiality of the data transmitted in LoRaWAN communications. Table 3.1 shows the message's encryption in the LoRaWAN network.

Table 3.1: Message encryption block in LoRaWAN

Ai	**0x01**	**4x0x00**	**Dir**	**DevAddr**	**FCnt**	**0x00**	**i**
Size (bytes)	1	4	1	4	4	1	1

3.3 Message authentication code (MAC)

LoRa employs MIC, often interchangeably referred to as Message Authentication Codes (MAC), to ensure the integrity of messages and authenticate the sender of a message. A MAC functions as a keyed hash function that relies on a symmetric algorithm. This means that only entities possessing the secret key can both generate and validate the MAC using the message and the secret key. The key distinction between a keyed hash function and a regular hash function lies in that, with a keyed hash function, only parties with knowledge of the secret key can create and verify the hash. In contrast, a hash function allows anyone to generate and validate the hash without the need for a secret key. Notably, LoRa employs hash functions for key management since it does not utilize a secret key in its processes. This approach streamlines key management while maintaining message integrity and security. The primary objective of a MAC is to ensure the integrity of a message, rather than safeguarding its confidentiality. If the transmitted message is not encrypted, then using a MAC alone does not guarantee the secrecy of the message content. When applying a MAC in conjunction with authenticated encryption, three general approaches are possible. To guarantee both the confidentiality and integrity of the ciphertext, it is necessary to encrypt the message before applying the MAC. This approach is commonly referred to as "Encrypt-then-MAC." When the primary concern is guaranteeing the integrity of the plaintext rather than the ciphertext, the MAC should be generated over the plaintext before encrypting the message. There are two common approaches in this scenario:

MAC-then-Encrypt: In this approach, the MAC is generated over the plaintext and included with the plaintext. Afterward, the entire plaintext, including the MAC, is encrypted.

Encrypt-and-MAC: Here, the plaintext is first encrypted to create the ciphertext. Then, the MAC is generated over the ciphertext. The MAC remains unencrypted and is sent alongside the ciphertext. The choice between these approaches depends on specific security and operational requirements, with each method offering different advantages and trade-offs.

3.4 Advance encryption standard (AES) operation

LoRaWAN relies solely on XOR operations and lacks the inclusion of AES, the reality is more nuanced. As previously emphasized, AES plays a crucial role in LoRaWAN security, specifically in the standardized CTR (Counter) mode. This mode, like various other cryptographic modes such as CBC (Cipher Block Chaining), involves XOR operations as part of its process. What distinguishes this application of AES in LoRaWAN is the practice of using a distinct AES key for each block cipher. This approach significantly enhances security by ensuring that each block of data is encrypted with its own unique key. This level of granularity in key management adds an extra layer of protection to the system, making it more resilient to potential security threats. In a nutshell, LoRaWAN doesn't exclusively rely on XOR; instead, it combines AES and XOR operations to create a robust and multifaceted security architecture. This approach helps address various security challenges and contributes to the overall integrity of the LoRaWAN protocol.

3.5 LoRaWAN counter management

The efficacy of LoRaWAN security measures relies on the premise that specific numeric values, when conveyed as plain text data, must be treated as invalid if they are repeated. To simplify the implementation of the LoRaWAN specification, these values are systematically incremented. When utilized as counters, it becomes the responsibility of the end-device to retain only the most recent value, ensuring that duplication is prevented. To guarantee the integrity of these coun-

ters, they are securely stored in a persistent, non-volatile, and tamper-resistant memory within the end device. The network's security protocols are designed to reject data packets that attempt to reuse numbers, which are mandated to be unique for each end-device. Notably, the subsequent numbers are designated as persistent counters. LoRaWAN possesses two types of counters such as Uplink Frame Counter (FCntUp) and Downlink Frame Counter (FCntDown). The FCntUp is maintained by the end-device and is incremented with each uplink message transmission. It helps prevent replay attacks by ensuring that duplicate or out-of-sequence messages are detected and discarded by the network server. Similarly, the FCntDown counter is managed by the network server and is used to track downlink messages sent to the end-device. It aids in maintaining synchronization and preventing replay attacks from the server side.

3.5.1 Counter overview

LoRaWAN exhibits two types of counters for packets's advancement in the network. The first counter such as, the DevNonce experiences an incremental increase with each successive Join attempt associated with a particular JoinEUI. In cases where a device alters or modifies its JoinEUI during the Join procedure, it is essential to maintain distinct and persistent DevNonce counters for each unique JoinEUI. This practice ensures that the security and uniqueness of Join requests are upheld even when JoinEUI variations occur. The second one, also known as the frame counters exhibit persistence within each network session. In the case of an end-device utilizing OTAA, there is an option to reset the frame counters at the initiation of each session, but it's imperative that no frame counter values are repeated with the same session keys. Conversely, for an end-device employing ABP, it must consistently uphold the practice of not reusing any frame counter values, necessitating the maintenance of persistent values. This approach safeguards the security and integrity of data transmission within the LoRaWAN network.

3.5.2 Counter reset

LoRaWAN allows the use of both 16-bit and 32-bit frame counters. When a frame counter reaches its maximum value and overflows, it is reset to 0. As an RN2483 end device with a default uplink transmission setting of roughly 50 seconds per message. In this scenario, an overflow occurs approximately every 38 days, calculated as 216 divided by 50 seconds. In compliance with the LoRaWAN specification, for end devices, the frame counter value is reset to zero after an overflow. This practice ensures that the frame counter remains within the appropriate range and upholds the integrity of data transmission.

3.6 Packet acknowledgment

In LoRaWAN, packet acknowledgment is a feature that applies to both uplink and downlink transmissions. Specifically, for class A devices, the downlink transmission occurs only after an uplink transmission. If a negative acknowledgment (NACK) is received from a class A device, the downlink message is automatically re-encrypted and resent. This mechanism helps ensure the successful delivery of messages in the network. On the other hand, he server's response to confirmed downlinks varies slightly for class B and class C devices. This variation is due to the ability of class B and class C devices to receive downlink messages as part of the network-initiated downlink slot, regardless of whether they have sent an uplink message previously. These devices may transmit uplinks, which carry the ACK/NACK flag, at irregular intervals. Once the Network Server dispatches a confirmed downlink to an end device, it will refrain from generating the downlink ACK/NACK message until the next uplink is received. Consequently, the downlink will not be retransmitted automatically if the end device fails to acknowledge the initial transmission. This distinct behavior accommodates the unique characteristics of class B and class C devices within LoRaWAN networks. Repeatedly reattempting downlinks without restrictions for class B and class C devices is not advisable, as it may lead to undesirable outcomes. In cases where the end device is powered off or out of the gateway's range, this practice could potentially trigger an infinite loop of retransmission attempts. Such a scenario would have

several adverse consequences, including needlessly saturating the network with transmissions, which can create noise in the area, and also significantly increase the consumption of the gateway's duty cycle, leading to inefficient use of network resources. The LoRaWAN packet types can be categorized as either confirmed or unconfirmed, and they are outlined below in the table 3.2.

Table 3.2: Packet types specified in LoRaWAN specification.

Mtype (Binary)	**MType (Decimal)**	**LoRaWAN specification**	**Description**
000	0	Join request	Join request OTAA uplink
001	1	Join request	Join accept OTAA downlink
010	2	Unconfirmed data up	Uplink dataframe, confirmation not required
011	3	Unconfirmed data down	Downlink dataframe, confirmation not required
100	4	Confirmed data up	Uplink dataframe, confirmation requested
101	5	Confirmed data down	Downlink dataframe, confirmation requested
110	6	RFU	Reserved for future use

Continued on next page

Table 3.2: Packet types specified in LoRaWAN specification. (Continued)

Mtype (Binary)	MType (Decimal)	LoRaWAN specification	Description
111	7	Proprietary	Proprietary

3.6.1 Acknowledgement method

In a LoRaWAN network, when an end device sends an uplink-confirmed message, the server carefully assesses the message for compliance with network standards. If the message meets the required criteria, the server responds with a frame that incorporates ACK (Acknowledgment) bits in the frame control field (FCtrl field). This frame serves as a downlink ACK message, and it is noteworthy that it does not contain a frame payload. However, it's important to highlight that for class A end devices with scheduled downlink messages, the frame payload will be included in the message as shown in the table. 3.3. This process ensures reliable and efficient communication between end devices and the LoRaWAN network while maintaining data integrity.

Table 3.3: PHY packet acknowledge.

Size (bytes)	MHDR	DevAddr	FCtrl	FCnt	FPort	MIC
Size (bytes)	1	4	1	2	1	4

3.6.2 Re-transmission

In the context of an uplink confirmed message in LoRaWAN, the end device diligently awaits an acknowledgment (ACK) from the network server within its predefined receive windows. In cases where the expected ACK does not happen within the designated time windows, the end device promptly initiates a retrans-

mission sequence. This sequence is typically repeated several times, underscoring the network's commitment to ensuring message delivery. However, if despite these diligent efforts, the ACK remains difficult to catch and out of reach even after multiple retransmissions, the end device accepts the situation and considers the message lost or in some cases rejected by the network. This intrinsic capability of LoRaWAN fortifies the dependability and resilience of data transmission, adapting gracefully to network challenges such as congestion or interference.

Chapter 4

Cyber-Security Vulnerabilities and Privacy Issues in LoRaWAN

Summary

LoRaWAN devices come into play with solving the problems of long range and high consumption of energy. However, such devices have limited computational power and are not capable of handling large scale complexity. As a result, these devices are vulnerable to security vulnerability and privacy concerns in the network. For any IoT network, security and privacy are actually top priorities. As a result, effective countermeasures are crucial for enabling LoRaWAN's widespread adoption in the IoT ecosystem.

The authors of [35] examined the LoRaWAN architecture, use cases, and security issues. Additionally, they have provided a list of a number of potential mitigation strategies to address the current LoRaWAN security flaws and thereby stop the associated threats. A thorough Security Risk Analysis (SRA) of the protocol has been provided by the authors in [15], which also includes a number of countermeasures to the outlined security risks. Their analysis identifies crucial practical threats like end-device physical capture, rogue gateways, and self-replay that demand special consideration from companies building LoRa networks. In a continued work, through formal security analysis via Scyther prover[1], Eldefrawy

[1]In the literature, the Scyther verification tool has already been useful in analyzing the

et al. identified several vulnerabilities of LoRaWAN including replay attacks, especially for the version 1.0 [36].

The network-layer and the application-layer securities are both defined by LoRaWAN. The integrity between the nodes and the gateway is provided by the network-layer security, which guarantees the authenticity of each node (gadget) in the system. Third parties cannot access the application data packets attributable to end-to-end encryption provided by the application-layer security between the device and the application server [26]. Designers and programmers have embraced LoRaWAN because it is promoted as a secure protocol, adopting a well-designed architecture and continuing security improvements of the protocol to disseminate data and produce key pairs in an inbuilt security [15] [35] [26].

Although LoRaWAN has observable advantages like lower costs, easy setup and maintenance, and long-range connection, it also has known flaws and poses immediate threats. Implementations of LoRaWAN frequently run into issues with keys and identity management. Once the keys are stolen, the LoRaWAN framework is exposed because encryption, the network's only security measure, depends on them [16]. For instance, in [37], the authors outline the LoRaWAN v1.0.2 standardized security features and analyze the effectiveness of the security measures in place under hostile conditions. They have identified five significant flaws that could jeopardize a LoRaWAN deployment's availability, confidentiality, and integrity. Additionally, the authors in [38] define and consider the potential energy attack vectors before conducting an experiment to confirm whether or not one of these vectors is feasible for an energy attack. The findings unequivocally demonstrate that LoRaWAN attacks involving energy are feasible and might even cause a compromised device to lose a significant amount of energy. The demonstrated attack specifically risen the overall power consumption during a point-to-point event from 36 percent to 576 percent depending on the device's SF (Spreading Factor). Notably, the attack used during the demonstration can be used against any LoRaWAN device and does not necessarily involve the attacker having any keys or other sensitive information.

security vulnerabilities of some communication standards, such as WiMAX and EAP protocols. For more information about the Scyther prover, please visit: https://people.cispa.io/cas.cremers/scyther/

