LoRaWAN: Vulnerability Analysis and Practical Exploitation

Xueying Yang

July 21, 2017

LoRaWAN: Vulnerability Analysis and Practical Exploitation

by

Xueying Yang

in partial fulfillment of the requirements for the degree of

Master of Science

in **Electrical Engineering**

at Delft University of Technology,

to be defended publicly on Friday July 28, 2017 at 13:30 PM

Student number:	4476220	
Project duration:	Oct 24, 2016 – July 28, 2017	
Thesis committee:	Dr. ir. Fernando A. Kuipers,	TU Delft
	Dr. Christian Doerr,	TU Delft
	Ir. Evgenios Karampatzakis,	Brightsight B.V.

Acknowledgements

I would like to express my thanks for the continuous guidance and valuable suggestions from Dr. ir. Fernando A Kuipers. He has always been supportive and patient. Without him this thesis would not have occurred.

I want to express my gratefulness to my friendly and helpful colleagues from Brightsight. I want to thank Evgenios Karampatzakis in particular. He has been given me suggestions, resources and knowledge, and encouraged me to achieve my goals. My gratefulness is extended to Meng, Yuzhu and Fabian. We shared the intern's room and spent good time together.

Many thanks to all the friends I made in the Netherlands. These 2 years in TU Delft is amazing, and I will always cherish the memories here.

My last, and most heartfelt words are to my parents and my boyfriend, for their support, and for providing me the means to follow my dreams.

Xueying Yang
Delft, July 2017

Abstract

Internet of Things (IoT) applications nowadays have a wide impact on people's daily life while the size of IoT has been increasing rapidly. Millions of devices huge amount of data and different kinds of new protocols can bring many security issues.

LoRaWAN is a MAC layer protocol for long-range low-power communication dedicated to the IoT. It can be used to transmit messages between IoT end devices and gateways. However, since the development of LoRaWAN is still at an early stage, the security level of the protocol is not well developed, and the need for analyzing and developing the security level of LoRaWAN is necessary and urgent.

This research summarizes the secure features of LoRaWAN in the aspects of activation methods, key management, cryptography, counter management and message acknowledgement. Then, vulnerabilities of LoRaWAN are found and analyzed. 4 Attacks based on these vulnerabilities are designed and described via an attack tree method. These attacks are (1) replay attack, (2) eavesdropping, (3) bit flipping and (4) ACK spoofing. As a poof-of-concept, the attacks are implemented and executed in a LoRaWAN environment. Afterwards, mitigation and secure solutions against attacks are given to protect the security of LoRaWAN networks.

The result of this research can be used in developing the security level of LoRaWAN protocol and setting the standard criteria for evaluating security of LoRaWAN devices.

CONTENTS

List of Figures

LIST OF TABLES

1

Introduction

1.1. Motivation

The word “Internet of Things” has been gaining popularity in recent years. The basic idea behind Internet of Things is connecting things, such as mobile devices and sensors, through a unique addressing scheme to reach a common goal [6]. IoT applications nowadays have a wide impact on people’s daily life, including not only indoor applications such as smart household objects, but also long-range applications like smart grid, street lights and connected smart cars [31]. It is predicted that there will be 13.5 billion connected objects in use by 2020 [13].

As IoT covers a wide range of application products, the number of protocols that are used in IoT also keeps on increasing. Vendors can select protocols based on different capabilities and features. In the meantime, security is also an important feature to be taken into consideration.

With the development of IoT, there emerge some high level needs for IoT product security. One of the important needs is to protect privacy [28]. This is because IoT devices can generate large amounts of data, which contain private information. Another important need is to safe guard IoT products from being used in DDoS attacks or as launching points in the network [1]. IoT devices are attractive to attackers because so many of these devices are shipped with insecure defaults or insecure, remotely exploitable code [1].

However, to fulfill these needs in IoT security, many challenges need to be overcome. For example, there is a lack of defined standards for secure IoT development. Also, there is no accepted reference architecture among vendors. Moreover, IoT products and services need cooperation of many technologies and protocols, making security of IoT even harder to be guaranteed [22]. Other challenges include limited security planning in development methodologies, IoT product deployment in insecure or physically exposed environments, and resource constraints in embedded systems which may limit security options [31].

Among IoT technologies, Low-Power-Wide-Area (LPWA) technologies are designed to connect IoT devices with low power requirements, long range and low cost. The Long-Range Wide-Area Network (LoRaWAN) is a new MAC layer protocol in the family of Low-Power Wide-Area Networks (LPWAN). It is based on LoRa technology, which is a wireless modulation for low-power low-data-rate and long-range applications [4]. LoRaWAN is designed for wireless battery-operated network, and it fills the gap between the short-coverage low-power-consumption network and the long-coverage high-power-consumption network [2] [11].

