2-1-2021

# LoRaWAN device security and energy optimization

John A. Stranahan Jr.
*Rowan University*

# LORAWAN DEVICE SECURITY AND ENERGY OPTIMIZATION

by

John A. Stranahan Jr.

A Thesis

Submitted to the
Department of Computer Science
College of Science and Mathematics
In partial fulfillment of the requirement
For the degree of
Master of Science in Computer Science
at
Rowan University
January 21, 2021

Thesis Chair: Vahid Heydari, Ph.D.

## Dedications

This manuscript is dedicated to my friends and family who have always believed in me during this journey.

**Acknowledgments**

I would like to express my appreciation for all of the people who have taught me life lessons along the way. Special thanks to Dr. Vahid Heydari for the guidance, support, and opportunities throughout the years. I am a better learner, teacher, and professional with my committee and peers at Rowan University.

## Abstract

John A. Stranahan Jr
LORAWAN DEVICE SECURITY AND ENERGY OPTIMIZATION
2019-2020
Vahid Heydari, Ph.D.
Master of Science in Computer Science

Resource-constrained devices are commonly connected to a network and become "things" that make up the Internet of Things (IoT). Many industries are interested in cost-effective, reliable, and cyber secure sensor networks due to the ever-increasing connectivity and benefits of IoT devices. The full advantages of IoT devices are seen in a long-range and remote context. However, current IoT platforms show many obstacles to achieve a balance between power efficiency and cybersecurity. Battery-powered sensor nodes can reliably send data over long distances with minimal power draw by adopting Long-Range (LoRa) wireless radio frequency technology. With LoRa, these devices can stay active for many years due to a low data bit rate and low power draw during device sleep states. An improvement built on top of LoRa wireless technology, Long-Range Wide Area Networks (LoRaWAN), introduces integrity and confidentiality yet, protocol and implementation vulnerabilities still exist within the network, resulting in security risks to the whole system. In this research, solutions to these vulnerabilities are proposed and implemented on a LoRaWAN testbed environment that contains devices, gateways, and servers. Configurations that involve the transmission of data using AES Round Reduction, Join Scheduling, and Metadata Hiding are proposed in this work. A power consumption analysis is performed on the implemented configurations, resulting in a LoRaWAN system that balances cybersecurity and battery life.

## Table of Contents

**Table of Contents (Continued)**

**Table of Contents (Continued)**

## List of Figures

## List of Figures (Continued)

Figure Page

## List of Tables

# Chapter 1

# Introduction

## 1.1 Background

Wireless technologies are continually evolving and improving device connectivity, power efficiency, design flexibility, and cost reduction [1]. The interconnection of devices that can relay information to other devices and even trigger software-based events is a technology that has seemingly endless opportunities. The world's growing dependency on devices known as "things," which are physical devices that are constantly connecting and exchanging information with each other, is undeniable. These "things" are part of a subset of devices connected to the internet and make up the Internet of Things [2].

The Internet of Things (IoT) is a connected network of devices that includes sensors, wearable devices, smartphones, and many other types of sensors that send and receive data from the internet. The usefulness and need for data acquisition have been ingrained into our society, commerce, and daily lives. Figure 1 below shows that according to Statistica, the total number of IoT connected devices will reach around 75 billion by 2025 [3]. Society has become reliant on wireless devices to improve quality of life, and the industries are dependent on more efficient and interconnected technology to improve overall system functions. IoT devices typically gather and send information to remote servers, allowing the analysis of that data and improving organizational operation. The data may trigger webhooks or automated actions in the cloud.

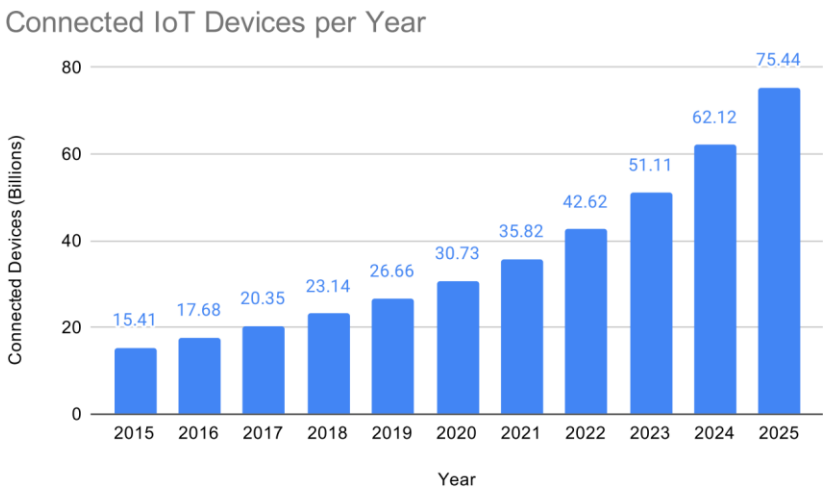


*Figure 1.* IoT Connected Devices From 2015 to 2025 [3]

To improve these wireless connections, many unique types of hardware and software solutions exist. Long-Range Low-Power wireless devices focus on providing sensor information from one remote device to a remote gateway. The data is routed to a network and application server from the gateway, where the data can be used fittingly. These devices belong to a type of wireless telecommunications known as Low-Power-Wide-Area-Networks (LPWAN). LPWAN's are a low-power network that contains devices that transmit data over a long distance with low data rate and low bandwidth. This technology's core fundamentals are low power consumption, low transceiver chip cost, and an extensive coverage area [4].

These wide-area networks are considered low-powered because the devices involved are not powered by a mains power source but rather a battery. For the devices to send data over a long period, the data bitrate (the number of bits over time) is reduced,

and therefore, the power consumption over time is reduced. Low power consumption results in the ability to set up these devices in remote areas and trust that the overall battery life can be as long as multiple months or even numerous years. A ten-year battery life due to the low data rate and careful selection of parameter configurations and duty cycling is possible [5]. Real-world results will vary and most likely will be lower than that rate depending on frequency, microcontroller hardware, firmware, and software. An important radio frequency modulation scheme that will focus on the experiment phase is LoRa, or "Long-Range."

### 1.2 Problem Statement

External microcontrollers are increasingly being connected to the internet, and unfortunately, with this expanded functionality, security has been given a lower priority than connectivity [6]. When data is transferred between multiple devices or systems, each step requires a layer of protection. Protection includes the prevention of alteration of data or unauthorized access by attackers who do not have permissions to access the various resources [5]. The attack surface, or the vector in which inputs can be provided, include the physical device itself, networking hardware, servers, and other remote network resources.

The prevention of cyber-attacks and detection and response is critical to ensure a system is cyber secure. To deem a system secure, the following information security tenants must be satisfied. Confidentiality of information means that data should never be disclosed to unauthorized parties [7]. The integrity of information is defined as data that should never be modified by those who cannot change it. Availability of data is defined

as the assurance that data is accessible when the user requests it. Authentication, particularly mutual authentication, in an IoT context is a verification of identity commonly seen when IoT devices join a network. Nonrepudiation is the assurance that actions such as join procedures, activation, and data transmission can be traced back to the device doing the operation, and these actions are commonly logged by IoT servers [7].

Since LPWAN transmits data over a public unlicensed radio frequency band, it is essential to ensure the data's confidentiality. Confidentiality could be achieved through encryption but comes with the drawback of increased energy consumption and a shorter availability of the device [8]. It may not be feasible to replace the battery of sensor nodes as they can be deployed in hard to reach or even potentially unsafe environments. It follows with such a device's battery life must also be prioritized.

As with physical access to any device, there is an inherent threat to this technology's security. This flaw is due to the portability and accessibility of the devices, which may grant the ability to retrieve the devices' credentials. Protection of the keys and ensuring that installing these devices results in a secure system is paramount [9]. The options to secure a device must be analyzed. Solutions must be offered to ensure a greater level of physical security and less associated risk for provisioning devices in settings such as city rooftops, parking garages, streets, and more.

The first purpose of this work is to examine the security posture of a subset of LPWAN devices, called LoRaWAN devices. The distinction between these types of devices will be made in Chapter 2 of this work. The second purpose is to evaluate the power

expenditure of LoRaWAN devices within a testbed environment. The testbed will be used to find a set of optimal software changes to the LoRaWAN devices. The third purpose is to improve the security of the LoRaWAN devices and show that the changes are valid by analyzing the overall battery consumption, battery life, and device availability. The result is to propose a set of changes to the LoRaWAN protocol that will help strike a balance between cybersecurity and battery life.

### 1.3 Strategy and Approach

This section details background information is critical to the topics that will be addressed in the experimental phase of this work. To become familiar with and test the hardware, software, firmware, and protocols: a LoRaWAN testbed was created. The testbed contains a network of portable devices, concentrators, and servers and is described in detail. The goal of the testbed is to test improvements to a real-world application of a LoRaWAN network.

Due to a LoRaWAN network's modularity, the testbed itself is modular and many future improvements to the LoRaWAN protocol can be tested. This work's experiment phase is an iterative look at the power expenditure and security posture when utilizing various existing and newly proposed configurations. Power and current measurements were gathered from multiple types of LoRaWAN devices from a power monitor. A statistical analysis is performed from the data collected to discover an optimal configuration for a LoRaWAN device that balances both security and availability. This work will address security issues on the theoretical side as well as the experimental side. Conclusions on what changes to LoRaWAN systems will be made. The conclusions are

found by analyzing real power data and the theoretical framework of the LoRaWAN protocol.

### 1.4 Assumptions

A few assumptions about the environment must be considered when devising a LoRaWAN testbed. The first assumption is that the devices used to test the protocols may not be the most energy-efficient devices. The tradeoff here is the ease of customization and configuration of the hardware, firmware, and software. Next, there may be a loss of accuracy with battery calculations over more extended periods. The equations are a guideline of how long the battery *should* last rather than *will* last. Lastly, there may be a margin of error within real-world data points as environmental factors, and quality factors such as the real-world reliability, availability, and maintainability of the end nodes are difficult to predict.

# Chapter 2

## Theoretical Framework

Chapter 2 outlines the LoRa and LoRaWAN specification and details the relevant information critical to keeping this long-range, low-power technology secure. The behavior of various settings and implementations are evaluated, and the importance to the security vs. battery life problem is considered. The primary source of information for sections 2.4 and 2.5 that outlines the LoRaWAN protocol is from the LoRa Alliance LoRaWAN Specification v1.0.2 document [10].

### 2.1 LoRa Background

LoRa (Long Range) is a patented radio frequency modulation scheme derived from chirp spread spectrum modulation to transmit radio signals. LoRa's name is an acronym for "long-range," which accurately expresses this technology's core functionality and purpose. The LoRa protocol, which has been acquired by the communications company Semtech, is commonly used to send sensor data via radio waves from an IoT device or sensor to a gateway/concentrator. This technology is agnostic from the higher Open Systems Interconnection (OSI) model layers, allowing interoperability with existing or new network architectures. The radio module has regulatory approval to operate in the 868 and 915 MHz industrial, scientific, and medical (ISM) spectrum [11].

As acknowledged, long-range is an important feature, yet other essential features include low-data transfer rate, low-power, and low-cost [11]. Semtech brand LoRa modules are available for only a few dollars online, with complete controllers available

for a few times the chip price. There are estimates that the battery life of these devices is up to 10 years, depending on the hardware involved. The typical range of LoRa devices is up to 10km, yet it relies heavily on the surrounding environment, power mode, antenna quality, and more. In 2019 a transmission distance of 766 km (476 miles) was reached by a LoRa module connected to a balloon, as shown in Figure 2.

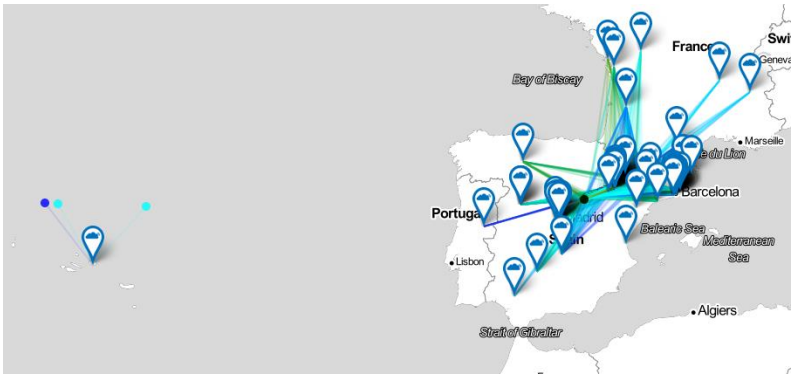


*Figure 2.* Map Depicting Longest Distance a LoRa Frame Was Received [19]

The required amount of data sent over time is assumed to be relatively low if these devices are chosen. There are many use cases in which this is appropriate, be it periodic data transmission or reactions to events. Low power consumption is achieved with small amounts of bit data sent through short radio transmissions followed by a sleep cycle to conserve battery life. The remaining core functionality is low-cost, which is possible because of the inexpensive manufacturing of transceivers and the inexpensive microprocessors required to process the sensor's data.

## 2.2 LoRa Physical Layer Specifications

As seen in Figure 3 below, LP-WAN (sic) devices have a longer maximum communication distance than Bluetooth, Wi-Fi, and NFC [12]. Cellular data is best for high baud rates and long distances than LoRa, but it is met with a higher overall cost due to the need for a SIM card for each device, subject to internet service provider rates. LoRa is one of the best and most cost-effective alternatives if small amounts of data over time is warranted. If there are no Wi-Fi access points in range and years of battery life is required, LoRa would be a desirable communication protocol. Outdoor, where local area networks are limited, the full benefits of this technology can be seen.

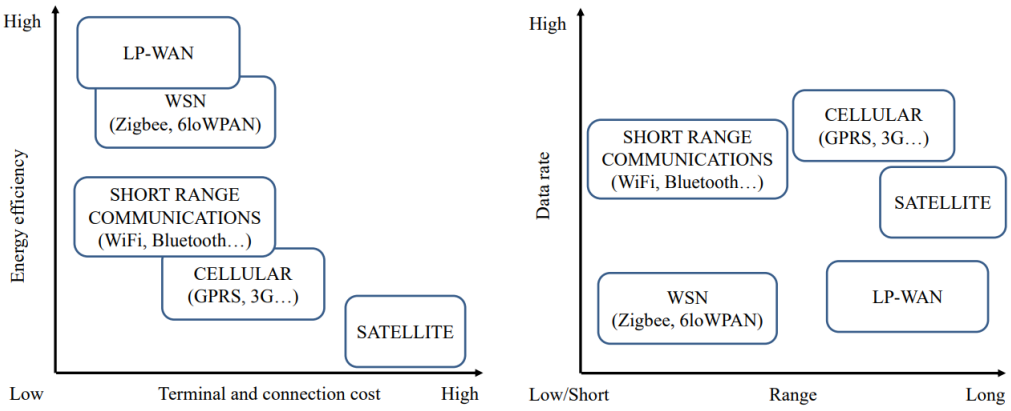


*Figure 3.* Energy Efficiency vs. Data Rate and Coverage Range [11]

LoRa is a low-power communication protocol due to the LoRa physical layer, or the LoRa PHY. The LoRa PHY is defined as the radio interface that creates the bits from spread spectrum modulation. The physical layer communication performance is

customizable with bandwidth, coding rate, and spreading factor. As far as LoRa, there are various bands that these devices can communicate with depending on their deployed country. Radio frequency modules are provisioned with a specific carrier frequency by region, and the chip and antenna are manufactured and transmit on the same band (915 MHz for North America) [13]. Table 1 below shows that the adjustable settings that impact the performance of LoRa communications are outlined [14].

Table 1

*LoRa North America Regional Specification Summary*

| Setting | Values | Description |
|---|---|---|
| Carrier Frequency | 915 MHz | Operating Frequency used by Semtech SX1276 chips. |
| Bandwidth | 125...500kHz | Higher bandwidths increase data rates but reduce receiver sensitivity and range. |
| Spreading Factor | 7-12 | Higher spreading factors increase radio signal sensitivity and larger packets. |
| Coding Rate | 4/5 . . . 4/8 | A higher coding rate increases resilience to interference and decoding errors but results in larger packets. |
| Transmission Power | +20dBm to +30dBm | Not recommended to change, increasing power immensely affects battery life. |

*Note*. Information is retrieved from the LoRaWAN Specification [10].

### 2.3 LoRaWAN Network Architecture

As an improvement of LoRa, LoRaWAN was developed to add confidentiality of data, the integrity of the frames being sent, and availability to LoRa end nodes. LoRaWAN gateways, also known as concentrators, receive the radio transmissions from the LoRa end nodes. LoRaWAN is a MAC protocol that defines the network architecture and communication protocol to send data from LoRa radio frequency modules securely. The expected battery life of LoRaWAN nodes is decreased due to the overhead of securing the payload and the activating of devices.

There are several core components of a LoRaWAN network that make it useful. A wide range of configurations and settings available because the network is designed as a "Star of Stars" network topology [15]. Each component has a wide array of functionality that can meet the consumer's needs. As seen below in Figure 4, there are four main components within a LoRaWAN network: end nodes, concentrators, network servers, and application servers.

