AMERICAN UNIVERSITY OF BEIRUT

Counter Measures Against Replay Attacks On LoRaWAN Based Devices In IoT Networks

by

Maher Jallad

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Computational Sciences program
of the Faculty of Arts and Science
at the American University of Beirut

Beirut, Lebanon
December 2019

AMERICAN UNIVERSITY OF BEIRUT

Counter Measures Against Replay Attacks On LoRaWAN Based Devices In IoT Networks

by

Maher Jallad

Approved by:

Haidar Safa

Dr. Haidar Safa, Professor — Advisor — Computer Science

Dr. Wassim El Hajj, Associate Professor — Member of Committee — Computer Science

Dr. Mohamad Nassar, Assistant Professor — Member of Committee — Computer Science

Haidar Safa

Dr. Alexandre Guitton, Professor — Member of Committee — Computer Science

Date of thesis defense: December 2^{nd}, 2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:	JALLAD	MAHER	ELIE
	Last	First	Middle

⊗ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☒ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

□ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**

Two ___ years from the date of submission of my thesis , dissertation or project.

Three ___ years from the date of submission of my thesis , dissertation or project.

	February 20th, 2020
Signature	Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I would like to thank greatly my advisor Professor Haidar Safa for the support of my Masters study and related research.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Wassim El Hajj, Associate Professor Mohammad Baker, and Professor Alexandre Guitton, for their insightful comments and encouragement, but also for the hard questions which incented me to widen my research from various perspectives.

My gratitude to Professor Ali Chehab and Professor Hassan Noura who provided me with a wide information about my thesis subject which made it possible to conduct this research and reach a final conclusion.

Last but not the least, I would like to thank my family: my Wife, parents and kids for supporting me spiritually throughout writing this thesis and my life in general

An Abstract of the Thesis of

Maher Jallad for Master of Sciences
Major: Computational Sciences

Title: Counter Measures Against Replay Attacks On LoRaWAN Based Devices In IoT Networks

Recently, LoRaWAN became one of the most important low power network technologies. In fact, it can be considered as one of the most efficient technologies for the Internet of Things (IoT). Indeed, LoRaWAN offers long range, low-power, and low-data-rate. However, it faces different challenges such as availability, authentication and integrity attacks that were presented recently. One of the dangerous attacks is a replay attack in the Activation by Personalization (ABP) that can lead to a selective denial-of-service on individual end devices. In this thesis we aim to strengthen the IoT security measures in the ABP or Activation by Personalization joint procedure during key generation and exchanging process, by introducing a modification on the key management exchange of both the network and application keys.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The fundamental value proposition of the Internet-of-Things (IoT) is to enable new value cases by remotely monitoring and controlling distributed embedded systems, which together with their low cost will result in pervasive deployments increasingly dominating the technology studies [10]. To reach a mature level of IoT security, many challenges need to be addressed such as the lack of standards for secure IoT development. In addition, there is no accepted reference architecture among vendors up till now. Moreover, IoT products and services need the cooperation of many technologies and protocols, making the security of IoT even harder to be guaranteed. Therefore, IoT devices have to implement a set of security requirements to be considered secure mainly from cybersecurity prospect. LoRa on the other hand is a proprietary spread spectrum modulation scheme that is derived of Chirp Spread Spectrum modulation which allows for data transfer over long distances and low energy consumption. In this chapter, we present the IoT basic concepts and history, then specify the motivation for choosing this subject. Also we define our problem, objectives and thesis plan.

Basic Concepts and History

In the early 2000's, Kevin Ashton was laying the groundwork for what would become the Internet of Things (IoT) at MIT's AutoID lab [11]. Ashton was one of the pioneers who conceived this notion as he searched for ways that Proctor and Gamble could improve its business by linking RFID information to the Internet. The concept was simple but powerful. If all objects in daily life were equipped with identifiers and wireless connectivity, these objects could communicate with each other and be managed by computers.

At the time, this vision required major technology improvements. After all, many questions were raised such as: how would we connect everything on the planet? What type of wireless communications could be built into devices? What changes would need to be made to the existing Internet infrastructure to support billions of new devices communicating? What would power these devices? What must be developed to make the solutions cost effective?

Motivation

The Internet of Things is a fact now, as it is already spreading all over the world. Sensors and embedded devices in automobiles, phones, homes, roads, bridges, and appliances and farm equipment are already making new kinds of information available and changing the way the information is produced and experienced. IoT obviously represents a great opportunity for advances in information analysis. The connections between IoT and data storage and processing as well as machine learning are obvious and gaining attention already, except for the security, where this field is considered as the major problem that is being faced by vendors and technologists of which the most important is “how to protect our devices” from being penetrated by hackers or controlled by any malicious software. The security mechanism designed to protect communications must provide appropriate assurances in terms of confidentiality, integrity, authentication and non-repudiation of information flows. Security of IoT on the other hand is becoming very critical in terms of protection of data transfer between the end devices and the servers via transparent gateways. Such security problems are related to different type of attacks that might jeopardize the data transmission between the IoT devices and the network.

Problem Definition

A major attack is the ability to break the key generation procedure and control the end devices, in addition to the ability to locate the device physically via a GPS since the device can also send its location when installed at very large distances ranging in km's. Many approaches have proposed to protect the data confidentiality and integrity, but still no final solution has been adopted to strengthen the end devices security protection. In this thesis, we focus on a new approach for the Activation by Personalization (ABP) procedure to minimize the security problem especially in the "key management" between the end devices and the servers under LoRa protocol using LoRaWAN network. The ABP security problem is that it directly connects end-devices to the specified network without initiating a join-request and accept procedure since all keys are embedded inside, hence it does not generate any keys and can directly encrypt messages using these keys. If the keys are compromised, all communication between the device, gateway, and network server can be decrypted by third party entities for the lifetime of the device.

Objectives and Contribution Summary

As mentioned earlier, we aim at proposing a new approach that may solve the security related to "key management". The proposed solution will be based on

“Light Weight Algorithms” domain by introducing an additional hash function and more than one input for the key to be calculated. It enhances the possibility of adding different type of protection or enforcing the existing type of encryption being used nowadays by the IoT devices. Our objectives and contributions can be summarized as follows: 1. Survey the related works found in the literature and identify their advantages and limitations. 2. Propose a new approach to solve the protection of the keys interchanged between the devices and the network 3. Implement the new approach in the existing IoT design if possible 4. Evaluate the performance of the proposed approach and compare it with other types of approaches such as “Light Weight Algorithms”. The contribution of this thesis is to show the re-initialization of the frame counter in the ABP procedure where the ***Nwkeys*** and ***AppKeys*** are embedded in the end device and try to enhance the mitigation technique that prevents replay attacks,by implementing a new scheme of cryptography protocol based on hashing in LWC rather than ECC to the existing embedded keys during the re-joining sessions procedure, as well as this preposition may be a candidate for securing end devices that have their keys embedded in the future.

Thesis Plan

Chapter 2 introduces the basic concepts of IoT and LoRa. In Chapter 3 we survey the related work, addressing the different types of vulnerabilities and investigate most of the existing defense solutions; we identify the advantages and limitations of each one of them and we emphasize on the limitations that will be solved in the proposed algorithm. In Chapter 4 we introduce the proposed approach and show how it can overcome the limitations of the existing solutions. In Chapter 5 we introduce the evaluation between AES-128bits and the ECC 160-bits, in addition to the equipment used for the tests. Finally Chapter 6 concludes this thesis.

Chapter 2

IoT BACKGROUND AND BASIC CONCEPT

At a very basic level, “Internet of Things” means devices that can sense aspects of the real world like temperature, lighting, the presence or absence of people or objects, etc. IoT devices report real-world data, act on it and resulting more and more information produced and consumed by machines. IoT devices communicate between themselves to improve the quality of our lives, due to this, IoT is now being embraced by different type of technologies as will be mentioned in the following discussion.In this chapter, we will point out the basic concepts of the IoT as well the background and how it was thought of and introduced.

2.1 IoT Technologies

The adoption of technologies supporting the IoT has been increasing worldwide in a tremendously rate. Examples of such technologies are Long Range Wide Area Network (LoRaWAN) [4], Narrow-Band (NB-IoT) [4], and Wireless Smart Ubiquitous Network (Wi-Sun) [4], in addition to many others that are less famous. However, as the IoT market is rapidly expanding, LPWA (Low Power Wide Area) became one of the fast growing spaces in IoT. LoRaWAN (long range wide area network) which is part from LoRa, is being widely implemented in most IOT deployments.

2.2 LoRa

LoRa is a wireless technology developed to create the low-power, wide-area networks (LPWANs) required for machine-to-machine (M2M) and Internet of Things (IoT) applications. The technology offers a very compelling mix of long range, low power consumption and secure data transmission and is gaining significant attraction in IoT networks being deployed by wireless network operators, detailed information about the physical features are shown in Table 2.1 [7].

LoRa devices communicate with LoRa gateways in a star topology, which send data to network server and onto an application server accessible by owners of LoRa devices as shown in Figure 1.1, in which we can see the presence of end devices connecting via transparent bridges to the network server. The gateway here acts as a transparent receiver and transmitter by sending the data from the end device onto a backhaul system. It contains multi-channel transmitters for processing signals which are bi-directional. The network server on the other hand manages the network by setting up schedules, adapting data rates, storing and processing received data.

LoRa modulation technique [12] (Rb) relies on the relationship between the required data bit rate with the Chirp rate and symbol rate. This is defined in eq.(2.1):

$$Rb = SF \times \frac{BW}{2^{SF}} bits/s \times CR \tag{2.1}$$

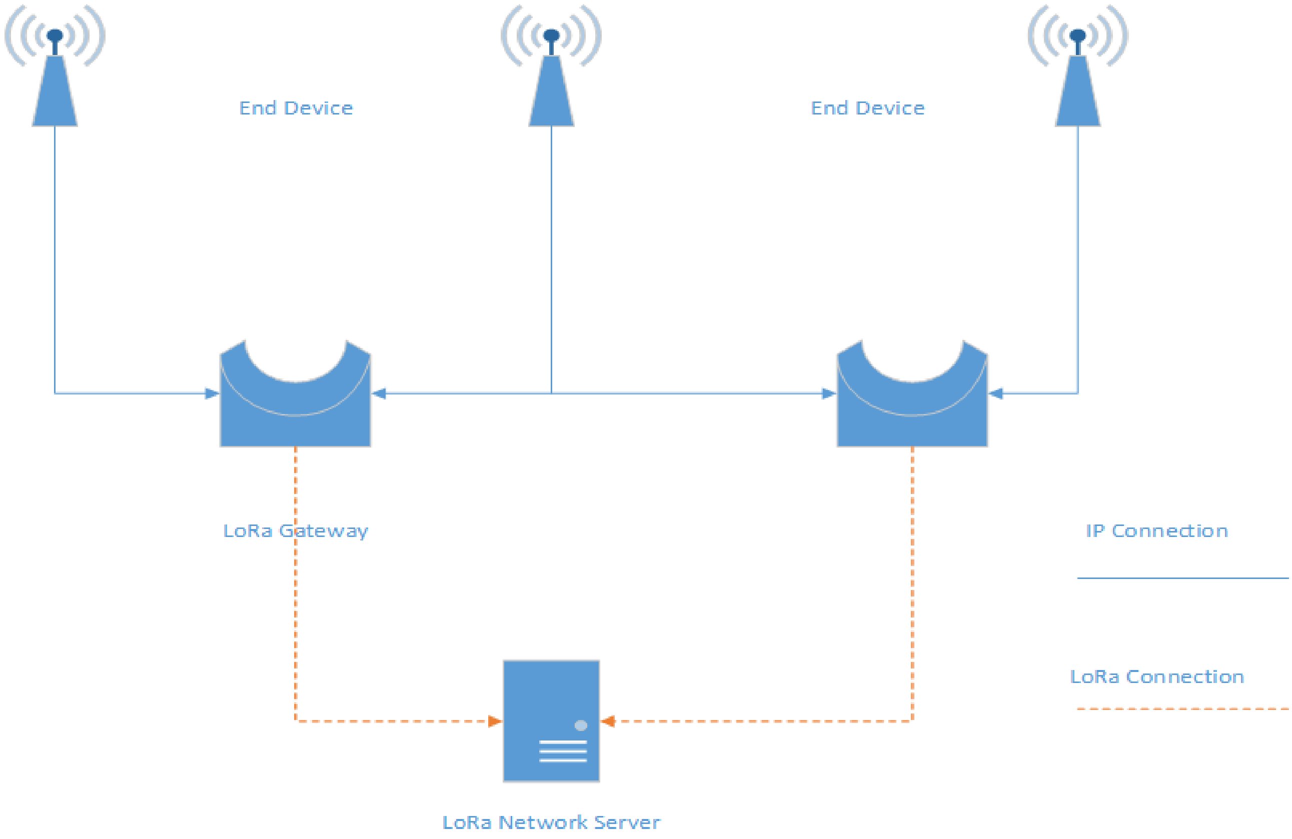


Figure 2.1: LoRa Network Architecture [1]

where SF is spreading factor, BW is modulation bandwidth (Hz) and CR is the code rate which is equal to 4/(4+n), with n $\in$ (1,2,3,4).The bandwidth is the most important parameter of the LoRa modulation. A LoRa symbol is composed of 2^{SF} chirps, which cover the entire frequency band. It starts with a series of upward chirps. When the maximum frequency of the band is reached, the frequency wraps around, and the increase in frequency starts again from the minimum frequency. The chirp rate is thus equal to the bandwidth (one chirp per second per Hertz of bandwidth) [13].

Table 2.1: Physical Features of LoRa [7]

Parameters	**LoRa**
Spectrum	Unlicensed
Modulation	CSS
Bandwidth	500-125kHz
Peak data rate	290bps-50kbps(DL/UL)
Link budget	154dB
Max.# message/day	Unlimited
Duplex Operation	N/A
Power efficiency	Very high
Mobility	Better than NB-IoT
Connection density	Utilized with NB-IoT
Energy efficiency	$\eta \geq$ 10 years battery life
Spectrum efficiency	Chirps SS CDMA better than FSK
Area traffic capacity	Depends on gateway type
Interference immunity	Very high
Peak Current	32mA
Sleep current	1 microA
Standardization	De-Facto standard

2.3 LoRaWan Structure

LoRaWAN, in its turn, is a low-power wide-area data-link protocol for multi-node networks [14]. Its network architecture is laid out in a star-of-stars topology [14] with no repeaters and mesh connections. It has end nodes through which gateways acting as transparent bridges relay messages to the central network server as shown in Figure 2.2 which shows end nodes connecting via gateway bridges to the internet cloud servers and then toward the https clients such as e-mail, customer IT and remote management. In this architecture the gateways and the central sever are assumed to belong to network operators with end nodes belonging to subscribers. Subscribers are provided with an opportunity of transparent bidirectional and secured data transfer to end nodes. The LoRaWAN specification provides seamless interoperability among smart Things without the need of complex local installations. It uses the 868MHz and 900MHz ISM bands and is able to transmit over several kilometers depending on environment. LoRaWAN implements the MAC layer as shown in Figure 2.4, where we see the different types of layers such as the application, LoRa MAC, LoRa options, modulation and lastly the regional ISM bands that it works on while LoRa technology at the physical layer. LoRa uses wide bandwidth to help protect against deliberate interference or environmental noise. LoRa and LoRaWAN devices are capable of providing data rates from between 0.3kbps to 50kbps which varies based on required range and interference.