The first version of LoRaWAN specification was released in Jan. 2015 by the LoRa Alliance. So far LoRaWAN specification version 1.0.2 has been released. Although the development of LoRaWAN networks is still in an early stage, it begins to be rolled out in multiple countries. For example, the KPN LoRa network is available throughout the Netherlands, making the Netherlands the first country in the world to have a nationwide LoRa network for Internet of Things (IoT) applications [14]. LoRaWAN is also planned to be used for a number of purposes like railway level crossings, burglar alarms and monitoring Industrial Control Systems (ICS) [27]. In this case, the potential market of LoRaWAN is huge.

As LoRaWAN protocol is relatively new, the security level of it is not well-developed yet, and the need for analyzing and developing the security level of LoRaWAN is necessary and urgent.

Research has been done for LoRa technology in different perspectives, but mostly in a performance perspective, e.g. see [7]. Up till now, there is still no study to assess LoRaWAN vulnerabilities in a systematic way. Despite the fact that LoRa technology provides security mechanisms, such as encryption and signature, its security level is not yet as developed as common systems. We aim to fill this gap, and provide vulnerability analysis, possible attacks as well as security solutions for LoRaWAN protocol.

1.2. Thesis Objective

The objective of this research is to answer the following research questions:

- How secure is LoRaWAN?
  - What is the research status of LoRaWAN security?

  - What is the security architecture of LoRaWAN?
  - What are the known security issues in LoRaWAN?
- What are vulnerabilities of LoRaWAN?
  - What are the security requirements of LoRaWAN?
  - Are there any design flaws in LoRaWAN?
- What kinds of attacks are possible on LoRaWAN?
  - What is the possible attack scenario?
  - Is it possible to implement such an attack?
- How to produce secure solutions for LoRaWAN? What features of LoRaWAN can be reviewed/improved to offer acceptable security?

1.3. State of the Art

Research has been done towards security of IoT. [28] introduces privacy and security issues in IoT applications, architectures, etc. [1] classifies threat types and analyzes and characterizes intrusions and attacks in IoT. [13] introduces some use-case scenarios and compares safety, security, and privacy among these cases. [31] compares LoRaWAN, sigfox and Symphony for their different IoT architecture and security requirements. It also suggests solutions from existing technologies as a starting point for establishing a standardized security paradigm in IoTs.

Some studies have investigated different perspectives for LoRaWAN. [34] analyzes the transmission delay and power consumption of LoRaWAN in the procedure of over-the-air activation. [36] introduces the indoor and outdoor performance evaluation via a use case, and indicates LoRa technology is reliable for low cost applications especially in remote regions. [25] studies the capacity and scalability of LoRaWAN, and shows that LoRa can be effectively utilized for the moderately dense networks of very low traffic devices. [22] compares the wireless technologies of LoRaWAN, Sigfox and OnRamp Wireless, and discusses the how the technology works in different scenarios and whether the technology is suitable for low-throughout networks. [7] proposes a range-scale performance evaluation of LoRaWAN. In particular, performance of LoRaWAN is modelled and estimated in the aspects of packet payloads, radio-signal quality, and spatiotemporal.

While IoT security is studied and addressed, vulnerabilities in LoRaWAN need to be found. There is also some research about LoRaWAN vulnerabilities. [27] gives a briefly overview of LoRaWAN security and provides security guidance in LoRaWAN implementation. However, some of the flaws and attacks mentioned in this white paper are only for particular systems, and are not realistic or possible for other solutions. [37] analyzes security threats in LoRaWAN by focusing on the generation of DevNonce, which is a random value in join request. It studies the randomness of DevNonce and provides alternatives for generation methods. [24] focuses on solving the privacy problem that multiple users may have, when they attempt to access the same application server. It compares encryption algorithms and modes based on time to compute and resources used. [26] compares existing key management protocols for IoT, and proposes to use one to enhance the security mechanism of LoRaWAN. Though some research has been done to study vulnerabilities in LoRaWAN, it is still not sufficient. This thesis aims to fill in this gap and provide an in-depth security evaluation of LoRaWAN.

1.4. THESIS OUTLINE

This thesis answers the research questions in section 1.2 in a systematic manner. This work is categorized into six chapters: In chapter 2, key security features of LoRaWAN are summarized. Chapter 3 introduces the vulnerability analysis method for LoRaWAN. 6 steps are described, and steps are used in chapter 4 to excute found attacks. Chapter 4 introduces 4 possible attacks against LoRaWAN. The attack goal, attacker's capability and attack scenarios are given. Moreover, experiments that prove the findings are implemented and analyzed. Chapter 5 suggests solutions against the attacks. In chapter 6, a final conclusion and future work recommendations are provided.