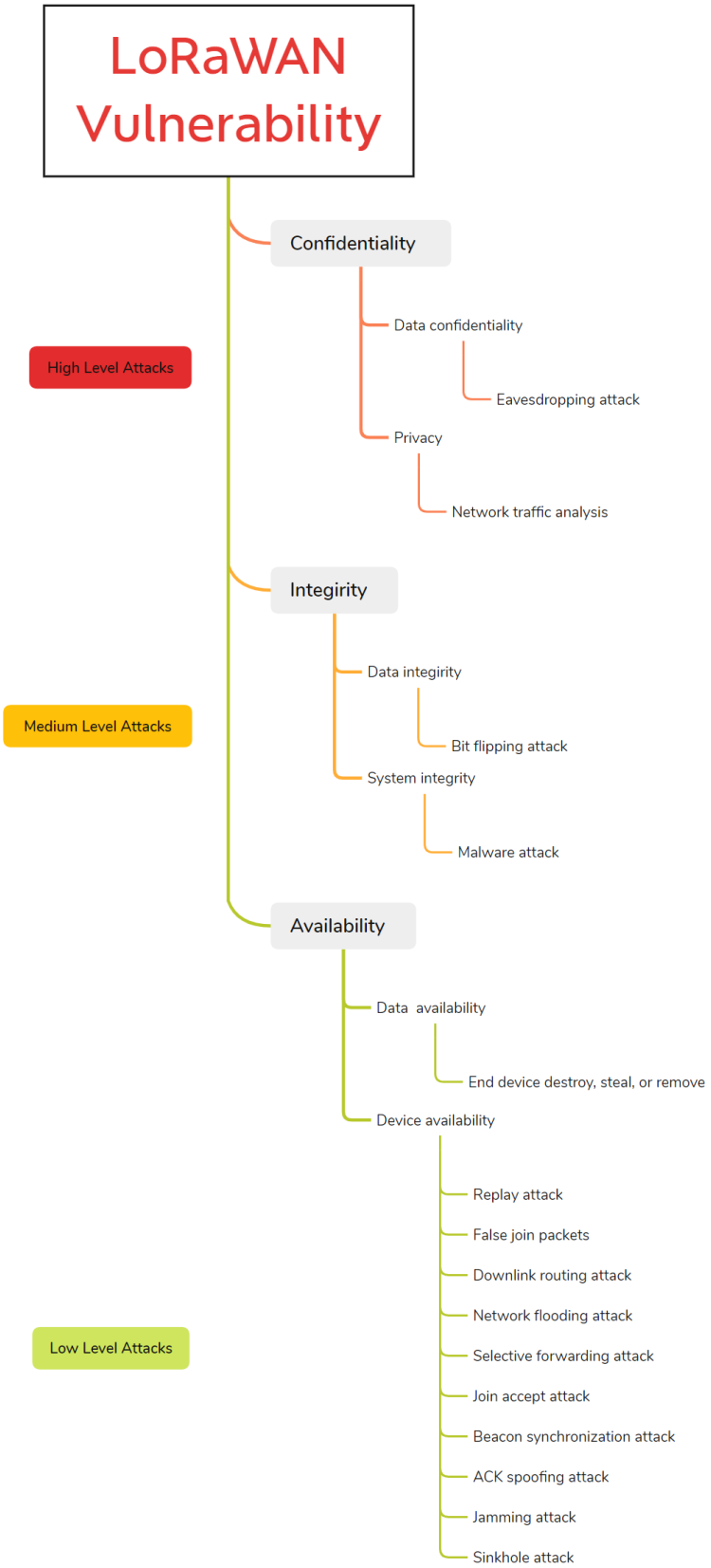


Figure 4.1: LoRaWAN vulnerabilities

4.1 Confidentiality

Confidentiality, in essence, pertains to safeguarding data from unauthorized access or exposure. In other words, data loses its confidentiality when it becomes accessible to unintended recipients, thereby potentially disclosing sensitive information to potential adversaries. Within the realm of LPWAN, this section delves into two fundamental aspects of confidentiality: data confidentiality and privacy. These aspects play a pivotal role in ensuring the secure transmission and protection of data in LPWAN networks.

4.1.1 Data confidentiality

Data confidentiality stands as a fundamental and indispensable strategy for safeguarding data from audiences who have no legitimate concern with it. In the context of LoRaWAN technology, data confidentiality assumes paramount importance due to several factors. These include transmitting packets over the Internet, remote data storage, engagement of third-party services, and considerations regarding packet latency. Notably, devices within the LoRaWAN infrastructure transmit packets, introducing potential security vulnerabilities owing to the broadcast nature of these packets within the network. An eavesdropping adversary can exploit this scenario to gain unauthorized access to confidential data, underscoring the critical need for robust data confidentiality measures.

4.1.2 Eavesdropping attack

This attack involves an intruder passively monitoring the network's traffic at a vulnerable point in order to gather information about data transmissions. Vulnerable points may include device cloning or key sniffing, which provide unauthorized access to the network. Acquiring the encryption key allows intruders to not only intercept but also manipulate the packets exchanged between the sender and receiver. Within the realm of LoRaWAN security threats, gateways play a pivotal role in collecting packets from end devices and forwarding them to network servers. Consequently, a rogue gateway in LoRaWAN has the potential to expose the network to significant security breaches.

4.1.3 Data privacy

Privacy constitutes a fundamental pillar of data protection within the IoT paradigm, where it primarily revolves around safeguarding the Personal Identifiable Information (PII) of end users from falling into unintended hands [39]. The exposure of PII can have significant repercussions, making data privacy a critical concern.

Data privacy breaches specifically pertain to the disclosure of sensitive information, and they involve the detection and understanding of the correlations among devices at the network's periphery. Privacy attacks pose notable challenges when developing new applications within this context. Fortunately, various algorithms and techniques documented in the literature offer potential solutions to address these privacy breaches and fortify the security of IoT systems. (Reference: [39])

4.1.4 Network traffic analysis

This process involves the detection, learning, and examination of data as it traverses the network. In the context of LoRaWAN, this scenario unfolds when an end device disseminates a packet to the gateway. In practical terms, a diverse array of network analysis tools and software, including but not limited to Scapy, Wireshark, and NMAP, serve as instrumental resources for dissecting and scrutinizing these transmitted packets. This comprehensive analysis aids in understanding the content, structure, and potential security implications of the data in transit.

4.2 Integrity

Integrity serves as a foundational element of data security, with its primary objective being the preservation of data accuracy, completeness, and reliability as it traverses the network. Its overarching goal is to ensure that data remains unaltered during its journey from source to destination, safeguarding it from potential tampering, deletion, or unauthorized additions. The concept of integrity can be dissected into two distinct parts: data integrity and system integrity.

4.2.1 Data integrity

Within the realm of LoRaWAN, data integrity emerges as a pivotal concern due to the inherent vulnerability of the devices used in this network. LoRaWAN devices are often scattered randomly across expansive and open environments, rendering them susceptible to a range of data integrity vulnerabilities. Among these vulnerabilities, two prevalent threats take center stage: replay attacks and bit-flipping attacks. These issues become particularly pronounced in large-scale deployments, necessitating robust measures to safeguard data integrity in LoRaWAN networks.

4.2.2 Bit flipping attack

Bit flipping attacks, characterized by their relatively low complexity and high incentive for malicious actors, involve a careful analysis of the packet content followed by strategic alterations to specific portions of the ciphertext. This manipulation occurs without the need for decryption. The tampering of ciphertext introduces distortions that can mislead the packet's intended recipient, leading to the dissemination of inaccurate information.

In LoRaWAN, end devices employ counter mode during packet transmission to gateways. This mode utilizes XOR operations to safeguard plaintext, ensuring that the data remains secure without undergoing reordering, which could otherwise render it vulnerable to bit flipping breaches (as discussed in [8]).

4.2.3 System integrity

System integrity in the context of LoRaWAN refers to the system's ability to operate its functions without encountering risks, mitigating potential misuse, and preventing unauthorized security breaches. This aspect forms the foundation for establishing trust and ensuring the integrity of communication between the device and the system. It encompasses a range of measures and practices aimed at maintaining the reliability and security of the overall system.

4.2.4 Malware attack

A malware attack at the application layer presents a significant security threat to the system, employing various malicious tools such as ransomware, trojans, and worms. In this type of attack, the attacker monitors the end user's activities and can manipulate information by introducing viruses into the system. Additionally, the attacker may have the capability to remove end users from the application system, compromising their access and data. This form of attack poses a severe risk to the integrity and security of the system's application layer.

4.3 Availability

Availability is a difficult aspect of ensuring that the system, information, and network are accessible to users whenever they need it within the specified time frame. However, availability can be compromised if issues arise in either the software or hardware components of the system. In the context of LoRaWAN, even a relatively simple attack, such as draining the battery of an end device, can have a detrimental impact on the availability of successful communication. This emphasizes the importance of safeguarding the network's resources to maintain reliable and uninterrupted service.

4.3.1 Data availability

Data availability is a critical aspect that ensures the data remains reliable and easily accessible for use. It focuses on the seamless continuity of information within the network. In the context of the LoRaWAN system, data availability challenges can arise due to the deployment of end nodes at the network's edge, often at a significant distance from the gateway. The distance between these nodes and the gateway may introduce potential availability issues, including signal disruptions, the risk of end device removal, or even theft. These factors highlight the need for strategies to maintain robust data availability in the LoRaWAN network, particularly in scenarios with challenging physical environments.

4.3.2 End device destroy, steal or remove

This type of attack leverages the end device to compromise the network's security by tampering with the root keys that are established during the manufacturing process. The exposure of a session key generated from these root keys within the LoRaWAN system can have detrimental effects on the transmitted data, potentially impacting aspects such as authenticity and availability. This underscores the critical importance of safeguarding root keys in LoRaWAN to prevent such security breaches and maintain the overall integrity and reliability of the network.

4.3.3 Device availability

Device availability in LoRaWAN is crucial to ensuring that devices are ready to receive and respond to messages from the gateway in a timely manner. If an end device is in a low-power state, it may miss down-link messages sent by the gateway. This can have a significant impact on the overall network's efficiency and the successful communication between devices and the gateway. Therefore, maintaining device availability is a key consideration for a well-functioning LoRaWAN network.

4.3.4 Replay attack

A replay attack happens when an intruder eavesdrops in the secure network and retransmits the valid data maliciously upon getting hands on it. The edge of this attack is to have no sophisticated knowledge required for an attacker to decrypt the packet seized from a network. This attack affects communication in the LoRaWAN system, and the attacker intends the weak point, for instance, to jam the OTAA joining method using the selective RF jamming method [25]. In this attack, the end device sends the join-request message with DevNonce to the network server. After waiting for a response from the network server in the given time frame, it retries another joining request message. In the meanwhile, the attacker acknowledges the network server for the first joining request. While the legitimate end device still waiting for a join-accept message. In this way, the attacker talks with the network server illegitimately.

4.3.5 False join packet

The attack's impact on network availability is generally considered unlikely, but it can have severe consequences if it occurs. To carry out a false join attack in LoRaWAN, the attacker typically needs two critical pieces of information: the JoinEUI and DevEUI of the end devices. Both these elements play a crucial role in the Over-The-Air Activation (OTAA) joining procedure, making them valuable targets for potential attackers. If an attacker gains access to these identifiers, they can exploit this information to disrupt the network's security and availability.

4.3.6 Down-link routing attack

This attack happens by adding a compromised gateway to the network. The downlink routing attack occurs when an attacker eavesdrops on successful communication. The end device sends an up-link message to the network server using the authentic gateway, at the same time the attacker eavesdrops and acknowledges the same message through the compromised gateway [35]. Consequently, a down-link error might occur when receiving duplicate packets from both gateways. The network server, however, de-duplicates the packets but the established network between the authentic device and gateway may not be ensured.

4.3.7 Network flooding attack

Butun *et al.* in [15] focused on the issue of flooding in LoRaWAN. They pointed out that an end device could potentially launch a flooding attack against the entire network by repeatedly sending the same packet. This could significantly disrupt the reliability of network communication. To mitigate this attack, end devices can be equipped with mechanisms to limit the airtime for packet transmissions. By setting airtime limitations, the network can better defend against flooding attacks.

4.3.8 Selective forwarding attack

Selective forwarding attacks have become a concern with the introduction of LoRaWAN 1.1 and are considered a minor risk. In this type of attack, packets are

selectively forwarded or repeatedly transmitted, disrupting smooth communication within the network. This attack can pose a threat whether the end device is using the Over-the-Air Activation (OTAA) or the Activation by Personalization (ABP) mode. By sending packets frequently with illegitimate methods poses blackhole threats on the network [40]. To address this issue, the network can benefit from the implementation of an Intrusion Detection and Prevention System (IDPS) to enhance security.

4.3.9 Joint accept attack

This attack has a critical impact on LoRaWAN technology, especially during the exchange of packets between the end node and the server. In this scenario, the end device communicates with the network server using the Over-the-Air Activation (OTAA) method, and the gateway facilitates the transmission of packets from the end device to the network server. When the network server receives a request, it sends a join accept message through a gateway. The attacker, motivated by this attack, attempts to intercept the join accept message before the legitimate end device and responds illegitimately to the network server. As a result, the network establishes connections with unauthorized devices, compromising the link between the end node and the server [9].

4.3.10 Beacon synchronization attack

Beacon synchronization poses a challenge for Class B devices within the LoRaWAN framework. This attack involves the broadcasting of counterfeit beacons by an attacker, achieved by deploying a rogue gateway within the network. As a result, Class B devices initiate multiple windows to receive downlink messages in a flooding manner, without proper synchronization with the legitimate gateway. This disruptive behavior escalates the likelihood of packet collisions during data exchange. To address this issue, Martinez and his co-authors (as cited in [41]) propose a potential solution by implementing a key at the gateway to mitigate beacon synchronization problems.

4.3.11 ACK spoofing attack

In LoRaWAN, a spoofing attack occurs when a malicious node impersonates another node, gaining unauthorized access to the network through the manipulation of data using illegitimate means. This type of attack is primarily focused on downlink messages and the acknowledgment of those messages. In a spoofing attack, the attacker initially intercepts a downlink message and subsequently alters it, causing the device it pretends to be to generate excessive uplink confirm messages, leading to a flooding of the network with spurious traffic [42]. Spoofing attacks can have significant consequences, particularly at the physical layer, and can result in severe network disruption and denial of service

4.3.12 Jamming attack

Jamming attacks pose a significant and critical challenge in LoRaWAN technologies. In a jamming attack, the attacker initially identifies the transmission frequency being used in the network and then tunes to the same frequency with the intention of disrupting communications. This disruption can be achieved by transmitting a high number of bits, packets, and non-continuous signal transmissions on a specific channel. Jamming attacks are categorized into four different classes, which include constant jamming, deceptive jamming, random jamming, and reactive jamming [43]. In LoRaWAN, jamming is particularly effective because end devices utilize the Chirp Spread Spectrum (CSS) modulation technique, which spreads the signal over a wide frequency range on its way to the gateway. Detecting jamming attacks in the network can be quite challenging, but practical methods involve measuring the signal strength to identify anomalies.

4.3.13 Sinkhole attack

In sinkhole attacks, the attacker diverts the network's entire traffic along a specific route, essentially creating a "sinkhole" for the network's communication. This type of attack involves setting up a dedicated path for communication with other nodes as a means to carry out malicious activities. By doing so, the attacker causes the network to create a tunnel, negatively impacting the performance of

nodes within the network [44]. Sinkhole attacks are considered intermediate-level attacks, and they can have significant consequences, including the potential for denial-of-service within the network.

4.3.14 Rogue-gateway attack

The research conducted by Mohamed and colleagues [45] sheds light on the sophisticated tactics employed by attackers to compromise LoRaWAN networks. These Rogue Gateway Attacks involve the impersonation of legitimate gateways, a strategy that enables a range of disruptive actions within the network. These actions encompass packet dropping, which can result in data loss and unreliable communication, as well as black-hole attacks that divert network traffic into a void. Furthermore, the attackers employ worm-hole and selective forwarding techniques, which pose severe security threats to the network's integrity and can lead to data compromise and network disruptions. The study highlights the importance of addressing these evolving threat vectors to ensure the continued security and reliability of LoRaWAN deployments.