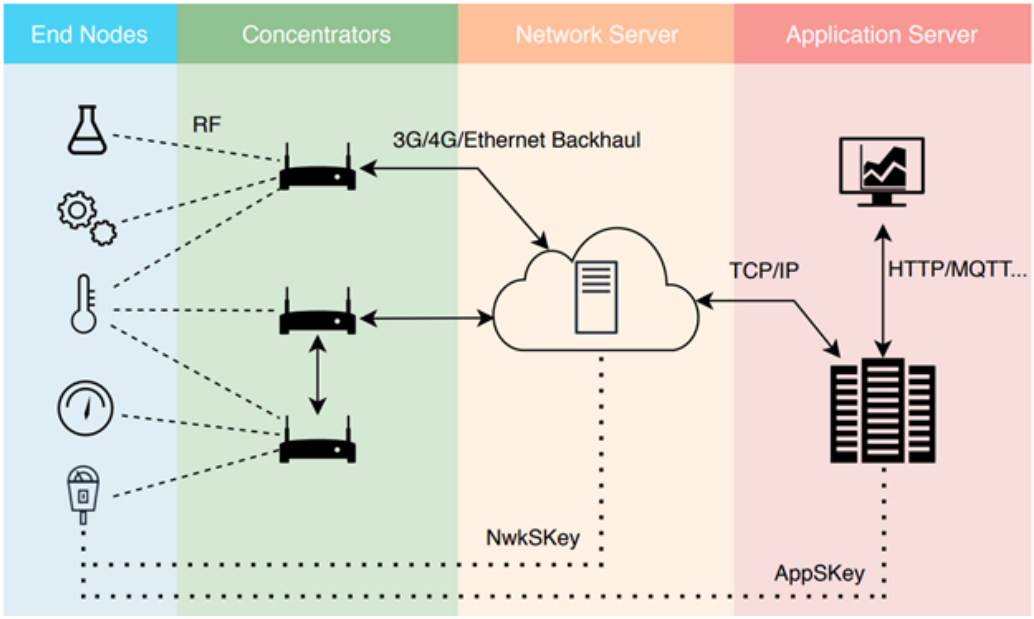


*Figure 4.* LoRaWAN Network Architecture [22]

LoRaWAN end nodes, or end devices, are microcontroller units that contain LoRa radio frequency transceivers and sensors. The transceivers are optimized for low current consumption and high interference resistance. These devices have an overall low cost because the radio frequency modules use an inexpensive crystal oscillator. End nodes are also portable, easy to provision, and available with commercial off the shelf technology. Many open-source platforms allow the framework's customization to test the interaction between the microcontroller and the transceiver code. Gateways collect the radio frequency transmissions from end nodes. Before a packet is sent to the gateway, the data field is encrypted by an AppSKey, and the Message Integrity Code (MIC) is generated by a Network Session Key (NwkSKey) [10].

Gateways, or concentrators, are devices that allow networking operations and, in this case, act as LoRaWAN base stations and even sensor nodes. The gateways enable large scale deployment of end nodes and have a wide range of flexibility and customization options. Gateways that are not connected to the internet can act as beacons to relay end node data to another internet-connected gateway. Gateways typically offer multiple backhaul options such as Wi-Fi, LTE, and Ethernet as well as built-in GPS modules to keep track of remote gateways. A gateway is managed by and sends data to a network server located in the cloud, or if a gateway-embedded network server is in use, directly to that local network server [10].

Network Servers provide the security and scalability of the LoRaWAN network. Network Servers provide integrity checks of each message via the MIC. When similar payloads are sent, the MIC changes because the frame counter is incremented in each following message. The NwkSKey is provided to the end node and network server upon activation. The network server uses the NwkSKey and the AES-CMAC algorithm to compute the MIC from the frame [8][10]. If the MIC calculated matches the MIC attached to the frame, the integrity of the payload is intact. If the message was altered, then the MIC would change, meaning that it may not be from a known source or modified in transit. The network server can send messages downlink to the end node and interface uplink with the application server. Most application servers provide a network management interface to adjust parameters and settings. Network servers also interact with databases to store device information and allow interfacing with Application Servers [10].

Application Servers allow the decrypting of application payloads to store, collect, or visualize the incoming data. Much like the network server uses a NetSKey, the application server has an AppSKey, yet it uses AES-128 CTR mode to decrypt the data [8]. Typically, an application server has a web-interface that allows the devices, users, and services to be configured, and a RESTful API is exposed to gather the data [27][29]. Application servers can retrieve the state of the gateway to see if it is up, show gateway location, and show real-time logs of live frames or events coming into the server [10].

### 2.4 LoRaWAN Protocol Stack

The LoRaWAN Protocol stack in Figure 5 shows the hierarchy between the MAC and physical layer. Starting from the bottom up, the Physical Layer defines the Regional ISM Band, which outlines the allotted frequency bands for the US, EU, AS, KR, and other geographical regions. The LoRa Modulation Layer contains the radio and modulation standards, with software, hardware, and radio frequency regulations. This LoRa layer deals explicitly with the bit layer implementation of the radio frequency. Next, the MAC layer is simply the media access control layer that the LoRaWAN protocol adds to the system. The primary purpose of a LoRaWAN node is to transmit Application Layer data to the medium to collect and visualize the data. In this layer, security is essential to ensure the network operator has no access to the application data. Each layer has specific functionality and standards to ensure that the protocol operates as intended by Semtech.

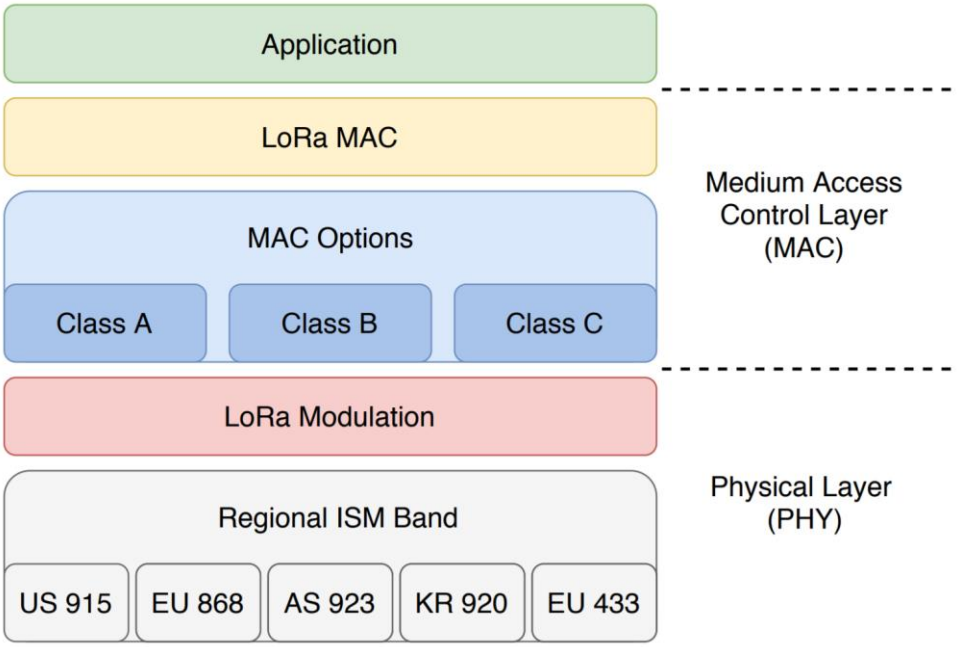


*Figure 5.* LoRaWAN Protocol Stack [10]

In Figure 6 below, The LoRaWAN message format is displayed. The Physical Layer of the Payload seats the Preamble, PHY Header, PHY Header CRC, PHY Payload, and CRC. The LoRaWAN aspect of the message is nested inside of the PHY Payload within the physical layer. Within the network layer, there is the MAC Header, MAC Payload, and Message Integrity code. The MAC payload contains application layer data: A Frame Header, Frame Port, and Frame Payload. The Frame Header contains unencrypted metadata of the device, which is the Device Address, Frame Control, Frame Counter, and Frame Options.

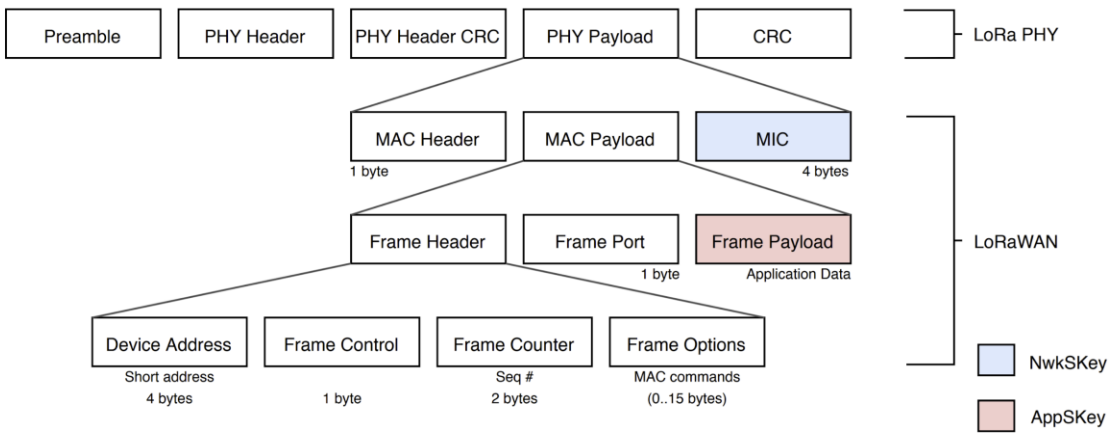


*Figure 6.* LoRaWAN Message Format [10]

In Tables: 2, 3, and 4 below: each component of the network and application layer of the LoRaWAN message is detailed. The knowledge of the specific functions that each field provides within the PHY Payload allows a better understanding of the effects they have on the system's safety and efficiency. It is essential to understand the function of public fields versus encrypted fields or message integrity codes to understand what metadata can be collected. Public fields allow faster processing for the gateway and LoRaWAN Network Server and enable the LoRaWAN Network Server to quickly determine NwkSKey's use on the payload to check the MIC. Other public fields are useful, yet the most identifying them are the Device Address and Frame Counter. The LoRa PHY fields in Figure 6 are not a feature of the LoRaWAN protocol but rather the LoRa radiofrequency physical layer.

Table 2

*LoRaWAN PHY Payload Structure*

| PHY Payload | Size (bytes) | Description |
|---|---|---|
| MAC Header | 1 | Distinguishes between six different MAC messages. |
| MAC Payload | 1 to M | Contains the Application Layer Data. |
| Message Integrity Code | 4 | Calculated over the MAC Header and MAC Payload( Frame Header, Frame Port, and Frame Payload). |

*Note*. Information is retrieved from the LoRaWAN Specification [10].

Table 3

*LoRaWAN MAC Payload Structure*

| MAC Payload | Size (bytes) | Description |
|---|---|---|
| Header | 7 to 22 | Contains device specific data. |
| Port | 0 to 1 | An application-specific, fully customizable field. |
| Payload | 0 to N | The AppSKey encrypts the sensor data payload. |

*Note*. Information is retrieved from the LoRaWAN Specification [10].

Table 4

*LoRaWAN Frame Header Structure*

| Frame Header Field | Size (bytes) | Description |
|---|---|---|
| Device Address | 4 | A 32-bit identifier for end-devices contains a NwkAddr and NwkID. |
| Frame Control | 1 | Network control information that controls data rate and acknowledgments of previous messages. |
| Frame Counter | 2 | Field used for numbering the sequence of the frame. A unique Frame Counter adds MIC variability. |
| Frame Options | 0 to 15 | Used to change data rate, transmission power, and check connection. |

*Note*. Information is retrieved from the LoRaWAN Specification [10].

Within the MAC Options layer, the architecture provides three device classes of connectivity node to gateway connectivity: Class A, Class B, and Class C. Class A stands for “All” device functionality, Class B stands for “Beacon” communications, and class C represents “Continuous” communications. The experiments in this work are solely on Class A devices as they are more commonly used and are used when longer battery life is desired. The varying device classes and their receive states are visualized in Figure 7 below.

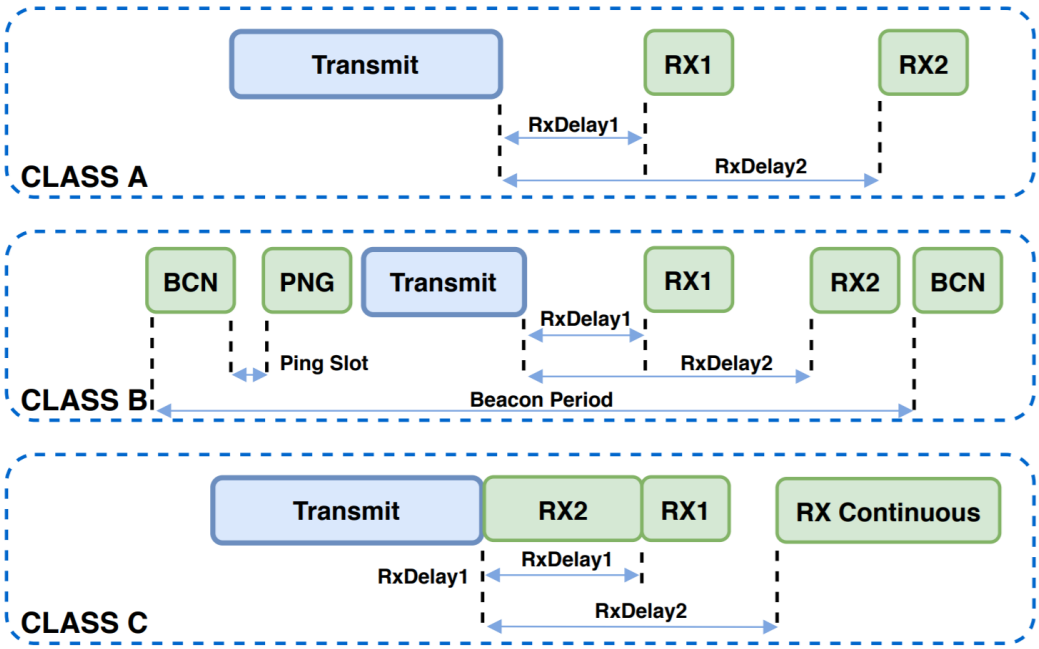


*Figure 7.* LoRaWAN Communication Classes [24]

Class A devices are optimized to minimize power expenditure and to have high energy efficiency. Class A devices are battery-powered and bidirectional: where the nodes can transfer data uplink to a gateway and accept downlink transmissions momentarily. These devices are useful when sensors or actuators are used that do not have latency constraints and cannot be hardwired or connected to an external cellular or Wi-Fi network. Every LoRaWAN device supports this base type of communication. The end node opens two slots of time in which the device can receive transmissions from a gateway. If a class A device does not receive a message, the node will perform without interruption and still send uplink messages to the gateway [11][16].

Class B or Beacon devices are battery-powered and bidirectional, much like Class A devices, yet these devices also open up extra received windows at scheduled intervals. Class B devices have two receive slots like Class A devices. In addition to the two receive slots, Class B devices open additional receive slots at scheduled intervals to collect downlink messages. The gateway sends a network beacon transmission to the end node to start the beacon period. Within this beacon period, the end device opens a ping reception slot at a given interval. When a downlink ping is received from the gateway, the end node sends the response. It is worth noting that a Class B device has inherent downlink latency restrictions and cannot support Class C functionality [11][16].

Class C, or Continuous, devices are mains powered bidirectional devices and optimal for actuators and sensors. The two-way communication is maximized due to lower latency and a more continuous listening of downlink messages/commands. There is a large power draw because there is always an open receive window for the node if it does not transmit data. Class C devices have the two receive slots activated after a transmission, much like Class A devices. Still, after the second receive slot, the receive window remains open until the next uplink message to the gateway. The majority of the research within this document will not consider these devices because they are not battery resource-restricted and not impacted by changes to the LoRaWAN protocol [11][16].

### 2.5 LoRaWAN Device Activation

LoRaWAN devices can be activated on the network by multiple procedures. These procedures include Activation by Personalization (ABP) and Over the Air

Activation (OTAA) [10]. Each of these activation methods has its benefits and drawbacks regarding convenience, security, and power-efficiency.

Activation by Personalization (ABP) is the hardcoding of the Device Address (DevAddr), NwkSKey, and AppSKey onto the end-device and Network Server [10]. As detailed in Figure 7, ABP does not require a handshake between the node and the network server; as soon as the node joins the network and the network server has the keys mentioned earlier, the node can start sending data to the network. The DevAddr is a 32-bit device address that is assigned to the device manually. The AppSKey is used to encrypt the frame payload (our data) using AES-CTR within the MAC payload nested inside the PHY Payload [11]. Once the data is encrypted, the MIC is computed over the entire MAC frame payload. The MIC is appended to the MAC frame payload to ensure that the packet has not been modified in transit. When using ABP, the device is locked to specific network and application servers. Not having a schedule to update the session keys always will save the battery life of the node over time, but this will add more physical security risk to the system. The downside to this scheme is that the keys are not session keys anymore. This downside means that there are preloaded keys, in which sending new keys to the device is only possible if you have physical access to the device.