2

Relevant concepts

2.1. Concepts in LoRaWAN

2.1.1. LoRa technology

LoRa is a modulation scheme that is similar to Chirp Spread Spectrum modulation (CSS) [9]. It is a proprietary spread spectrum modulation method. LoRa is a physical layer technology, and it has the features of variable data rate, scalable bandwidth, high robustness and orthogonal spreading factors. With LoRa technology, the transceiver can achieve long transmission range and low power consumption.

2.1.2. LoRaWAN architecture

LoRaWAN has a star-of-stars topology. Gateways relay messages between network server and end-devices. Between End-devices and gateways, LoRa or FSK modulation is used. Between gateways and server, standard IP connection is used. The LoRaWAN architecture is shown in figure 2.1.

2.1.3. LoRaWAN operation modes

There are 3 operation modes in LoRaWAN [4]:

Class A: End-devices will open 2 downlink receive windows after their uplink message transmission. Downlink messages can be sent and received during these 2 windows. This is the most power-efficient mode.

Class B: Besides 2 receive windows, there will be extra scheduled receive slots. The

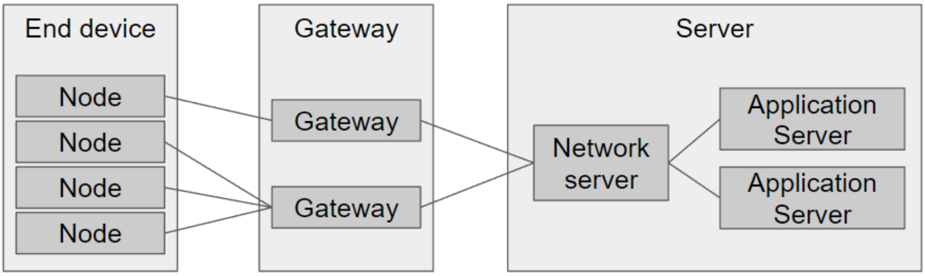


Figure 2.1: Architecture of a LoRaWAN network

gateway will send time synchronized beacons to the end-device to provide time reference.

Class C: End-devices will have nearly continuous receive windows. This is the most power-consuming option.

2.1.4. LORA® TECHNOLOGY EVALUATION KIT

The LoRa® technology evaluation kit - 800 (part number: DV164140-1) from Microchip is used in an experiment validation process to set up LoRa networks. The kit is developed to test LoRa transmission and LoRa network performance. The kit includes a LoRa gateway, 2 LoRa motes and an example LoRa server [18].

The Microchip LoRa gateway has the LoRa module SX1301, which is the base band processor and data concentrator. The gateway provides communication with the Microchip supported example LoRa network and application server. Uplink messages are issued according to the LoRaWAN specification, and they are captured and forwarded by Microchip's Gateway. The gateway has 6 channels, and it includes an LCD screen, SD Card for Configuration Data, Ethernet connection, 868 MHz antenna, and full-band capture radios [17].

The LoRa evaluation kit also includes two RN2483 Mote boards (part number: DM164138). The LoRa motes are LoRa end devices. The LoRa Mote is a demonstration board, and it includes a transceiver module RN2483, which is developed according to LoRa technology. The module accepts commands via UART interface. Communication with the module is achieved through two methods of power supply, USB and Battery [16].

2.1.5. THE THINGS NETWORK

The Things Network is an Internet of Things data network, which uses the LoRaWAN network technology to provide low power wireless connectivity over long range. It is a

Figure 2.2: LoRa technology evaluation kit components [18]

global network community, and is open sourced. It connects sensors and actuators with transceivers to servers.

With The Things Network, people are free to set up end devices with sensors and connect to gateways that may or may not be their own. The Things Network provides gateways and server backend services for developers to build their own LoRa applications.

2.2. PROTOCOL VULNERABILITY ANALYSIS

2.2.1. PROTOCOL VULNERABILITY ANALYSIS METHODS

A network protocol is "a specification for the format and relative timing of the messages exchanged" in a spatially distributed system [35].

There are different methods in assessing protocol security like using ProVerif [20], building attack tree [29], creating honeypot to find new attacks [3], using fuzz testing to test software [33], etc. In this thesis, attack tree is mainly used .

2.2.2. ATTACK TREES

Attack trees form a convenient way to systematically categorize the different ways in which a system can be attacked [29]. Attack tree notation is graphical and structural, providing a promising method to automate the threat analysis process.