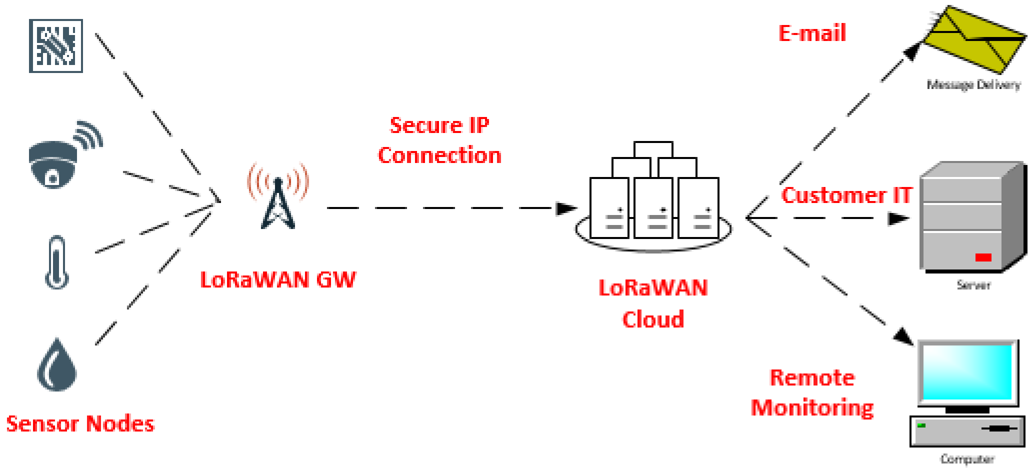


Figure 2.2: LoRaWAN Structure [2]

The LoRaWAN network applies an adaptive modulation technique with multi-channel multi-modem transceiver in the base station to receive a multiple number of messages from the channels. The spread spectrum provides orthogonal separation between signals by using a unique spreading factor to the individual signal.

This method provides advantages in managing the data rate.

2.4 LoRaWAN Elements

The LoRaWAN consists of the following elements and as shown in figure 2.3 [15],[16]:

1. End node fulfills controlling and measuring functions. It contains a set of necessary sensors and controlling elements.

2. LoRa Gateway is a device receiving the communications from the end nodes and then transferring them onto the backhaul system. This network can be Ethernet, cellular or any other telecommunications channels. Gateways and end devices build up a star network topology. Normally, this device contains multi-channel transmitters for processing signals simultaneously or even several signals through one channel. Consequently, several devices of this kind provide for network coverage and transparent bi-directional data transfer between end nodes and the server.

3. Network Server manages the network: setting up schedules, adapting data rates, storing and processing received data.

4. Application Server provides remote control over end nodes and collects data from them

2.5 LoRaWAN Classes

A LoRa network distinguishes between a basic LoRaWAN Class A devices and optional Classes B and Class C as shown in Figures 2.4 and 2.5 [15]. These classes are explained as following:

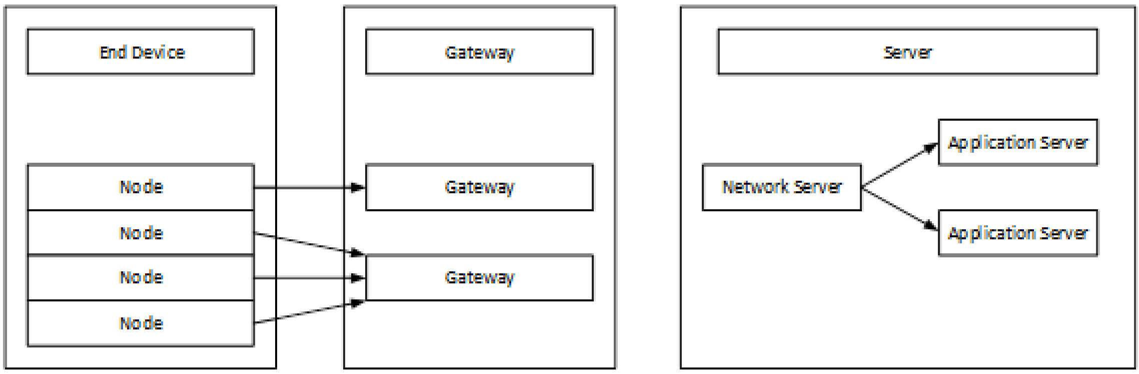


Figure 2.3: The structure of LoRaWAN Network

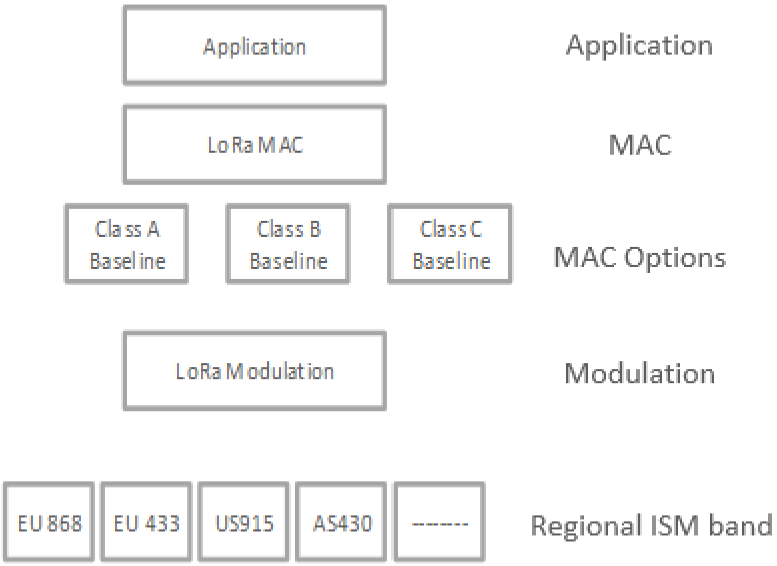


Figure 2.4: LoRaWAN Classes

1. Bi-directional end-devices (Class A): End-devices of Class A allow for bi-directional communications whereby each end-device's uplink transmission is followed by two short downlink receive windows. The transmission slot scheduled by the end-device is based on its own communication needs with a small variation based on a random time basis (ALOHA-type of protocol). This Class A operation is the lowest power end-device system for applications that only require downlink communication from the server shortly after the end-device has sent an uplink transmission. Downlink communi-

cations from the server at any other time will have to wait until the next scheduled uplink.

2. Bi-directional end-devices with scheduled receive slots (Class B): End-devices of Class B allow for more receive slots. In addition to the Class A random receive windows, Class B devices open extra receive windows at scheduled times. In order for the End-device to open the receive window at the scheduled time, it receives a time synchronized beacon from the gateway.

3. Bi-directional end-devices with maximal receive slots (Class C): End-devices of Class C have nearly continuously open receive windows, only closed when transmitting. Class C end-device will use more power to operate than Class A or Class B but they offer the lowest latency for server to end-device communication.

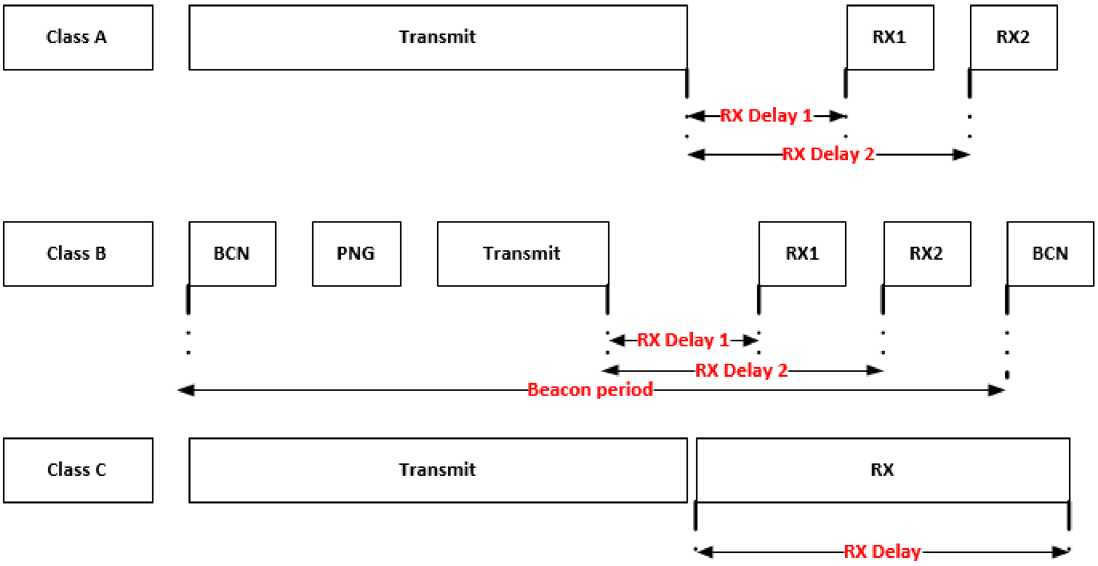


Figure 2.5: LoRa Class Types [3]

2.6 LoRaWAN Protocol Architecture

Figure 2.6 shows the protocol architecture of LoRaWAN. As shown in this figure, LoRaWAN's protocol consists of a MAC layer and an application layer, and operates based on the LoRa physical layer. The packet format is shown in Figure 2.7 where the maximum payload lengths vary with the data rate [17]. The main elements of Figure 2.7 can be defined as following:

1. MAC layer: the packet processed in the MAC layer consists of a MAC Header (MHDR), a MAC Payload, and a Message Integrity Check (MIC). In a join procedure for end node activation, the MAC Payload can be replaced by join request or join accept messages. The entire MAC Header and MAC Payload portion is used to compute the MIC value with a network session key (Nwk-SKey).The MIC value is used to prevent the forgery of messages and authenticate the end node.

2. Application layer: the MAC Payload handled by the application layer consists of a Frame Header (FHDR), a Frame Port (FPort), and a Frame Payload (FRM). The FPort value is determined depending on the application type. The FRM Payload value is encrypted with an application session key (App-SKey). This encryption is based on the AES-128bits algorithm

2.7 End Node Activation

LoRaWAN defines two joining procedures for end-devices: Over-the-Air Activation (OTAA) and Activation by Personalization (ABP) [18]. OTAA requires 64 bit end-device identifier, EUI-64 ***(DevEUI)***, the 64 bit application identifier, EUI-64 ***(AppEUI)***, and an Application Key ***(AppKey)***. An end-device must follow this procedure every time it joins a new network. OTAA is the most secure authentication method because a network session key for that end-

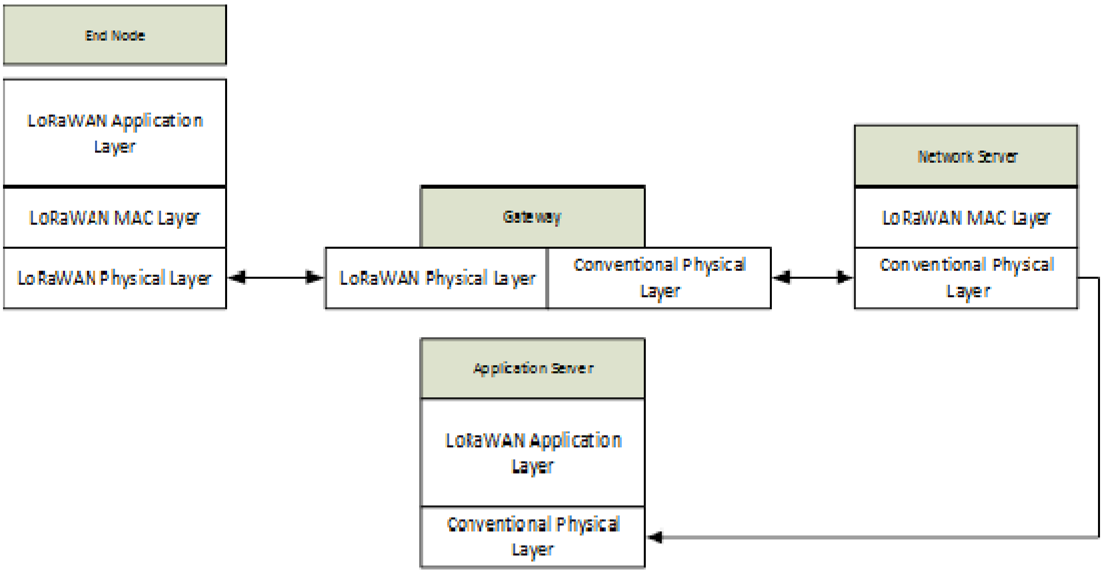

End Node
LoRaWAN Application Layer
LoRaWAN MAC Layer
LoRaWAN Physical Layer
Gateway
LoRaWAN Physical Layer
Conventional Physical Layer
Network Server
LoRaWAN MAC Layer
Conventional Physical Layer
Application Server
LoRaWAN Application Layer
Conventional Physical Layer

Figure 2.6: LoRaWAN Protocol Architecture

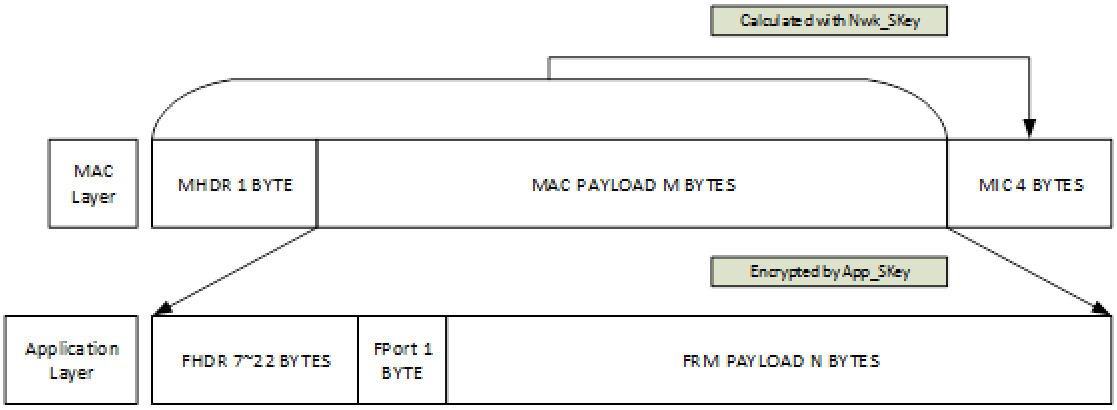


Figure 2.7: LoRaWAN Packet Format

device is generated each time the device joins the network, which allows roaming between networks belonging to different providers. Moreover, having two keys makes tampering with or reading application data harder, even if one of the keys is compromised. The end-device initiates the OTAA procedure by a sending join request message as shown in Figure 2.7 . The message includes the ***AppEUI***, ***DevEUI***, and nonce ***(DevNonce)*** of the end device. The DevNonce is a random generated two byte value which is tracked by the network server and used to reject any join request with an invalid nonce value. This mechanism prevents replay attacks. The ABP joining procedure directly connects end-devices to the specified network without initiating a join-request and accept procedure. The device address ***(DevAddr)***, network session key ***(NwkSKey)***, and application session key ***(AppSKey)*** are directly defined and stored in the end device. Therefore, it does not generate any keys and can directly encrypt messages using these keys. If the keys are compromised, all communication between the device, gateway, and network server can be decrypted by third party entities for the lifetime of the device.