4.4 Countermeasures addressing LoRaWAN vulnerabilities

In light of the significant security challenges that affect the LoRaWAN system, this section provides a comprehensive set of recommendations, as illustrated in Figure 4.2. These recommendations are pivotal when it comes to crafting a secure LoRaWAN system. Further down in the subsections lists detailed and comprehensive recommendations for addressing each specific security vulnerability. These guidelines are helpful in ensuring that the system not only remains resilient but is also well-prepared to effectively counter potential threats and breaches.

4.4.1 Eavesdropping

This attack directly impacts the confidentiality of the network, where the attacker exploits plain text to intercept other plain text information being exchanged. To

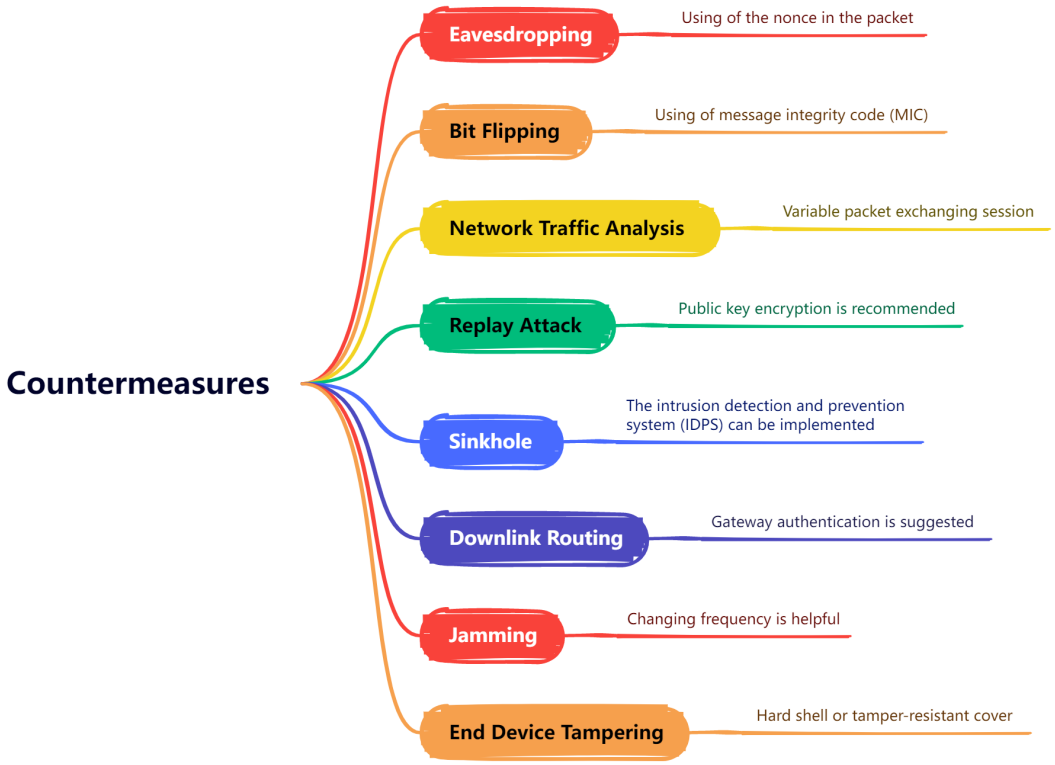


Figure 4.2: Countermeasures associated with some of the LoRaWAN vulnerabilities.

safeguard the system against this form of attack, it's imperative to incorporate nonces in the packets. Nonces serve as a protective measure to mitigate this threat effectively.

4.4.2 Bit Flipping

In this type of attack, the network layer can be exploited to manipulate the message content. To mitigate this threat, it is recommended to implement a message integrity code (MIC) at the application server. MIC plays a crucial role in ensuring the message's integrity and protecting it from unauthorized alterations.

4.4.3 Network Traffic Analysis

The analysis of network traffic often involves the use of various software and tools, which are employed to monitor successful communications. To make this type of attack more challenging, it is advisable to introduce variability in the packet exchange sessions. This approach can enhance the security of the network by introducing unpredictability and thwarting potential attackers.

4.4.4 Replay attack

This attack adversely impacts the MAC (Medium Access Control) layer within the network, primarily during the end device's joining process with the network server. To counter this attack, it is advisable to employ public key encryption, as detailed in the paper [46]. This encryption method serves as an effective countermeasure to mitigate the effects of this particular attack.

4.4.5 Sinkhole

In a sinkhole attack, a dedicated path is established to channel overall network traffic through a single compromised node. To address this concern, implementing an Intrusion Detection and Prevention System (IDPS) within the network is a recommended strategy. The IDPS plays a critical role in detecting and preventing such attacks, and enhancing network security and resilience.

4.4.6 Downlink routing

A downlink routing attack can be executed through a compromised gateway. To prevent this type of attack, it is essential to authenticate the gateway before integrating it into the network. Gateway authentication serves as a critical security measure to thwart potential threats and maintain the network's integrity.

4.4.7 Jamming

The jamming attack is a widely recognized threat in RF-based communication technologies. Attackers can execute a jamming attack by analyzing the trans-

mission frequency and altering the number of bits in the intercepted packet. To mitigate jamming issues in LoRaWAN, changing the frequency can be a valuable tactic. By shifting frequencies, the network can effectively address and mitigate the impact of jamming attacks.

4.4.8 End device tempering

To enhance the security of the network, the physical tampering of end devices can be made more challenging by incorporating hard shells or tamper-resistant covers. This type of attack is more likely to occur because end devices are typically deployed at the network's edge, making them easily accessible to potential attackers. Implementing protective measures like hard shells or tamper-resistant covers helps safeguard the end devices and ensures the overall integrity of the network.

4.4.9 Rouge gateway attacks

In the paper [45], the authors introduced a novel technique aimed at enhancing the cybersecurity of LoRaWAN gateways. This approach involves the adoption of a Public Key Infrastructure (PKI) comprising a 2-tier Certificate Authority (CA) solution. The 2-tier CA resolves the vulnerabilities associated with a single-point failure setup by implementing a root-CA and intermediate-CA configuration. According to the authors, simulation results demonstrated the effectiveness of this technique in successfully mitigating malicious attacks originating from rogue gateways, including Selective Forwarding Attacks.

4.5 Replay Attacks in LoRaWAN

In the realm of LoRaWAN, a replay attack refers to a type of cyber attack where an adversary captures and subsequently replays a legitimate message sent between an end node and a network server. Numerous research studies have been published in the literature that tackle replay attacks in the LoRaWAN network. The following research discusses replay attacks:

4.5.1 Protecting end-device from replay attack on LoRaWAN

Sung *et al.* [7] analyzed the replay attack in the LoRaWAN implementation during the join procedure of the end device and the network server. To protect the network against the replay attack, the LoRaWAN standard recommends a user identification method also known as DevNonce, but this may not be an optimal solution to protect the network. Therefore, the authors proposed a method by using a user's physical characteristics of a network known as RSSI (received signal strength indicator) and proprietary handshaking. The concept of this work is good; however, it has not been verified through practical implementation.

4.5.2 A simple and efficient replay attack prevention scheme for LoRaWAN

Kim and Song [6] discussed that recent studies reveal that the current techniques used to prevent replay attacks in LoRaWAN can mistakenly identify legitimate messages as replay attacks. To address this issue, various replay attack prevention methods have been suggested. Nonetheless, these existing approaches come with limitations. Some are not compatible with the current packet structure, while others fail to account for exceptional scenarios like device resets. For this reason, they proposed a new LoRaWAN replay attack prevention scheme that effectively tackles these issues. This approach maintains the existing packet structure and takes into account scenarios like device resets. Through calculations, it is shown that their scheme substantially decreases the possibility (by 60-89%) of wrongly identifying a normal message as a replay attack, surpassing the performance of the current LoRaWAN approach. Their results are also substantiated by real-world experiments. By employing the designed detector, it is anticipated that users who have encountered a replay attack can be protected while still being able to maintain their ongoing connections.

4.5.3 Protecting gateway from ABP replay attack on LoRaWAN

In [48], the authors conducted an analysis of the attack surface for end devices that were activated using the ABP authentication method. Their investigation led to the discovery of countermeasures aimed at detecting and mitigating replay attacks. The authors began by considering the attacker's behavior, using a simulated experimental attack as their basis. And, condensed this attack into specific points, which were then replicated across the campus network. Subsequently, they elucidated how to prevent such an attack by incorporating this approach into an algorithm. In the implementation phase, the authors integrated the detector into the gateway and conducted real-time testing with a dataset. They analyzed the outcomes to describe how the received data behaved on the application server, when the detector was applied.

4.5.4 Scenario and countermeasure for replay attack using join request messages in LoRaWAN

SeungJae *et al.* [49] reveal that the join request messages are unable to be encrypted due to the absence of a session key before the completion of Over-the-Air Activation (OTAA). Consequently, the contents of these join request messages are extracted without undergoing decryption. And malicious entities can readily seize these messages using sniffing tools, making it uncomplicated to pilfer and exploit their contents without needing decryption. They proposed a countermeasure using the XOR masking in the join request message. The end device exchanges the masked join request message with the network server, and revert back by applying the dedicated token. In this case, if the attackers sniff the join request message, they may not be able to exploit it since each message is concealed with distinct tokens.

4.6 Denial-of-Service Attacks in LoRaWAN

Denial-of-Service (DoS) attacks in LoRaWAN can disrupt the regular functioning of LoRaWAN networks and the associated IoT devices. These attacks are designed to flood the network infrastructure, gateways, or end nodes with an overwhelming volume of traffic, rendering them inaccessible to genuine users. The following research offers a thorough and comprehensive analysis of DoS attacks.

4.6.1 Denial-of-service attacks on LoRaWAN

The researchers in [9] presented a comprehensive security analysis in which a series of vulnerabilities were identified, thereby revealing susceptibilities that could potentially be leveraged for the perpetration of Denial of Service (DoS) attacks targeting end-devices. The researchers employed the Colored Petri Nets model to verify vulnerabilities related to beaconing, downlink routing, and join accept replay. These vulnerabilities were found to lead to DoS attacks within the LoRaWAN implementation.

4.6.2 Detecting denial-of-service attacks in LoRaWAN

In [50], the authors conducted a comprehensive investigation into the susceptibility of LoRaWAN protocols to packet collision and reactive jamming. They utilized datasets collected from an experiment to explore this issue. Through empirical research and the application of machine learning techniques, they aimed to determine whether a signal transmission was normal or tampered with. Furthermore, various metrics were employed to detect attacks, including Packet Delivery Ratio (PDR), Packet Inter-Arrival Time (IAT), and a comparison of modem settings at the transceiver and gateway to provide insights into how these attacks impacted the system. Additionally, binary classifiers were developed using specific metadata to identify abnormal changes and differentiate between collision and jamming attacks. The research underscores the significance of in-depth profiling of LoRa metadata for a better understanding of LoRa's security landscape. This knowledge can be instrumental in devising effective countermeasures against DoS attacks, which is crucial in the context of the rapidly growing IoT platform

that LoRa represents.

4.7 Key Management in LoRaWAN

4.7.1 An improved Key distribution and updating mechanism for low power wide area networks (LPWAN)

The authors in [51] discussed the vulnerabilities in the existing LoRaWAN's key exchanging method and proposed the Key distribution and updating mechanism (KDUM) to strengthen the network's security. The KDUM protocol makes use of updating the root and session keys to become the network more resilient against cyber-attacks. Furthermore, the authors argued that asymmetric algorithms for instance; RSA is demanding high computational power, while IoT devices have low computational abilities. Therefore, they considered the DH algorithm to exchange the session key remotely. To configure the algorithm robust, two varieties i.e., discrete logarithms in a finite field (DH), and ECDH were available. However, they considered ECDH as it utilizes less number of bits and is high-security level. ECDH consumes less energy as compared to the DH and RSA algorithms, and hence the sensor node can be active for a long time. The KDUM protocol possesses a good security framework. However, the key limitations of the LoRaWAN *ED* are not addressed with this algorithm.

4.7.2 ECDH based Key management for LoRaWAN considering sensor node limitations

The conventional LoRaWAN implementation relies on security keys, as the root keys are hardcoded in the *ED*s at the time of the manufacturing process. The root keys are then used to create the session keys to secure communication. However, there might be security risks if the security keys are generated for a long duration. Therefore, the keys management method is proposed in [52] that is practically applied on the sensor node. The ECDH protocol with Micro-ECC setup is tested with various curves and evaluates the network's efficiency, battery power consumption, and security strength. The ECDH implementation is em-

ployed with the MCCI-LMIC library and disabled some extra features that cause energy consumption. This protocol achieves good security with the cost of low flash memory consumption.

4.7.3 Secure session key generation method for LoRaWAN servers

Lin Tsai *et al.* in [53] pointed out an issue with the existing LoRaWAN's key commissioning method. According to the authors, the network entity in LoRaWAN v1.0.x generates and utilizes the AES based key to authenticate communication and guarantee the integrity between *ED*s and *AS*. However, the communication of different servers is not defined in the specification. Therefore, a server session key generation (S2KG) approach is presented to securely establish communication among different servers. In the S2KG method, the *NS* utilizes symmetric key cryptography, while the *JS* makes use of elliptic curve cryptography (ECC) to enhance the security of the communication. The S2KG method achieves good confidentiality, packet integrity, and mutual authentication while generating session keys.

4.7.4 Enhancing LoRaWAN security through a lightweight and authenticated key management approach

In [54], the authors discussed security vulnerabilities in the LoRaWAN v1.0 and proposed a key management scheme. The proposed scheme is based on the DH key exchange method which is considered a convenient remedy because of the less computational overhead and use of fewer cryptographic bits. This scheme is then compared with the IKEv2 [55], DTLS [56], and EDHOC [60], which is found a feasible solution to enable the key exchange process. However, the work is solely compliant with the LoRaWAN v1.0.x and it does not cover the newer version of LoRaWAN v1.1. Therefore, this work needs to be updated considering the security requirement of version v1.1.