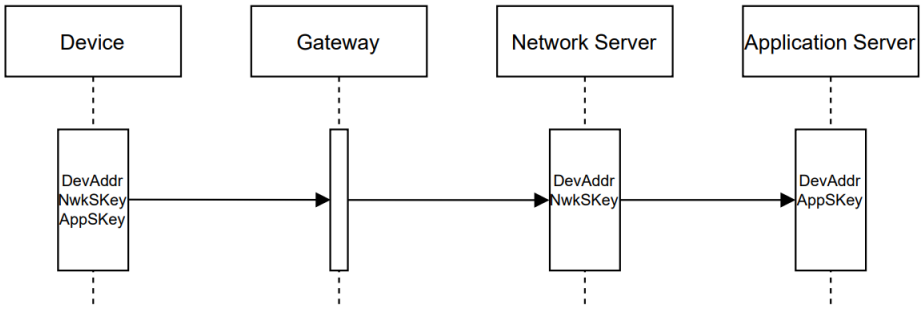


*Figure 8.* LoRaWAN: Activation by Personalization [10]

Over the Air Activation (OTAA) is the most recommended method for a node to join the network [10]. As seen in Figure 8, OTAA is safer because it allows mutual authentication between the device and the network server. With OTAA, the session keys are unique per device per session and are regenerated during every join. OTAA is a procedure that relies on the preloading of a DevEUI, an AppEUI, and an AppKey. The DevEUI is an Extended Unique Identifier that is a EUI-64-based unique identifier attached to the node upon manufacturing. The AppEUI is a global application ID that uniquely identifies the node and is used for the activation procedure. The AppKey, or application key, is a 128-bit AES key that helps generate the AppSKey and NwkSKey. AppKey privacy is important because if an attacker gets the AppKey, they could generate the AppSKey for themselves. Once OTAA is initiated, the NwkSKey and AppSKey are calculated from the AppEUI and AppKey to join the network successfully. When the device is rejoined to the network, a DevAddr, NwkSKey, and AppSKey are now provisioned. This process is unlike in ABP, where the keys remain the same for the node's life.

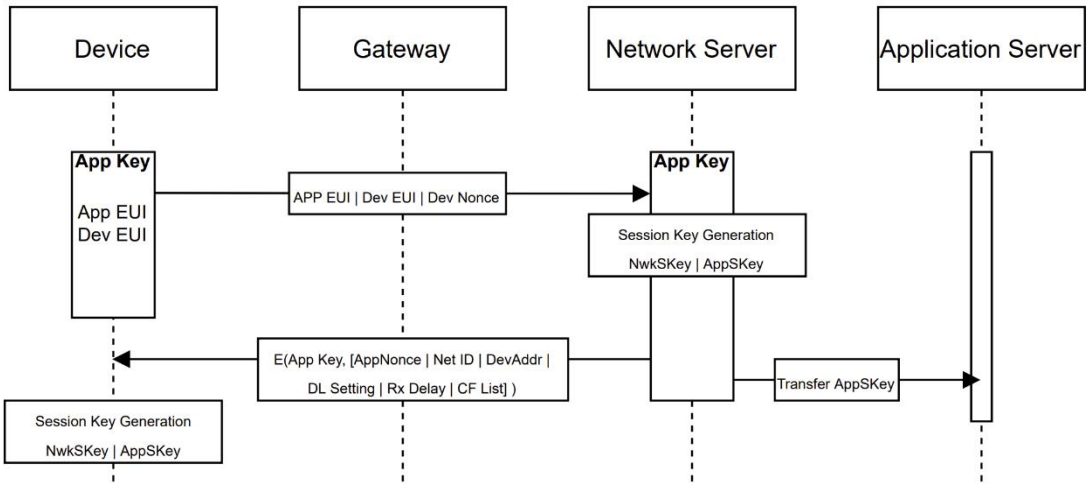


*Figure 9.* LoRaWAN: Over the Air Activation [10][26]

# Chapter 3

## Literature Review

This chapter summarizes the technologies, theories, and strategies relevant to the proposed solutions within this work. LoRaWAN technology is relatively new and continually improving, and there are a few key experiments that helped create the problem statement.

### 3.1 LPWAN Security Survey

In the paper "A Survey on the Security of Low-Power Wide-Area Networks: Threats, Challenges, and Potential Solutions," many research gaps about LPWAN networks are detailed [9]. Recent research on key LPWAN security challenges was introduced, such as replay attacks, denial-of-service attacks, and more. The effect of the attacks and various approaches proposed to solve them are listed within the publication. There are multiple types of LPWAN protocols and technologies, and they do share some similarities with LoRaWAN networks which are mainly the focus of this work.

Side Channel attacks are a threat to LPWAN devices because the secret keys are stored in EEPROM on the nodes. The paper highlights the use of better encryption, faster encryption, tamper-proof cases for nodes, and authentication measures to combat this threat. Device key management flaws or a compromised key affect the data's confidentiality in the system [9]. A solution this publication considered is the usage of a cryptographic co-processor for key storage. This allows each end node to store the keys securely to prevent physical attacks that involve extracting the key from the end node.

This requires the use of an external hardware security module or one integrated into the node itself.

### 3.2 Encryption and Power Consumption

Via experimental results, it was concluded that the selection of encryption methods used for data transmissions would affect the overall battery consumption for a mobile device [17]. The authors of the paper entitled: “Energy-aware Security in M-Commerce and the Internet of Things” found that the usage of the AES-128 increased battery consumption by 75% and processing time by 65% compared to non-encrypted transmissions. Compared to AES-128 to encrypt data, using AES-192 increased both battery and time consumption by 8%. Compared to AES-128 to encrypt data, using AES-256 increased both battery and time consumption by 16%. All of the tests were completed on a Windows XP Based laptop, and results will vary between different types of hardware and software, but generally, these figures show an expected trend. The publication also concluded that the ability to select what level of encryption primitive or configuration used would be beneficial to allow an “energy-aware” policy. Thus, the future of battery and non-battery powered IoT devices requires some variance in what protocols to use for specific devices if this technology is here to stay [17].

### 3.3 AES Cryptanalysis

D'souza et al. [18] address the AES algorithm's drawbacks and proposes improvements by using a Dynamic Key Generation and Dynamic S-Box generation. One concern surrounding AES, a symmetric encryption standard recommended by the US National Institute of Standards and Technology (NIST), is that the security of the data

will eventually be broken. AES attacks are still being researched because cyber-attacks are continuously developing, and security specialists must keep proposing new systems and schemes to keep practical attacks from being successful. The authors state that brute-force attacks, differential attacks, algebraic attacks, and linear attacks exist. The breaking of the symmetric algorithm means that private information can be intercepted and is no longer confidential. The dynamic key generation utilizes the value of time at the start of the generation and salt to create the key. The dynamic S-Box makes it more difficult for attackers to study static sets of S-Boxes.

Alanazi et al. [30] have completed research that addresses that AES security is not absolute and that the relationship between cost and time enforces AES's security. The idea is to retrieve data by finding the valid key; it will cost a large amount of money and take a large amount of time. Speculations that government intelligence services may have the technical and economic means to attack keys equivalent to about 90 bits. An investment budget of 1 million dollars can handle key lengths of about 70 bits. It is noted that if the speed of computing devices doubles every year, the higher end of Moore's Law, a crack can be made within the next 10-20 years with a high budget. If an attacker can try keys at a rate of one billion keys per second, the attacker will need around 10,000,000,000,000,000,000,000 years to break AES 128, the weakest of the AES algorithms. Increasing to 50 billion keys per second would take $5 \times 10^{21}$ years to check all possible keys, which equals about five thousand billion billion years.

When deciding what key length to use, it is essential to know the length of time the data will need to be confidential. If the data must be secured for a few hours or a few days and the information is no longer interesting or important, the encryption scheme's

choice does not need to be the optimal solution. On the other hand, if data must be secured for a lifetime, an algorithm that will stand the test of time must be chosen. An exhaustive search of AES's keyspace is impractical for many decades unless the computational power of practical attacks increases exponentially and there are no shortcuts found to bypass the need to default to a brute force attack [30].

Bogdanov et al. [31] have publicly published the best-known key attack on AES called the biclique attack. The biclique approach was originally used in hash cryptanalysis and was applied to the AES primitive with Meet-in-the-Middle attacks. The best meet-in-the-middle attack on AES-128 is only applicable to as few as four transformation rounds. The rounds add confusion to the data, and the transformation round structure of AES contains the following operations: SubBytes, ShiftRows, MixColumns, and an AddRoundKey. The best key-recovery attack on AES-128 is only applicable to as few as seven rounds. The computation complexity of brute force AES is related to the key size. The complexity to brute force, or simply to check every possible key until the plaintext is found, is $O(2^{128})$. The security of full AES has not been broken because the computation complexity required to run this theoretical attack is $2^{126}$ [31]. If the complexity was reduced from $2^{128}$ to $2^{127}$ the keyspace is reduced by a factor of two, or simply, divided in half. This means the keyspace that will need to be searched to find the correct key is smaller by a factor of four with the biclique key-recovery attack. When considering reduced rounds, the best-known attack on 8 round AES-128 is the biclique attack with a computation complexity of $2^{124.9}$ required to break it [31]. When brute-forcing full round AES-128, there are 340,282,366,920,938,463,463,374,607,431,768, 211,456 possible keys to check. When running a biclique key recovery attack on full

round AES there are about 85,070,591,730,234,615,865,843,651,857,942,052,864 possible keys. When running a biclique attack on 8 round AES there is a about 39,686,834,346,933,596,647,877,615,529,848,297,995 possible keys. This attack means that practical attacks on a reduced round version of AES 128 will take a longer time than it is worth, and therefore using reduced rounds is not unsafe.

### 3.4 Round Reduction

J.-P. Aumasson, a presenter at Real-World Crypto 2020, concluded in his paper "Too Much Crypto" that many cryptography primitives would not be less safe with fewer rounds [32]. The various primitives considered to have round reduction implemented includes AES-128, which is the only scheme used by the LoRaWAN standard. The paper concludes that if performance matters, the number of rounds chosen in creating a primitive should be picked after careful cryptanalysis. This fact is even more true in the case of LoRaWAN devices because they are resource-constrained devices. Also stated is the assumption that the paper will not cause any immediate change in the algorithms mentioned. Yet, data scientists and risk management experts should have a say in the conversation [32].

As far as security is concerned, increasing the number of rounds is a safe way to ensure more security, yet this does not benefit resource-constrained devices. More detail about the safety of reduced round AES will be prominent in Chapter 4. In a realistic scenario, a standard attacker that does not have seemingly infinite resources is more likely to find issues with implementation, protocols, and hardware before any attack on AES with reduced rounds is worth it. Aumasson argues that if your device uses AES-128

and you say that reducing rounds of AES may result in a supercomputer running a $2^{80}$ attack, then the overall problem of the security of a system cannot be answered by cryptography. You have a bigger problem on your hands [32]. When applying this logic to LoRaWAN: if sensor data is sent over public radio waves with the LoRaWAN standard AES-128 encryption and adversaries from intelligence organizations are a threat, the data should probably not be collected and sent publicly with radio transmissions at all.

Encryption works because it is unlikely that data will be decrypted in an efficient and realistic period by realistic attackers. Reducing AES rounds to the suggested nine rounds of encryption does not make it more likely that data will be decrypted in an efficient and practical time. The time to brute force even 9 rounds of AES is unrealistic inefficient. If the data is eventually decrypted, all of the required confidentiality and integrity of the data is void. Luckily, the LoRaWAN protocol was set up such that the cryptography involved is not just for data hiding but also for ensuring that the data is coming from the correct source; therefore, two keys would need to be cracked to cause the maximum amount of malice to the system.

### 3.5 Literature Review Summary

The idea presented in this work is to utilize current research factors and implement various improvements that favor battery life and those that favor security and find a balance between these two. Multiple configurations of a LoRaWAN node are tested and the results are assessed. These configurations offer to the end-user the ability

to choose to use a higher security policy based on their data security needs or a higher battery efficiency policy based on their availability needs.

# Chapter 4

## LoRaWAN Security Vulnerabilities and Solutions

### 4.1 Overview

The LoRaWAN protocol is secure in terms of data itself because the protocol implements cryptographic mechanisms to ensure payloads remain secure throughout the frame's life. The security implementation is future-proofed and makes LoRaWAN devices safe to use in many industries. Though the protocols used are secure, security vulnerabilities are exposed due to improper implementation of a LoRaWAN Network. Another reason is the use of metadata. The threat types that can affect a LoRaWAN network are disclosure threats, alteration threats, and denial or destruction threats. If proper precautions are not taken, the LoRaWAN network can become exposed to physical attacks, metadata collection, triangulation, malicious gateways, and replay attacks.

### 4.2 Physical Attacks

Due to the portable and remote nature of LoRaWAN end nodes, the most challenging security issue in IoT is resiliency against physical attacks. This type of attack is when a malicious actor tampers with hardware components or access credentials on a device. LoRaWAN devices are typically unattended and distributed, making them more vulnerable to these types of attacks, especially when they are provisioned for outdoor applications [9].

A common threat type on LoRaWAN devices is denial or destruction threats. These threats remove the devices from the network and affect the availability of the system. Alteration threats such as sensor manipulation are another way to add false data into a system that compromises the data's integrity. Next, with LoRaWAN devices, keys can be read from the devices and utilized to create a malicious node. Simple deterrent measures can be added to individual nodes within a LoRaWAN network and are listed below.

First, tamper-proof containers for the nodes can add a layer of physical protection to the hardware. If more physical security beyond that is required, movement sensors near the nodes can be implemented. The activation of the sensors can send a message to the network server to run logic that blocks uplink data or can set up a downlink payload to the nodes nearby. Similarly, data streams can be analyzed on the application server, and if the payloads received are unusual or do not fit within an input validation scheme, a downlink message can be sent to deactivate the node on the network server-side. As a last resort, kill switches, which disable a device when an attack is detected, can also be used to combat physical attacks [9].

To ensure the most significant cyber-physical attack protection, protecting the keys on the device is critical. We must protect the keys on the device that allow confidential data to be sent to the network server using a hardware security module or cryptographic coprocessor that can be added to a node. A hardware security module allows secure key storage that ensures the keys cannot be extracted from the node. The result of this is more robust authentication security to the node. The ATECC508A is a chip that offers 10Kb EEPROM memory for key storage, certificates, and data. To

protect the microcontroller's physical security, a cryptographic co-processor such as the ATECC508A has several security features built-in [28].

If a malicious entity wishes to obtain secret information from the chip via modification, manipulation or probing: the use of active shields is a countermeasure to the attack. There is a feature called Active Shield Circuitry, which provides the chip with self-detect physical attacks and helps protect the memory on the device. The integrated circuit has metal lines covering it, and there are predefined random data ran across the lines. From then, a receiver observes the data [33]. The physical attacks will disturb the integrity of the shielding lines that contain random test data as the receiver will not get the correct data because the lines will short circuit or open [33].

The ATECC508A chip also has internal memory encryption to protect the EEPROM memory that contains keys, certificates, and data from being read [28]. The chip additionally offers Glitch Protection by use of filters when active or sleep. The alternating current pulses are monitored to ensure the device ignores any shorter or longer pulses than the filter values [28]. The supply voltage and logic clock are generated in the chip, which prevents any voltage tampering on the device's pins. Included is a NIST CAVP certified Random Number Generator with a dynamic internal seed stored in the EEPROM [28].

### 4.3 Malicious Gateways

Malicious gateways are another potential threat to the security of a LoRaWAN network. Malicious gateways capture the data sent on the radio frequency or could go as far as to spoof existing trusted gateways. Through malicious gateways, attackers can filter

out packets that are upstream to the network server or downstream to a gateway and falsely confirm uplink messages so that they never make it to the network server. Utilizing confirmed uplink and downlink messages can help combat this attack [9][11].

In terms of LoRaWAN node technology, confirmed uplink messages mean our LoRaWAN device requires the network to confirm that the message was received by sending confirmed downlink messages. Confirmed downlink messages are simply the opposite, meaning the LoRaWAN network sends a confirmation frame to the node that received the message. The implementation of any of these will keep the radio frequency transmitter powered on longer and increase the overhead and thus the battery power draw over time. In this case, Class C devices that are hooked up to the main power line will benefit the most from this configuration. Class B devices implement confirmed downlink messages and, as a result, are less efficient compared to Class A devices.

Suppose the node requires downlink messages from the gateway, and there is a concern with malicious gateways sending false downlink messages. In that case, the solution is to remove the need for downlink messages entirely on standard data transmissions. This will not affect the OTAA scheme as there is a separate MAC command for that. An issue with removing downlink messages to the node is that some data may be lost if it is not requested to be resent.

### 4.4 Untrusted Network Servers

To ensure that the application can trust the network server, the same organization must create the network server. The network server and the application

server are designed to be end-to-end encrypted, but this is possible to override in reality. A malicious untrusted network server can derive the AppSKey during the OTAA stage.

If an untrusted network server does this operation, the end-to-end encryption is broken. Therefore, only trusted network servers should be used if a LoRaWAN network is required because application data is now exposed to the network server. This does limit the scalability of the LoRaWAN node because the usage of a third-party network server limits the number of available options.

### 4.5 Metadata Collection

Because LoRaWAN technology uses a public frequency, public metadata fields exist, as seen in Chapter 2.4, Figure 6. Public fields in the metadata are cleartext that is visible to all gateways that capture the radio frequency. Anyone can set up a gateway and collect the activity of end nodes. An attacker can access this and make assumptions about what may be in the payload, even though they cannot decrypt the data field within the payload. Though this is not a direct attack on confidentiality, this information may be helpful in certain situations during the reconnaissance phase. Metadata collection is difficult to solve because the metadata is used by gateways to determine if the packet should be processed. However, metadata collection can be combated with a few measures.

One method of reducing metadata collection is to utilize a highly directional radio frequency transmitter, which targets the direction of the frequency [38]. By default, LoRa devices use an omnidirectional transmitter. Targeting the frequency in a specific direction ensures that a limited number of gateways will pick up the signal. Another method to

combat metadata collection is to pad all payloads to be a set size to not reveal any hints at what is being sent by the packet based on external data observed. Padding the packets does add overhead to the LoRaWAN transmissions due to the processing involved. The extra bytes added to pad the payload will need to be processed by the encryption algorithm, resulting in a longer processing time and higher energy expenditure.