2.7.1 Over-the-Air-Activation

In the OTAA mode [19], an end node communicates with the network server to perform the activation process, which is called join procedure. According to the LoRaWAN specifications [20], the OTAA mode is used when an end node is deployed or reset were the AppKey is used to cryptographically sign the Join Request. All three values (AppEUI, DevEUI and Devnonce) are then made available to the application server to which the device is supposed to connect. OTAA from security side uses the AES cryptographic primitive combined with several modes of operations such as CMAC, CTR, AES-CMAC, and CBC. Confusingly enough, the AppKey is used to generate the session keys, NwkSKey and AppSKey which are used when the node sends a Join Request message. Figure 2.8 shows the LoRaWAN join procedure. A detailed explanation of each step is as follows:

1. Join request message: by sending a join request message, the end node starts the join procedure. ***DevEUI***, ***AppEUI***, and ***DevNonce*** are included in the join request. ***DevEUI*** and ***AppEUI*** refer to the global end node and application identifier, respectively. They follow the IEEE EUI-64 address space format. The ***DevNonce*** is a random number generated by the end node and starting from zero when the device is joined. The MIC

value of join request is calculated by the following formula:

$$cmac = aes128{-}cmac(\boldsymbol{AppKey}, \boldsymbol{MHDR}|\boldsymbol{AppEUI}|\boldsymbol{DevEUI}|\boldsymbol{DevNonce}) \tag{2.2}$$

$$MIC = cmac[0\ldots3] \tag{2.3}$$

where an application key ***(AppKey)*** is pre-shared between the end node and the network server. The “cmac” or calculated MAC is composed of the Application Key, MAC header, Application end-device-identifier, device end-device-identifier and the device plain text counter all encrypted by the aes-128bit.

2. After the network server receives the join request, it performs the replay attack prevention process (described in chapter 3), which is based on the ***DevNonce***. If the ***DevNonce*** in the join request is previously used, the network server determines that the message is invalid and that the join process will fail. If the message is valid, the network server authenticates the end node with the MIC value. If the end node passes the authentication, the network server generates an ***NwkSKey*** and an ***AppSKey*** by the following formula:

$$Nwk{-}Skey = aes128{-}encrypt(\boldsymbol{AppKey}, 0x01|\boldsymbol{AppNonce}|\boldsymbol{NetID}|\boldsymbol{DevNonce}|\boldsymbol{pad16}) \tag{2.4}$$

$$App{-}Skey = aes128{-}encrypt(\boldsymbol{AppKey}, 0x02|\boldsymbol{AppNonce}|\boldsymbol{NetID}|\boldsymbol{DevNonce}|\boldsymbol{pad16}) \tag{2.5}$$

where an ***AppNonce*** is a random number generated by the network server. NetID is a 24-bit field. Its 5 least significant byte or ***LSBs*** are called ***NwkID*** which is used to separate addresses of geographically duplicated LoRa networks. The other bits of ***NetID*** can be freely determined by the network server.

3. Join accept message: a join accept message contains ***AppNonce***, ***NetID***, ***DevAddr***, ***DLSettings***, ***RxDelay***, and ***CFList***. The DevAddr is a 32-bit identifier of the end node within the current network. The 7 most significant byte or ***MSBs*** of DevAddr are referred to as the ***NwkID***, which is also contained in NetID. The other bits can be arbitrarily chosen by the network server. ***DLSettings*** contains several values related to the downlink configuration. ***RxDelay*** is a delay between the transmission and reception process. ***CFList*** is an optional field that is about channel frequencies. Finally, the whole join accept message is encrypted with the ***AppKey***.

4. Transfer AppSKey: since the AppSKey is devised to secure end-to-end communications between the end node and the application server, it should be transferred from the network server to the application server. The LoRaWAN specification does not specify when and how to exchange AppSKey with the application server, since it is an essential part hence included it in the join procedure.

5. After receiving the join accept message, the end node decrypts it and generates session keys using extracted parameters.

2.7.2 Activation by Personalization

Activation by Personalization skips connection and confirmation requests. In this case, before activation, the device is assigned unique parameters (***DevAddr***, ***NwkSKey*** and ***AppSKey)***, which are stored on the end device and the network server. When activated, the end device sends these values to the server directly. At the same time, messages are encrypted and signed with a digital signature. It is assumed that only a pre-configured network server with the appropriate parameters, (Figure 2.9) can process this data as text.

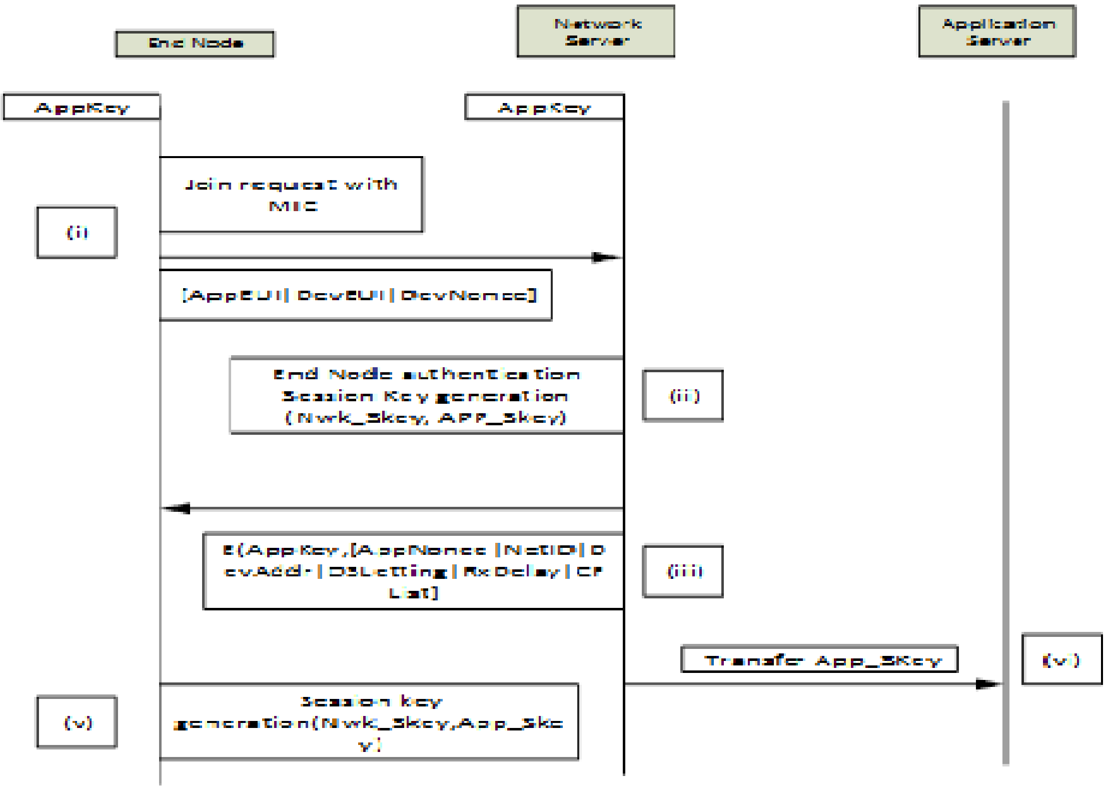

End Node
Network Server
Application Server
AppKey
AppKey
Join request with MIC
(i)
[AppEUI| DevEUI | DevNonce]
End Node authentication Session Key generation (Nwk_Skey, APP_Skey)
(ii)
E(AppKey,[AppNonce |NetID| DevAddr | DLSetting | RxDelay | CF List]
(iii)
Transfer App_SKey
(v)
(v)
Session key generation(Nwk_Skey,App_Skey)

Figure 2.8: OTAA LoRaWAN join procedure [4]

1. *Network Session Key* **(NwkSKey)**: This is a network layer security mechanism. This key is unique to each end device and shared between the end device and the network server. The network session key provides message integrity during communication and security for end device to network server communication.

2. *Application Session Key* **(AppSKey)**: This key is responsible for end-to-end (application to application) ciphering of the payload. This is also an AES 128-bit key, unique to each end device. It is shared between the end device and application server. The application session key encrypts and decrypts application data messages and provides security for application payloads.

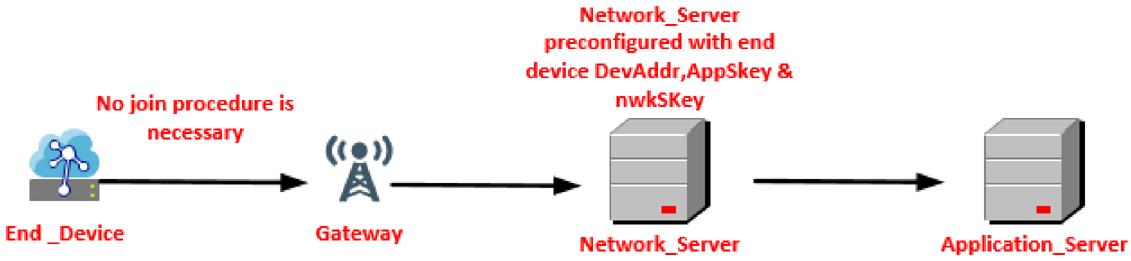


Figure 2.9: ABP procedure [5]

2.8 Protection of data over LoRa

Once a "Node" is joined to a LoRa network, by OTAA, all messages will be encrypted using the ***NwkSKey*** and ***AppSKey***. As these keys are only known by the Network Server and the joined node, hence the possibility of another node to take over the session or a "man in the middle" attack to recover the clear-text data is extremely difficult [21]

2.8.1 Data Encryption

During communications, frame payloads are encrypted first. If a frame payload contains only MAC commands, NwkSKey is used for encryption. Otherwise, AppSKey is used. The encryption process is seen in the following Figure 2.10 .

- Define blocks Ai
- i =1 to k, k=ceil (length (frame-payload)/16).
- Si = AES128-encrypt (K, Ai)
- for I = 1 to k, k=NwkSKey or AppSKey
- Truncate (frame-payload—pad16) XOR S to the first length (frame algorithm-payload) octets.

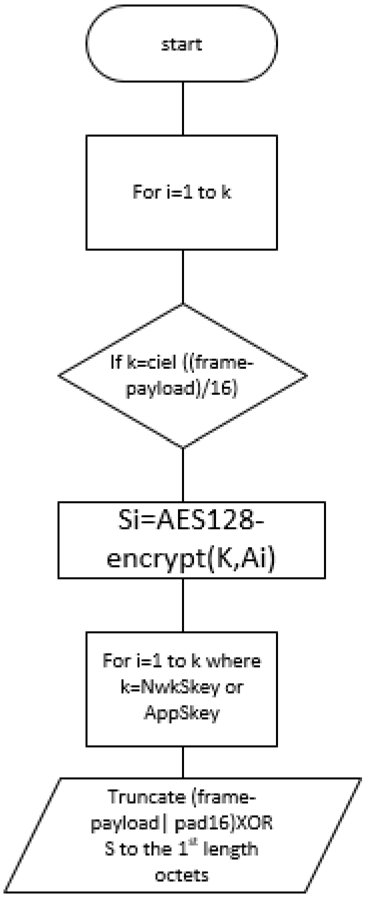

start
For i=1 to k
If k=ciel ((frame-payload)/16)
Si=AES128-encrypt(K,Ai)
For i=1 to k where k=NwkSkey or AppSkey
Truncate (frame-payload| pad16)XOR S to the 1st length octets

Figure 2.10: Frame payload Encryption

Table 2.2: Encryption block of a message in LoRaWAN network

Ai	0*01	4*0*00	Dir	DevAddr	Fcnt	0*00	I
Size(bytes)	1	4	1	4	4	1	1

This encryption method is Advanced Encryption Standard or AES, which is a symmetric encryption algorithm. It supports a block length of 128 bits and key lengths of 128, 192, and 256 bit. The block cipher mode of operation here is very similar to the Counter (CTR) mode. It can be observed that for CTR mode, there is a nonce and a block counter. The Electronic Codebook (ECB) mode is usually used in encryption. Each block is encrypted separately as shown in Figure 2.11 . But the disadvantage of this method is that identical plaintext blocks are encrypted into identical cipher text blocks, thus it doesn't hide the data patterns being transmitted well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all [22] unless a single message block is to be encrypted only once. However, since ECB mode is only used for join request, and the message will never be repeated because of the nonce, it is still secure for LoRaWAN to use ECB mode (where the message is divided into blocks, and each block is encrypted and decrypted separately) [23].

2.8.2 Data Decryption

In "Data Decryption" the method of encryption is switched, starting from the cipher text toward the plaintext in a similar process as "Data Encryption".

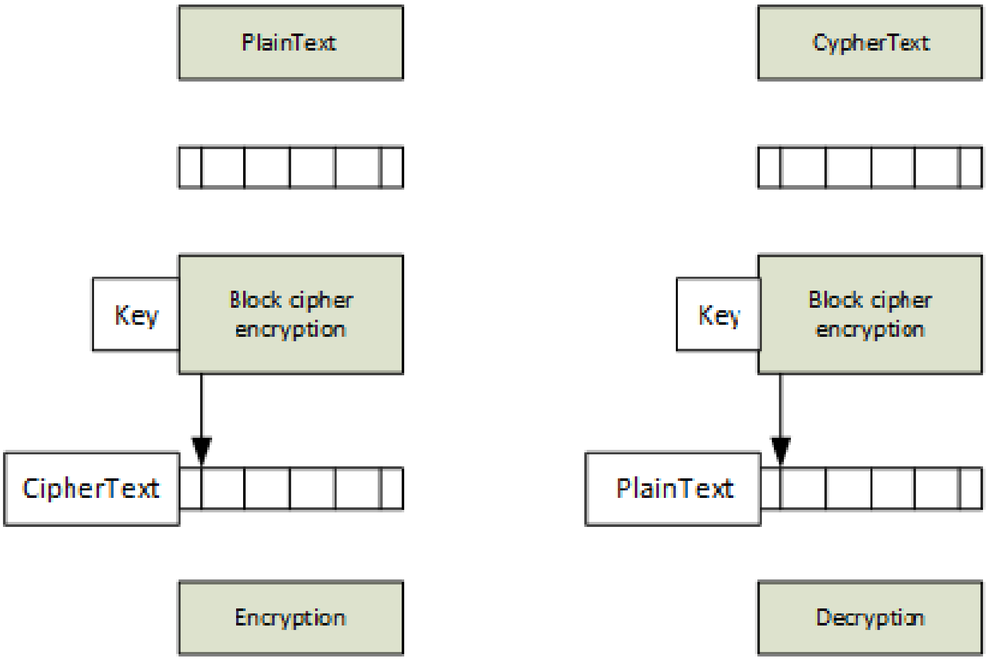


Figure 2.11: ECB cipher block Encryption/Decryption mode

Chapter 3

IoT SECURITY, VULNERABILITIES,ATTACKS AND MITIGATION

The rapid development of the IoT market and the need to secure and protect the traffic from vulnerabilities resulted in lots of testing from different vendors and institutions of which the most important is the "LoRa Alliance Institution". In this chapter we elaborate some attacks mentioned prior in chapter two in a more detailed explanation to pinpoint how these attacks are being generated and applied when attacking IoT networks.

3.1 Analysis Of Vulnerabilities

LoRa devices are susceptible to several security attacks such as ABP attack, compromising device and network keys attack, jamming techniques attacks, replay Attacks, wormhole attacks and side attacks. These attacks are briefly described as following:

- One of the potential vulnerability is using ABP for joining, because deriving its keys from publicly available information such as the end device itself, can be used to launch ABP attacks. Indeed, this could be worked out through reverse engineering of one device, then all other communications to any device would then be compromised. Therefore, we believe a unique set of keys (***NwkSKey*** and ***AppSKey***) must be derived for each device to protect the communication of other devices. In addition, LoRaWAN packet structure does not include any time based data or signature to validate the time of the message, and this might create a vulnerability to perform replay or/and wormhole attacks on LoRaWAN networks.

- Compromising device and network keys attack is the kind of attack where a hacker gains physical access to a device and extracts the keys.
- In Jamming technique attacks, malicious entities can transmit a powerful radio signal in the proximity of application devices to disrupt the radio transmissions. Typically, such attacks require dedicated hardware to counter measure them.
- A replay attack occurs on a security protocol by re-transmitting the valid data by a malicious entity. The main purpose of this attack is fooling the device or module by using handshake messages or old data from the network.
- A wormhole attack could be performed by using two types of devices, sniffer and jammer. The sniffer captures packets and signals to the jammer to notify that it captured the packet. The captured packet never reaches to the gateway and, validation of captured message stays valid.The captured message can be replayed any time. Gateway and Network server forwards to the packet to the application layer.
- A side attack is an attack were an intruder eavesdrops on the device's side channel emissions and take note when an encryption key is used to access

the device. This tiny amount of information can then be used to, in effect, duplicate the key.