4.7.5 An enhanced Key management scheme for LoRaWAN

Han and Wang in [19] argued that the security of LoRaWAN v1.1 comes up with the basic security requirements. The current key generation method in LoRaWAN uses AES-ECB which is considered not a viable choice as it is vulnerable to pattern analysis. In other words, the AES-ECB consumes more battery power from the sensor node. Therefore, they proposed the Rabbit Cipher-based key management technique to enhance the complexity of cryptoanalysis of the security keys. The rabbit is a stream cipher that is used in the two-step key derivation function to obtain the pseudo-random number generator. Simulation results reveal that the proposed method achieves good performance in terms of less computing than the LoRaWAN key generated mechanism. Also, it provides a high number of randomness which makes the protocol more robust against cryptoanalysis.

4.7.6 A dual key-based activation scheme for secure LoRaWAN

Kim and Song in [57], stressed the security loopholes in the existing LoRaWAN's key update and session key generation. They considered a dual key-based approach for LoRaWAN and fixed the issue of key updates not receiving full support. In this approach, the NwkKey and AppKey are used to generate the initial join request process. Then the session keys that are derived from the previous join process are considered for the second join process. The process solves the issue of the update which is not taken into consideration in the existing LoRaWAN implementation. Furthermore, this approach allows each layer to establish a unique session key so that the layers can operate separately. The real-world experiment demonstrates that the proposed approach outperforms in terms of battery consumption and latency as compared to the original LoRaWAN implementation. However, this approach does not guarantee perfect forward secrecy, since the session keys are generated from the previous session.

4.7.7 Activation of LoRaWAN end devices by using public key cryptography

Another interesting research carried out by Marlind and Butun in [46]. According to the authors, the *ED*s in LoRaWAN come up with reconfigured root keys. And the session keys are generated using the root key, so there is a potential of key disclosing if the *ED* is susceptible to security breaches. Changing the root key in the *ED* is cumbersome since it requires a physical presence to the device. They provide an alternative remedy whereas the root key can assign remotely using public key cryptography. This method is also known as the public key over the air activation (PK-OTAA) since it uses elliptic curve cryptography (ECC). The main reason for using the ECC is because of the shorter key length size. Furthermore, the root key is generated using ECDH that exchanges the secret value. This protocol provides a significant security improvement to the LoRaWAN joining procedure. However, it consumes more battery power compared to the original LoRaWAN.

4.7.8 A complete key management scheme for LoRaWAN v1.1

The authors in [58] argued that the new release of LoRaWAN i.e., v1.1 has significantly improved the security framework; however, the process of key delegation is still vague and needs to be defined explicitly. They proposed a new key management setup to tackle the issues with key updating, generation, and exchanging from the node through the server. The proposed setup considered the Rabbit Cipher stream that takes a 128-bit secret and 64-bit initial vector as input. Simulation reveals that the Rabbit Cipher algorithm is much faster than the AES-ECB mode and consumes less battery power.

4.7.9 A secure and efficient blockchain-based key management scheme for LoRaWAN

To enhance the authenticity and availability of LoRaWAN, the permissioned-blockchain-based key management technique is proposed in [59]. The new architecture of the LoRaWAN is considered whereas the *NS* and the *JS* are controlled through blockchain and the key exchange process is accomplished using ECDH which provides high security with minimal resource usage. To assess the achievement, this approach is compared with the ChirpStack *NS*, and the formal security prover known as AVISPA is used to verify the security. Simulation results reveal that the proposed approach outperforms in terms of packet processing latency.

4.7.10 A novel secure root key updating scheme for LoRaWANs based on CTR_AES DRBG 128

Hayati *et al.* in [18] proposed an approach that used the Photon-256 algorithm to produce a unique session key for secure communication in LoRaWAN. This approach first considers the initialization stage for the session key at the *ED*. And the contents of the session key are collected at the *JS*. In addition, a set of key pairs i.e., NwkSKey and AppSKey are created at the *ED* and the *NS*. This approach is verified and validated via GNY logic and the protocol prover known as Scyther. The proposed approach achieves a good security framework and economical solution. However, it is not fully compliant with the LoRaWAN implementation. The key features of comparison is analyzed in table 4.1.

Table 4.1: Analysis of different approaches for key update in LoRaWAN.

Article	Algorithm's Approach(s)	Achievements	Flaws	Year	Ref
M. Leent *et al.*	Proposed a key generation algorithm	The protocol possesses a good security framework that enhanced the existing LoRa end	The formal security verification is not considered	2017	[51]
E. Jayasuriya *et al.*	keys management method is proposed that is practically applied on the sensor node	Evaluates the network's efficiency, battery power consumption, and security strength	Disabling existing LoRaWAN feature may lost the quality of service	2021	[52]
Lin Tsai *et al.*	A server session key generation (S2KG) approach is presented to securely establish communication among different servers	The algorithm achieves good confidentiality, packet integrity, and mutual authentication while generating session keys	The algorithm is not considered on practical implementation, therefore, it may put extra burden on node's memory	2020	[53]

Continued on next page

Table 4.1: Analysis of different approaches for key update in LoRaWAN. (Continued)

Article	Algorithm's Approach(s)	Achievements	Flaws	Year	Ref
Sanchez Iborra *et al.*	Proposed a scheme based on the DH key exchange method which is considered a convenient remedy	A feasible solution to enable the key exchange process	Compliant with the LoRaWAN v1.0.x and it does not cover the newer version of LoRaWAN v1.1	2018	[54]
Han and Wang	Proposed the Rabbit Cipher-based key management technique to enhance the complexity of cryptoanalysis of the security keys	Outperforms in terms of less computing than the LoRaWAN key generated mechanism	No perfect forward secrecy is guaranteed	2018	[19]

Continued on next page

Table 4.1: Analysis of different approaches for key update in LoRaWAN. (Continued)

Article	Algorithm's Approach(s)	Achievements	Flaws	Year	Ref
Kim and Song	Considered a dual key-based approach for LoRaWAN and fixed the issue of key updates not receiving full support	Outperforms in terms of battery consumption and latency as compared to the original LoRaWAN implementation	This approach does not guarantee perfect forward secrecy, since the session keys are generated from the previous session	2017	[57]
Marlind and Butun	Public key over the air activation (PK-OTAA) is proposed to delegate the keys securely	It saves the join request message from external attacks	Consumes more battery power compared to the original LoRaWAN	2020	[46]
Chen *et al.*	This algorithm considered the Rabbit Cipher stream that takes a 128-bit secret and 64-bit initial vector as input	Achieves less good battery power management	Requires high computing efficiency	2021	[58]

Continued on next page

Table 4.1: Analysis of different approaches for key update in LoRaWAN. (Continued)

Article	Algorithm's Approach(s)	Achievements	Flaws	Year	Ref
Tan *et al.*	The permissioned-blockchain-based key management technique is proposed	Enhanced the authenticity and availability of LoRaWAN by keeping in view the low latency of processing	Hyperledger fabric is more energy-demanding, as LoRa nodes operate with limited battery power	2021	[59]
Hayati *et al.*	Proposed an approach that used the Photon-256 algorithm to produce a unique session key for secure communication in LoRaWAN	Achieved good security as it has validated via GNY logic and the protocol prover known as Scyther	Not fully compliant with the LoRaWAN implementation	2022	[18]

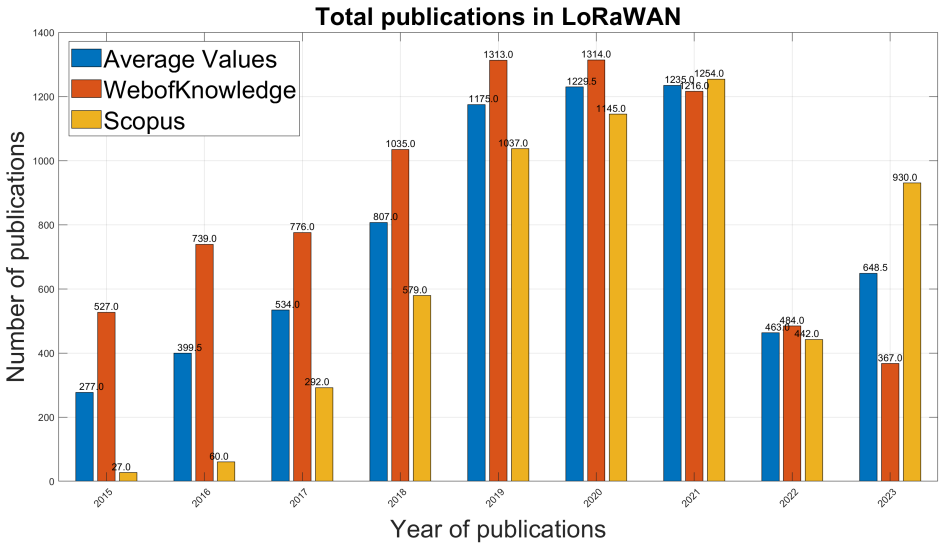
4.8 The bibliometric overview of cyber risks and threats in LoRaWAN

This section offers a comprehensive bibliometric analysis of LoRaWAN, drawing data from two prominent databases, Scopus and Web of Knowledge. To conduct this analysis, a meticulous search query was devised, incorporating relevant terms such as ("LoRa" OR "LoRaWAN") AND ("Security" OR "Cybersecurity"). The data retrieved from both databases was systematically gathered, and the results are eloquently visualized using Matlab. Figure 4.3(a) unveils a chronological publication record spanning the years 2015 through 2023, revealing notable growth in publications with a significant peak observed in 2020. This trend underscores the increasing relevance and interest in LoRaWAN technology. In Figure 4.3(b), the focus shifts to the specific realm of LoRaWAN security, providing insights into the number of publications in this domain. This graphic representation underscores the increasing importance of security considerations in LoRaWAN implementations. For an even deeper understanding of the landscape, Table 4.2 presents specific details about the number of papers addressing distinct attack vectors within the realm of LoRaWAN security. This granular insight is valuable for researchers, practitioners, and stakeholders interested in the nuances of security challenges in LoRaWAN networks.

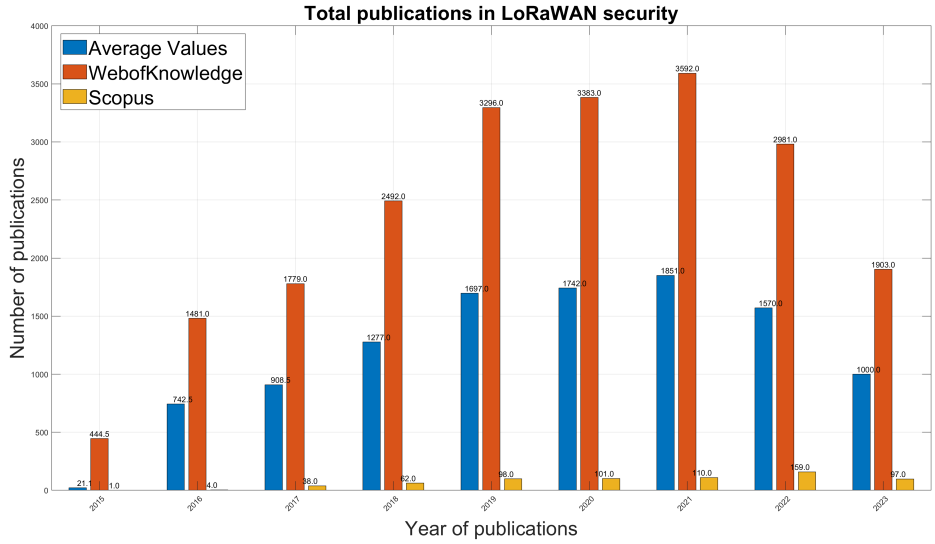
Table 4.2: Papers dealt with various attacks

String searched	Papers dealt with attacks		Type of documents	
Attacks	Scopus	WebofKnowledge	Article (S+W)	Conference proceedings (S+W)
"LoRAWAN" AND "Key related vulnerabilities"	1	1	(1),(1)	(0),(0)
"LoRAWAN" AND "Plain-text Key Capture"	0	0	(0),(0)	(0),(0)
"LoRAWAN" AND "Eavesdropping Attack"	11	8	(3),(3)	(8),(5)
"LoRAWAN" AND "Bit Flipping Attack"	5	3	(1),(1)	(4),(2)
"LoRAWAN" AND "Device Cloning"	1	1	(0),(0)	(1),(1)
"LoRAWAN" AND "Replay Attack"	35	25	(11),(10)	(24),(15)
"LoRAWAN" AND "Wormhole Attack"	2	1	(0),(0)	(2),(1)
"LoRAWAN" AND "Selective Forwarding Attack"	0	0	(0),(0)	(0),(0)

(S+W) = Scopus +WebofKnowledge



((a)) String searched ("LoRa" OR "LoRaWAN")



((b)) String searched ("LoRa" OR "LoRaWAN")AND("Security" OR "Cybersecurity")

Figure 4.3: a) Total number of papers published in LoRaWAN, b) Total papers published in LoRaWAN security.

Chapter 5

The key generation and distribution (KGD)

Summary

In this section, a detailed explanation is provided regarding the intricate processes involved in generating and distributing keys within the infrastructure of LoRaWAN. It offers a comprehensive overview, outlining the intricate mechanisms and operations essential for this crucial aspect of the system's functionality.

5.1 Key generation

The first step that makes up the KGD implementation into practice is the key generation process. The KGD method uses the NIST-approved encryption algorithm named AES counter-based deterministic random bit generator (AES-CTR-DRBG), which is a type of random number generator based on a block cipher, and uses a counter mode to generate random numbers [61]. The strength of AES-CTR-DRBG depends on the block cipher used, and AES-128 is considered to be a secure choice. This algorithm turns a fixed-length block cipher into a stream cipher. It generates a keystream by encrypting a counter and then XORing the keystream with plaintext. The AES-CTR provides a high degree of randomness that makes the system more difficult to predict as it was verified in [62]. Also,

the research in [18] utilized this method for obtaining random bits in a key updating procedure in LoRaWAN. However, researchers in [63] and [64] stressed the vulnerability found in the AES-CTR. A practical attack is possible on this technique and could obtain the internal input information. Cohney *et al.* suggested that this attack could be avoided by updating the input seeds frequently. But, frequent update of the input seeds is not an optimal solution and the LoRa devices are equipped with limited power and processing unit.