The simplest way to protect against device and frame counter metadata collection is to upload code onto the node that allows OTAA on a schedule [39]. This work will reference this scheme of OTAA on a schedule as Join Scheduling. OTAA not only requests a new NetID, DevAddr, and AppNonce, but it also resets public fields such as the frame counter and public dev address. The AppNonce allows the generation of the new NwkSKey and AppSKey. OTAA rejoins the node to the network, and the device address and the frame counter are no longer the same. If metadata is being collected and the device address and frame counter always change while using Join Scheduling, there are fewer frames with public data that can be mapped to a single node.

A more advanced method to reduce the impact metadata collection has than Join Scheduling is a change to the protocol to ensure that the device address and frame counter are not public. The original LoRaWAN standard PHY payload creation method makes device addresses and frame counters public and allows data to be tracked during the reconnaissance stage of an attack. A proposed Metadata Hiding scheme will ensure that the packet's sequence number or frame count cannot be tracked, as well as the device address.

The encryption scheme must be understood and mapped to analyze improvements to the encoding process of the PHY payload on an end node. The following steps required for payload encoding are listed below, and they are illustrated in Figure 10. The information is based on the LoRaWAN specifications for MAC message formats [10]. Figures 10 to 15 include ciphertext, cleartext, and plaintext. The ciphertext is defined as encrypted data, cleartext is information that is not intended to be encrypted or transformed even when privacy is being used, and plaintext is information that is intended to be encrypted [28].

**Step 1:** The Frame Port is set to 0x1. If the Frame Port is not set to zero, the protocol dictates that the Frame Payload is encrypted by the AppSKey.

**Step 2:** The Data field is the data that is measured by a sensor, and it is then encrypted using the AES-128 algorithm with the AppSKey as the key. Every cleartext field is populated before Step 3.

**Step 3:** Once the PHY payload is created, the MIC is computed using AES-CMAC, and it is attached to the PHY payload. The end node contains both the keys, and the network server has the NwkSKey, and the application server has the AppSKey to ensure the end-to-end encryption.

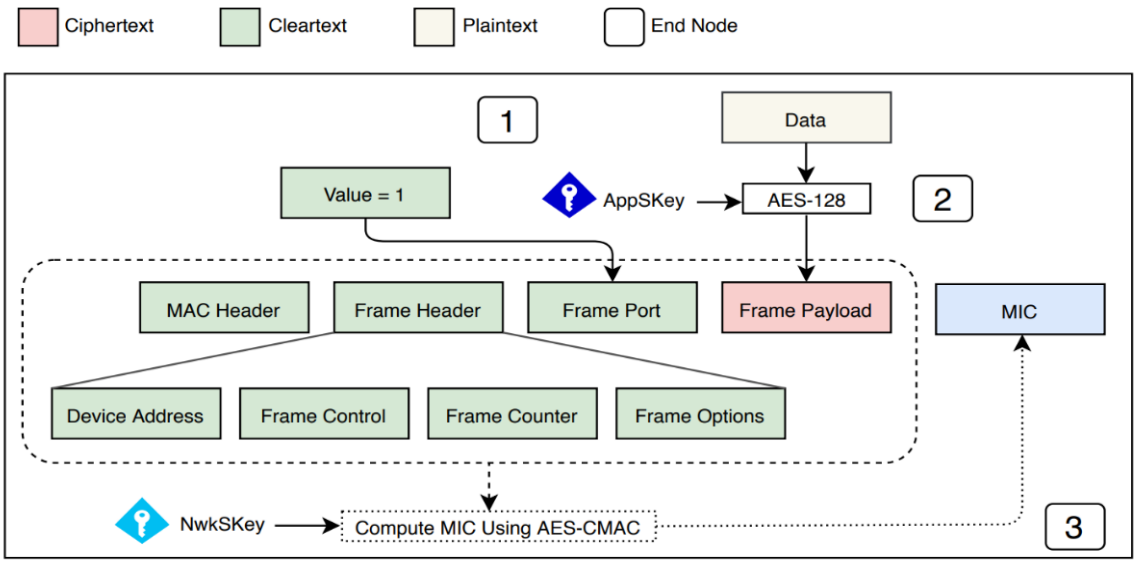


*Figure 10.* LoRaWAN Standard End Node PHY Payload Encoding

Concerning the goal of finding improvements to the PHY payload's decoding process within the network server, the decryption scheme must be understood and mapped. The following steps required for payload decoding in Figure 11 below are as follows:

**Step 1:** The Frame Port is set to 0x1, which dictates that the AppSKey was used to encrypt the frame payload.

**Step 2:** The PHY payload is sent through the AES-CMAC algorithm with the NwkSKey, and a MIC is generated.

**Step 3:** The network server verifies that the calculated MIC matches the MIC of the PHY payload. If they are both the same, then the data was not modified during transfer, and the device address and device counter were not changed.

**Step 4:** The packet has now been cleared to be sent to the Application Server, which has the AppSKey required to decrypt the payload. This step helps ensure that the network server does not have access to the encrypted frame payload, thus satisfying end-to-end encryption.

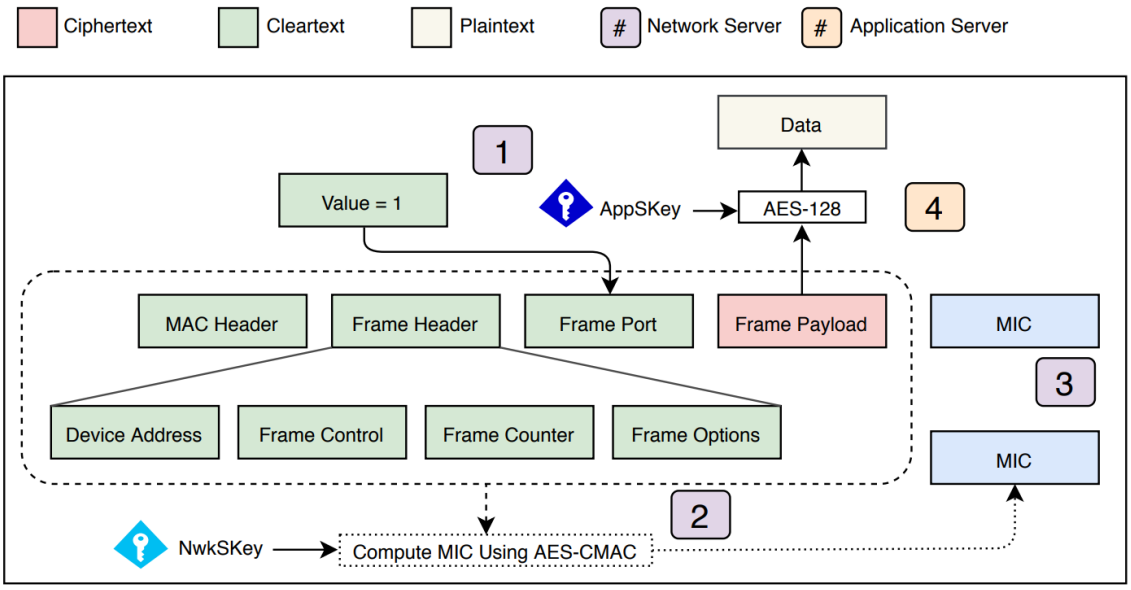


*Figure 11*. LoRaWAN Standard PHY Payload Decoding

In Figure 12 below, a proposed method of hiding the device address and frame counter is outlined to favor end-to-end encryption. A random device address and frame counter are also generated to add variability. The following steps required for improved node frame encoding with an untrusted network server are as follows:

**Step 1:** The first step is to set the Frame Port to 0x2. The value 0x2 is chosen to differentiate that the network server shall not use the standard PHY payload decoding scheme.

**Step 2:** A random Device Address and Frame Count is generated and placed within their prospective fields. Each of these values is generated with a random seed to allow a pseudo-random sequence of pseudo-random numbers. This seed is to be shared with the Network Server during the activation phase.

**Step 3:** The sensor data is encrypted with the AppSKey and placed into the unencrypted Frame Payload that contains the real Device Address and Frame Counter. The Encrypted Data field's size is equal to the size of the plaintext data field as the AES-128 CTR mode of operation is used.

**Step 4:** This step involves the encrypting of the frame Payload that contains encrypted sensor data with the NetSKey and device address, frame counter, and sensor data with the AppSKey and placing it into the PHY Payload.

**Step 5:** The final step involves computing the MIC using the AES-CMAC algorithm and the NwkSKey and attaching it to the frame. The packet can now be transmitted over the air safely.

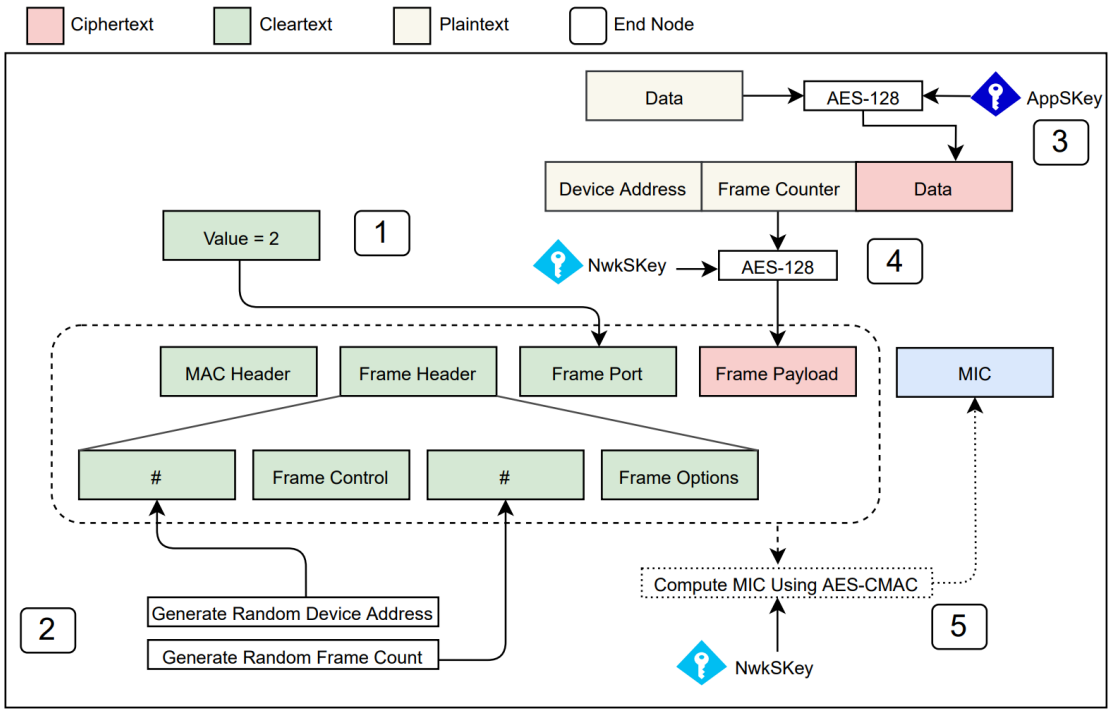


*Figure 12.* LoRaWAN Improved Node Frame Encoding with Untrusted Network Server

Within Figure 13, the proposed changes to the PHY Payload decoding scheme are made for an untrusted network server. There is overhead involved with this decoding method because the device address and frame count need to be decrypted before the packet can be considered a packet from an authenticated node. This action means that every packet that it captures will need to be checked. The overhead is negligible if the gateway is connected to a wall power supply. The steps required are as follows:

**Step 1:** The Frame Port field must be verified to contain the value 0x2 to continue to Step 2. If the value is not 0x3 (reserved) or 0x2, then the cleartext device address and frame count should be considered accurate, and the standard decoding method should be running.

**Step 2**: The proper NwkSKey for the payload can be found by matching the spoofed device address with the PRNG result after the seed is supplied. The result is precomputed before the incoming packet arrives. This allows a quick mapping from the spoofed device address attached to the packet to the NwkSKey and the real DevAddr. This step also involves using the NwkSKey and the AES-128 CTR algorithm to decrypt the Frame Payload data. The resulting data is the plaintext Device Address and Frame Counter and the encrypted sensor Data field that can only be unencrypted by the AppSKey.

**Step 3:** If the Frame Counter extracted from the decrypted frame payload is the expected value, the packet has not been replayed. If the value Frame Counter is not the expected value, the packet is thrown out.

**Step 4:** The PHY payload, except for the MIC, is sent through the AES-CMAC algorithm with the NwkSKey, then a MIC is generated.

**Step 5:** The generated MIC is compared to the MIC of the frame, and if they match, the data and device address were not modified.

**Step 6:** The Data field is encrypted by the AppSKey and placed into the Frame Payload field.

**Step 7:** The real Device Address and real Frame Counter are added to the packet and the spoofed versions are thrown out. This action ensures the Application Server knows what device the data is coming from and the correct order.

**Step 8:** Indicates a frame with the correct Device Address, Frame Counter, and Encrypted Data and can be decrypted on the Application Server by the AppSKey. The Network Server does not need to map which device address matches with an AppSKey, and end-to-end encryption is maintained.

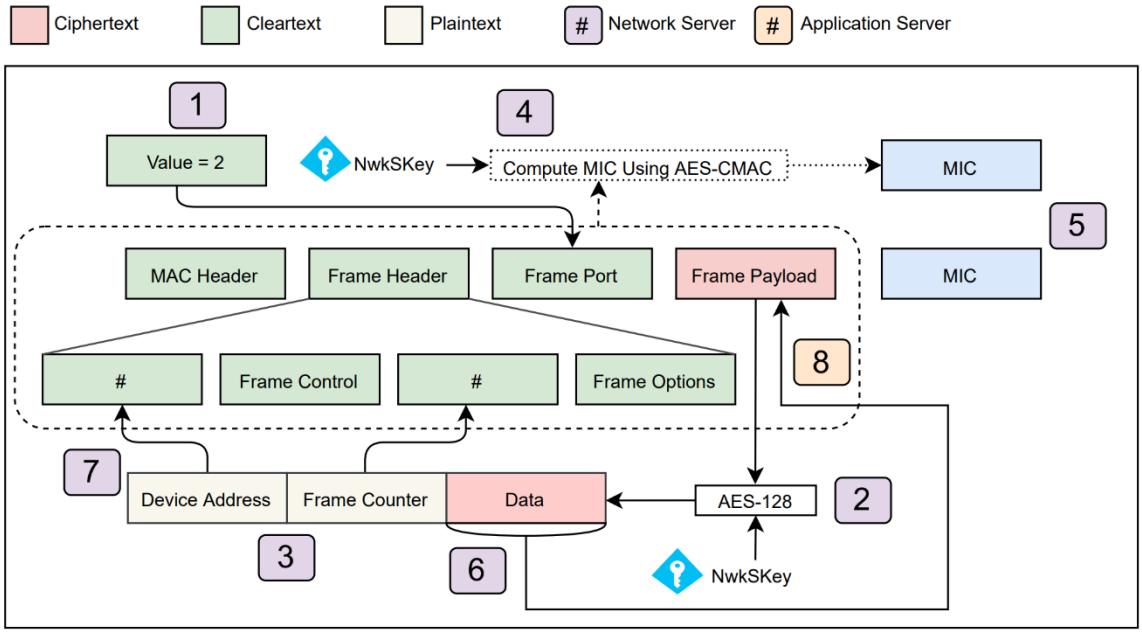


*Figure 13.* LoRaWAN Improved Network Server PHY Payload Decoding with Untrusted Network Server

In Figure 14 below, a proposed method of hiding the device address and frame counter is illustrated for a trusted network server. The following steps required for encoding a frame that will be sent to a trusted network server is as follows:

**Step 1:** The Frame Port is set to 0x3. The Frame Port can be set to any number between 1 to 255; this is to dictate that a packet with data is being sent. The value 0x3 is chosen to differentiate that the network server will not use the standard PHY payload decoding or the improved frame decoding for untrusted network servers. This allows the original method to be used to encode and decode data as well as additional procedures.

**Step 2:** The cleartext Device Address and Frame Count are replaced with random addresses and counts to add confusion to the device's true identity. Each of these values is generated with a random seed to allow a pseudo-random sequence of pseudo-random numbers. This seed is to be shared with the Network Server during the activation phase.

**Step 3:** Involves encrypting the real device address, frame counter, and sensor data with the AppSKey and placing it into the PHY Payload.