3.2 Replay attack for ABP activated nodes

This attack is designed to achieve spoofing and denial of service or ***DoS***. For the server, the attack goal is to achieve spoofing. After the attack, the server will accept a malicious replayed message from the attacker's end device, and it will believe the message is from an accepted working end device. For the victim end device, the attack goal is to achieve ***DoS***. After the attack, the message that the victim end device sends, will not be accepted in the server. The period of ***DoS*** depends on the selection of replayed message as shown in Figure 3.1 in which the end device is sending traffic to the gateway and then to the server. It also shows a malicious device that is sniffing and replicating the traffic in a DoS attack pattern by using it's own mirror bridge.

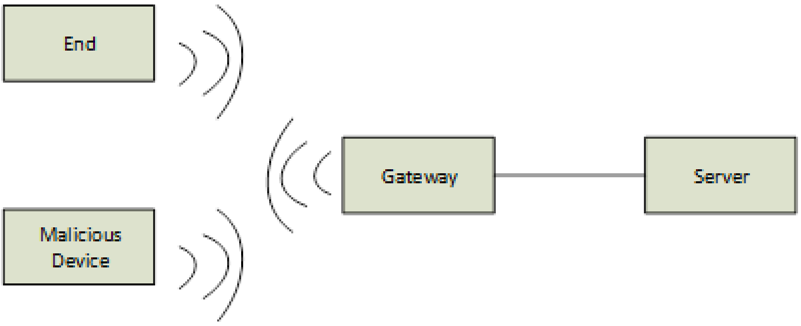


Figure 3.1: LoRaWAN replay attack

In order to achieve this attack, the attacker should be capable of:

- Having knowledge of the physical payload format of LoRaWAN messages.
- Knowing the wireless communication frequency band of the victim end device

- Having a device to capture LoRaWAN wireless messages.

- Having a device to send LoRaWAN messages in a certain frequency.

- Storing and reading plaintext of LoRaWAN messages

If the attacker does not have a specific victim target, in a large LoRaWAN network, the attacker has to wait for an overflow in order to start its attack. However, if the attacker is performing attacks in a relatively small network, the best for him is to be able to reset the victim end device to reduce the waiting time.

3.3 Protocol Vulnerabilities

Many system features constitute vulnerabilities that can be exploited and lead to replay attacks. Indeed, ABP activation method has many security flaws! ABP activated end-devices use static keys, which means after resetting, the keys will stay the same and will not be changed. Also, unlike OTAA activated end devices, there is no join procedure for ABP activated devices. So for a malicious message, it can be accepted by LoRaWAN network server when it meets below requirements:

1. Session keys are the same as one accepted end device

2. ***DevAddr***(Device Address) is the same as one accepted end device

3. Counter value is acceptable

In this case, the attacker can choose and resend the messages before a reset, and the server cannot tell whether these messages are from this session or the session before resetting. Moreover, counters are not used in a secure way. The protocol specification, states that:

"After a JoinReq – JoinAccept message exchange or a reset for a personalized end-device, the frame counters on the end-device and the frame counters on the network server for that end-device are reset to 0." [24]

Therefore, after resetting, the ABP activated end-device will reuse the frame counter value from 0 with the same keys. In this case, the attacker grab messages in the last session with larger counter values and reuse it in the current session.

To operate the attack, these steps should be followed:

1. Capture messages: Use a device to capture uplink messages of an ABP activated node, and save them into the attacker's database

2. Get Frame Counter ***FCnt*** value: Read the uplink counter value from these messages since counter values are not encrypted.

3. Wait till the end device resets or counter overflows

4. Find a suitable message. Select a captured message with suitable counter value from attacker's database.

The criteria to select a suitable message is based on the attacker's goal. Assume the uplink counter value in malicious message is ***Cm***, and the uplink counter value in end device is ***Ce***. The maximum counter gap is ***Gap***.

Let Cm = x and Ce = y

- If (x - y) $\leq$ Gap: Malicious message will be accepted. Messages from end device with the counter value in $[x, y]$ will be ignored.

- If (x - y) > Gap: Malicious message will be ignored.

The most harmful attack is to select the counter value $\mathbf{x = Gap + y}$, since it will take the longest time to wait till it is recovered.

- Replay. Resend the message to the gateway

3.4 Eavesdropping Attack

The attack is designed to compromise the encryption method of LoRaWAN. By sniffing the wireless traffic between the gateway and the end device, the attacker can use the corresponding relationship between 2 messages with the same counter value to decrypt the cipher text. After the attack, the attacker can compromise the confidentiality of the system, and obtain sensor data transmitted in the system. If LoRaWAN is used to transmit secret data, this attack can cause serious privacy issues.

In order to perform the attack, the attacker should have the capabilities of:

- Having a LoRaWAN wireless sniffer device to sniff wireless packets.
- Having basic knowledge of end devices such as message type and message format.
- Having a database to store and compare LoRaWAN traffic.

3.4.1 Protocol Vulnerabilities

The root reason here is similar to the reason in attack 1. There are 2 vulnerabilities in the protocol to achieve this attack: First, ABP activation method has security flaws, and second, counters are not used in a secure way. Another vulnerability is that the cipher block mode is not secure.

LoRaWAN uses "block cipher" mode similar to CTR when it comes to data messages transfer. Instead of using a nonce in the block, a counter value is used. After the resetting, since the key is statistic and the counter value will be reused, the key stream will be the same for messages with same counter value. If we have 2 messages with same key stream, then:

Plaintext1 **XOR** Keystream = Ciphertext1

Plaintext2 **XOR** Keystream = Ciphertext2

Then we have:

Plaintext1 **XOR** Plaintext2 = Ciphertext1 **XOR** Ciphertext2

The attack setup illustrated in Figure 3-2 shows how a malicious gateway and server is built by the attacker to capture wireless packets from the target network.

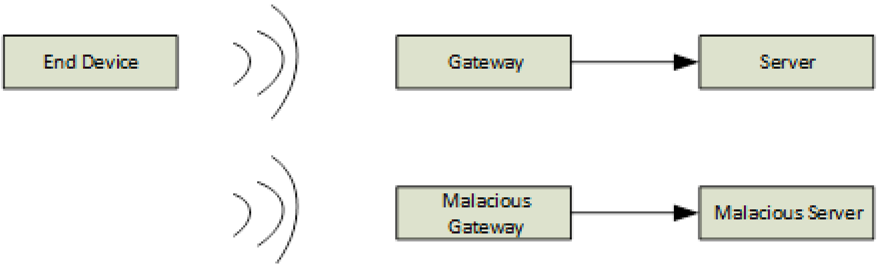


Figure 3.2: Eavesdropping attack

The attack works as follows:

1. The attacker captures and stores LoRaWAN wireless packets, and logs basic information.

2. After resetting, attacker continues to collect packets, compare packets before and after resetting and pair packets with same counter value.

3. Coding with method crib dragging, see the result.

Figure 3-3[16] shows an example of conducting an eavesdropping attack in a LoRaWAN network. A malicious gateway with appropriate frequency can receive messages from end device. Pairing the messages before and after resetting with same counter value, makes it possible to crib ragging to derive the plaintext.

As summarized before in this chapter, there are several varieties of possible attacks in Wireless Sensor Network (WSN) such as below attacks shown in table 3.1:

All these attacks are still not solved to protect the IoT, yet lots of experiments are being done to overcome them in the future.

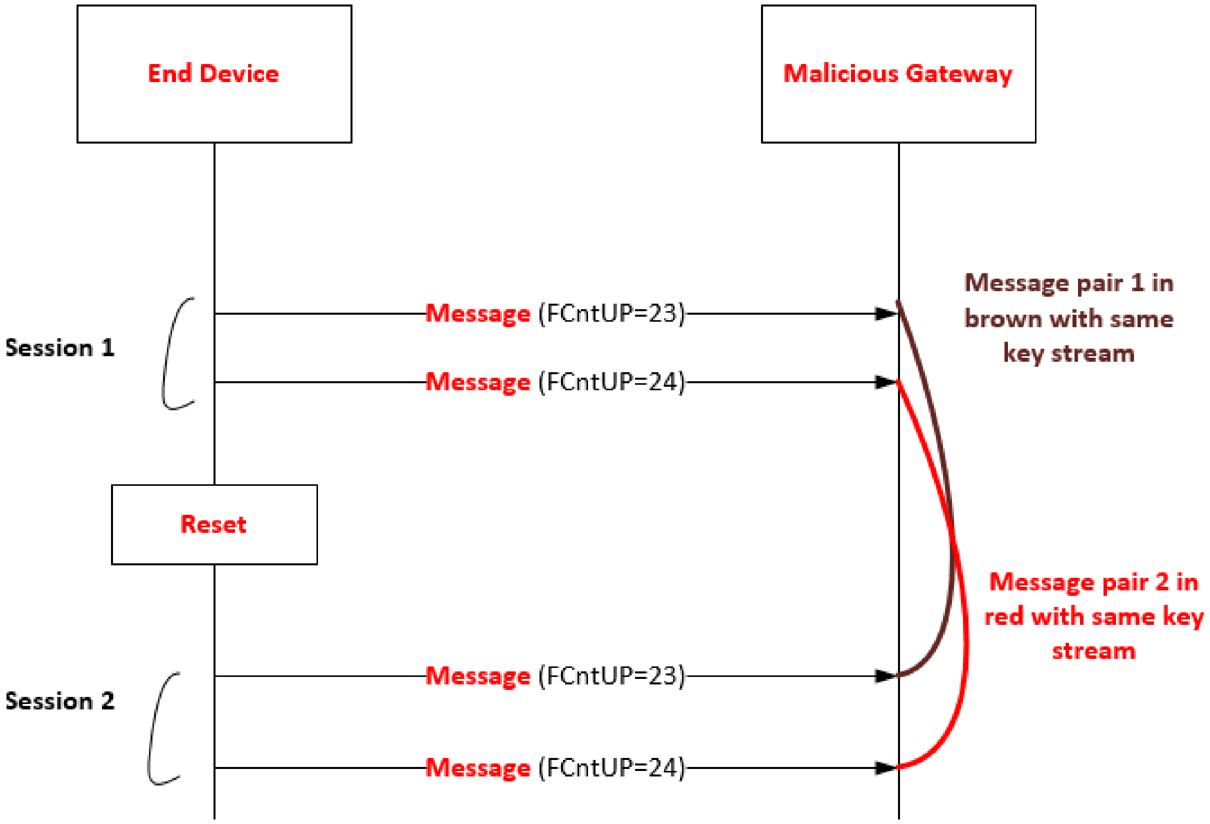


Figure 3.3: Eavesdropping Attack Session

[16]

When using LoRaWAN devices such as ABP, it should be used only in certain circumstances, because during the resetting time, the embedded static keys used are always the same in addition to the counter which will be reset back to zero. In this case, counter values will be reused every time the end device resets. Hence, we can strengthen the block cipher mode that LoRaWAN uses which is similar to AES-CTR. Although AES-CTR mode is secure based on the assumption that the nonce will never be reused. But in LoRaWAN network, counter overflow and counter reset will both cause the counter reuse. One type of encryption which is now being studied to enforce the security for IoT is the ECC or Elliptic Curve Cryptography. The ECC is a light weight encryption type to

Table 3.1: Type of WSN attacks [8]

Physical/Link layers	**Network/transport layers**	**Application layers**	**MultiLayer Attacks**
Jammers	NWK/transport	Injection	Side channel attacks
Relay Attacks	Sinkhole	Buffer overflows	Replay Attacks
Sybil Selective Forwarding	Unfairness False Routing		Traffic Analysis Crypto Attacks
Synchronization Attack	Hello Session Flooding Eavesdropping		

complete the authentication and establish a secure session key between the end device and the application server. Subsequently, over-the-air traffic including the MAC commands and the application payload are the origin authenticated, integrity and replay protected, and encrypted. Furthermore, application payload is end-to-end encrypted between the end-device and the application server. All of these procedures rely on AES-128 cryptographic keys and algorithms. Therefore, coming with a more secure encryption type as add on layer to AES yet light to be embedded in the end devices is not an easy task. Using ECC may be the next-generation approach to cryptography that uses a mathematical formula to enable the use of relatively small cryptographic keys to provide the same or a greater level of security compared to the larger RSA keys. IoT devices usually have small memory and CPU. In addition, it requires minimal time to send the data in order to preserve the power in its batteries. Using large encryption type other than the symmetric methods such AES will drain the batteries' juices, and this is not recommended at all since the devices may be located in areas which are not easy to reach. So using methods which require less computation then asymmetric methods are highly recommended in key exchange between the devices and the servers while preserving the power. ECC and RSA are both asymmetric encryption algorithms. The RSA asymmetric key of 2048-bits is equivalent to a symmetric AES key of 128-bits. The NSA (National Security Agency) [25] usually requires suit B cryptography to be implemented which requires a 3072 bit RSA key, which is equivalent to 256-bit symmetric key encryption, but the computation time for increasingly higher levels of security increases exponentially using RSA. In Table 3.2 [26] the difference between ECC and RSA key equivalencies are shown in details, where we can see the key size differences and equivalencies and which is considered as good encryption suite.

ECC allows devices with limited processing power to achieve a high level of security without sacrificing expensive computing cycles and with minimal effect on application performance.

ECC has three types of cryptography [9]:

1. Elliptic Curve Digital Signature Algorithm (ECDSA) is a digital signature algorithm that is primarily used to authenticate digital content, and identify the author of that content

2. Elliptic Curve Integrated Encryption Scheme (ECIES) is an integrated encryption scheme that provides security against chosen plain text and chosen cipher text attacks

3. Elliptic curve Diffie-Hellman (ECDH) allows two parties, each with public-private key pairs, to share a secret over an insecure channel

As for our required scheme, it will be the 3rd option based on Diffie-Hellman choice to check.

The security strength of ECC as shown in Figure 3.4, depends on the Elliptic Curve Discrete Logarithm Problem (ECDLP) based on scalar multiplication, which includes point doubling and adding operation. From mathematical point of view, the ECC formula is:

Table 3.2: ECC and RSA key equivalencies [9]

Key & Encryption	**Public key length**	**Equivalent symmetric key length**	**Subjective Security**
RSA	1024	80	Not Recommended
RSA	2048	112	Good
RSA	3072	128	Suite B Top Secret
RSA	15360	256	Future
ECC	163	80	Not Recommended
ECC	224	112	Good
ECC	256	128	Great
ECC	384	192	Suite B Top Secret
ECC	521	256	Future

$$y^2 \equiv x^3 + ax + b(modp) \tag{3.1}$$

$$4a^3 + 27b^2 \neq 0(modp) \tag{3.2}$$

$$b \in (Z - p) \tag{3.3}$$

where a and p is a prime number, a & b are non-negative numbers less than p, in addition we need to have a point G or generator of points such that when required to create a key exchange system, an encryption/decryption system requires to have point G in addition to an elliptic group

$$E_p(a, b) \tag{3.4}$$

as parameters, where each user A selects a private key

$$n_A \tag{3.5}$$

and generate a public key:

$$P_A = n_A x G \tag{3.6}$$

Algebraically a curve is non-singular if and only if the discriminant:

$$\triangle = -16(4a^3 + 27b^2) \neq 0 \tag{3.7}$$

and this is usually needed in cryptography.