In practice, we used a well-vetted, cryptographic-grade random number generator library to seed the DRBG, such as Argon2, a password hashing winner in 2015 [66]. Argon2 itself is a key derivation function that is designed to be secure against various attacks such as brute-force and dictionary attacks. It is not a DRBG itself, but it can be used to derive a cryptographic key that can be used as a seed for a DRBG. Argon2 can help AES-CTR DRBG in terms of security by using it to derive a key for the AES-CTR DRBG from a password or a secret. This would provide an additional layer of security because an attacker would need to know both the password and the key in order to predict the output of the DRBG. In other words, it can help AES-CTR DRBG in terms of security by using it to derive a nonce for the AES-CTR DRBG. Using a nonce together with a key in AES-CTR DRBG increases the security of the generated random bits. The Argon2 lists several input parameters as shown in Fig. 5.1, whereas T is the time cost that controls the number of iterations, M shows the amount of memory being used. The parameters P and H show parallelism that controls the number of threads, and hash length, consequently. The last two parameters P_s show the input password to be taken, while S is the salt. All these steps can make the Argon2 algorithm more suitable for a 32-bit microprocessor. DRBG takes all these parameters as input and passes through AES-CTR mode with the static key S_k to obtain both the NwkKey and AppKey at both the *JS* and end node. The JoinNonce is a random variable that is generated at the *JS* while DevNonce is at the end node. Now, we have root secrete keys as shown in the red and green colors in Fig. 5.2, but updating them at the *ED* is challenging because it is often deployed in a scattered remote area where physical access is not possible. To update the keys, both instances needed to perform a remote key exchanging method which is discussed in the next section.

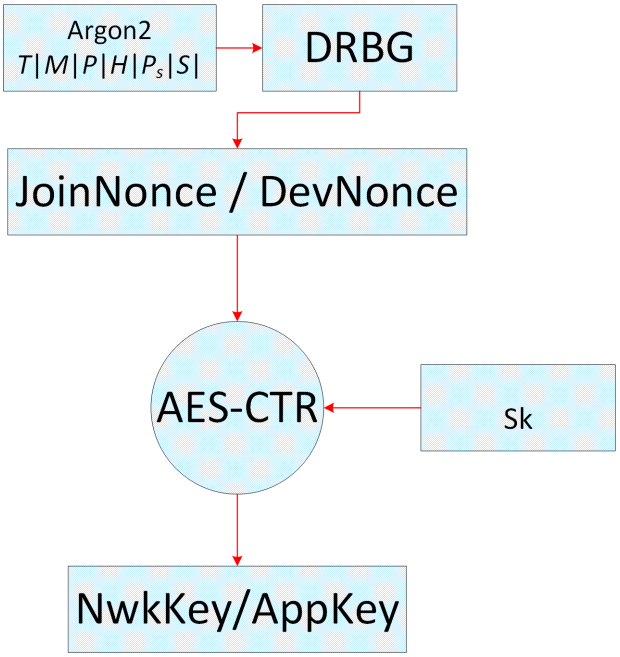


Figure 5.1: Root keys generation in the *JS*/*ED*.

5.2 Key distribution

The root keys are then used to generate several session keys that are distributed among several servers and the *ED*. The purpose of each session key is discussed in Chapter 3 section 3.2. To exchange these session keys remotely, a key exchange algorithm is required that distributes the session key to the LoRa end. Several key exchange methods are available in the literature; however, this work considers the Diffie-Hellman (DH) key exchange method. Other key exchanging methods like RSA, which consumes more power and is highly computationally demanding. To use the DH method, two variants such as DH with discrete logarithms in a finite field and ECDH could be utilized. The description of each method is explained below.

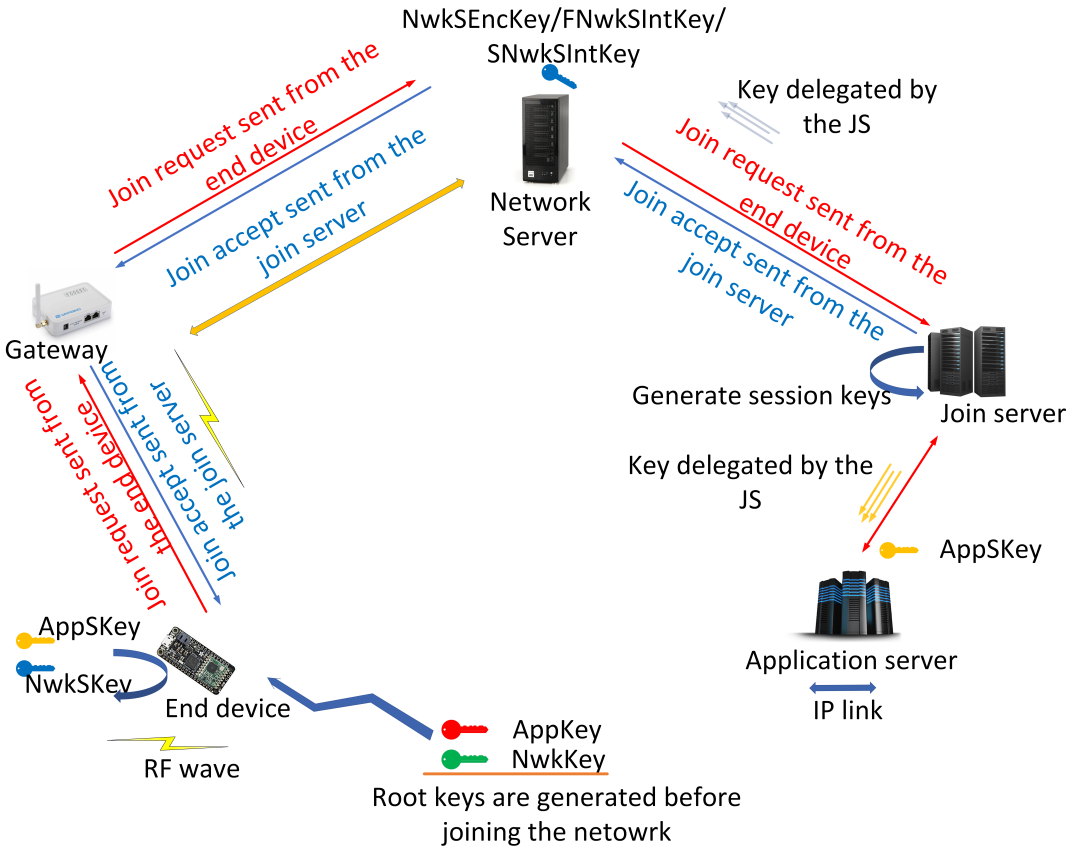


Figure 5.2: Root and session key delegation at the *JS*/*ED*.

Table 5.1: Notations used in the proposed approach.

Notation	Definition
ED_{pub}	Public key of the *ED*
ED_{sec}	Secrete/private key of the *ED*
G	Generator point
JS_{pub}	Public key of the *JS*

Continued on next page

Table 5.1: Notations used in the proposed approach. (Continued)

Notation	Definition
JS_{sec}	Secrete/private key of the JS
ED_{sh}	Shared key of the ED
JS_{sh}	Shared key of the JS
$\mathbb{I}$	Identity element
h	Hashed message
k	Random bit generated
ED_ith	The number of EDs
J_sr	The number of EDs

5.2.1 Key distribution with Diffie Hellman

DH is the earliest mathematical model that is used to exchange cryptographic keys securely over a public/insecure channel. In the context of LoRa session keys, the devices (ED_{ith}) and the JS (J_{sr}) generate a private key also known as a secret key. To employ DH key exchanging algorithm, all the ED_{ith} and J_{sr} agree on a prime P and generator point G. The term G must be a primitive root of P and must consider lower than P. Using these parameters, ED_{ith} and J_{sr} exchange a 128-bit long shared key as shown in the equations below.
ED_{ith} calculates public key (ED_{pub}) from its secrete key ED_{sec}.

$$ED_{pub} = G^{ED_{sec}}(modP) \tag{5.1}$$

J_{sr} calculates public key JS_{pub} from the secrete key JS_{sec} in the same fashion as the ED_{ith}.

$$JS_{pub} = G^{JS_{sec}}(modP) \tag{5.2}$$

ED_{ith} computes the shared key ED_{sh} from its ED_{sec} and JS_{pub}.

$$ED_{sh} = JS_{pub}{}^{ED_{sec}}(modP) \tag{5.3}$$

J_{sr} computes the shared key JS_{sh} from its JS_{sec} and ED_{pub}.

$$JS_{sh} = ED_{pub}{}^{JS_{sec}}(modP) \tag{5.4}$$

The shared keys i.e., ED_{sh}==JS_{sh} works in the same fashion as the current root keys do in the LoRa end. Using shared keys, several session keys i.e., $FNwkSIntKey$, $SNwkSIntKey$, $NwkSEncKey$, and $AppSKey$ can be generated to put an extra security layer on the payload being exchanged between the ED_{ith} and multiple servers. The key notations used in these equations are depicted in table. 5.1.

5.2.2 Key distribution with Elliptic Curve Diffie Hellman

Elliptic curve: An elliptic curve is an algebraic curve that consists of points (x,y) over finite fields $\mathbb{F}$p (where p is a prime number) and is generated by the following cubic degree

$$Ax^3 + Bx^2y + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J = 0, \tag{5.5}$$

however, cryptography uses the simplest, as such the Weierstras form [**?**] which is given as

$$Y^2 = x^3 + ax + b(modp), \tag{5.6}$$

and the visual plot is shown in Fig. 5.3(a). Elliptic curves such as NIST curve secp256k1 are widely used in bitcoin's public key cryptography defined by the standard for efficient cryptography (SEC) from the Certicom research [67]. All curves must make sure if they satisfy the condition given in the equation below

$$4a^3 - 27b^2 \neq 0. \tag{5.7}$$

The mathematical intuition of elliptic curves over a group G contains elements along with a single point of addition (denoted by $+$) and has the following properties:

- The addition operation of two points a and b from G results in another point c such as $a + b = c$ denotes $a, b, c \in G$
- The commutative operation from G is $a + b = b + a$
- Three points from G follow associative operation, $a+(b+c) = (a+b)+c = a + b + c$
- There is possible identity element $\mathbb{I}$ such that $a + \mathbb{I} = a$
- There exists an inverse element for every element from G as $-a$ such that $a + (-a) = 0$

If a line passes through the two points $P + Q$ as shown in Fig. 5.3(a) it will also intersect the third point $P + Q + R = \mathbb{I}$ on the curve as drawn in Fig. 5.3(b). There is a projection of the third point R. Thus, the addition property $P + Q$ results in an inverse point $-R$, as shown in Fig.5.3(c). Furthermore, if another point S is added to the resultant inverse point $-R$ as can be seen in Fig. 5.3(d), it will touch another point and will produce another inverse in the same fashion. Thus, the security of a system depends on the difficulty of solving a problem involving many points on elliptic curves. Elliptic curves can be used for a variety of cryptographic tasks i.e., key exchange, digital signatures, encryption, etc. We now distribute all the keys (root and session keys) between the ED_{ith} and J_{sr} using the ECDH algorithm which is explained in the subsection below.

LoRaWAN's Keys with ECDH: ECDH provides a convenient and secure method of key exchanging between the ED_{ith} and J_{sr} in LoRaWAN standard. As discussed the key generation process in Section **??**(A), we now exchange the key in the given KGD solution. To take the ECDH into consideration, the ED_{ith} and J_{sr} must choose a private key that will use in the public key of each other. To generate the shared key, the ECDH uses the curve given in the equation 5.6 that helps in determining the shared keys. To perform the ECDH, the ED_{ith} and J_{sr}

must agree on the prime (p) number generator (G) point publicly. Both parties must satisfy the requirements given in equation 5.7; otherwise, the points may lie outside of the chosen curve. The key exchanging of LoRaWAN *ED* and server using ECDH follows as:

- If E is the elliptic curve over a finite field with a prime number such as $E\mathbb{F}p$, then the key exchange takes place as
- The ED_{ith} and J_{sr} chooses a point P in $E\mathbb{F}p$
- The ED_{ith} selects a secrete key ED_{sec} and calculates the public key ED_{pub}. In mathematical terms,

$$ED_{pub} = ED_{sec}.P \in E\mathbb{F}p, \tag{5.8}$$

- The ED_{ith} shares ED_{pub} with the J_{sr}
- The J_{sr} chooses a secrete key JS_{sec} and computes the public key JS_{pub}. In mathematical terms,

$$JS_{pub} = JS_{sec}.P \in E\mathbb{F}p, \tag{5.9}$$

- The J_{sr} shares JS_{pub} with the ED_{ith}
- The ED_{ith} computes the shared key received from the J_{sr} as

$$\begin{aligned} ED_{sh} &= ED_{sec}.JS_{pub} \in E\mathbb{F}p, \\ &= ED_{sec}.(JS_{sec}.P) \in E\mathbb{F}p, \\ &= ED_{sec}.JS_{sec}.P \in E\mathbb{F}p, \end{aligned} \tag{5.10}$$

- The J_{sr} computes the shared key received from the ED_{ith} as

$$\begin{aligned} JS_{sh} &= JS_{sec}.ED_{pub} \in E\mathbb{F}p, \\ &= JS_{sec}.(ED_{sec}.P) \in E\mathbb{F}p, \\ &= JS_{sec}.ED_{sec}.P \in E\mathbb{F}p. \end{aligned} \tag{5.11}$$

- Both the ED_{ith} and J_{sr} can communicate securely using the shared key $ED_{sh} == JS_{sh}$