**Step 4:** Involves computing the MIC using the AES-CMAC algorithm and the NwkSKey and attaching it to the frame.

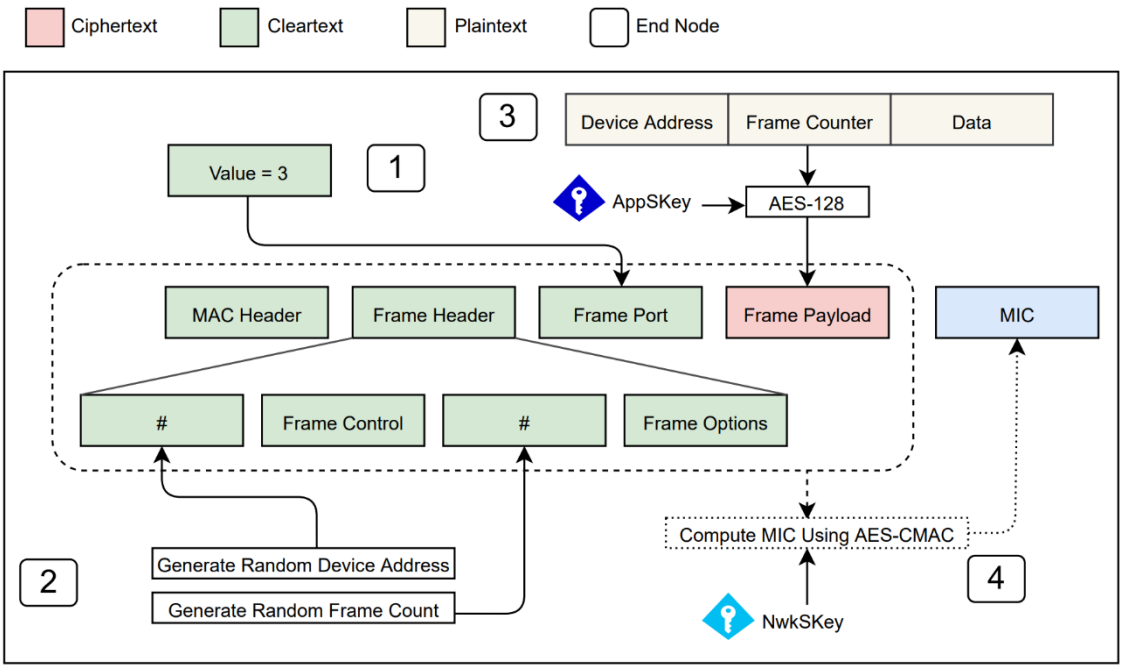


*Figure 14.* LoRaWAN Improved Node Frame Encoding with Trusted Network Server

In Figure 15 below, the proposed changes to the PHY Payload decoding scheme are made for a Trusted Network Server. Like the untrusted decoding scheme, there is overhead. Every LoRaWAN transmission that is received must have the decryption algorithm ran to extract the correct device address and frame count. The following steps required for decoding a frame that will be sent to a trusted network server is as follows:

**Step 1:** The Frame Port field must be verified to contain the value 0x3. If the field is 0x3, then the following steps should be taken. Otherwise, the original decoding scheme should be completed if the value is not 0x3 or 0x2.

**Step 2:** The proper AppSKey and NwkSKey for the payload can be found by matching the spoofed device address with the PRNG result after the seed is

supplied. The result is precomputed before the incoming packet arrives. This action allows a quick mapping from the spoofed device address attached to the packet to the AppSKey and the real DevAddr. This step also involves the AppSKey and the AES-128 algorithm to decrypt the data from the Frame Payload. Typically, the network server should not have access to the AppSKey to ensure that end to end encryption is enforced. If the application server trusts the network server and this is not required, the network server can save the NwkSKey with the node's device address. This step involves the AppSKey and the Frame Payload's decryption and unveils the proper Device Address, the Frame Counter, and sensor data.

**Step 3:** If the Frame Counter extracted from the decrypted frame payload is the expected value, the packet has not been replayed. If the value Frame Counter is not the expected value, the packet is thrown out.

**Step 4:** The PHY payload, except for the MIC, is sent through the AES-CMAC algorithm with the NwkSKey and a MIC is generated.

**Step 5:** The generated MIC is compared to the MIC of the frame, and if they match, the data and device address were not modified.

**Step 6:** The plaintext Data field is encrypted by the AppSKey and placed into the Frame Payload field. This action ensures that the Frame Payload is encrypted when it is sent to the application server.

**Step 7:** The real Device Address and Frame Counter are placed inside the Frame Header in their respective positions. This ensures the Application Server will know what device the payload comes from and the sequence it arrives.

**Step 8:** This indicates a frame with the correct Device Address, Frame Counter, and Encrypted Data and can be sent to and be decrypted by the Application Server with the AppSKey.

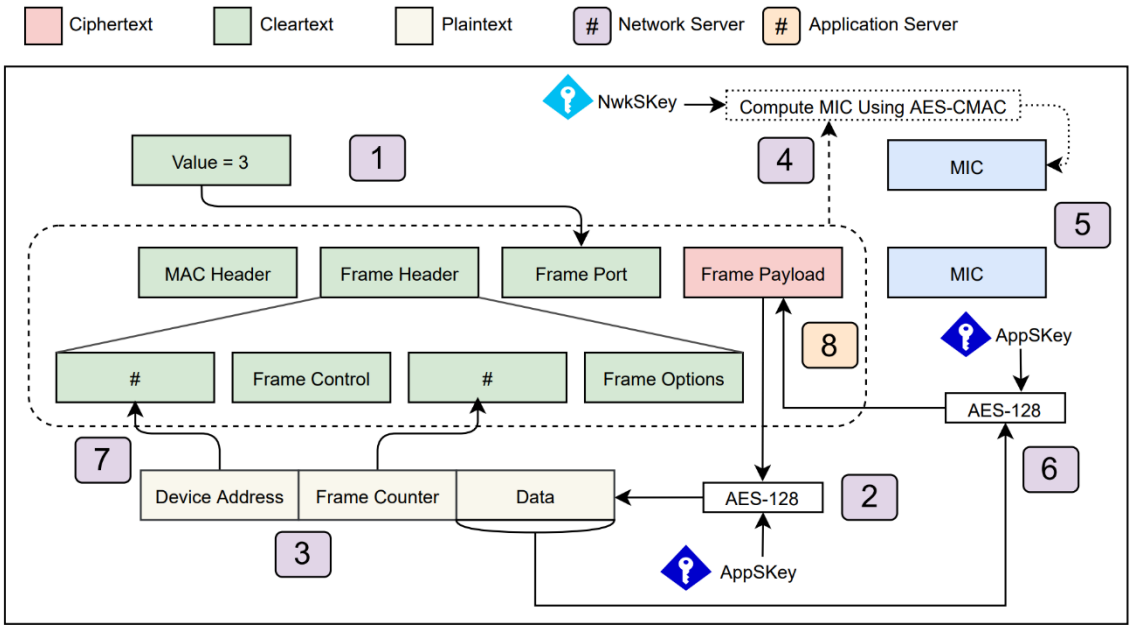


*Figure 15.* LoRaWAN Improved PHY Payload Decoding with Trusted Network Server

The encoding schemes that utilize the metadata hiding in the frame payload add time and computational overhead to the encryption method. The maximum Frame Payload size depends on the Spreading Factor required for transmission [10]. For a

Spreading Factor of 10-12, 51 bytes is the maximum frame payload size. For a Spreading factor of 9, the maximum frame payload size is 115 bytes. For a Spreading Factor of 7-8, the maximum frame payload size is 222 bytes.

The total size added to the payload with the device address and frame counter is 6 bytes. The size of the plaintext data that goes through the AES encryption is the same as the ciphertext output size due to AES-128 CTR mode [8]. With the extra 6 bytes added to the Frame Payload, the maximum application data size changes. For a Spreading Factor of 10-12, 41 bytes is the maximum frame payload size. For a Spreading factor of 9, the maximum frame payload size is 105 bytes. For a Spreading Factor of 7-8, the maximum frame payload size is 212 bytes.

### 4.6 Replay Attacks

Replay Attacks are the capturing, storing, and sending of messages that were originally sent to a gateway [9]. These messages can be publicly captured from a malicious gateway and then re-sent to a malicious node later. Replay attacks allow the attacker node to replay action from the past and send it to the server. A common mistake when implementing the LoRaWAN network is to set it up without packet filtering. One way to combat replay attacks is to ensure that the network server can keep track of the packet counter and filter out the repeated message. The MIC provides a level of replay protection because the device address and frame counter will need to be correct to calculate the correct MIC [11].

# Chapter 5

## LoRaWAN Testbed Design

### 5.1 Overview

Conductive to evaluating the cybersecurity posture and power expenditure of a LoRaWAN sensor network, the physical hardware and software that is commonly used in a real-world sensor network must be obtained and configured. It was a critical requirement to develop a platform for rigorous testing of the LoRaWAN protocol that allows complete customization and reliable measurements. A testbed implementation of a standard LoRaWAN network, as illustrated in Figure 16 below, was created. Multiple hardware and software configurations were configured to add variability in the experimental phase of this work. The testbed's core elements include the network server stack, gateway, end nodes, and power monitor.

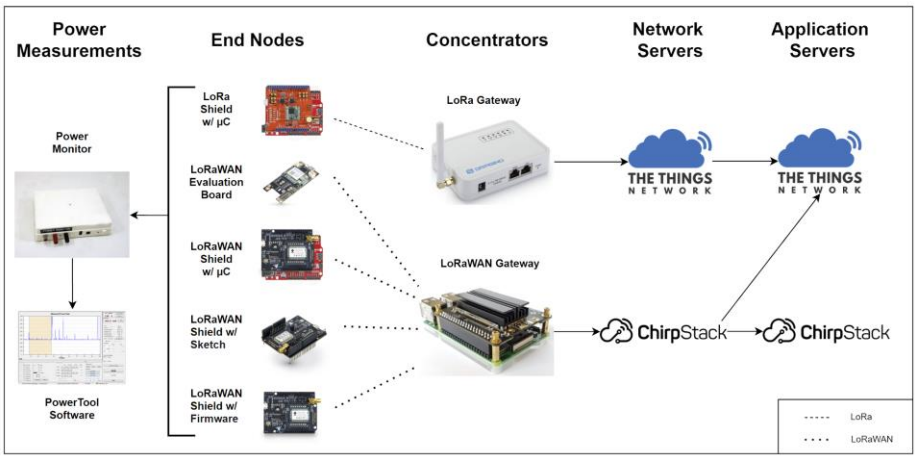


*Figure 16.* LoRaWAN Testbed Architecture

### 5.2 LoRa Gateway and LoRa Node

To get a better understanding of a base LoRa system's features, a LoRa gateway was set up to receive base LoRa data. The open-source gateway that was chosen was the Dragino LG01 [34]. With the gateway, a built-in web server is available to view, customize, and set up a means of data transfer to external platforms such as The Things Network (TTN) [35]. The web server GUI can be accessed via a Wi-Fi Hotspot or LAN. The maximum range for signal transmission to the gateway has a maximum range of about three to six miles.

For the sake of analyzing LoRa PHY transmissions without any of the benefits of LoRaWAN, the Dragino Arduino Shield featuring LoRa technology is used. The 915 MHZ band is preprogrammed into the radio frequency transmitter for the US region’s communications. The Dragino shield utilizes an Arduino Uno form factor and is controlled by a sketch loaded onto the Arduino Uno. Also required is the RadioHead Arduino library, which allows access to the Arduino radio hardware and supports the Semtech SX1276 within the shield. Figure 17 shows the form factor of the Dragino LoRa Shield.

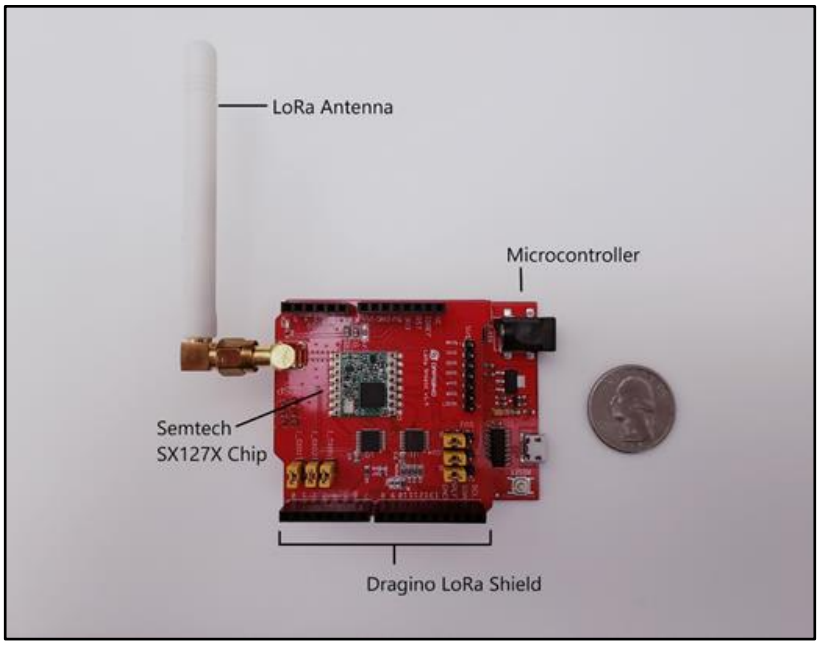


*Figure 17.* Dragino Shield with Microcontroller Board

### 5.3 Network Server Stack Architecture

Due to the star topology implementation between nodes and gateways, LoRaWAN nodes may be selected from many different manufacturers. Many different types of capture data and ways to utilize the radio transmitter hardware. LoRaWAN nodes are available in open source and commercial ready packages that allow fast development and easy integration into an existing LoRaWAN network. As long as the LoRaWAN node supports the same LoRaWAN standard and can set NwkSKeys and AppSKeys, the gateways can gather the activation by personalization context. Most of the existing stacks have the option of ABP, yet OTAA is suggested for real-world implementations due to dynamic session keys' security benefit.

The open-source network stack that is selected for analysis is the ChirpStack LoRaWAN stack. This stack was chosen because of the compatibility and extensive documentation while interfacing with various models of LoRaWAN modules. This open-source software also contains all of the software needed to gather data from the transmitter, send it to the network server, and route it to an application server. The ChirpStack LoRaWAN server's architecture contains many components that have a separation of concerns, collectively allowing data to be captured, forwarded, validated, saved, accessed, and more. Figure 18 details the full ChirpStack server stack, including Gateway components, servers, and external integrations.

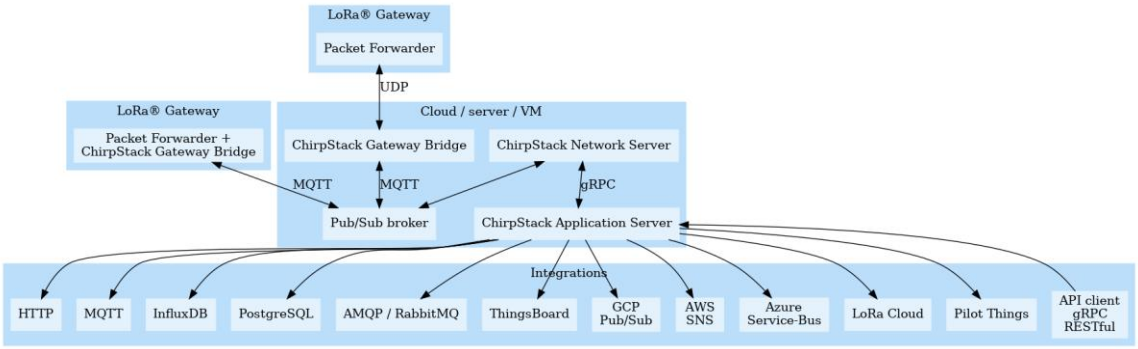


*Figure 18.* ChirpStack Architecture [20]

The components within the Cloud/server/VM section of the stack include the ChirpStack Gateway Bridge, ChirpStack Network Server, and the ChirpStack Application server. The specific software, hardware, and configuration settings for the End Nodes that send data and the gateway itself will be detailed. They offer an extended

level of complexity and a much larger scale of customization. Moreover, it is worth noting that the “Cloud/server/vm” components can all be hosted remotely or within the gateway itself, and in the case of this testbed, it is all located on the gateway itself.

The ChirpStack Application Server is responsible for the management and inventory of devices and handling join requests and the encryption and decryption of application payloads. The web interface allows access to the network server functionality and a way to create organizations, users, applications, and devices. Additionally, a REST API enables endpoints to manage the LoRaWAN network when it comes to integrations, devices, and more. Due to the device data being encrypted with the AppSKey and the NwkSKey, device data can be decrypted, and the data can be viewed for testing purposes. Figure 19 below shows the format of the Application Server and the various ways to visualize application data.

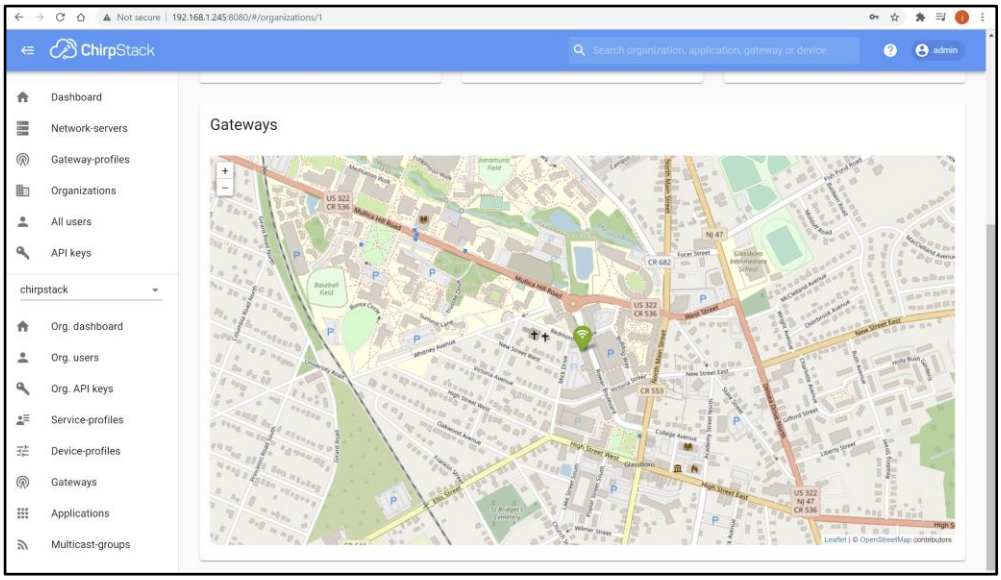


*Figure 19.* ChirpStack Application Server

The ChirpStack Network Server is responsible for all authentication, encryption/decryption, the LoRaWAN mac-layer, application server communication, and the scheduling of downlink frames. The network server makes device activation possible in an ABP or OTAA context. One advanced setting is called ADR (adaptive data rate) that lets the server control the device's transmission power and data rate. A control panel that allows the setup of devices and configuration of the network is hosted by the server and accessed at port 8080. Other features include the ability to create device profiles, set up battery status requests, synchronize time with the end nodes, manage the entire gateway, and even send firmware updates to gateways or nodes.