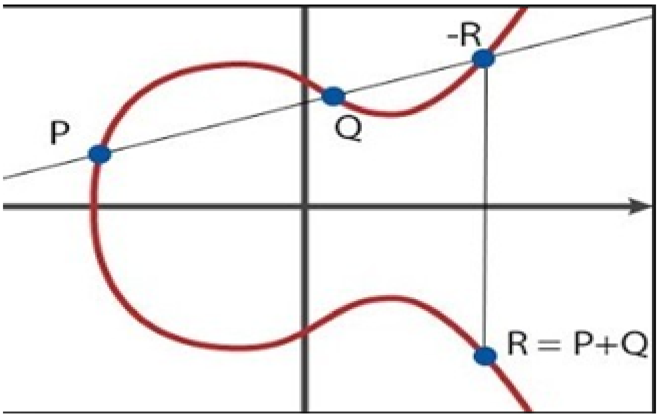


Figure 3.4: The ECC form graph

Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks and requires additional protocols such as the TLS protocol. Indirectly, they can be used for encryption by combining the key agreement with a symmetric encryption scheme if possible. Hence applying the ECDH in an IoT sensor will require a good memory and storage and most of all CPU power, which is difficult up to this time to be implemented in class A and B of the IoT devices, whereas class C can be able to hold ECC cryptography due to its continues supply of power. In other words, handling of asymmetric cryptographic keys such as ECC is very important but these keys are still often very large compared to the AES-128bit symmetric cryptography and run in quadratic time, rather than on substitution-permutation network as in AES-128bits, where its calculation are done in a particular finite field. Thus for the IoT its better to stay under the AES 128bit encryption rather then going to ECC.

3.5 Encryption & Decryption

1. Key Generation: is an important part where two keys has to be generated the public and private key. We select a number ‘d’ within the range of ‘n’, then using the following equation

   $$Q = d * P \tag{3.8}$$

   where d is a random number selected between 1 & n-1, P is a point on the EC, Q is the public key and ‘d’ is the private key

2. Encryption: let ‘m’ be the message to be send, hence we need to represent this message on the curve. We consider ‘m’ has the point ’M’ on the curve ‘E’, randomly select ‘k’ from (1-(n-1)).Two cipher texts will be generated let it be C1 and C2.
   C1 = k*P
   C2 = M + k*Q (C1 and C2 will be sent)

3. Decryption: to get back the message ‘m’, $M = C2 - d * C1$, M can be represented as $C2 - d * C1 \Rightarrow C2 - d * C1 = (M + k * Q) - d * (k * P)$

4. Calculation1: $C2 = M + k * Q$ and
   $C1 = k * P$

5. Calculation2: $M + k * d * P - d * k * P$ (cancelling out k * d * P)

6. Calculation3: M (Original Message is retrieved)

3.6 Limitation of ABP in LoRaWAN

As discussed so far, the IoT devices need to operate robustly and to provide an adequate level of security. The security mechanisms should be constructed to work efficiently on very constrained devices with possibly the highest protection.

The elliptic curves cryptography (ECC)-based solutions are ideal for such scenarios, due to the security equivalence of the Rivest, Shamir and Adleman (RSA) public-key cryptosystem. Public key scheme, but with significantly smaller keys and computational requirements that are related to constraints on the amount of energy available to them as well the storage.These low-energy environments are so constrained that commonly-implemented RSA or ECC and Diffie-Hellman-type public-key cryptography protocols either can't address the power constraints, or their performance is so slow that the devices are unsuitable for real-world activities. Furthermore, limited energy storage on some devices results in a finite amount of runtime available to complete all operations, placing additional constraints on the resources available to run a protocol such as the ECC. When it comes to the limitation of Activation by Personalization (ABP) in LoRaWAN, we see that Key management has been a big problem, where In this procedure we have the ***NwkSKey*** and ***AppSKey*** embedded inside the end devices in addition to the ***DevAddr***, hence once a new end device is installed it will transmit its keys under AES128bit ***CTR*** mode to the [27] network server where the keys and DevAddr are stored as well via a transparent bridge and starts associating with the network server that provides the management of gateways and endpoints, authentication and authorization of endpoints, network encryption and decryption, data routing, adapting data rates, eliminating duplicate packets, and interfacing with applications as shown in Figure 3.5 IoT Topology. Here is where an attacker can built his work on detecting traffics and data flows in order to introduce the attack, either by replay, DDoS or eavesdropping and much more types of attacks that can jeopardize the network and data transmission. A notorious problem in protocol security is the insufficient use of randomness or nonce ("number used once") which is used in the join request. Following the classical definition of security, we care about the confidentiality, integrity, and availability of a system. As the communication channel is wireless and thus available to anyone for injection and modification, also the authenticity of communication – in other words do the packets indeed originate from the alleged source and the protection against originally legitimate but maliciously re-injected traffic become a concern.

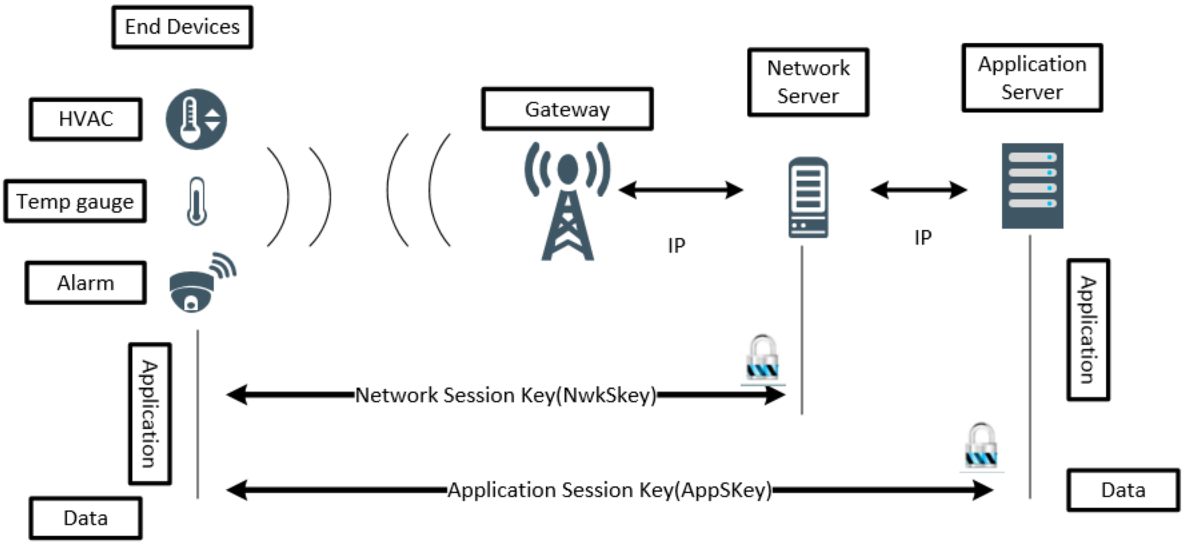
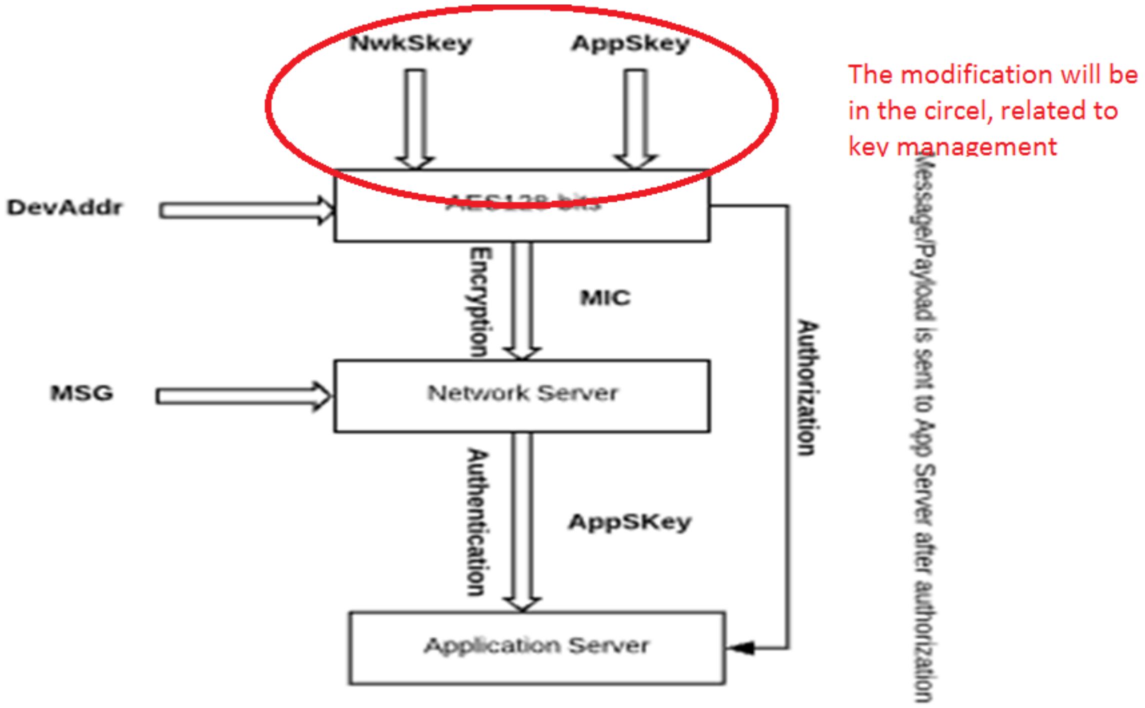


Figure 3.5: IoT Topology

Chapter 4

Proposed Work

In this chapter, we plan to tackle the general key infrastructure of the ABP keys as shown in Figure 4.1 by modifying the circled part and see how feasible it is to be implemented under the IoT ABP procedure:



The modification will be in the circel, related to key management

Figure 4.1: modification section

4.1 LoraWan Security Issues

As stated before, LoraWan is subject to various kind of attacks based on confidentiality, integrity, and availability, and one of the most dangerous attacks is the "replay attack" [28] part from man in the middle in the ABP joining procedure which is an attack on security protocol or integrity by re-sending or repeating the valid data transmission by the malicious entity thus fooling the device or module by using handshake messages or old data from the network. The problem with replay attack is that devices process it as legitimate messages although encrypted. A replay attack can gain access to the network from the end device either by gaining physical access to it or by rebooting the device and sniffing the traffic then proceeds through the gateway to the network server, also the attacker may gain access to the device if he or she were able to connect to the network infrastructure the end device belongs to.

In order to perform the attack in wireless networks, the entity should know the communication frequencies and channels to sniff data from transmission between devices. In LoRaWAN, it is not possible to decrypt transmissions between end-devices and gateways without ***AppSKey*** [29], since the entire payload of the LoRaWAN message is encrypted by it. Additionally, since tampering with the data will make the MIC check fail, it is not possible to do it without ***NwkSKey***. Although the malicious entity can resend the message consecutively, using frame counters which are defined in LoRaWAN specifications these messages or attacks can be detected and discarded. Once the end-device is activated, these counters are both set to 0 and each message coming from the gateway or the device increments counters. If a message is received with a lower frame counter than the last message, it is ignored. However, the LoRaWAN specification handling off frame counters is specifically left to the application and developer. Therefore, networks which do not track these frame counters could be vulnerable to replay attacks.

Counters are an important component in replay protection and, as the mes-

sage counter is used in LoRaWAN to generate the key stream, are also essential to the confidentiality of the communication channel. If the counter overflows, it will be started from 0 again. Also to the LoRaWAN specification, the counter value will be set to zero after resetting.
Therefore, an ABP-activated end device will reuse the frame counter value from 0 with the same keys. In this case, an attacker can grab messages in the last session with larger counter values and reuse it in the current session. Besides resetting, another method to restart the counter is a counter overflow. After the counter value reaches its maximum value, the counter will be reset and will restart from 0. With counter values from the last session and the same session keys, an attacker can also replay previous messages to cut off the communication between the end device and the server. A message replay [30] is trivial to implement for an adversary. First, monitor and store the uplink messages of an ABP activated node. Second, wait until the device has reset the counter value Frame Count ***(FCnt)***, which is sent in clear text. Assume the uplink counter value of the malicious message is $\boldsymbol{FCntm}$, the uplink counter value of the end device is $\boldsymbol{FCntcurr}$, and the maximum accepted counter gap is Gap. Third, replay any message with $FCntm - Fcntcurr = Gap$ to fit the running window algorithm of LoRaWAN and thus be accepted by the network if replayed. The most harmful attack is to select the counter value $FCntm = Gap + FCntcurr$ since it will take the devices the longest time to recover, Figure 4.2 and Figure 4.3.

In Figure 4.2 we show the original ABP messages between the end devices and the gateways and the injected malicious message is the message in the last session with the same device address, session keys, and larger counter value. As long as the attacker sends this message in this session to the network server, and it is accepted, the messages from the victim with a counter value smaller than 70 will be ignored, whereas in Figure 4.3 we see how the attack scenario is implemented in order to achieve the replay attack and collect the traffic signatures. For the attack, minimal hardware is required: a traffic sniffer as well as a LoRa transmitter to

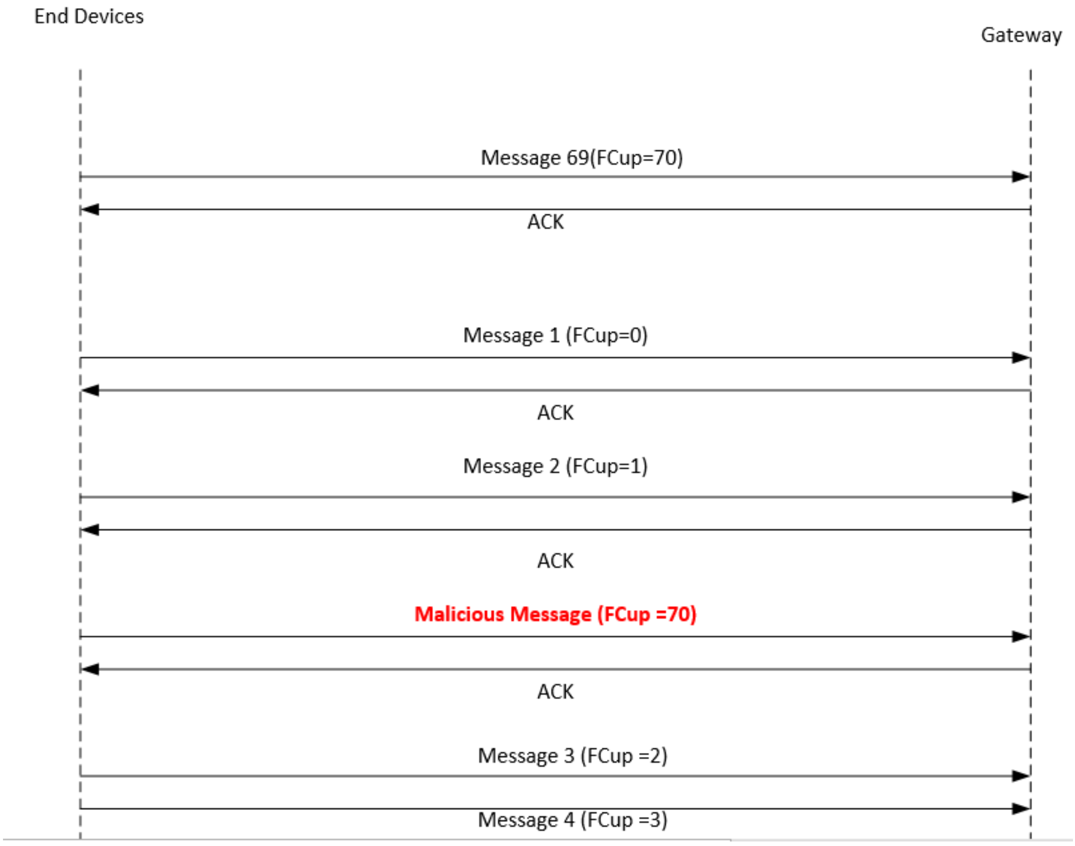


Figure 4.2: ABP Original Message

replay messages. While in a small LoRaWAN with only a few end devices, the attacker may need to wait a long time for a counter overflow, the attack can be efficiently conducted for ABP-activated end devices in a large deployment. Once the attacker gets the largest possible counter value for one end device, it can periodically replay the message and block the end device permanently (or until the session keys of the end device are changed, which requires for ABP a separate channel or physical access). This replay vector thus implements a denial-of-service attack on the availability of an LPWAN deployment.