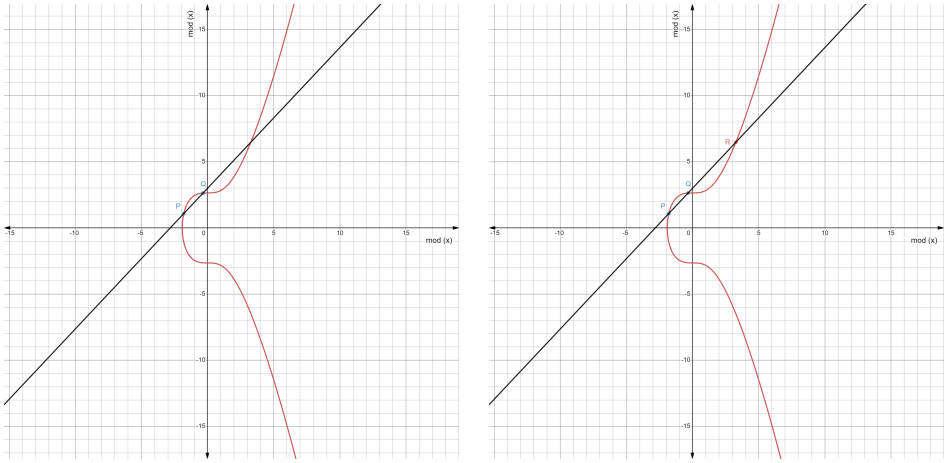
5.3 Key authentication

The proposed KGD implementation considers the key authentication phase, as we discussed the key generation and distribution between the ED_{ith} and J_{sr} over an insecure channel using the ECDH method. However, the authentication of the key being exchanged is important as there is a possibility that the ED_{ith} may share the key with an adversary or vulnerable J_{sr}, and vice versa. To cope with the problem, the authenticated key exchange is a key solution that could potentially avoid key revealing breaches. The proposed KGD uses the elliptic curve digital signature algorithm (ECDSA) since it comes up with the same foundation as in ECDH. The ECDSA consumes less energy of the node as compared to other rivals such as RSA, ElGamal, and DSA; because it takes a smaller key length size and low computation power. ECDSA algorithm is based on the mathematics of equation 5.6. Likewise the ECDH, the ED_{ith} and J_{sr} are required to disclose some information publicly.

In ECDSA, a private key is a randomly generated number, which is used to generate a public key through a mathematical operation. The public key is then used to generate a digital signature for a message. The ECDSA algorithm follows two parts such as signature generation and verification. To generate a signature, it is important to first compute a hash of the message. In the proposed LoRa KGD solution, we compute the hash function of the public key of the ED_{ith} and J_{sr} using SHA-256 hashing function. It is worth noting the hash function results in a larger bit length; however, we kept only the leftmost bits to compute the signature. For the sake of simplicity, we present a simple example that we considered as a key authentication process in the proposed KGD solution using the ECDSA algorithm. The simple method is as follows:

Key Generation:

- The ED_{ith} generate(s) a secret/private key ED_{sec} from the random or deterministic random number generation such as [1, n-1].



((a)) $Y^2 = x3 + a_x + b$ ((b)) $P + Q + R = \mathbb{I}$

((c)) $P + Q = -R$ ((d)) $-R + S - S = \mathbb{I}$

Figure 5.3: Elliptic curve with points on different positions.

- In this case, the ED_{sec} is for instance the number "7".
- Using the ED_{sec}, the public key ED_{pub} is calculated as

$$ED_{pub} = ED_{sec}.G \tag{5.12}$$

  where, G is the generator point on the curve having x and y coordinates for instance $(x, y) = (3, 7)$
- For example the ED_{pub} is a point on the curve having having coordinates $(x_1, y_1) = (14,28)$

Message Hashing:

- Using the cryptography library in python "Hashlib" [68], we compute the hash function for the message being transmitted to the J_{sr}
- In this example, we hashed the message using SHA3-256

$$h = \text{hashlib.sha3\_256(message)} \tag{5.13}$$

  where hash *"h"* is for instance $\boxed{h = 12345}$

Signature Generation:

- The ED_{ith} generate(s) a random number for instance $\boxed{k = 3}$
- The signature in ECDSA uses two values (r,s), where the x_1 coordinate is used to compute the value r

$$r = x_1 \,\text{modulo}\, n \tag{5.14}$$

- The variable n denotes the infinity points on the curve. Here we suppose its value is $\boxed{n = 29}$

- The x_1 coordinate is already given such as $x_1 = 14$; therefore, equation 5.14 can be written as

$$\begin{aligned} r &= x_1 \,\text{modulo}\, n \\ r &= 14 \,\text{modulo}\, 29 \\ \boxed{r &= 14} \end{aligned} \tag{5.15}$$

- It is also worth mentioning that if r is equal to zero then k must be regenerated. The value of r is computed now we define the value of s as follows below

$$\begin{aligned} s &= k^{-1}(h + ED_{sec} * r) \,\text{modulo}\, n \\ s &= 3^{-1}(12345 + 7 * 14) \,\text{modulo}\, 29 \end{aligned} \tag{5.16}$$

- The modular inverse of k with respect to n is calculated as is $k * k^{-1} \equiv 1$ (modulo n) or equivalently $k * k^{-1}$ (modulo n) $= 1$

$$\begin{aligned} 3 * 0 &\equiv 0 \,\text{modulo}\, 29 \\ 3 * 1 &\equiv 3 \,\text{modulo}\, 29 \\ 3 * 2 &\equiv 6 \,\text{modulo}\, 29 \\ 3 * 3 &\equiv 9 \,\text{modulo}\, 29 \\ .. \quad .. \quad .. \quad .. \quad .. \\ .. \quad .. \quad .. \quad .. \quad .. \\ 3 * 8 &\equiv 24 \equiv 24 \,\text{modulo}\, 29 \\ 3 * 9 &\equiv 27 \equiv 27 \,\text{modulo}\, 29 \\ 3 * 10 &\equiv 30 \equiv \boxed{1} \,\text{modulo}\, 29 \\ 3 * 11 &\equiv 33 \equiv 4 \,\text{modulo}\, 29 \\ .. \quad .. \quad .. \quad .. \quad .. \\ .. \quad .. \quad .. \quad .. \quad .. \\ 3 * 27 &\equiv 81 \equiv 23 \,\text{modulo}\, 29 \\ 3 * 28 &\equiv 84 \equiv 26 \,\text{modulo}\, 29 \end{aligned}$$

- The modular inverse of k with respect to n is "10", but to check the signature validity, let's take an incorrect integer "9". So, from equation 5.16

$$\begin{aligned} s &= 3^{-1}(12345 + 7 * 14) \text{ modulo } 29 \\ s &= 9(12345 + 98) \text{ modulo } 29 \\ s &= 9(12443) \text{ modulo } 29 \\ \boxed{s &= 26} \end{aligned} \tag{5.17}$$

- Equations 5.15 and 5.17 show the signature $(r,s) = (14, 26)$ is calculated that will further append to the public key of the J_{sr} and, vice versa.

Signature Verification:

- After the signature generation at the ED_{ith}, the J_{sr} then verify the signature and check the authenticity of the message being exchanged.
- To verify the signature, the J_{sr} takes the following parameters as input such as; the signed message, the signature $(r,s) = (14,26)$ and the $ED_{pub}=(x_1, y_1)=(14,28)$ of the ED_{ith}.
- The J_{sr} authenticate the message using the modular inverse of s with respect to n. The modular inverse is computed in the same fashion as in equation 5.17.

$$\begin{aligned} w &= s^{-1} \text{modulo } n \\ w &= 26^{-1} \text{modulo } 29 \\ w &= 13 \end{aligned} \tag{5.18}$$

- The J_{sr} then calculate u_1 and u_2 to recover the random points used in the signature generation step.

$$
\begin{aligned}
u_1 &= (h * w) \operatorname{modulo} n \\
u_1 &= (12345 * 13) \operatorname{modulo} 29 \\
\boxed{u_1 &= 8} \\
u_2 &= (r * w) \operatorname{modulo} n \\
u_2 &= (14 * 13) \operatorname{modulo} 29 \\
\boxed{u_2 &= 5}
\end{aligned}
\tag{5.19}
$$

- The J_{sr} uses the ED_{pub} of the ED_{ith}, and the G point. By scalar multiplication, we get

$$
\begin{aligned}
(x_2, y_2) &= u_1 * G + u_2 * ED_{pub}, \\
(x_2, y_2) &= 8 * (3, 7) + 5 * (14, 28), \\
(x_2, y_2) &= (24, 56) + (70, 140), \\
(x_2, y_2) &= (112, 196).
\end{aligned}
\tag{5.20}
$$

- When the ED_{ith} send(s) a message towards the J_{sr}, it will first check if the message being sent is original or not. If the value of r is equal to x i.e., $r \equiv x$, then it means the received signature is valid. Otherwise, it will consider as tempering or integrity has been attacked. Hence, with the KGD mechanism, the LoRaWAN key authenticity is proven where $r = 14 \neq x_2 = 112$

$$
r = x_2 \operatorname{modulo} n \tag{5.21}
$$

Figure. 5.4 provides a visual representation of the key authentication process. The process involves the *ED* signing the message using the JS_{pub} and sending it to the *JS*. The *JS* then uses the JS_{sec} to authenticate the message received from the *ED*. Similarly, the *ED* authenticates the messages received from the *JS* using the JS_{sec}. This two-way authentication process ensures the security and integrity of the communication between the *ED* and *JS*.

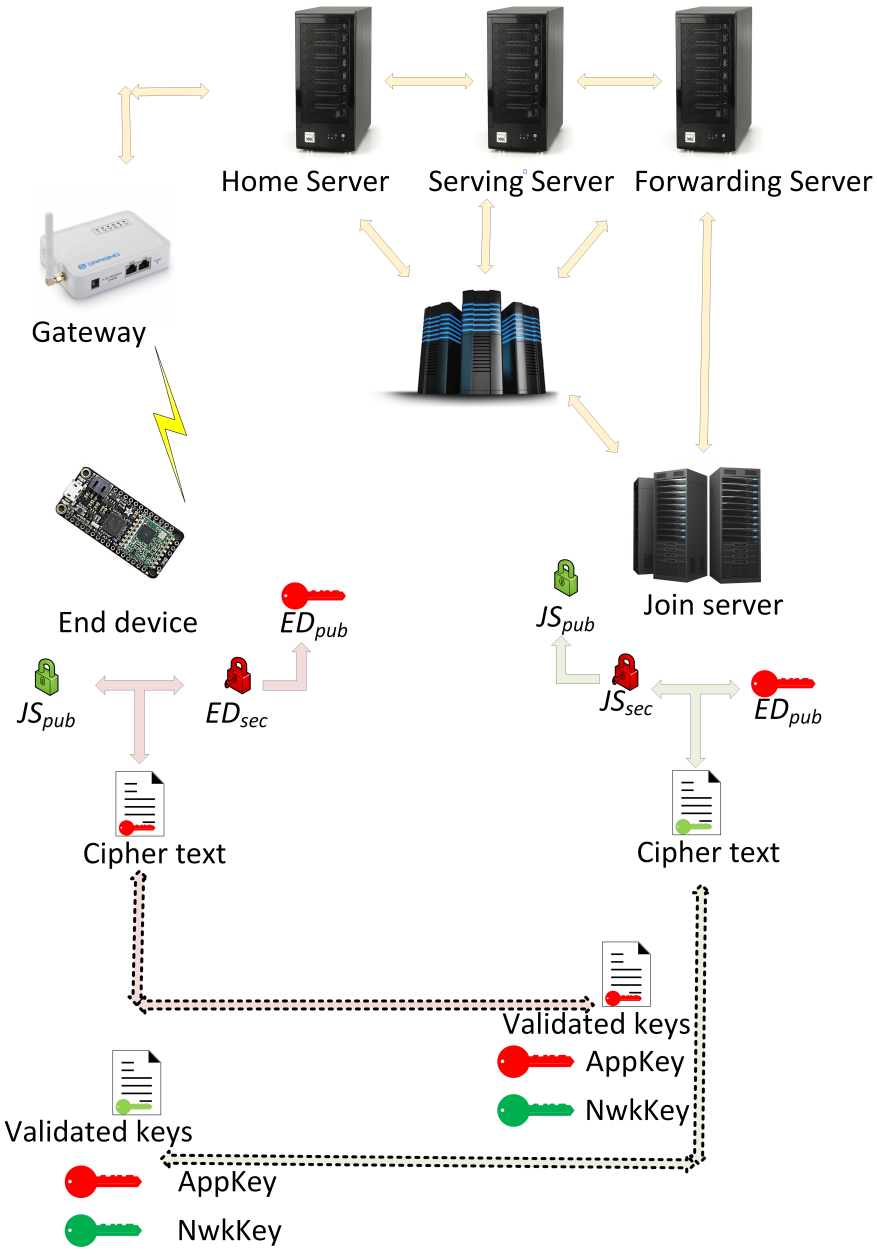


Figure 5.4: Authentication of the keys using ECDSA in the *JS*/*ED*.

5.4 Security verification

Verification of the proposed KGD algorithm is one of the most important steps that have been evaluated by the automated prover tool Scyther [69], [70], [71] developed by researchers in CISPA: the Helmholtz Center for Information and Security [72]. It is worth noting that several other tools like Avispa [73], ProfVerif [74], and Tamarin [75] are available in the literature. However, this work considers Scyther as it is easy to use, lightweight and has an attractive GUI that uses the Security Protocol Description Language (SPDL) programming language to characterize the role of security analysis. The KGD protocol is scrutinized and validated using Scyther tool as shown in Fig. 5.5. The Scyther tool verified and passed several performance tests that show the KGD algorithm is secured against several attacks. The non-injective agreement (Niagree) parameter is verified that reflects both the ED_{ith} and JS are agree to exchange the packets. If they did not exist within the range, then the Scyther tool shows "Fail" instead of "Ok", and graphically visualizes possible attacks. Furthermore, the feature of non-injective synchronization (Nisynch) indicates that the content exchanged between both entities is similar and no integrity attack has been devised. The parameter "Alive" shows the two parties are available and ensures the existence for content exchanging between ED_{ith} and the JS. The parameter "Secret" verifies the secrecy of the communication and indicates that no activity is being attacked.