The Pub/Sub broker utilizes the MQTT( Message Queuing Telemetry Transport) protocol, a lightweight, standard publish-subscribe network protocol that enables the bidirectional message transport between devices [36]. The broker is the Eclipse Mosquitto MQTT broker, which is open-source, efficient, and scalable from lightweight devices to full-size servers. The specific purpose of this integration is to ensure that the data that it receives is published as JSON (JavaScript Object Notation), a standard data exchange format and file format [37]. The broker's data may be sent uplink to the network server or downlink to the ChirpStack Gateway Bridge. The ChirpStack Network Server Must subscribe to the application and device id to gather data from the broker. Figure 20 below shows the packets being accepted by a network server and application server.

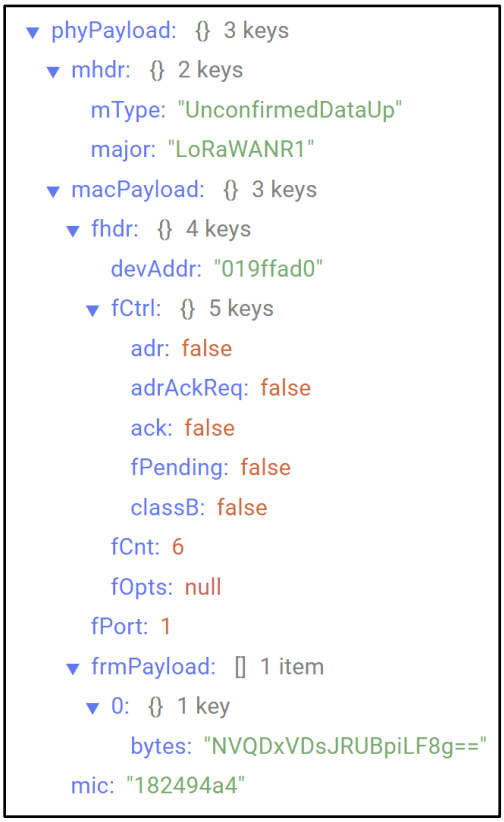


*Figure 20.* Network Server Payload


*Figure 21.* Application Server Payload

### 5.4 LoRaWAN Gateway

The open-source, commercial gateway chosen to receive the LoRaWAN node radio signal in this work is the RAK2245 RPi HAT Edition WisLink LPWAN Concentrator. This concentrator is compatible with a Raspberry Pi Form factor and is compatible with Raspberry Pi 3 Model B+ or Micro Computers for DIY setup of a commercial LoRaWAN Network. The gateway supports 903-927.5Mhz radio frequencies for the LoRaWAN standard and supports eight channels that allow different uplink frequencies to be processed by the radio transmitter. While this is a concentrator, the radio module can also send downlink LoRaWAN messages with two Semtech SX125X front end radio chips. The RAK2245 Pi Hat board arrives with an IPEX Antenna and GPS Active antenna to track the gateway's location. The hardware required to set up the gateway is detailed in Table 5.

Table 5

*LoRaWAN Gateway Components*

| Hardware Type | Description |
|---|---|
| Single Board Computer | Raspberry Pi 3B+ |
| LPWAN Concentrator | RAK2245 Pi HAT |
| Memory Card | 16GB SD Card |
| Power Supply | 5V Power Supply |

For the sake of simplicity, the Network Server and Application Server are both hosted on the Gateway. The packet forwarder processes the LoRaWAN data received by the Semtech LoRa Radio Module on the RAK2245. The packet forwarder used is the ChirpStack Concentratord [sic], a concentrator daemon decoupled from the gateway hardware, allowing the ability to run multiple packet-forwarding applications simultaneously. The ChirpStack Gateway Bridge connects the packet forwarder to the ChirpStack Network server. The bridge converts the hardware abstraction layer protocol that the packet forwarder uses to a network server readable JSON format. The data is transmitted from the packet forwarder to the gateway bridge via UDP by default but can be modified to MQTT to ensure no packet loss.

Each of these components is required to be running to accept data from end nodes. The RAK2245 is a Pi HAT (Hardware Attached on Top) that plugs directly without the need to solder into the Raspberry Pi that hosts the operating system. The gateway can act as a node to transmit data from the gateway's location as well without the need to transfer to another gateway. The RAK2245 also includes a GPS module, as seen in Figure 21 below.

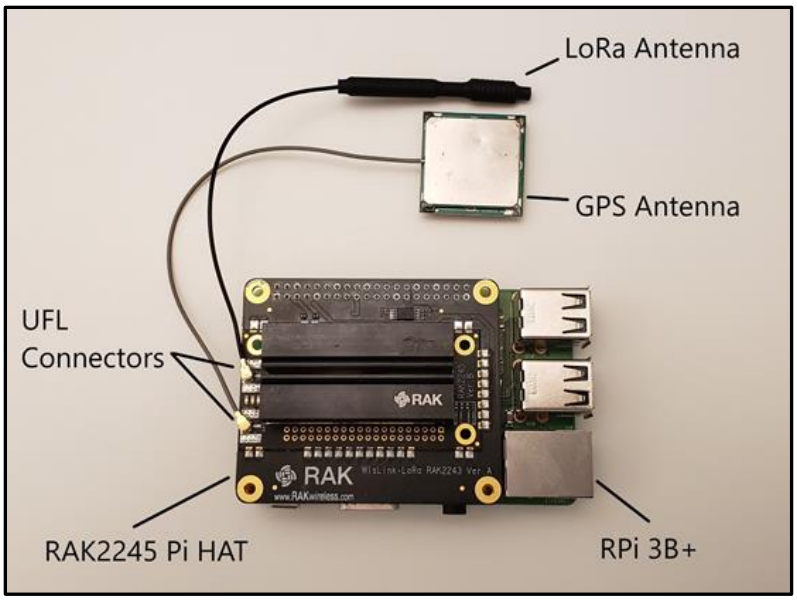


*Figure 22.* LoRaWAN Gateway

### 5.5 LoRaWAN Evaluation Board

The RAK4600 Evaluation Board is a LoRaWAN development board made up of the RAK4600 LPWAN module and a RAK5005 Base. The LPWAN module allows the usage of LoRaWAN on the 915 MHz band. The WisBlock Base offers an overall ultra-low power consumption and UART/GPIO interface for sensors. This RAK4600 EVB also has Bluetooth capabilities to update the firmware or send commands to the device during the provisioning process. The manufacturer's power consumption rating is 2.0 µA in sleep mode, which is extremely low and allows a long battery life. The supply voltage is required to be 2.0 to 3.6V.


*Figure 23.* RAK4600 Evaluation Board [21]

### 5.6 LoRaWAN Node with Stock Firmware

The WisDuino Evaluation Board (RAK811) is a development board that can optionally be attached to the GPIO pins of a microcontroller to control the radio frequency module on the HAT. The logic that controls the join, transfer, and receive functionality is all by calling AT commands provided by the shield's firmware. The name AT is an abbreviation for Attention, and these commands were initially designed in the early 80s for controlling modems. The firmware of this device uses these commands, and they are still in use in most modern smartphones to support telephony functions. Below is an image of this configuration.

*Figure 24.* RAK811 Shield

In this configuration, the RAK811 utilizes the onboard factory firmware to directly carry out AT commands from the RAK Serial Port Tool. Through the serial port tool, various settings can be extracted from the RAK811, such as the firmware version, device status, LoRa radio status, etc. Through this tool, the setup of a device can be completed without the need to upload code. The device can be set to restart, sleep, or send mode. For OTAA, the AppKey, DevAddr, APP EUI, and Dev EUI can be set up. The configuration for this node looks just like the previous configuration.

### 5.7 LoRaWAN Shield with Microcontroller Board

The RAK811 Shield with Microcontroller Board is a LoRaWAN Node that is attached to a SparkFun RedBoard Qwiic. The RedBoard has less idle current draw than an Arduino Uno and has a very similar form factor. The main benefit of using a RedBoard is adding external devices quickly via a Qwiic connector. The code on the

Microcontroller controls the firmware on the RAK811 by calling AT commands. The Node is shown below in Figure 24.

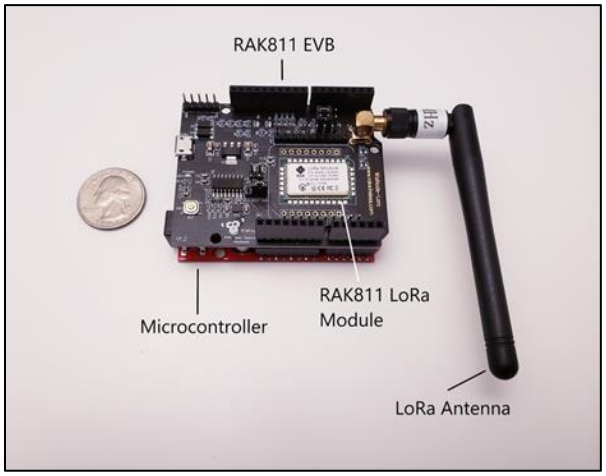


*Figure 25.* RAK811 Shield with Microcontroller Board

### 5.8 LoRaWAN Node with Arduino Sketch

The RAK811 can have an Arduino Sketch loaded onto it without the need for a base microcontroller. This configuration can handle the measurement of the sensor data through the GPIO pins. The logic that controls when to utilize the radio hardware is controlled by the sketch on board the microcontroller and the LMIC, HAL, and SPI Libraries. The configuration for this node looks just like the previous configuration without the microcontroller.

### 5.9 Power Monitoring

The equipment used to monitor the current, power, and voltage is the Monsoon Power Monitor (MPM). The measurements are measured through the main channel through the alligator clips. The alligator clips are attached to the output power and ground pins to capture the power data over time. The voltage that powers the microcontroller can be set anywhere from 3.35V to 4.5V. The fine current scale accuracy is +/- 1% or +/- 50 µA (whichever is greater). The fine current scale resolution is 1µA, which is suitable for the monitoring of LoRaWAN nodes. The MPM is pictured below in Figure 25.


*Figure 26*. Monsoon Power Monitor Hardware

The software used to interface with the Monsoon Power Monitor (MPM) is the PowerTool software. The PowerTool software is used to analyze the performance of

devices by monitoring the power data statistics. The graph shows the instantaneous metrics over time and can be zoomed in or out. Specific time frames can be selected, and the power data for that period can be displayed. Specific time frames can be chosen to power on and off the power monitor in a particular time frame. The samples captured over time can be saved in a capture file as well as a CSV file. Figure 26 displays a capture of the current overtime for a device.

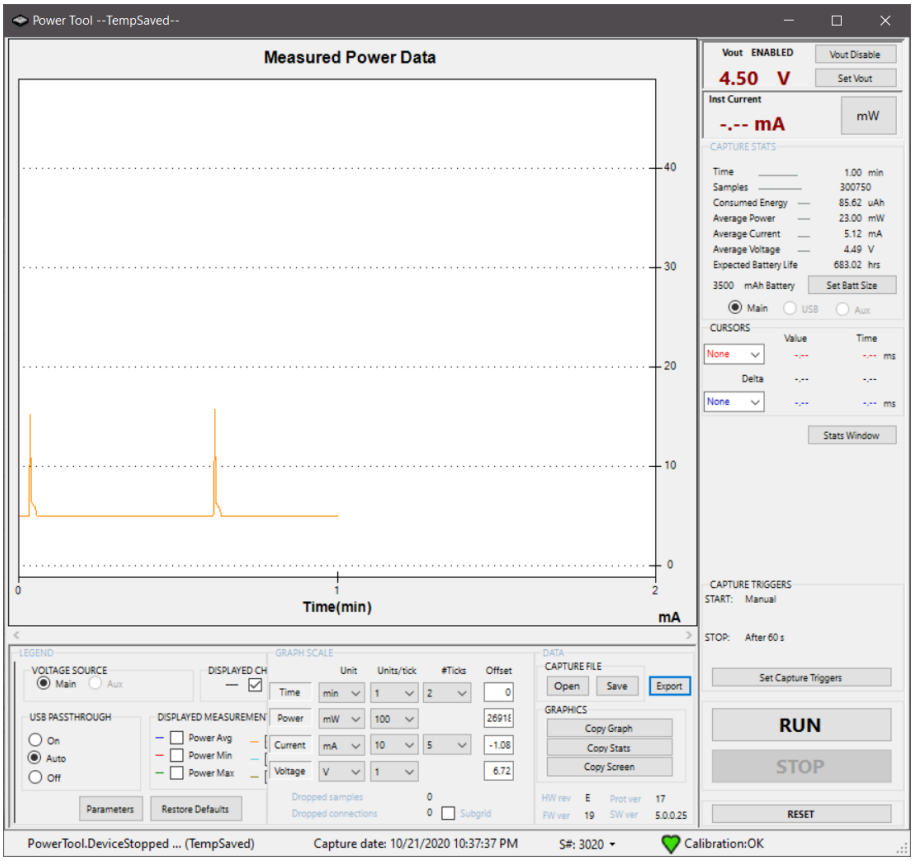


*Figure 27*. Monsoon Power Monitoring PowerTool Software

Figure 27 below looks at a real LoRaWAN transmission captured by the software as a CSV file, converted into a graph to visualize the instantaneous current over time. The Class A device modes are all shown, and the sleep mode can be set to a required interval. The Active Mode Duration includes all the activity when the node wakes up until it goes back to sleep.

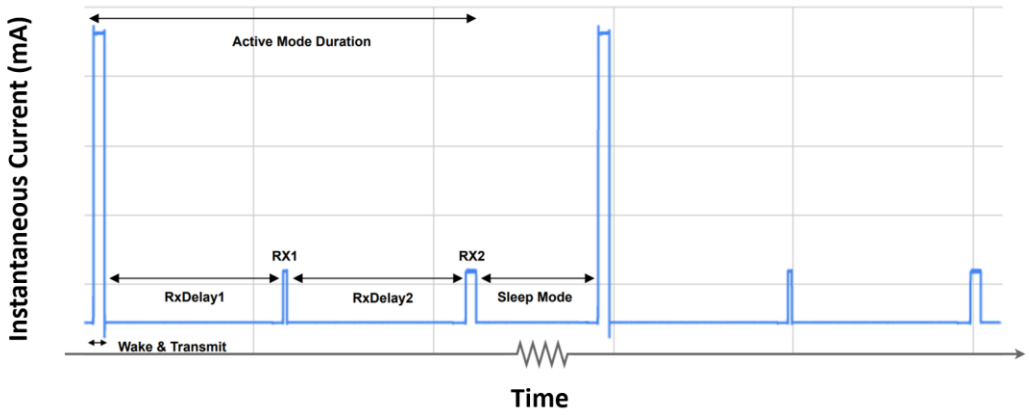


*Figure 28.* Class A Device Power Monitor Data Graph

# Chapter 6

## Experiments and Results

### 6.1 Procedure

To determine the impact of reduced rounds, join scheduling, and hidden public fields on the power draw of a LoRaWAN node, certain power data variables must be measured and analyzed. The various configurations of devices described in the previous chapter are measured to determine trends and find a viable solution to some of the detrimental side effects of the LoRaWAN networking protocol. This data gathering helps decide on optimal LoRaWAN network settings and configurations while still keeping the transferred data secure. The conclusions made will allow improved operating costs for industries using this technology. Additionally, they will also ensure that a LoRaWAN network's set-up is done with power efficiency and security in mind.

### 6.2 LoRaWAN Evaluation Board

The LoRaWAN Evaluation Board requires the lowest voltage to run and does not consume as much energy as a LoRaWAN module with an Arduino Uno form factor. The purpose of testing this board is to determine the best possible battery life for the LoRa chip used in all of the experiments. The data in Table 6 below is the average of 30 measurements and provides a benchmark for the best possible battery life over time.

Table 6

*LoRaWAN Evaluation Board Power Data*

| Power Metric | Idle Performance |
|---|---|
| Average Voltage | 3.35 V |
| Average Power | 0.96 mW |
| Average Current | 0.29 mA |
| Battery Capacity | 3500 mAh |
| Expected Battery Life | 12,151.50 hours (16.64 months) |

### 6.3 LoRaWAN Node with Stock Firmware

The LoRaWAN Shield with Stock Firmware was selected to test the firmware preloaded onto the hardware on arrival. There is no way to customize the firmware to test the round reduction or metadata hiding scheme's power data. However, OTAA is possible to measure. The data in Table 7 below shows the Power Data during the wakeup and transfer mode. The data in Table 8 below shows both the data for a full transmission and a full OTAA.