As a heartbeat to demonstrate the success of the attack and validate that the DoS outage matches the predicted value, scientists installed a sensor to report a field measurement every thirty seconds via LoRaWAN to a back-end server. The

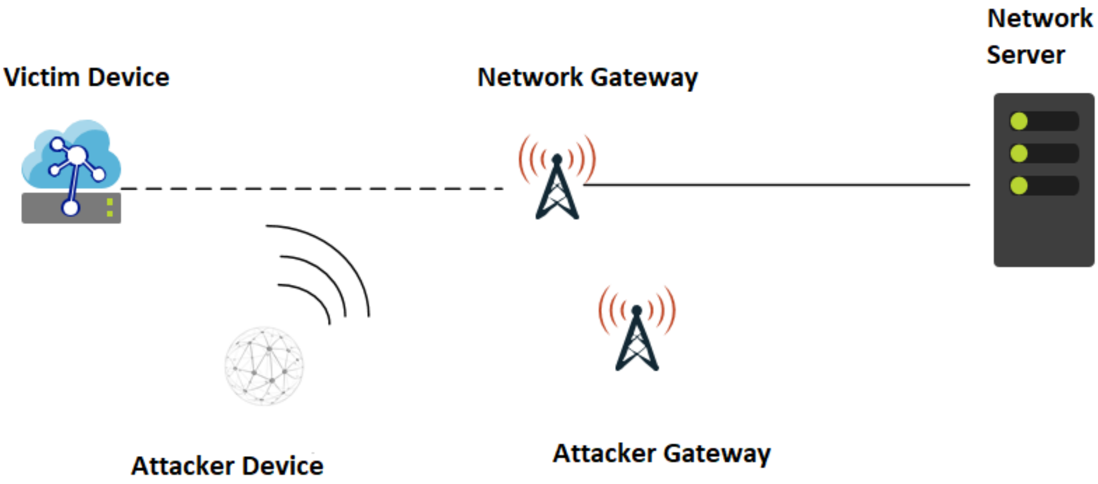
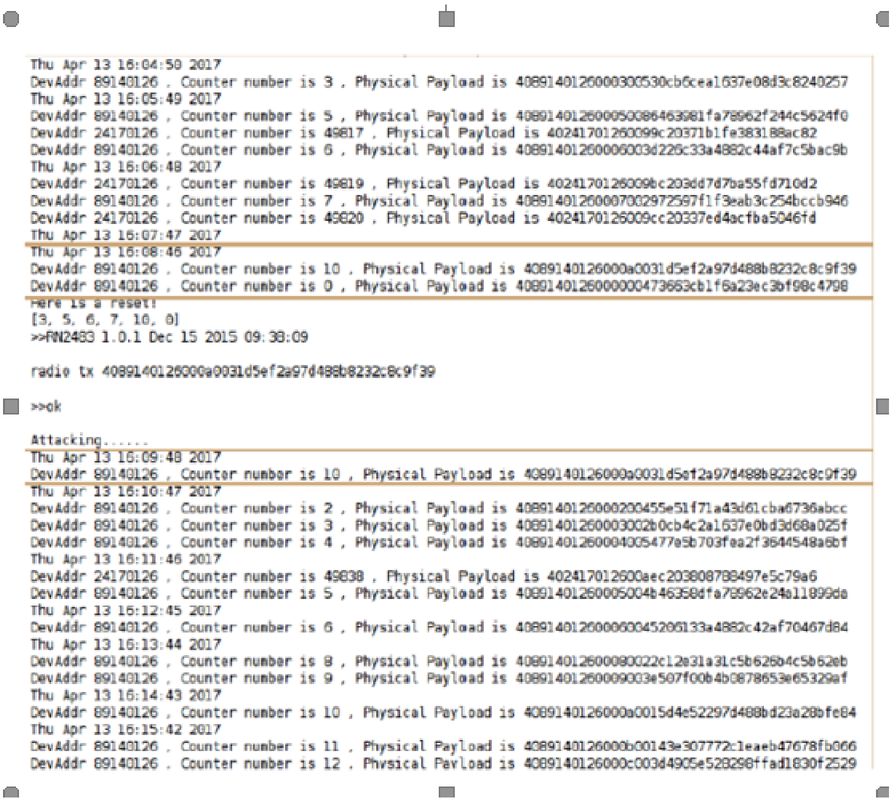


Figure 4.3: Showing how the victim device is attacked [6]

malicious gateway would monitor all frequencies in use by LoRa, and complete a dictionary. In the top highlighted area of the gateway trace shown in Figure 4.4, the attacker notices a device reset and simply re-injects a previously saved message (bottom highlighted area), in this case with the counter value 10. As the subsequent frames sent by the legitimate sensor are out of sequence, the sensor will need to increment and transmit until back in sync. As devices obey a specific low-volume duty cycle, the sensor is effectively blocked during this time. The reporting backend of the LoRa application service shown in Figure 4.7 confirms the replay and an outage for 5.5 minutes. Note that the denial-of-service was accomplished during the entire time by means of a single packet, in contrast to other DoS attacks such as SYN floods, this attack thus leaves no abnormal adversarial traffic such as flooding which is detectable by the network.



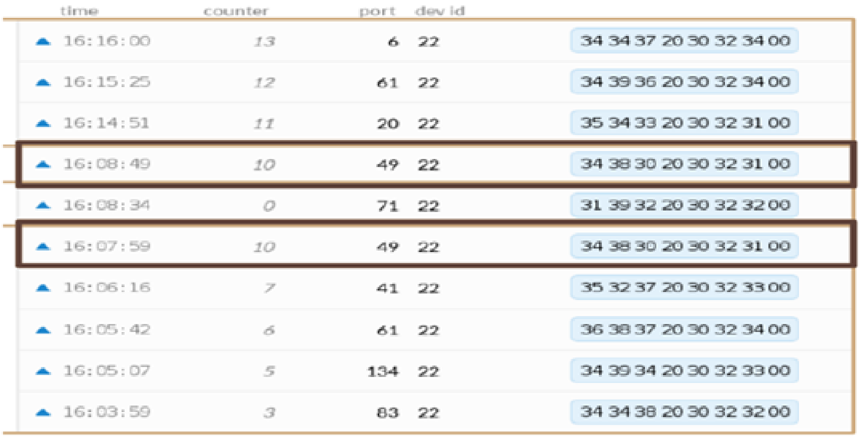
```
Thu Apr 13 16:04:50 2017
DevAddr 89140126 , Counter number is 3 , Physical Payload is 4089140126000300530cb6cea1637e08d3c8240257
Thu Apr 13 16:05:49 2017
DevAddr 89140126 , Counter number is 5 , Physical Payload is 40891401260005008646398 1fa78962f244c5624f0
DevAddr 24170126 , Counter number is 49817 , Physical Payload is 40241701260099c20371b1fe383188ac82
DevAddr 89140126 , Counter number is 6 , Physical Payload is 4089140126000600 3d226c33a4882c44af7c5bac9b
Thu Apr 13 16:06:48 2017
DevAddr 24170126 , Counter number is 49819 , Physical Payload is 4024170126009bc203dd7d7ba55fd710d2
DevAddr 89140126 , Counter number is 7 , Physical Payload is 408914012600070029725 97f1f3eab3c254bccb946
DevAddr 24170126 , Counter number is 49820 , Physical Payload is 4024170126009cc20337ed4acfba5046fd
Thu Apr 13 16:07:47 2017
Thu Apr 13 16:08:46 2017
DevAddr 89140126 , Counter number is 10 , Physical Payload is 4089140126000a0031d5ef2a97d488b8232c8c9f39
DevAddr 89140126 , Counter number is 0 , Physical Payload is 40891401260000000473663cb1f6a23ec3bf98c4798
Here is a reset!
[3, 5, 6, 7, 10, 0]
>>RN2483 1.0.1 Dec 15 2015 09:38:09

radio tx 4089140126000a0031d5ef2a97d488b8232c8c9f39

>>ok

Attacking......
Thu Apr 13 16:09:48 2017
DevAddr 89140126 , Counter number is 10 , Physical Payload is 4089140126000a0031d5ef2a97d488b8232c8c9f39
Thu Apr 13 16:10:47 2017
DevAddr 89140126 , Counter number is 2 , Physical Payload is 408914012600020 0455e51f71a43d61cba6736abcc
DevAddr 89140126 , Counter number is 3 , Physical Payload is 40891401260003002b0cb4c2a1637e0bd3d68a025f
DevAddr 89140126 , Counter number is 4 , Physical Payload is 40891401260004005477e5b703fea2f3644548a6bf
Thu Apr 13 16:11:46 2017
DevAddr 24170126 , Counter number is 49838 , Physical Payload is 402417012600aec203808788497e5c79a6
DevAddr 89140126 , Counter number is 5 , Physical Payload is 40891401260005004b46358dfa78962e24a11899da
Thu Apr 13 16:12:45 2017
DevAddr 89140126 , Counter number is 6 , Physical Payload is 40891401260006004520 6133a4882c42af70467d84
Thu Apr 13 16:13:44 2017
DevAddr 89140126 , Counter number is 8 , Physical Payload is 40891401260008 0022c12e31a31c5b626b4c5b62eb
DevAddr 89140126 , Counter number is 9 , Physical Payload is 40891401260009003e507f00b4b0878653e65329af
Thu Apr 13 16:14:43 2017
DevAddr 89140126 , Counter number is 10 , Physical Payload is 4089140126000a0015d4e52297d488bd23a28bfe84
Thu Apr 13 16:15:42 2017
DevAddr 89140126 , Counter number is 11 , Physical Payload is 4089140126000b00143e307772c1eaeb47678fb066
DevAddr 89140126 , Counter number is 12 , Physical Payload is 4089140126000c003d4905e528298ffad1830f2529
```

Figure 4.4: Log file of malicious gateway [6]



time	counter	port	dev id	
16:16:00	13	6	22	34 34 37 20 30 32 34 00
16:15:25	12	61	22	34 39 36 20 30 32 34 00
16:14:51	11	20	22	35 34 33 20 30 32 31 00
16:08:49	10	49	22	34 38 30 20 30 32 31 00
16:08:34	0	71	22	31 39 32 20 30 32 32 00
16:07:59	10	49	22	34 38 30 20 30 32 31 00
16:06:16	7	41	22	35 32 37 20 30 32 33 00
16:05:42	6	61	22	36 38 37 20 30 32 34 00
16:05:07	5	134	22	34 39 34 20 30 32 33 00
16:03:59	3	83	22	34 34 38 20 30 32 32 00

Figure 4.5: Log file of the victim's server [6]

4.2 Proposed Mitigation Technique to Prevent Replay Attack

In order to defend against replay attacks, some simple countermeasures exist such as the use of time-stamps, onetime passwords, and challenge-response cryptography. Nevertheless, these schemes are inconvenient and with doubtful efficiency considering the vulnerabilities to which challenge-response protocols are susceptible to. Another approach is the use of RF shielding on readers in order to limit the directionality of radio signals and subsequently the appearance of a ghost, whereas another approach is based on the distance between the information requestor and the information owner. Implied that the signal-to-noise ratio of the end device system can reveal even roughly the distance between an end device and a gateway. This information could definitely be used in order to make discrimination between authorized and unauthorized devices and subsequently mitigate replay attacks. Thus we state below some important types of countermeasures being worked on:

- Attach a sequence number to each message used in an authentication exchange. A new message is only accepted if its sequence number is in the proper order

- Team(A) accepts a message as fresh only if the message contains time-stamp that, in A's judgment, is close enough to A's knowledge of the current time, this requires to have the clock connected to a unified **NTP** (network time protocol) server

- Team(A) expecting a fresh message from team(B), first sends team(B) a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value

- Sequencing of messages and non-acceptance of duplicated messages

- Creating random session keys which are time and process bound

- One-time password

- Adding Additional Byte or more with Light cryptographic algorithm depending on the length of the packet in which it can withhold the traffic

- Tag the frame or packet with a special key generation that can be only decrypted by the Network server, once done, the ED join request message is accepted and receives a join accept message. This tag may be based on both time-stamp and additional byte value such as point 5 or point 6

The LoRaWAN specification explicitly warns developers about generating secure network and application keys, because compromising the keys of one device should not compromise the security of the communications of other devices. [30]

The process to build those keys should be such that the keys cannot be derived in any way from publicly available information (like the Node address for example)

The characteristic of the proposed countermeasure are listed in the following and shown in Figure 4.6:

- **Lightweight scheme:** which uses at least a minimum number of iterations and applied only during the reset operation. Our aim is to have it light in comparison to the memory and power consumption since the sensor is Class A, where the power should last at least 10 years. The algorithm should be flexible round function without using any diffusion operation, in which it reduces the computational complexity of the proposed cipher and consequently the required latency and resources. Moreover, the proposed encryption scheme can be realized in parallel, while the decryption algorithm can be partially parallelized. [27]

- **Simple hardware and software implementations:** The proposed key derivation function can use any secure hash function, which renders the corresponding hardware and software implementations of the proposed key derivation scheme to be simple and efficient. [31]

- **Dynamic Key Approach:** In contrast to the existing cipher solutions, the proposed approach is based on a dynamic key, which is variable and changes in a pseudo-random manner after each new reset operation. In addition, changing the dynamic key produces different cipher text and MIC(s). The dynamic nature of the proposed cipher provides high robustness against any kind of attacks [32]

The produced dynamic key depend on the stored secret key ***(SK)*** that embedded in the end device and a reset counter number.

Equation 4.1 represents the proposed dynamic key derivation scheme. In fact, the proposed technique concatenates the network (NK) and application (AK) keys with the number of counter sessions CS. Then, each corresponding output is hashed to produce a new confidentiality secure key counter(SKC_{CS}) and secure key integrity(SKI_{CS}) session keys for each new session as described in the following equation:

$$SKI_{CS} = h(NK||CS), \qquad SKC_{CS} = h(AK||CS) \tag{4.1}$$

where h represents any secure cryptographic hash function such as SHA-512.