5.5 Hardware implementation

The section discusses the practical implementation of the KGD algorithm with the LoRaWAN testbed. For convenient performance, we considered the Adafruit RFM95 radio chip that is packaged with the SX1276 [76] LoRa transceiver. The LoRa transceiver is then connected with the Raspberry Pi (RPi) 4 model B which makes a complete LoRa node as used in [45]. This transceiver uses a few spread spectrum versions, such as SF7-SF12, to enable ultra-long-range communication. Furthermore, it provides good sensitivity with consuming very little energy. The LoRaWAN gateway acts as a bridge, that forwards data from the *ED* to the *NS* and vice versa. This paper uses the PicoCell SX1308 [77] LoRa concentrator from

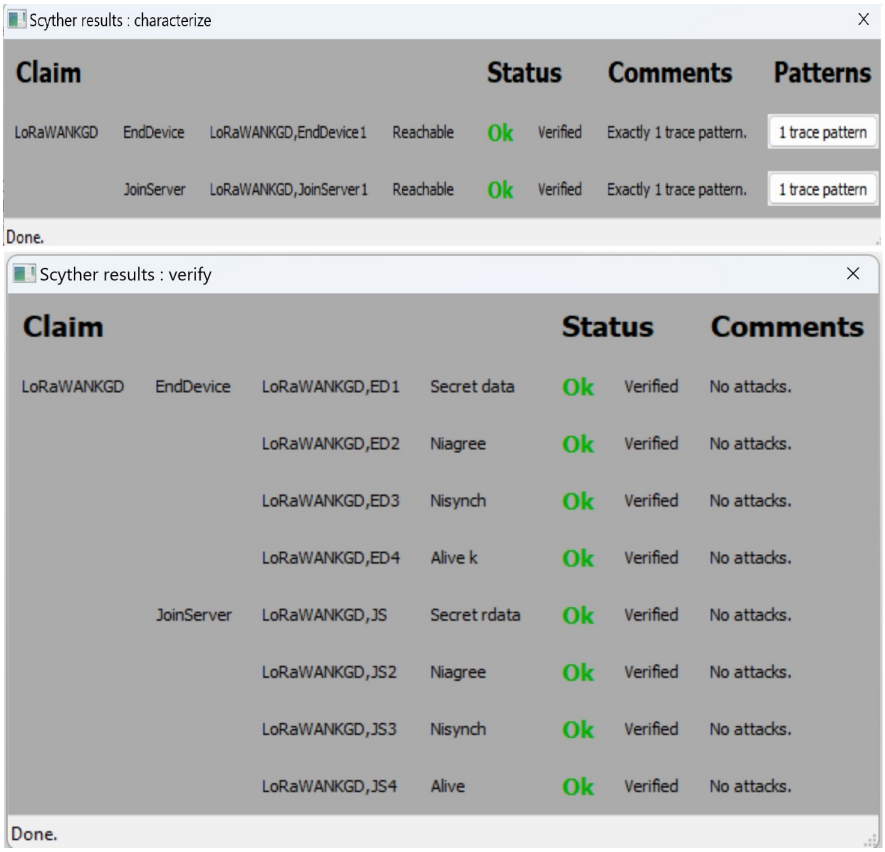


Figure 5.5: The KGD protocol validation using Scyther tool.

the Semtech corporation. The PicoCell is connected with the RPi 3 model B, as shown in Fig. 5.6. The gateway uses a LoRa packet forwarder [78] that forwards the packet using IP/UDP link, shown in Fig. 5.7. In this paper, we used the server from the TheThingsNetwork (TTN) [1] that provides a convenient solution using built-in integrations such as AWS, Azure, MQTT to process the data come from the *ED*.

5.6 Performance evaluation

This section elucidates the evaluation of the performance of the KGD solution, which is implemented on a Raspberry Pi 4 Model B in conjunction with the

[1] https://eu1.cloud.thethings.network/console/

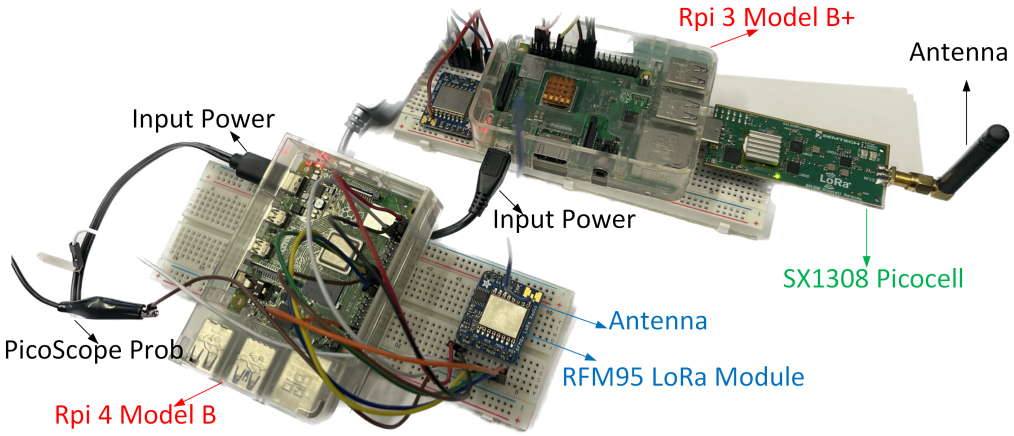


Figure 5.6: The testbed used in the experiment.

```
pi@raspberrypi: ~/lorhackste
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov  8 19:15:25 2022
pi@raspberrypi:~ $ cd lorhackster/picoGW_packet_forwarder/lora_pkt_fwd/
pi@raspberrypi:~/lorhackster/picoGW_packet_forwarder/lora_pkt_fwd $ ./lora_pkt_fwd
*** Packet Forwarder for Lora PicoCell Gateway ***
Version: 0.1.0
*** Lora concentrator HAL library version info ***
Version: 0.2.3;
*** MCU FW version for LoRa PicoCell Gateway ***
Version: 0x010A0006
***
INFO: Little endian host
INFO: found global configuration file global_conf.json, parsing it
INFO: global_conf.json does contain a JSON object named SX1301_conf, parsing SX1301 parameters
INFO: lorawan_public 1, clksrc 1
INFO: antenna_gain 0 dBi
INFO: Configuring TX LUT with 16 indexes
INFO: radio 0 enabled (type SX1257), center frequency 867500000, RSSI offset -164.000000, tx enabled 1
INFO: radio 1 enabled (type SX1257), center frequency 868500000, RSSI offset -164.000000, tx enabled 0
INFO: Lora multi-SF channel 0>  radio 1, IF -400000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 1>  radio 1, IF -200000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 2>  radio 1, IF 0 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 3>  radio 0, IF -400000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 4>  radio 0, IF -200000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 5>  radio 0, IF 0 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 6>  radio 0, IF 200000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 7>  radio 0, IF 400000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora std channel> radio 1, IF -200000 Hz, 250000 Hz bw, SF 7
INFO: FSK channel> radio 1, IF 300000 Hz, 125000 Hz bw, 50000 bps datarate
INFO: global_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to 383438344C004100
INFO: server hostname or IP address is configured to "eu1.cloud.thethings.network"
INFO: upstream port is configured to "1700"
INFO: downstream port is configured to "1700"
INFO: downstream keep-alive interval is configured to 10 seconds
INFO: statistics display interval is configured to 30 seconds
INFO: upstream PUSH_DATA time-out is configured to 100 ms
INFO: packets received with a valid CRC will be forwarded
```

Figure 5.7: The raspberry pi terminal shows the PicoCell SX1308 receives and transmits packets.

RFM95 LoRa module. In order to assess the LoRa capability, we disabled several peripheral features of the Raspberry Pi, such as HDMI, Ethernet, and external ports, except for the GPIO pins. This was done to minimize any potential interference and to allocate more system resources toward the LoRa module's operation, thereby providing a more accurate evaluation of its performance. To evaluate the performance, we determined the total energy consumption of the Raspberry Pi when integrated with the RFM95 module. We established a connection between the PicoScope 2203 [79] and our experimental setup, and proceeded to acquire measurements while compiling our Python script. Figure 5.8 shows the total energy consumption of the Raspberry pi and RFM95 module when applying the KGD algorithm by using a secure shell (SSH) protocol. The graph shows a line that represents the total energy consumed over time in milliseconds, including the execution of the KGD algorithm, the transmission of a join request message, and the reception of a join accept message. These measurements provide insights into the energy efficiency of the KGD algorithm and its impact on the overall energy consumption of the system.

5.6.1 Perfect forward and backward secrecy

The perfect forward and backward secrecy depends on the breach of the root keys either at the *ED* or *JS*. By hands-on the NwkKey or AppKey, an intruder can potentially disclose the secrecy, which in turn, poses a high risk to the entire network. To mitigate such risks, the proposed scheme employs periodic updates of the root keys, which are generated through the use of time T and input password P_s parameters of Argon2 in the key generation process. Consequently, even if one or both root keys are compromised, the intruder is unable to predict future keys or retrieve previous ones. This renders the network more resilient against potential attacks, since new keys are introduced periodically to enhance security.

5.6.2 Statistical Randomness test

In this section, the proposed scheme is validated through a randomness test, which is an important aspect of evaluating the security of cryptographic schemes. This

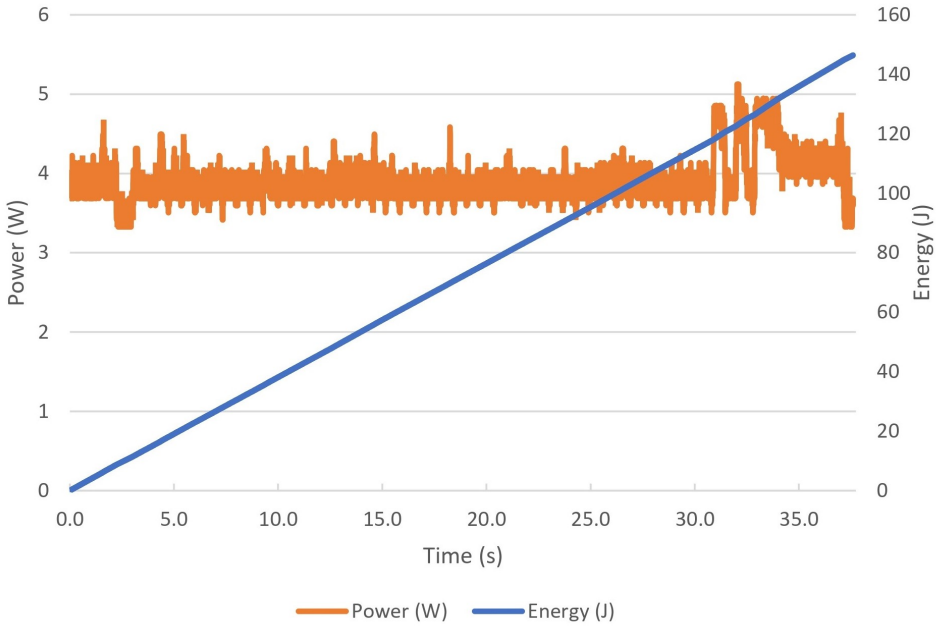


Figure 5.8: Total energy consumption of the Raspberry Pi connected to the RFM95 LoRa module.

test helps to ensure that the generated keys are truly random and unpredictable, and therefore suitable for use in secure communication protocols. Therefore, to evaluate the statistical randomness of the proposed approach, we conducted two tests for the randomness that complied with the requirements of the National Institute of Standards and Technology (NIST) [80] and Diehard [81], respectively. Due to the limited computing capacity of the 32-bit microprocessor, we tested the proposed scheme on Ubuntu 22.04.2 LTS, with an Intel® Core™ i5-10500 CPU @ 3.10 GHz x 12 and 8.0 GiB of RAM. We stored the input values in a transcendental number (.pi) file from NIST's suite and called the iteration 40196 times, whereas each iteration generates Additionally, we executed the Diehard test to ensure that all tests were passed and that no anomalies were detected. As demonstrated in table 5.2, the key generated by the proposed algorithm satisfied with the NIST's recommended randomness tests.

Table 5.2: NIST statistical test [2] suit for the uniformity of P-value.

No	Statistical Test	Proportion	P-Value	Assessment
1	Frequency	10/10	0.534146	Passed
2	BlockFrequency	10/10	0.739918	Passed
3	CumulativesSums (a)	10/10	0.911413	Passed
4	CumulativeSums (b)	10/10	0.911413	Passed
5	Runs	10/10	0.122325	Passed
6	LongestRun	10/10	0.213309	Passed
7	Rank	10/10	0.911413	Passed
8	FFT	10/10	0.739918	Passed
9	Serial (a)	9/10	0.350485	Passed
10	Serial (b)	9/10	0.534146	Passed
11	LinearComplexity	10/10	0.739918	Passed

Chapter 6

Conclusions

6.1 Conclusions

LoRaWAN is an emerging technology that has opened a large pool of innovation in the field of IoT. It has diminished the constraints in long-distance packet advancement by keeping in view a very less energy consumption. LoRaWAN incorporates advanced security features for both the network and the application layers by employing several cryptographic keys. The *ED*, however residing on the edge of the network is a primary target for cyberattackers. This paper presents a remedy called the KGD algorithm that mitigates cyberattacks in the light of secure key management. The KGD algorithm is accomplished in three steps. At first, it generates the secret keys with a cryptographically secured deterministic random bit generator method. The generated keys are then exchanged between the *ED* and *JS* using the ECDH key exchanging algorithm. To check if the keys are exchanged with the authentic entities, a key authentication process such as the ECDSA algorithm is considered to verify if the keys were exchanged with the right parties. Finally, a protocol verification is executed using the Scyther protocol prover that shows the proposed KGD algorithm is secured against cyberattacks and possesses mutual authentication, integrity, availability, and perfect forward secrecy.