Table 7

*LoRaWAN Node with Stock Firmware Transfer Power Data*

| **Power Metric** | **Idle Performance** |
|---|---|
| Average Time | 62.864 ms |
| Average Current | 100.211 mA |
| Average Consumed Energy | 1.717 µAh |

Table 8

*LoRaWAN Node Average Power Data*

| **Power Metric** | **Result** |
|---|---|
| Average Tx Time | 2.133 s |
| Average Tx Current | 14.507 mA |
| Average Tx Consumed Energy | 8.589 µAh |
| Average OTAA Time | 5.149 s |
| Average OTAA Current | 12.545 mA |
| Average OTAA Consumed Energy | 17.909 µAh |

### 6.4 LoRaWAN Shield with Microcontroller Board

The LoRaWAN Shield with Microcontroller Board was selected to test external control of the firmware via a microcontroller. There is no way to customize the firmware to test the power data for the round reduction or metadata hiding scheme, yet OTAA measuring is possible. The benefit to this configuration is the ability to easily add additional logic to the node, such as using an external cryptographic coprocessor or sending data at specific intervals. The data in Table 9 below is the average of 30 measurements and has the most considerable overhead due to the microcontroller needing power.

Table 9

*LoRaWAN Shield with Microcontroller Transfer Power Data*

| **Power Metric** | **Idle Performance** |
|---|---|
| Average Time | 69.511 ms |
| Average Current | 104.363 mA |
| Consumed Energy | 1.991 µAh |

Table 10

*LoRaWAN Shield with Microcontroller Average Power Data*

| Power Metric | Idle Performance |
|---|---|
| Average Tx Time | 2.141 s |
| Average Tx Current | 14.987 mA |
| Average Tx Consumed Energy | 8.908 µAh |
| Average OTAA Time | 5.131 s |
| Average OTAA Current | 18.430 mA |
| Average OTAA Consumed Energy | 17.90 µAh |

### 6.5 LoRaWAN Node with Arduino Sketch

The LoRaWAN Node with Sketch was selected to modify the LMIC code responsible for the AES encryption of the data. The number of rounds can be decreased, and the power data measurements such as time, average current, and consumed energy are below in Figures 28, 29, and 30. Each measurement was completed 30 times to ensure accuracy. The average was calculated to ensure the results are less atypical. Table 11 below shows the power data metrics for the LoRaWAN Shield with the Arduino Sketch and libraries loaded onto it. There are variances in the power metrics for the sketch, mainly due to the LMIC library usage. These variances are due to the devices running on a higher average current for slightly more extended periods, especially regarding OTAA. When the devices are in the receive state, there is a higher "idle" current draw. This

results in more energy consumed over a more extended period than the nodes that utilize the stock firmware.

Table 11

*LoRaWAN Shield with Arduino Sketch Average Power Data*

| Power Metric | Idle Performance |
|---|---|
| Average Tx Time (10 Round) | 2.1207 s |
| Average Tx Current (10 Round) | 21.8209 mA |
| Average Tx Consumed Energy (10 Round) | 12.849 µAh |
| Average Tx Time (9 Round) | 2.1149 s |
| Average Tx Current (9 Round) | 21.4863 mA |
| Average Tx Consumed Energy (9 Round) | 12.614 µAh |
| Average Tx Time (8 Round) | 2.1129 s |
| Average Tx Current (8 Round) | 21.1335 mA |
| Average Tx Consumed Energy (8 Round) | 12.384 µAh |
| Average Tx Time (10 Round + MH) | 2.1516 s |
| Average Tx Current (10 Round + MH) | 22.1241 mA |
| Average Tx Consumed Energy (10 Round + MH) | 13.217 µAh |
| Average Tx Time (9 Round + MH) | 2.1494 s |
| Average Tx Current (9 Round + MH) | 22.0443 mA |
| Average Tx Consumed Energy (9 Round + MH) | 13.151 µAh |

Table 11 (continued)

| Power Metric | Idle Performance |
|---|---|
| Average Tx Time (8 Round + MH) | 2.1329 s |
| Average Tx Current (8 Round + MH) | 21.9209 mA |
| Average Tx Consumed Energy (8 Round + MH) | 12.990 µAh |
| Average OTAA Time | 5.5423 s |
| Average OTAA Current | 28.6813 mA |
| Average OTAA Consumed Energy | 44.110 µAh |

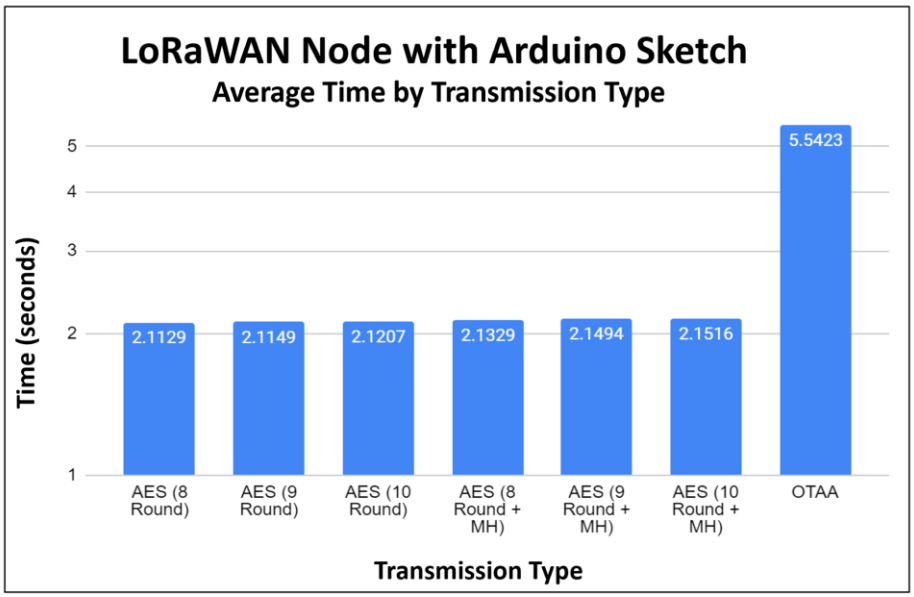


*Figure 29.* Average Time by Radio Transmission Type

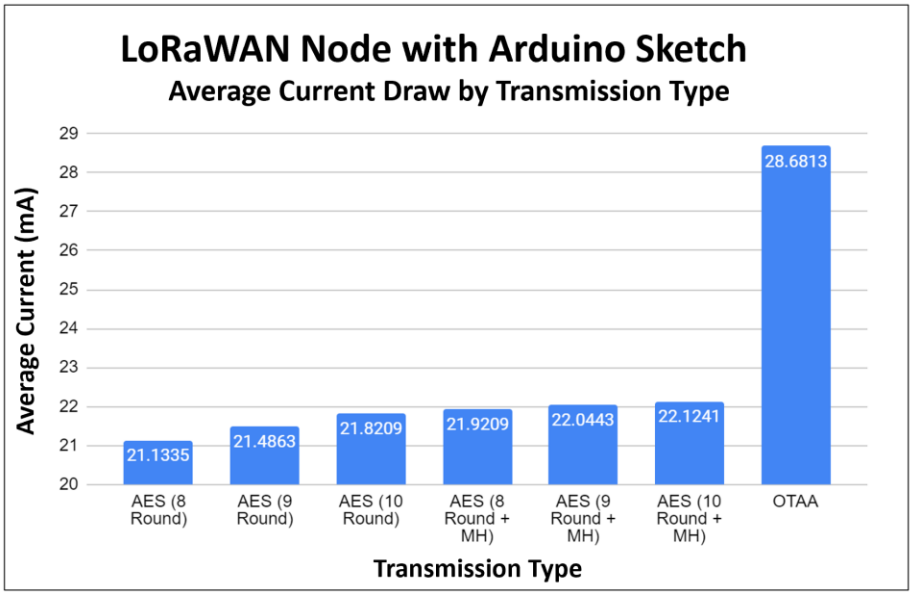


*Figure 30.* Average Current Draw by Radio Transmission Type

# Chapter 7

## Analysis and Statistics

### 7.1 Equations

The parameters used to calculate battery life over time are shown in Table 12 below. The power data and battery consumption parameters are gathered from the experiment section of this work using the power monitor. The result approximates the estimated battery life for a node at a given average current consumption.

Table 12

*Power Data and Battery Consumption Equation Parameters*

| Variable | Description |
|---|---|
| I | Interval of Data Transmission |
| $A_{Total}$ | Total Time in Active mode |
| $Tx_{Total}$ | Total Time in Wake + Transmit mode |
| $Rx_{Total}$ | Total Time in Receive mode |
| $S_{Total}$ | Time in Sleep Duration |
| $Rx_{Delay1}$ | Time in RxDelay1 |
| $Rx_{Delay2}$ | Time in RxDelay2 |
| Rx1 | Time in Rx1 |
| Rx2 | Time in Rx2 |

Table 12 (continued)

| Variable | Description |
|---|---|
| $AC_{Active}$ | Average Current during Active Mode |
| $AC_{Tx}$ | Average Current during Tx |
| $AC_{Rx1}$ | Average Current during Rx1 |
| $AC_{Rx2}$ | Average Current during Rx2 |
| $AC_{Rx1Delay}$ | Average Current during RxDelay1 |
| $AC_{Rx2Delay}$ | Average Current during RxDelay2 |
| $AC_{Sleep}$ | Average Current during Sleep Mode |
| $R_A$ | Ratio of Time in Active Mode |
| $R_S$ | Ratio of Time in Sleep Mode |
| BC | Battery Capacity in mAh |
| DC | Average Device Current in mA |
| $BL_{Hours}$ | Estimated Battery Life in Hours, based on ideal conditions |
| Variable | Description |
| I | Interval of Data Transmission |
| $A_{Total}$ | Total Time in Active mode |

Equation 1 below results in calculating the total amount of time the device is in receive mode during a transmission. The variables match those required according to the LoRaWAN Class A device transmission scheme. Equation 2 below results in calculating

the total amount of time the device is in an active mode. The total time in the wake and transmit mode and the result of Equation 1 (the total time in receive mode) are required. Equation 3 below allows the calculation of the ratio of the total time a node is in active mode between transmissions. The result requires the total amount of time throughout transmissions. Equation 4 below allows the calculation of the ratio of time in sleep mode between transmissions and requires the total amount of time the device is in sleep mode divided by the interval of transmission.

$$Rx_{Total} = Rx_{Delay1} + Rx1 + Rx_{Delay2} + Rx2 \quad (1)$$

$$A_{Total} = Tx_{Total} + Rx_{Total} \quad (2)$$

$$R_A = A_{Total} \div I \quad (3)$$

$$R_S = S_{Total} \div I \quad (4)$$

Equation 5 below results in calculating the average current during the active mode of the device, particularly the wake and transmit, and receive states. This equation requires the average current during wake and transmission, and the time the device takes in that state. Next, the equation requires the average current during the receive delay state and the time the state occurs. Next, the equation requires the average current during the receiving state and the time the device is in that state. Next, the equation requires the Second receive delay average current and the time it takes for that to occur. Next, the second receive state average current and the time it takes to receive are required. Finally,

this is all divided by the total time in active mode to get the entire activated state's average current draw.

Equation 6 below results in the calculation of the average current. The Average Current requires the ratio of time the device is active and the average active current. The average current calculation also involves the ratio of time the device is asleep, and the average sleep current measured by the power monitor as a constant.

$$AC_{Active} = ((AC_{Tx} \times Tx_{Total}) + (AC_{Rx1Delay} \times Rx_{Delay1}) + (AC_{Rx1} \times Rx1) + \quad (5)$$

$$(AC_{Rx2Delay} \times Rx_{Delay2}) + (AC_{Rx2} \times Rx2)) \div A_{Total} \quad (5\ \text{cont.})$$

$$DC = (R_A \times AC_{Active}) + (R_S \times AC_{Sleep}) \quad (6)$$

Once the average current overtime is calculated, the total amount of hours the device is estimated to be in operation can be calculated. The battery capacity in mAh is required as well as the Average device consumption in mA. Equation 7 below is used to estimate how long a battery will last. The estimate is based on the average current that a load is drawing from it and the nominal battery capacity. Most battery capacity figures are in milliamp-hours (mAh) or Amp-hours (Ah). Equation 7 below assumes a nominal battery capacity is the total amount of energy that can be withdrawn from a battery at a fully charged state at a particular current.

$$BL_{Hours} = BC \div DC \quad (7)$$

Table 13 below contains the power data and battery life equation parameters. This table outlines the parameters used for the average current for various modes of operation and their time. Some parameters are used for the calculation of OTAA when it comes to Join Scheduling.

Table 13

*Power Data and Battery Life Equation Parameters*

| Variable | Description |
|---|---|
| $A_{Total}$ | Total Time in Active mode |
| $S_{Total}$ | Time in Sleep Duration |
| $I$ | Interval of Data Transmission in hours |
| $JS_{rate}$ | Interval of OTAA in hours |
| $T_{Total}$ | Total Time per Period |
| $T_{No\ OTAA}$ | Total time without OTAA |
| $T_{OTAA}$ | Total time in OTAA mode |
| $AC_{Active}$ | Average Current during Active Mode |
| $CAC_{OTAA}$ | Average Current draw during OTAA |
| $AC_{Sleep}$ | Average Current draw during Sleep Mode |
| $AC_{Total}$ | Average Total current draw |
| $T_{Sleep}$ | Total Seconds in Sleep mode before OTAA |

Table 13 (continued)

| Variable | Description |
|---|---|
| $R_{Normal}$ | The ratio of Total Time under Normal Operation |
| $R_{OTAA}$ | The ratio of Total Time impacted by OTAA |
| $A_{Total}$ | Total Time in Active mode |
| $DC_{Total}$ | Average Total Device Current |
| $DC_{Normal}$ | Average Device Current during normal operation |
| $DC_{OTAA}$ | Average Device Current during OTAA |

The purposes of the following equations are to calculate the effect OTAA has on a schedule. To calculate the average current consumption of a device, the required equations are provided below. Equation 8 calculates the total time interval between data transmissions using the total time the device is active and the total time the device is in a sleep state. Equation 9 calculates the total time in minutes per period. Equation 10 calculates the total time where the device does not have a period that contains OTAA. Equation 11 calculates the ratio of time between the device not transmitting over the total period. Equation 12 calculates the ratio of time impacted by OTAA by dividing the interval by the total time per period. Equation 13 calculates the average current draw during the period where OTAA occurs with sleep states, wake, and transmit states, and OTAA states over the transmission interval. Equation 14 calculates the average device current with the ratio of time impacted by OTAA. Equation 15 calculates the average

device current during normal operation using the current during the normal operation and the ratio of the time during normal operation. Equation 16 factors in both the normal time and the time where OTAA occurs to get the average device current. This action factors in the OTAA scheme and the rate of normal transmissions at specific intervals.

$$I = A_{\text{Total}} + S_{\text{Total}} \tag{8}$$

$$\text{T}_{\text{Total}} = JS_{rate} \tag{9}$$

$$\text{T}_{\text{No OTAA}} = \text{T}_{\text{Total}} - I \tag{10}$$

$$\text{R}_{\text{Normal}} = \text{T}_{\text{No OTAA}} \div \text{T}_{\text{Total}} \tag{11}$$

$$\text{R}_{\text{OTAA}} = I \div \text{T}_{\text{Total}} \tag{12}$$

$$AC_{\text{OTAA}} = ((\text{T}_{\text{Sleep}} \times AC_{\text{Sleep}}) + (\text{T}_{\text{OTAA}} \times AC_{\text{OTAA}}) + (A_{\text{Total}} \times DC)) \div I \tag{13}$$

$$DC_{OTAA} = CAC_{\text{OTAA}} \times \text{R}_{\text{OTAA}} \tag{14}$$

$$DC_{Normal} = AC_{Active} \times \text{R}_{\text{Normal}} \tag{15}$$

$$DC_{Total} = DC_{OTAA} + DC_{Normal} \tag{16}$$

### 7.2 Battery Life with Round Reduction

Table 14 below contains the number of days the node will be operational when implementing the proposed Round Reduction method, based on the battery life calculations above. A lower sleep time means that the data is sent more frequently, resulting in diminished battery life. Round Reduction adds a few days of battery life when sleep time is low (data sent more regularly). Less data is sent, so more days are saved when round reduction is utilized. Round Reduction has a lower impact when sleep

times are over an hour. The idle current draw has a larger effect on battery life than transmissions.

Table 14

*LoRaWAN Shield with Sketch w/ Normal Transmissions & Reduced Rounds*

| Sleep Time | Days 10R | Days 9R | Days 8R |
|---|---|---|---|
| 15 sec | 43.74 | 44.48 | 45.20 |
| 30 sec | 80.48 | 81.73 | 82.95 |
| 60 sec | 138.75 | 140.61 | 142.41 |
| 300 sec | 329.79 | 331.87 | 333.86 |
| .25 hr | 427.99 | 429.16 | 430.27 |
| .5 hr | 462.42 | 463.10 | 463.74 |
| 1 hr | 481.80 | 482.16 | 482.51 |
| 12 hr | 501.04 | 501.08 | 501.11 |
| 24 hr | 501.95 | 501.97 | 501.99 |

### 7.3 Battery Life with Metadata Hiding

Table 15 below contains the number of days the node will be operational when implementing the proposed Metadata Hiding method, based on the battery life calculations above. Reduced Rounds combined with Metadata Hiding is also included in

the data set, and as the rounds decrease, the battery life increases. Inherently, a lower sleep time means that the data is sent more frequently, resulting in diminished battery life. There is more overhead with round reduction, and the full impact can be seen compared to other methods. Like Round Reduction, Metadata Hiding has a lower effect when sleep times are over an hour. The idle current draw has a more considerable impact than transmissions.