In fact, we propose to only update network and application keys for class A and B end devices. On the other hand, we propose also a new enhancement to reach a high level of security for LORA end devices of class C. This solution propose to use a new dynamic confidentiality and integrity keys for each new input message. These dynamic keys are generated by using the frame counter up and frame counter down as described in the following equation:

$$DKI_i = h(SKI_{CS}||FCup||FCdwn), \tag{4.2}$$

$$DKC_i = h(SKC_{CS}||FCup||FCdwn) \tag{4.3}$$

This means that the i^{th} message is encrypted by using DKC_i and authenticated by using DKI_i. The proposed scheme is based on the dynamic key approach will make LORA communication (between end devices and network/application

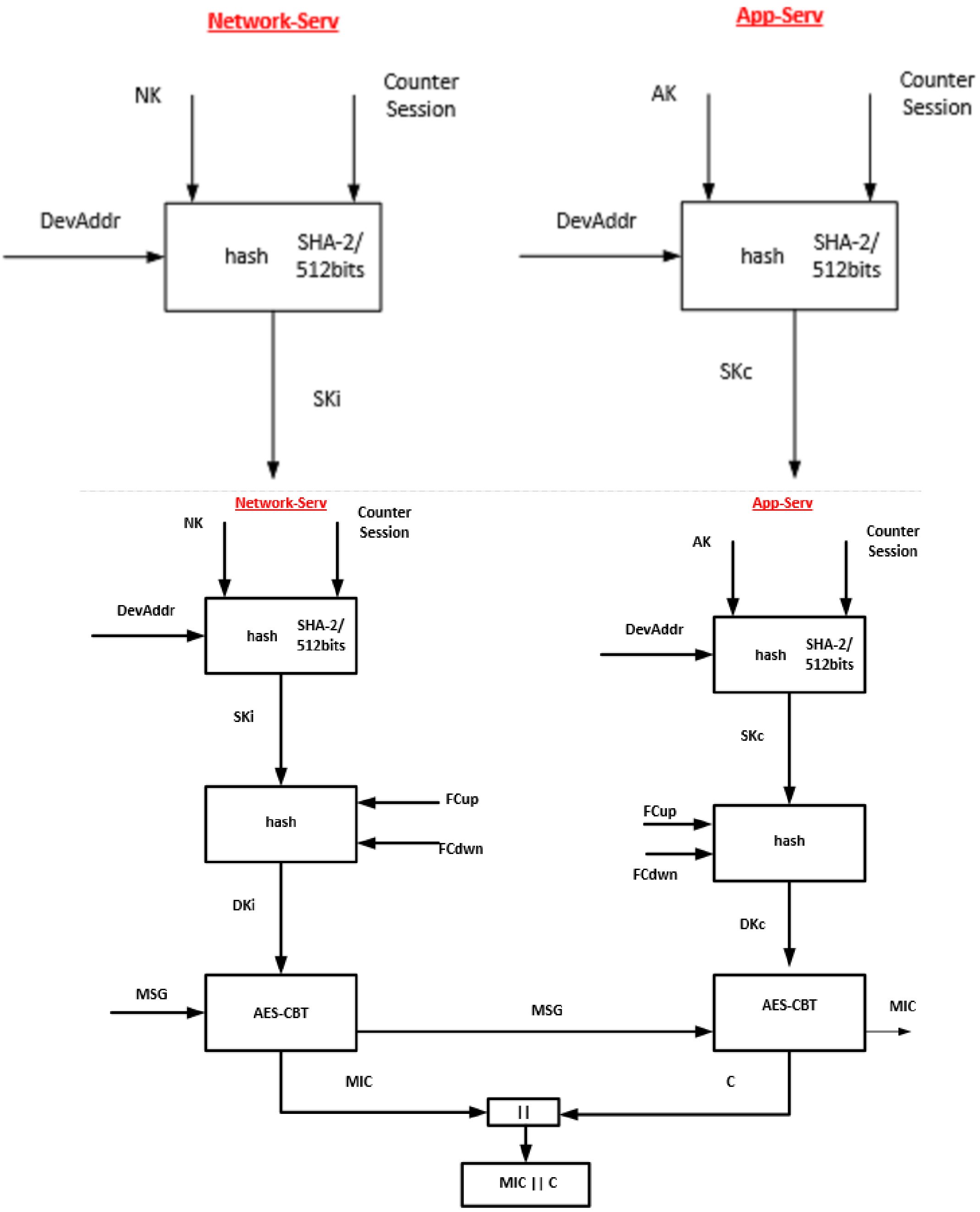


Figure 4.6: The proposed Dynamic Key Derivation Scheme for LORA end devices for class A and B (upper image) and for Class C respectively

servers) more secure.Two figures show the normal ABP joining traffic and the latter traffic after modification, Figure 4.7 and Figure 4.8. In Figure 4.7, in order for the ABP to join the network server, all the required is the NwkSKey,AppSKey and DevAddr since they are already embedded in the end device and allocated at the server end. In Figure 4.8, the modification is done for the join by replacing the NwkSkey, AppSkey and DevAddr by the DKC_i from equation 4.2 and the DKI_i from equation 4.3 .

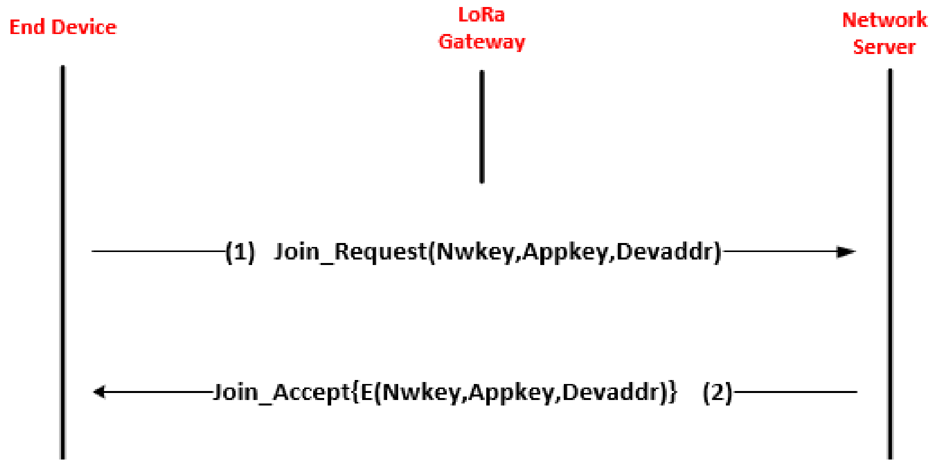

End Device
LoRa Gateway
Network Server
(1) Join_Request(Nwkey,Appkey,Devaddr)
Join_Accept{E(Nwkey,Appkey,Devaddr)} (2)

Figure 4.7: ABP Normal

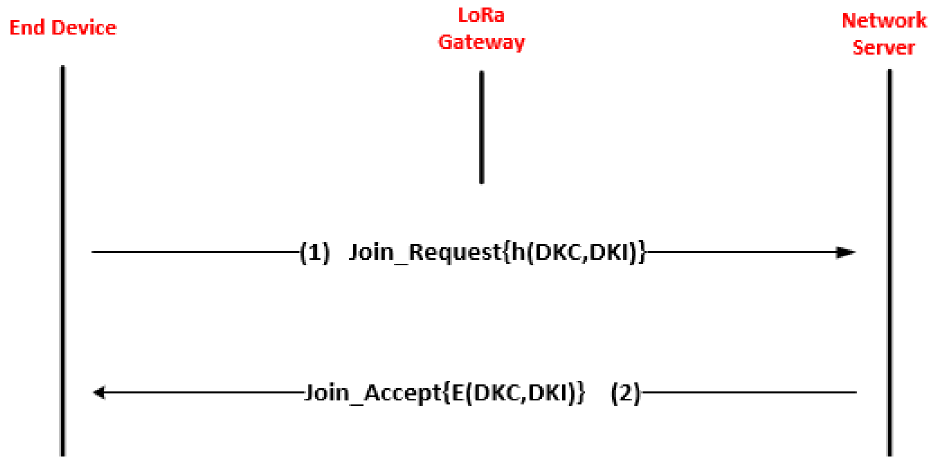


Figure 4.8: ABP Modified

Table 4.1: Table of Notations

Symbol	**Definition**
CS	The number of session
NK	Network Key
AK	Application Key
SCK_{CS}	The CS^{th} shared secret Confidentiality Session Key
FCup	Frame Counter UP
FCdwn	Frame Counter Down
DKC_i	The i^{th} dynamic confidentiality Key and it is used to encrypt the i^{th} input message during the CS^{th} session.
DKI_i	The i^{th} dynamic integrity key and it is used to the i^{th} input message during the CS^{th} session.
N M	Number of bytes in one block message The original message
C	The encrypted message
MIC	Message Integrity Code

4.3 Security Analysis

In this section, the security level of the proposed key derivation scheme is analyzed and assessed. More specifically, the produced update network and applications keys should reach a high level of randomness or degree of randomness, uniformity, and sensitivity. A randomness and uniformity tests in addition to sensitivity tests are applied to quantify the security level of the proposed solution.

4.3.1 Randomness of the Produced Dynamic Keys

The security level of the proposed update key solution depends on the robustness of the employed secure cryptographic hash function. Therefore, the randomness degree of the proposed update key scheme is tested by applying 100 iterations to have a size of 1 million bits, here we start to have sets of dynamic keys such as DK1, DK2, DK$_n$.

As an additional information, randomness tests in data evaluation, are used to analyze the distribution of a set of data to see if it can be described as random (patternless). In cryptography, the quality of the random numbers used directly determines the security strength of the system. The strength of the random number generator used by the security systems often determines how secure the systems are depending on the actual randomness of the bits generated [33].

For this purpose, the empirical NIST statistical test [34] is applied to 100 sequences of one million bits, produced with 100 different secret keys, hash and Nonces to validate the security of the proposed dynamic-key derivation scheme. In Figure 4.9, the obtained NIST proportion values and their corresponding P-values are shown. The obtained p-value (> 0.01) (P-value = 0.741948), which indicates that the null hypothesis is not rejected and the produced sequences reach a high level of randomness. As it can be inferred, the plotted proportion values (marked in blue) are above the threshold represented by the red line, which proves that the proposed dynamic key generation scheme passes all the statistical tests and a high randomness level is reached

In order to have a clearer view of the results, we suggest defining population areas in the [0,1) range where the p-values are distributed. We suggest dividing this range into three types of areas; Safe Area, Doubt Area, and Failure Area

These areas can be defined by the following limits: $0 < p - value \leq 0.1$ or $0.9 \leq p - value \leq 1$ fall in the Failure Area. $0.1 < p - value \leq 0.25$ or $0.75 \leq p - value < 0.9$ fall in the Doubt Area. $0.25 < p - value < 0.75$ which falls in the Safe Area. [35] Having more p-values in the Safe Area indicates that the tested sample is closer to randomness. On the other hand, having to many p-values in the Failure Area is an indicator of deviating from randomness.

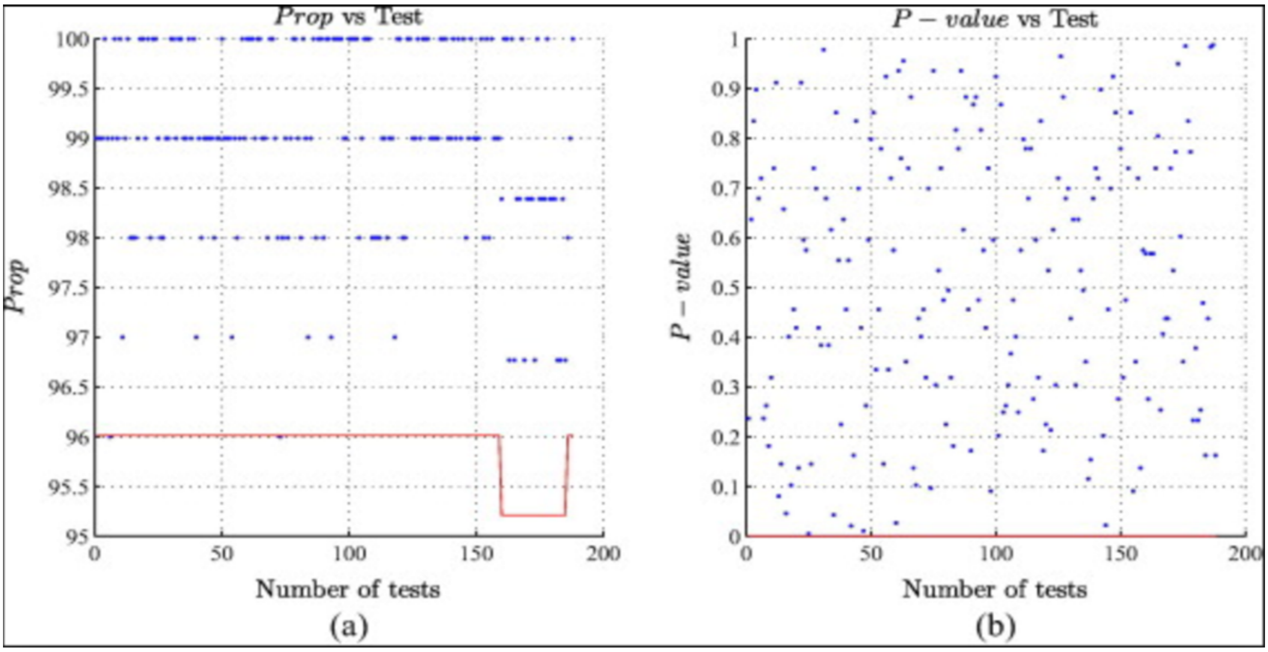


Figure 4.9: NIST test results: Proportion values

4.4 Key Sensitivity

The key sensitivity test evaluates the bit difference between original and update keys (slight change in the counter). This difference between both at the bit level should be close to 50%. Indeed, the sensitivity of the secret key SK is calculated as follows:

$$KS = \frac{\sum_{k=1}^{T} dec2bin(E_{SK}(P)) \oplus dec2bin(E_{SK'}(P))}{T} \tag{4.4}$$

where all the elements of CS are equal to those of CS', except for the Least Significant Bit (LSB) of a random byte, and T is the length of the secret keys (in bits).

We apply this test for 1000 different secret keys (SK_w and SK'_w) and the Hamming distance between original and update keys are computed at the bit level. In addition, the corresponding results are presented in Figure 4.10, where the $0 \leq$ KS ≤ 100 represents key avalanche effect toward the 50%.

This result prove that for all secret keys, the proposed update key scheme reaches a high level of key sensitivity since the obtained results are very close to the desired value, which is 50%. Therefore, the proposed update scheme confirms its security since the required sensitivity is attained.

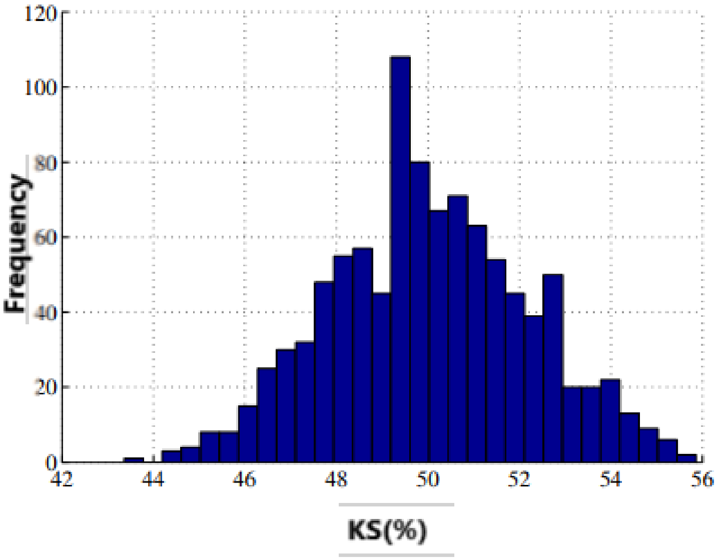


Figure 4.10: Histogram of the Key sensitivity results

4.5 Cryptanalysis

The proposed solution introduces the dynamicity by updating the network and application keys for each new reset. In addition, the produced network and server keys are produced by using a secure hash function such as SHA-512. This will prevent an attacker to recover the static network and application keys from the produced dynamic ones since it is a non-invertible transformation. In addition, a secure hash function ensures a high collision and any slight modification in any input can lead to produce different network and application keys. Moreover, the size of the counter session can be equal to 512 bits if SHA-512 is used.

4.6 Performance Analysis: Delay Overhead

The overhead delay is defined as the sum of processing overhead in each update network and application process.

1. C_{Conc} is the overhead of one concatenation operation,

2. C_{hash} is the overhead of one Hash operation,

The overhead delay of the proposed update mechanism for class A and B end devices is:

$$C_{prop(A,B)} = 2 \times C_{hash} + 2 \times C_{Conc} \tag{4.5}$$

The overhead delay depends of the number of packets transmitted between two reset as shown in Figure 4.11. Low overhead delay is introduced for a big number of packets transmitted between two reset.