6.2 Future work

The future work may consider malware attacks between the *JS* and the *AS* as the LoRaWAN specification has not yet disclosed the security consideration between these two entities. Using efficient Machine Learning (ML) algorithm might be feasible to detect such attacks.

References

[1] F. Hessel, "Lorawan security analysis: An experimental evaluation of attacks," Master's thesis, Technische Universität, 2021. 1, 15

[2] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., Booz-allen and hamilton inc mclean va, 2001. 2, 91

[3] J. Qadir, I. Butun, R. Lagerstrom, P. Gastaldo, and D. D. Caviglia, "Towards smart sensing systems: A new approach to environmental monitoring systems by using lorawan," in *2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC)*, pp. 176–181, IEEE, 2022. 5

[4] LoRaWAN, "Lorawan security™." [Online]. Accessed Feb 16 2023. 5

[5] M. Ralambotiana, "Key management with a trusted third party using lorawan protocol: A study case for e2e security," 2018. 5

[6] J. Kim and J. Song, "A simple and efficient replay attack prevention scheme for lorawan," in *Proceedings of the 2017 the 7th International Conference on Communication and Network Security*, pp. 32–36, 2017. 5, 57

[7] W.-J. Sung, H.-G. Ahn, J.-B. Kim, and S.-G. Choi, "Protecting end-device from replay attack on lorawan," in *2018 20th International conference on advanced communication technology (ICACT)*, pp. 167–171, IEEE, 2018. 5, 57

[8] J. Lee, D. Hwang, J. Park, and K.-H. Kim, "Risk analysis and countermeasure for bit-flipping attack in lorawan," in *2017 International conference on information networking (ICOIN)*, pp. 549–551, IEEE, 2017. 5, 47

[9] E. Van Es, H. Vranken, and A. Hommersom, "Denial-of-service attacks on lorawan," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pp. 1–6, 2018. 6, 51, 59

[10] S. Zulian, "Security threat analysis and countermeasures for lorawan join procedure," 2016. 6

[11] "Lorawan® specification v1.1®." [Online]. Accessed Feb 16 2023. 6

[12] F. Hessel, L. Almon, and M. Hollick, "Lorawan security: An evolvable survey on vulnerabilities, attacks and their systematic mitigation," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–55, 2023. 6

[13] J. Qadir, I. Butun, P. Gastaldo, and D. D. Caviglia, "Review of security vulnerabilities in lorawan," in *Applications in Electronics Pervading Industry, Environment and Society: APPLEPIES 2022*, pp. 248–254, Springer, 2023. 6

[14] Y. S. e. a. Philipp Hofmann, "Comparison and analysis of security aspects of lorawan and nb-iot." [Online]. Accessed Feb 19 2023. 6

[15] I. Butun, N. Pereira, and M. Gidlund, "Security risk analysis of lorawan and future directions," *Future Internet*, vol. 11, no. 1, p. 3, 2018. 6, 8, 42, 43, 50

[16] C. Cerrudo, E. M. Fayo, and M. Sequeira, "Lorawan networks susceptible to hacking: Common cyber security problems, how to detect and prevent them," *Accessed: Apr*, vol. 28, p. 2021, 2020. 7, 16, 43

[17] I. Butun, N. Pereira, and M. Gidlund, "Analysis of lorawan v1.1 security," in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*, pp. 1–6, 2018. 8

[18] N. Hayati, K. Ramli, S. Windarta, and M. Suryanegara, "A novel secure root key updating scheme for lorawans based on ctr_aes drbg 128," *IEEE Access*, vol. 10, pp. 18807–18819, 2022. 8, 64, 68, 72

[19] J. Han and J. Wang, "An enhanced key management scheme for lorawan," *Cryptography*, vol. 2, no. 4, p. 34, 2018. 8, 62, 66

[20] T. C. Dönmez and E. Nigussie, "Key management through delegation for lorawan based healthcare monitoring systems," in *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, pp. 1–6, IEEE, 2019. 8, 9

[21] J. Xing, L. Hou, K. Zhang, and K. Zheng, "An improved secure key management scheme for lora system," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 296–301, IEEE, 2019. 8

[22] X. Chen, J. Wang, and L. Wang, "A fast session key generation scheme for lorawan," in *2019 Australian & New Zealand Control Conference (ANZCC)*, pp. 63–66, IEEE, 2019. 9

[23] S. A. A. Hakeem, S. M. A. El-Kader, and H. Kim, "A key management protocol based on the hash chain key generation for securing lorawan networks," *Sensors*, vol. 21, no. 17, p. 5838, 2021. 9

[24] M. Vučinić, G. Selander, J. P. Mattsson, and T. Watteyne, "Lightweight authenticated key exchange with edhoc," *Computer*, vol. 55, no. 4, pp. 94–100, 2022. 10

[25] I. Butun, N. Pereira, and M. Gidlund, "Analysis of lorawan v1. 1 security," in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*, pp. 1–6, 2018. 13, 49

[26] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "Lorawan specification," *LoRa alliance*, vol. 1, 2015. 16, 43

[27] 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. 22

[28] J. B. Rydell, O. Otterlind, and I. Butun, "Delay considerations for reliable communications in lorawan," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, 2022. 22

[29] 3GPP, "2g, 3g, lte, and 5g specifications," 2022. 22

[30] LoRa Alliance, "Lorawan® security: Frequently asked questions," 2020. 23

[31] SigFox, "Make things come alive in a secure way," 2017. 23

[32] GSMA, "Security features of lte-m and nb-iot networks," 2019. 23

[33] N. Aftab, S. A. R. Zaidi, and D. McLernon, "Scalability analysis of multiple lora gateways using stochastic geometry," *Internet of Things*, vol. 9, p. 100132, 2020. 23

[34] J. J. Barriga and S. G. Yoo, "Securing end-node to gateway communication in lorawan with a lightweight security protocol," *IEEE Access*, vol. 10, pp. 96672–96694, 2022.

[35] H. Noura, T. Hatoum, O. Salman, J.-P. Yaacoub, and A. Chehab, "Lorawan security survey: Issues, threats and possible mitigation techniques," *Internet of Things*, p. 100303, 2020. 42, 43, 50

[36] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of lorawan," *Computer Networks*, vol. 148, pp. 328–339, 2019. 43

[37] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in lorawan," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 129–140, IEEE, 2018. 43

[38] K. Mikhaylov, R. Fujdiak, A. Pouttu, V. Miroslav, L. Malina, and P. Mlynek, "Energy attack in lorawan: Experimental validation," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pp. 1–6, 2019. 43

[39] I. Butun, "Privacy and trust relations in internet of things from the user point of view," in *The 7th annual computing and communication workshop and conference (CCWC)*, pp. 1–5, IEEE, 2017. 46

[40] 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, IEEE, 2017. 51

[41] A. Martínez, U. Zurutuza, R. Uribeetxeberria, M. Fernández, J. Lizarraga, A. Serna, and I. Vélez, "Beacon frame spoofing attack detection in ieee 802.11 networks," in *2008 Third International Conference on Availability, Reliability and Security*, pp. 520–525, IEEE, 2008. 51

[42] C. Salinesi, R. Mazo, O. Djebbi, D. Diaz, and A. Lora-Michiels, "Constraints: The core of product line engineering," in *2011 Fifth International Conference on Research Challenges in Information Science*, pp. 1–10, IEEE, 2011. 52

[43] P. I. R. Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the internet of things: Challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019. 52

[44] P. Bhale, S. Dey, S. Biswas, and S. Nandi, "Energy efficient approach to detect sinkhole attack using roving ids in 6lowpan network," in *International Conference on Innovations for Community Services*, pp. 187–207, Springer, 2020. 53

[45] A. Mohamed, F. Wang, I. Butun, J. Qadir, R. Lagerström, P. Gastaldo, and D. D. Caviglia, "Enhancing cyber security of lorawan gateways under adversarial attacks," *Sensors*, vol. 22, no. 9, p. 3498, 2022. 53, 56, 86

[46] F. Mårlind and I. Butun, "Activation of lorawan end devices by using public key cryptography," in *2020 4th Cyber Security in Networking Conference (CSNet)*, pp. 1–8, IEEE, 2020. 55, 63, 67

[47] J. J. Barriga and S. G. Yoo, "Securing end-node to gateway communication in lorawan with a lightweight security protocol," *IEEE Access*, vol. 10, pp. 96672–96694, 2022. 32

[48] E. Gresak and M. Voznak, "Protecting gateway from abp replay attack on lorawan," in *AETA 2018-Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, pp. 400–408, Springer, 2020. 58

[49] S. Na, D. Hwang, W. Shin, and K.-H. Kim, "Scenario and countermeasure for replay attack using join request messages in lorawan," in *2017 international*

conference on information networking (ICOIN), pp. 718–720, IEEE, 2017. 58

[50] S. J. Lagat, "Detecting denial of service attacks in lorawan," *Signature*, vol. 31, 2022. 59

[51] M. Leent, "An improved key distribution and updating mechanism for low power wide area networks (lpwan)," 2017. 60, 65

[52] E. Jayasuriya, *ECDH Based Key Management for LoRaWAN Considering Sensor Node Limitations.* PhD thesis, 2021. 60, 65

[53] K.-L. Tsai, F.-Y. Leu, L.-L. Hung, and C.-Y. Ko, "Secure session key generation method for lorawan servers," *IEEE Access*, vol. 8, pp. 54631–54640, 2020. 61, 65

[54] R. Sanchez-Iborra, J. Sánchez-Gómez, S. Pérez, P. J. Fernández, J. Santa, J. L. Hernández-Ramos, and A. F. Skarmeta, "Enhancing lorawan security through a lightweight and authenticated key management approach," *Sensors*, vol. 18, no. 6, p. 1833, 2018. 61, 66

[55] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, "Internet key exchange protocol version 2 (ikev2)," tech. rep., 2014. 61

[56] E. Rescorla and N. Modadugu, "Datagram transport layer security," tech. rep., 2006. 61

[57] J. Kim and J. Song, "A dual key-based activation scheme for secure lorawan," *Wireless Communications and Mobile Computing*, vol. 2017, 2017. 62, 67

[58] X. Chen, M. Lech, and L. Wang, "A complete key management scheme for lorawan v1. 1," *Sensors*, vol. 21, no. 9, p. 2962, 2021. 63, 67

[59] M. Tan, D. Sun, and X. Li, "A secure and efficient blockchain-based key management scheme for lorawan," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7, IEEE, 2021. 64, 68

[60] G. Selander, J. Mattsson, and F. Palombini, "Ephemeral diffie-hellman over cose (edhoc)," *Internet Engineering Task Force, Internet-Draft draft-ietf-lake-edhoc-09*, 2021. 61

[61] E. B. Barker, J. M. Kelsey, *et al.*, *Recommendation for random number generation using deterministic random bit generators (revised)*. US Department of Commerce, Technology Administration, National Institute of ..., 2007. 71

[62] M. Aljohani, I. Ahmad, M. Basheri, and M. O. Alassafi, "Performance analysis of cryptographic pseudorandom number generators," *IEEE Access*, vol. 7, pp. 39794–39805, 2019. 71

[63] V. T. Hoang and Y. Shen, "Security analysis of nist ctr-drbg," in *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part I*, pp. 218–247, Springer, 2020. 72

[64] J. Woodage and D. Shumow, "An analysis of nist sp 800-90a," in *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part II 38*, pp. 151–180, Springer, 2019. 72

[65] S. Cohney, A. Kwong, S. Paz, D. Genkin, N. Heninger, E. Ronen, and Y. Yarom, "Pseudorandom black swans: Cache attacks on ctr_drbg," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1241–1258, IEEE, 2020.

[66] J. Wetzels, "Open sesame: The password hashing competition and argon2," *arXiv preprint arXiv:1602.03097*, 2016. 72

[67] D. R. Brown, "Sec 2: Recommended elliptic curve domain parameters," *Standars for Efficient Cryptography*, 2010. 76

[68] G. P. Smith, "Secure hashes and message digests." [Online]. Accessed Feb 08 2023. 81

[69] K. Inoue, K. Wada, and Y. Ito, "Effective application of paro: Seal type robots for disabled people in according to ideas of occupational therapists," in *Computers Helping People with Special Needs: 11th International Conference, ICCHP 2008, Linz, Austria, July 9-11, 2008. Proceedings 11*, pp. 1321–1324, Springer, 2008. 86

[70] C. Cremers, S. Mauw, C. Cremers, and S. Mauw, "Operational semantics," *Operational semantics and verification of security protocols*, pp. 13–35, 2012. 86

[71] F. A. Putra, K. Ramli, N. Hayati, and T. S. Gunawan, "Pura-scis protocol: A novel solution for cloud-based information sharing protection for sectoral organizations," *Symmetry*, vol. 13, no. 12, p. 2347, 2021. 86

[72] C. Cremers, "The scyther tool." [Online]. Accessed Feb 08 2023. 86

[73] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, *et al.*, "The avispa tool for the automated validation of internet security protocols and applications," in *Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005. Proceedings 17*, pp. 281–285, Springer, 2005. 86

[74] B. Blanchet *et al.*, "An efficient cryptographic protocol verifier based on prolog rules.," in *csfw*, vol. 1, pp. 82–96, 2001. 86

[75] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*, pp. 696–701, Springer, 2013. 86

[76] Semtech, "Semtech corporation sx1276." [Online]. Accessed Feb 08 2023. 86

[77] SX1308P915GW, "Semtech corporation sx1276." [Online]. Accessed Feb 08 2023. 86

[78] LoRa-net, "Picocell lorawan gateway forwarder." [Online]. Accessed Feb 08 2023. 87

[79] "Picoscope 2203." [Online]. Accessed Feb 08 2023. 89

[80] S.-J. Kim, K. Umeno, and A. Hasegawa, "Corrections of the nist statistical test suite for randomness," *arXiv preprint nlin/0401040*, 2004. 90

[81] M. M. Alani, "Testing randomness in ciphertext of block-ciphers using diehard tests," *Int. J. Comput. Sci. Netw. Secur*, vol. 10, no. 4, pp. 53–57, 2010. 90