Table 15

*LoRaWAN Shield with Sketch w/ Metadata Hiding & Reduced Rounds*

| **Sleep Time** | **Days 10R** | **Days 9R** | **Days 8R** |
|---|---|---|---|
| 15 sec | 42.61 | 42.80 | 43.32 |
| 30 sec | 78.57 | 78.88 | 79.78 |
| 60 sec | 135.91 | 136.38 | 137.71 |
| 300 sec | 326.54 | 327.08 | 328.61 |
| .25 hr | 426.16 | 426.47 | 427.33 |
| .5 hr | 461.35 | 461.53 | 462.03 |
| 1 hr | 481.21 | 481.31 | 481.59 |
| 12 hr | 497.27 | 497.30 | 497.37 |
| 24 hr | 500.99 | 501.00 | 501.02 |

### 7.4 Battery Life with Join Scheduling

Table 16 below contains the number of days the node will be operational when implementing the proposed Join Scheduling method, based on the battery life calculations above. Reduced Rounds combined with Join Scheduling power data is also included in the data set, and as the rounds decrease, the battery life increases. The rate at which the OTAA operation is done in twenty-four hours. Like before, a lower sleep time means that the data is sent more frequently, resulting in diminished battery life. There is more overhead with OTAA. Like Round Reduction and Metadata Hiding, OTAA has a lower effect when sleep times are over an hour.

Table 16

*LoRaWAN Shield with Sketch w/ Join Scheduling & Reduced Rounds*

| Sleep Time | Days 10R + OTAA | Days 9R + OTAA | Days 8R + OTAA |
|---|---|---|---|
| 15 sec | 43.72 | 44.46 | 45.18 |
| 30 sec | 80.42 | 81.67 | 82.89 |
| 60 sec | 138.57 | 140.42 | 142.23 |
| 300 sec | 328.76 | 330.84 | 332.85 |
| .25 hr. | 426.27 | 427.45 | 428.58 |
| .5 hr. | 460.40 | 461.11 | 461.78 |
| 1 hr. | 479.61 | 480.01 | 480.39 |
| 12 hr. | 498.68 | 498.75 | 498.81 |

| | | | |
|---|---|---|---|
| 24 hr. | 499.58 | 499.63 | 499.69 |

Table 17 below shows the increase of average current consumption when join scheduling is considered. The values here can be used with the equations to calculate battery life. More frequent usage of OTAA will reduce battery life. Less frequent usage of OTAA will have a battery life comparable to the battery life of regular transmissions.

Table 17

*Increase in Average Current Consumption with Join Scheduling*

| AES Rounds | 24 Hours | 12 Hours | 6 Hours | 1 Hour |
|---|---|---|---|---|
| 10 | 0.001821 | 0.003642 | 0.007284 | 0.043709 |
| 9 | 0.001359 | 0.002719 | 0.005438 | 0.032632 |
| 8 | 0.001337 | 0.002675 | 0.005350 | 0.032101 |

### 7.5 Battery Life Cumulative Analysis

Table 18 below contains the number of days the node will be operational when considering Round Reduction and Metadata Hiding combined with Round Reduction. The results prove that eight rounds are the most efficient way to transfer packets to save battery life. Using ten rounds is a safe bet, yet adding metadata hiding and reducing the rounds to eight rounds has similar battery life. The most resource-intensive but most

secure version is ten rounds and metadata hiding. Table 19 below contains the number of days the node will be operational when considering Round Reduction and Join Scheduling. Rounds Reduction saves battery life when using eight rounds with OTAA at a 24-hour schedule compared to ten rounds with OTAA. OTAA adds some overhead, yet it is worth it if metadata hiding is not used.

Table 18

*Cumulative Analysis - Reduced Rounds vs. Metadata Hiding*

| Sleep Time | Days 8R | Days 9R | Days 10R | Days 8R + MH | Days 9R + MH | Days 10R + MH |
|---|---|---|---|---|---|---|
| 60 sec | 142.41 | 140.61 | 138.75 | 137.71 | 136.38 | 135.91 |
| 30 min | 463.74 | 463.10 | 462.42 | 462.03 | 461.53 | 461.35 |
| 1 hr. | 482.51 | 482.16 | 481.80 | 481.59 | 481.31 | 481.21 |
| 12 hr. | 501.11 | 501.08 | 501.04 | 497.37 | 497.30 | 497.27 |
| 24 hr. | 501.99 | 501.97 | 501.95 | 501.02 | 501.00 | 500.99 |

Table 19

*Cumulative Analysis - Reduced Rounds vs. Join Scheduling*

| Sleep Time | Days 8R | Days 8R + OTAA | Days 9R | Days 9R + OTAA | Days 10R | Days 10R + OTAA |
|---|---|---|---|---|---|---|
| 60 sec | 142.41 | 142.23 | 140.61 | 140.42 | 138.75 | 138.57 |
| 30 min | 463.74 | 461.78 | 463.10 | 461.11 | 462.42 | 460.40 |
| 1 hr. | 482.51 | 480.39 | 482.16 | 480.01 | 481.80 | 479.61 |
| 12 hr. | 501.11 | 498.81 | 501.08 | 498.75 | 501.04 | 498.68 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 24 hr. | 501.99 | 499.69 | 501.97 | 499.63 | 501.95 | 501.95 |

# Chapter 8

# Conclusion and Future Work

## 8.1 Conclusion

The most considerable power-efficiency improvements come from using Round Reduction when more frequent transmissions occur. As the data transmissions are spaced out, the power draw is more affected by the sleep state power draw than the savings during the less frequent transmissions. The most considerable security improvements are met with Metadata hiding. Join Scheduling should be implemented to every node when a network of devices is being provisioned.

This work introduces the concepts of Rounds Reduction, Join Scheduling, and Metadata Hiding for LoRaWAN networks. The final results of the experiments are as follows:

1. Use **8 Rounds & Join Scheduling** if a concern is Metadata Collection and long-term power expenditure.

2. Use **8 Rounds & Metadata Hiding** if you care about Metadata Collection and want to keep battery life close to that of a standard transmission with ten rounds.
3. Use **10 Rounds & Metadata Hiding** if you care about Metadata Collection and do not care about the battery life lost with the computational overhead.

In Figure XX below, the flowchart details what decisions to make when considering the methods proposed.

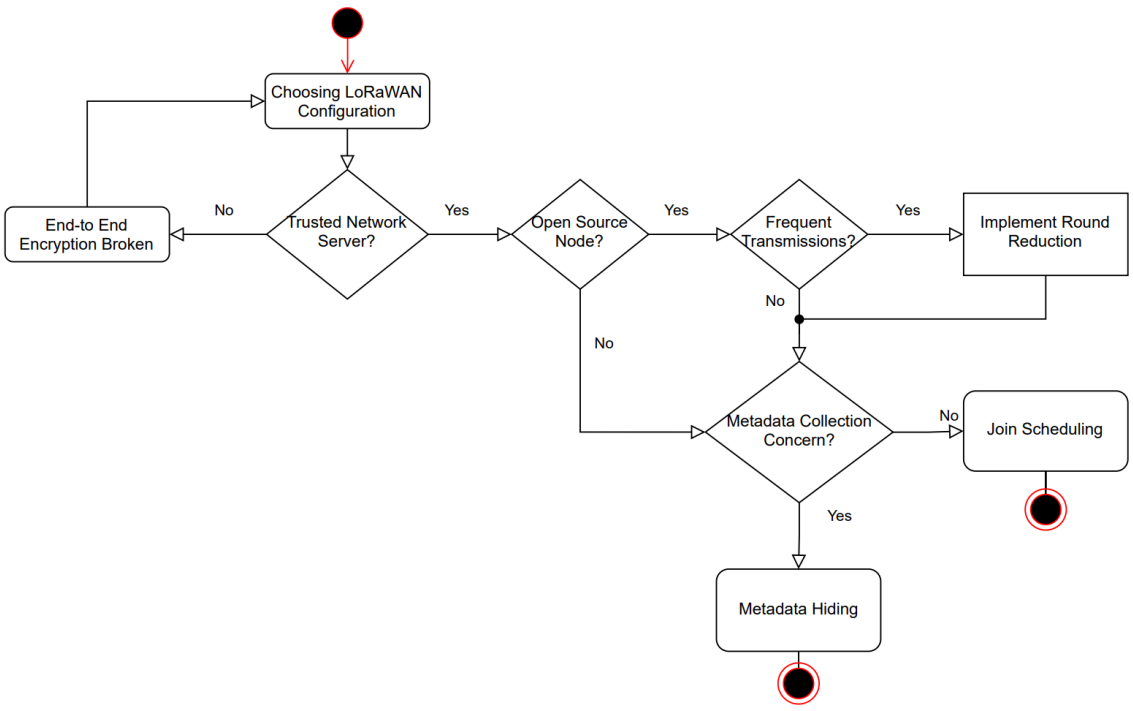


*Figure 31.* LoRaWAN Configuration Flow Chart

Though LoRaWAN is secure by design, many various vulnerabilities are inherent to the design. Some vulnerabilities are difficult to solve, such as the physical security of the keys. Some vulnerabilities can be mitigated with features inherent to the LoRaWAN protocol, such as OTAA or public fields. Modification to the source code of LoRaWAN and the ability to test real-world sensor nodes is critical to quantify the security vs. battery life of LoRaWAN nodes.

## 8.2 Future Work

The use of a cryptographic coprocessor, as described in Chapter 3, would add much needed physical security to the end nodes. An external cryptographic coprocessor would require pre-installed keys, and the data would need to be encrypted on the coprocessor and the generation of the message integrity code. Though it is easier to test with existing hardware by using an external cryptographic coprocessor, there is an inherent sleep state power draw when using the external cryptographic coprocessor. Additionally, there are many different modifications needed to the LMIC source code to ensure the encryption can occur on the external cryptographic coprocessor rather than the node itself.

Testing the optimization methods on Class B or Class C devices, mentioned in Chapter 2, is a useful endeavor. This idea will extend the security and power benefits to more resource-intensive device classes. There may be more considerable benefits of the schemes when using Class B and Class C devices.

Future-proofing LoRaWAN and discovering the overhead involved when switching from AES-128 to AES-192 or AES-256 is a useful concept. Eventually, AES-

128 will be deprecated and no longer safe to use; it will help replace the current encryption scheme and implement it into the LoRaWAN standard.

## References


[1] Rajotiya, Ravinder Nath. (2019). Advances in Wireless Technologies: A Survey, Chandigarh-4th International Conference on Science, Technology & Management.

[2] Aggarwal, Charu & Ashish, Naveen & Sheth, Amit, "The Internet of Things: A Survey from the Data-Centric Perspective," Managing and Mining Sensor Data, Boston, MA, 2013, pp. 383-428.

[3] Statista Research. (Nov 27). "Number of IoT Devices 2015-2025." Statista, 27 Nov. 2016, www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[4] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," 2016 10th International Symposium on Medical Information and Communication Technology (ISMICT), Worcester, MA, 2016, pp. 1-5, doi: 10.1109/ISMICT.2016.7498898.

[5] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the Security Vulnerabilities of LoRa," 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), Exeter, 2017, pp. 1-6.

[6] L. T. Dung, H. T. K. Tran and S. Choi, "Connectivity of Wireless Multi-hop Stochastic Networks under the Security Constraints in Rayleigh Fading Environment," 2020 22nd International Conference on Advanced Communication Technology (ICACT), Phoenix Park, PyeongChang, Korea (South), 2020, pp. 24-28, doi: 10.23919/ICACT48636.2020.9061415.

[7] A. A. Mishra, K. Surve, U. Patidar and R. K. Rambola, "Effectiveness of Confidentiality, Integrity and Availability in the Security of Cloud Computing: A Review," 2018 4th International Conference on Computing Communication and

Automation (ICCCA), Greater Noida, India, 2018, pp. 1-5, doi: 10.1109/CCAA.2018.8777537.

[8] K. Tsai, F. Leu, I. You, S. Chang, S. Hu, and H. Park, "Low-Power AES Data Encryption Architecture for a LoRaWAN," in IEEE Access, vol. 7, pp. 146348-146357, 2019.

[9] Adefemi Alimi, K.O.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S. A Survey on the Security of Low Power Wide Area Networks: Threats, Challenges, and Potential Solutions. Sensors 2020, 20, 5800.

[10] "LoRaWAN® Specification v1.0.2: LoRa Alliance®." LoRaWAN® Specification v1.0.2 | LoRa Alliance®, lora-alliance.org/resource-hub/lorawanr-specification-v102.

[11] Sanchez-Iborra, Ramon & Cano, Maria-Dolores. (2016). State of the art in LP-WAN solutions for industrial IoT services. Sensors. 16. 708. 10.3390/s16050708.

[12] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta and A. Bhatnagar, "CRAIoT: Concept, Review and Application(s) of IoT," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-4, doi: 10.1109/IoT-SIU.2019.8777467.

[13] S. Devalal and A. Karthikeyan, "LoRa Technology - An Overview," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, pp. 284-290, doi: 10.1109/ICECA.2018.8474715.

[14] "RP2-1.0.1 LoRaWAN® Regional Parameters: LoRa Alliance®." RP2-1.0.1 LoRaWAN® Regional Parameters | LoRa Alliance®, lora-alliance.org/resource-hub/rp2-101-lorawanr-regional-parameters-0.

[15] Ibrahim, Dina & Hussein, Dina. (2019). Internet of Things Technology based on LoRaWAN Revolution. 10.1109/IACS.2019.8809176.

[16] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic and A. L. L. Aquino, "LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities," 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, 2017, pp. 1-6.

[17] Hamad, F., Smalov, L., & James, A. (2009). Energy-aware Security in M-Commerce and the Internet of Things. IETE Technical review, 26(5), 357-362.

[18] F. J. D'souza and D. Panchal, "Advanced encryption standard (AES) security enhancement using hybrid approach," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 647-652, doi: 10.1109/CCAA.2017.8229881.

[19] The Things Network Global Team. (2019, July 30). LoRaWAN® distance world record broken, twice. 766 km (476 miles) using 25mW transmission power. The Things Network. https://www.thethingsnetwork.org/article/lorawan-distance-world-record

[20] ChirpStack. (2020). ChirpStack Architecture. Chirpstack.io. https://www.chirpstack.io/project/architecture

[21] Rak4600 Evaluation Board. (2020). Store.Rakwireless.Com. https://store.rakwireless.com/products/rak4600-evaluation-board

[22] LoRaWAN Architecture. (2020, November 24). The Things Network. https://www.thethingsnetwork.org/docs/lorawan/architecture.html

[23] A. Pop, U. Raza, P. Kulkarni and M. Sooriyabandara, "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254509.

[24] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic and A. L. L. Aquino, "LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities," 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, 2017, pp. 1-6.

[25] F. Mårlind and I. Butun, "Activation of LoRaWAN End Devices by Using Public Key Cryptography," 2020 4th Cyber Security in Networking Conference (CSNet), Lausanne, Switzerland, 2020, pp. 1-8, doi: 10.1109/CSNet50428.2020.9265530.

[26] S. Tomasin, S. Zulian and L. Vangelista, "Security Analysis of LoRaWAN Join Procedure for Internet of Things Networks," 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, CA, 2017, pp. 1-6, doi: 10.1109/WCNCW.2017.7919091.

[27] Application Server API. (2020, December 21). Chirpstack.Io. https://www.chirpstack.io/application-server/api/

[28] The IEEE Standards Dictionary: Glossary of Terms & Definitions. Piscataway, NJ: IEEE.2

[29] F. Paganelli, S. Turchi and D. Giuli, "A Web of Things Framework for RESTful Applications and Its Experimentation in a Smart City," in IEEE Systems Journal, vol. 10, no. 4, pp. 1412-1423, Dec. 2016, doi: 10.1109/JSYST.2014.2354835.

[30] H.O. Alanazi, B.B. Zaidan, A.A. Zaidan, H.A. Jalab, M. Shabbir, and Y. Al-Nabhani, "New Comparative Study Between DES, 3DES and AES within Nine Factors", J. Comput., vol. 2, no. 3, pp. 152-157, 2010.

[31] Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique cryptanalysis of the full AES. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 344–371. Springer, Heidelberg (2011)

[32] J.-P. Aumasson, "Too much crypto," Real-World Crypto 2020, Dec 2019.

[33] Umut Guvenc and Bilgem Tubitak , "Active Shield with Electrically Configurable Interconnections " , Securware 2013 : The Seventh International Conference on Emerging Security Information , Systems and Technology , 2013 , 43-45, Kocaeli, Turkey.

[34] Open Source LoRa WiFi Gateway Datasheet. (2019, August 31). Dragino.Com. https://www.dragino.com/downloads/downloads/datasheet/EN/Datasheet_LG01.pdf

[35] LG01 LoRa Gateway User Manual. (2018, April 3). www.dragino.com. https://www.dragino.com/downloads/downloads/UserManual/LG01_LoRa_Gateway_User_Manual.pdf

[36] MQTT: The Standard for IoT Messaging. (2020). www.mqtt.org. https://mqtt.org/

[37] Introducing JSON. (2020). www.json.org. https://www.json.org/json-en.html

[38] A. Hoeller, R. D. Souza, O. L. Alcaraz López, H. Alves, M. de Noronha Neto and G. Brante, "Analysis and Performance Optimization of LoRa Networks With Time and Antenna Diversity," in IEEE Access, vol. 6, pp. 32820-32829, 2018, doi: 10.1109/ACCESS.2018.2839064.

[39] Joining and Rejoining. (2020). Lora-Developers.Semtech.Com. https://lora-developers.semtech.com/library/tech-papers-and-guides/the-book/joining-and-rejoining/