Here, it is clear that the required delay overhead in this solution is low since the proposed scheme require a low number of operations.

While the required overhead delay for class C end devices is:

$$\begin{aligned} C_{prop} &= 2 \times C_{hash} + 2 \times C_{Conc} + 2 \times r \times C_{hash} + 2 && (4.6)\\ &\quad \times r \times C_{Conc} && (4.7)\\ &= C_{prop(A,B)} + 2 \times r \times C_{hash} + 2 \times r\ C_{Conc} && (4.8)\end{aligned}$$

Where r represents number of packets transmitted between two reset operations, where it can be seen that by increasing number of packets during the reset session, we see that sessions are persistent at low percentage. Here, it is clear that the proposed solution for class C is higher compared to class A and B since they require two number of hash and concatenation operations for each input packet. In terms of energy, end devices of class C are not limited, while in terms of delay, it depends of application. Let us indicate that this cost can be acceptable if a high level of security is necessary.

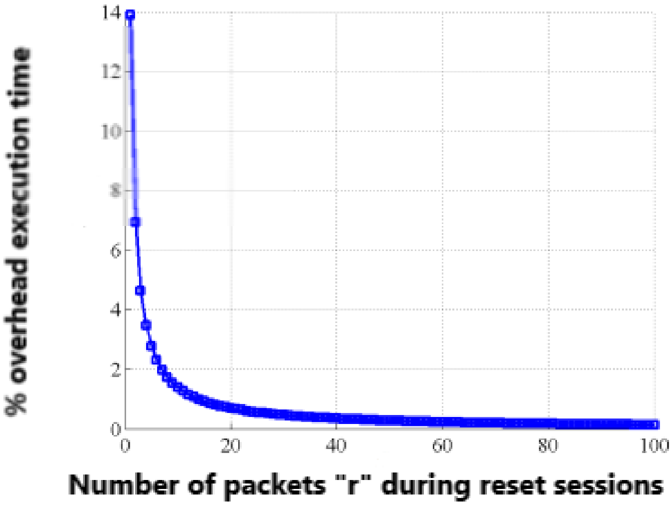


Figure 4.11: Variation of the overhead Delay in function of the number of packets between two reset period.

Chapter 5

Evaluation between AES-128bits & ECC 160-bits

In order to proceed with the evaluation of the security and what is best approach that we can use to overcome vulnerability breaches or sinkholes that may affect our equipment, we have to take in consideration the available technologies in cryptography and security hardening as well the hardware ability to have the codes implemented inside without causing high computation or power consumption. As for the ECC, from mathematical point of view, it's considered as a difficult title, but implementing it in IoT devices has to be very specific and precise based on the requirements.The ECC was supposed to be tested if it fits in sensors of type A and B, but after the testing, ECC showed that it requires more storage then the AES as well more computational power thus resulting in drainage of the existing sensor power, since the ECC uses a quadratic equation, whereas the AES uses finite field.

In the test lab we used two types of Arduino kits, one is Uno and the other is Mega 2560.We included the features of each Arduino used for more information:

Arduino Uno ATmega328:

- Microcontroller ATmega328

- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB of which 0.5 KB used by bootloader
- SRAM 2 KB EEPROM 1 KB
- Clock Speed 16 MHz

Arduino Mega 2560:

- Microcontroller ATmega25608
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 54 (of which 14 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA

- Flash Memory 256 KB of which 8 KB used by bootloader
- SRAM 8 KB EEPROM 4 KB
- Clock Speed 16 MHz

The two types of boards will be resembling an IoT devices with different memory types, one has 32KB memory and the other has 256KB memory.

Arduino Mega 2560[**36**]: The ATmega2560 (Figure 5.1) has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

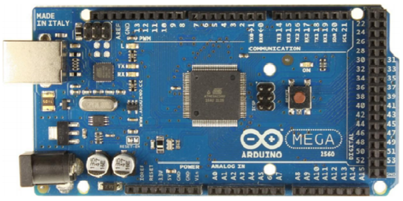


Figure 5.1: Arduino Mega2560

The Arduino Mega 2560 contains SRAM or Static Random Access Memory, which can be read and written from executing program. This is where temporary variables are stored. SRAM memory is used for several purposes by a running program:

- Static Data - This is a block of reserved space in SRAM for all the global and static variables from program. For variables with initial values, the runtime system copies the initial value from Flash when the program starts.
- Heap - The heap is for dynamically allocated data items. The heap grows from the top of the static data area up as data items are allocated
- Stack - The stack is for local variables and for maintaining a record of interrupts and function calls. The stack grows from the top of memory down

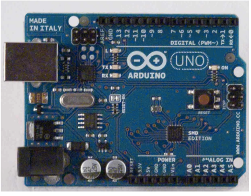


Figure 5.2: Arduino Uno 328

towards the heap. Every interrupt, function call and/or local variable allocation causes the stack to grow. Returning from an interrupt or function call will reclaim all stack space used by that interrupt or function. Most memory problems occur when the stack and the heap collide. When this happens, one or both of these memory areas will be corrupted with unpredictable results. In some cases it 26 will cause an immediate crash. In others, the effects of the corruption may not be noticed until much later.

Arduino Uno328 [36]: The ATmega328 (Figure 5.2) has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader), 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM Library).

5.1 Symmetric Cryptography Assessment

Back to the lab, experiments has been conducted on both the Arduino Uno and the Meg2560, and based on the hardware structure, the results of time testing as well storage changed. We started testing the time required by each code and started by testing AES-CBC encryption/decryption of a plaintext using key length of 128 bits.By testing 1 block 16 bytes code we saw the variation in time and storage(Table 5.1)

We then did a test on ECC by using TinyECC algorithm on both arduinos the Uno and the Mega2560 as well and got the following results (Table 5.2)

Table 5.1: AES-128bit Arduino Uno Vs Mega2560

Key AES128	**Uno**	**Mega2560**
Testing mode(encrypt-decrypt)	372us	376us

Table 5.2: TinyECC-160-bits Arduino Uno Vs Mega2560

Key ECC160	**Uno**	**Mega2560**
Testing mode(encrypt-decrypt)	1188us	1090us

5.1.1 AES-128bits Sketch Results

For the AES-128-bits sketches we see the following regarding the storage percentages:

Arduino Mega2560: Sketch uses 8822 bytes (3%) of program storage space. Maximum is 253952 bytes.Global variables use 718 bytes (8%) of dynamic memory, leaving 7474 bytes for local variables. Maximum is 8192 bytes.

Arduino Uno: Sketch uses 8634 bytes (28%) of program storage space. Maximum is 30720 bytes.Global variables use 718 bytes (35%) of dynamic memory, leaving 1330 bytes for local variables. Maximum is 2048 bytes.

5.1.2 ECC-160bits Sketch Results

As for the ECC160-bits sketches we see the following regarding the storage:

Arduino Mega2560: Sketch uses 9888 bytes (3%) of program storage space. Maximum is 253952 bytes. Global variables use 582 bytes (7%) of dynamic memory, leaving 7610 bytes for local variables. Maximum is 8192 bytes.

Arduino Uno: Sketch uses 9820 bytes (31%) of program storage space. Maximum is 30720 bytes. Global variables use 582 bytes (28%) of dynamic memory, leaving 1466 bytes for local variables. Maximum is 2048 bytes

From the lab experiments performed, we observed that the word length and architectural features are the causes of variations. From these findings and the experimental data, we can conclude that using ECC for small sensors will create a heavy burden on the devices from power and computation perspective.

Chapter 6

Conclusion

The work on this thesis is still rudimentary. The appearance of the LORA technologies in the last few years as an efficient candidate for IoT applications. However, LORA technology suffers from many vulnerabilities, threats, which lead to the presence of authentication and availability attacks, in addition, recent privacy issues. This thesis focus to provide a countermeasure against replay attack for configured LORA end devices with ABP activation mode. The proposed solution is based on a new dynamic key derivation approach that updates the network and application keys after each reset operation. Moreover, we propose to use dynamic confidentiality and authentication keys for each message in a session instead of session ones to reinforce the message confidentiality and authentication level. This solution is suitable for end devices of class C that can be employed in critical infrastructure. The proposed solution is designed to reach a good balance between security and system performance. A set of security and performance tests were presented to validate the efficiency and robustness of the proposed solution. The first priority for future work is evaluating other encryption algorithms in details in order to achieve a better performance for the Light weight Cryptography by enhancing the key management exchange in IoT especially in the ABP joining procedure whether ECC is used, or alternate solution is accepted internationally taking in consideration the requirement for low power and processing demand for longer battery life.

Appendix A

Abbreviations

LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
IoT	Internet of Things
Baron	Branch-And-Reduce Optimization Navigator
NB-IoT	Narrow Band IoT
Wi-Sun	Wireless Smart Ubiquitous Network
AES	Advanced Encryption Standard
RFID	Radio Frequency Identification
GPS	Global Positioning System
ABP	Activation By Personalization
OTAA	Over The Air Authentication
M2M	Machine to Machine
CSS	Chirp Spread Spectrum
BW	Bandwidth
SF	Spreading Factor
MAC	Media Access Control
ISM	Industrial Scientific and Medical
ALOHA	Additive Link On-line Hawaii Area

MHDR	MAC Header
MIC	Message Integrity Check
NWKey	Network Key
AppKey	Application Key
NWKSKey	Network Secure Key
APPSKey	Application Secure Key
DevEUI	Device Unique Identifier
APPEUI	Application Unique Identifier
DevNonce	Device Random Value
ECB	Electronic Code Book
FCnt	Frame Count
ECC	Elliptic Curve Cryptography
RSA	Rivest Shamir Adleman

Bibliography

[1] "A study of lora: Long range & low power networks for the internet of things." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038744/. (Accessed on 11/18/2019).

[2] "Working on wireless iot apps? lora might come handy. — inxee systems private limited." http://inxee.com/blog/working-on-wireless-iot-apps-lora-might-come-handy/. (Accessed on 10/18/2019).

[3] "Different lorawan classes. — download scientific diagram." https://www.researchgate.net/figure/Different-LoRaWAN-classes$_fig18_326134076.(Accessedon10/18/2019)$.

[4] "A dual key-based activation scheme for secure lorawan." https://www.hindawi.com/journals/wcmc/2017/6590713/. (Accessed on 08/03/2019).

[5] "Otaa-or-abp.key." https://static1.squarespace.com/static/-/560cc2c2e4b01e842d9fac18/t/5a938d38ec212d9451fbecf8/1519619387035/OTAA-or-ABPv3.pdf. (Accessed on 10/21/2019).

[6] "1fca8a62b83c4d99244eebc7ae7a4d22e336.pdf." https://pdfs.semanticscholar.org/893e/1fca8a62b83c4d99244eebc7ae7a4d22e336.pdf. (Accessed on 08/08/2019).

[7] B. Usmonov, O. Evsutin, A. Iskhakov, A. Shelupanov, A. Iskhakova, and R. Meshcheryakov, "The cybersecurity in development of iot embedded technologies," in *2017 International Conference on Information Science and Communications Technologies (ICISCT)*, pp. 1–4, IEEE, 2017.

[8] O. El Mouaatamid, M. Lahmer, and M. Belkasmi, "Internet of things security: Layered classification of attacks and possible countermeasures," *Electronic Journal of Information Technology*, no. 9, 2016.

[9] "Elliptic curve cryptography comes of age: Small keys unlock a big future - gemalto blog." https://blog.gemalto.com/security/2015/01/27/elliptic-curve-cryptography-comes-of-age-small-keys-unlock-a-big-future/. (Accessed on 12/10/2018).

[10] B. Henderson *et al.*, *Rethinking the Internet of Things: a scalable approach to connecting everything*. Apress, 2014.

[11] "Nsa suite b cryptography - wikipedia." https://en.wikipedia.org/wiki/NSA$_{S}uite_{BC}ryptography.(Accessedon08/03/2019)$.

[12] "mwri." https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf. (Accessed on 08/07/2019).

[13] "A study of lora: Long range & low power networks for the internet of things." https://www.ncbi.nlm.nih.gov/pmc/articles/PMc5038744/. (Accessed on 11/21/2019).

[14] "(14) (pdf) lorawan - a low power wan protocol for internet of things: a review and opportunities." https://www.researchgate.net/publication/318866065/LoRaWAN/-/A Low Power WAN Protocol for Internet of Things a Review and Opportunities. (Accessed on 10/21/2019).

[15] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *2016 23rd International Conference on Telecommunications (ICT)*, pp. 1–6, IEEE, 2016.

[16] M. Vanhoef and F. Piessens, "Advanced wi-fi attacks using commodity hardware," in *Proceedings of the 30th Annual Computer Security Applications Conference*, pp. 256–265, ACM, 2014.

[17] "lorawan image - google search." https://www.google.com.lb/search?q=lorawan+image. (Accessed on 08/03/2019).

[18] L. Alliance^{TM}, "Home page — lora alliance^{TM}." https://lora-alliance.org/. (Accessed on 08/03/2019).

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

[20] L. A. T. Committee *et al.*, "Lorawan 1.1 specification," 2017.

[21] "Analysis of vulnerabilities in lorawan by example of the lorawan", may, 2017, russia - google search." (Accessed on 12/05/2018).

[22] "Iot security - cryptography - embedded.com." https://www.embedded.com/iot-security-cryptography/. (Accessed on 10/21/2019).

[23] J. Katz, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography.* CRC press, 1996.

[24] "Notes on lorawan security - orne brocaar - medium." https://medium.com/@brocaar/notes-on-lorawan-security-7e741a8ee4fa. (Accessed on 08/04/2019).

[25] "Nsa suite b cryptography - wikipedia." https://en.wikipedia.org/wiki/NSA-Suite-B-Cryptography. (Accessed on 08/08/2019).

[26] "Elliptic curve cryptography comes of age: Small keys unlock a big future - gemalto blog." https://blog.gemalto.com/security/2015/01/27/elliptic-curve-cryptography-comes-of-age-small-keys-unlock-a-big-future/. (Accessed on 08/08/2019).

[27] R. Alvarez, C. Caballero-Gil, J. Santonja, and A. Zamora, "Algorithms for lightweight key exchange," *Sensors*, vol. 17, no. 7, p. 1517, 2017.

[28] X. Yang, *LoRaWAN: Vulnerability Analysis and Practical Exploitation.* PhD thesis, PhD thesis. 2017., 2017.

[29] R. Miller, "Lora security: Building a secure lora solution," *MWR Labs Whitepaper*, 2016.

[30] E. Sisinni, D. F. Carvalho, P. Ferrari, A. Flammini, D. R. C. Silva, and I. M. Da Silva, "Enhanced flexible lorawan node for industrial iot," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–4, IEEE, 2018.

[31] H. Noura, A. Chehab, L. Sleem, M. Noura, R. Couturier, and M. M. Mansour, "One round cipher algorithm for multimedia iot devices," *Multimedia Tools and Applications*, pp. 1–31, 2018.

[32] S. Naoui, M. E. Elhdhili, and L. A. Saidane, "Enhancing the security of the iot lorawan architecture," in *Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), International Conference on*, pp. 1–7, IEEE, 2016.

[33] "(1) why is randomness important in cryptography? - quora." https://www.quora.com/Why-is-randomness-important-in-cryptography. (Accessed on 12/10/2019).

[34] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., Booz-Allen and Hamilton Inc Mclean Va, 2001.

[35] "(pdf) testing randomness in ciphertext of block-ciphers using diehard tests." https://www.researchgate.net/publication/268414157-Testing-Randomness-in-Ciphertext-of-Block-Ciphers-Using-DieHard-Tests. (Accessed on 12/02/2018).

[36] "Arduino - home." https://www.arduino.cc/. (Accessed on 12/18/2018).