Figure 2.3 shows an example of an attack tree. In an attack tree, nodes represent attacks while the root node is the final goal of an attacker. Children of a node are refinements of this goal, and leafs therefore represent attacks that can no longer be refined [23].

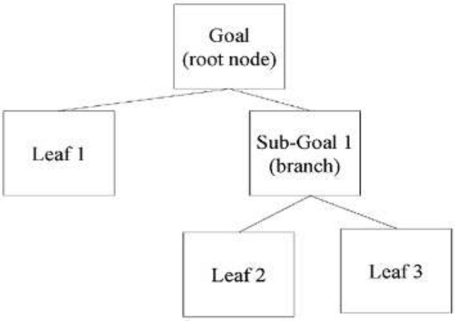


Figure 2.3: Architecture of an attack tree [29]

The advantage of the attack tree analysis is that it allows people to observe the underlying attack flow [19]. It is easy to see the attacker's behavior with the attack tree. However, compare to other security assessment methods such as fuzzing, the attack tree methods cannot tell the statistic result of the attack or the damage that the system may suffer from [33].

3

Security Features of LoRaWAN Protocol

Security has been built-in from the first version of LoRaWAN specification, and the specification offers security in different aspects.

3.1. Activation Methods

Before an end-device is able to communicate with the network server, the end-device should be activated and pass the join procedure. This mechanism is to control the access from unrecognized end-devices to a LoRaWAN network server and prevent these devices from participating in communications. From the LoRaWAN specification, there are two activation methods for end devices: Activation by Personalisation (ABP) and Over-the-Air Activation (OTAA).

3.1.1. Over-the-Air Activation (OTAA)

The Over-the-Air Activation consists of a “Join request” and a “Join accept” between an end-device and a server.

Join request

When the activation process starts, an AppKey should be assigned to both end-device and the network server. The end-device should know its AppEUI and DevEUI, and should be able to generate its DevNonce. AppKey is an AES-128 root key specified to an end-device [4]. AppEUI is an identifier of an application, while DevEUI is a global unique

MHDR	Join request or Join accept or MAC payload	MIC

Table 3.1: LoRaWAN message physical payload structure [4]

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

Table 3.2: Join request format [4]

identifier for an end-device. DevNonce here is a random number sequence and it is generated by issuing a sequence of Received Signal Strength Indicator (RSSI) measurements and it is assumed to be ideally random [4].

When the join procedure starts, it will first send a “Join request” over the air. Table 3.1 shows the physical payload structure of a join request message. Table 3.2 shows the format of a join request.

The message “Join request” is not encrypted, but it uses AppKey to generate the Message Integrity Code (MIC) to insure the integrity of the message.

JOIN ACCEPT

After the join request is received by the network server, the network server will check whether the end-device can be accepted or not. To make it simple, normally this process is done automatically. The network server will check the MIC as well as its DevEUI and AppEUI to determine if the end-device should be accepted and which application it belongs to. If the end-device is not accepted, there will be no response. If it is, then the network server will send a “Join accept” message to the end-device. The join accept message format is in table 3.3.

The Join accept contains a 3-byte AppNonce, which is generated by the network server. AppNonce can be a random value or a unique ID. DevAddr is the device address assigned by the network server to the end-device. NetID is a network identifier. DLSetting, RxDelat and CFList are used for physical layer settings.

The join accept message is first signed and then encrypted by Appkey. During the generation of the join accept message, two important session keys are also generated on the server's side using AppNonce from the server and DevNonce from the end-device. These two session keys will also be generated in the end-device after a join accept is

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

Table 3.3: Join accept format [4]

Key name	Key type	Length/bits	Generation	Usage
AppKey	Symmetric	128	By application	MIC for join request and accept Encrypt join accept Generate session keys
AppSkey	Symmetric	128	By AppKey	Encrypt data messages
NwkSKey	Symmetric	128	By AppKey	MIC for messages Encrypt command-only messages

Table 3.4: Key table of LoRaWAN

received by the end-device.

FEATURES OF OTAA

OTAA provides some security mechanisms.

First, it uses unique parameters. In OTAA, AppKey, DevEUI, AppEUI, AppNonce and DevNonce should all be unique between end-devices. In this case, compromising one end-device does not mean compromising the whole network.

Second, there is a buffer for DevNonce to prevent replay attack. Every time a new join request is received, the server should check the buffer to see if the nonce has been used before. If it has been used, then the end-device is not allowed to join the network. In this case, copying a join request and replaying it is not possible.

3.1.2. ACTIVATION BY PERSONALISATION (ABP)

Compare to OTAA, this activation method skips the join request and join accept. Before the activation, unique parameters, DevAddr, NwkSkey and AppSkey, are all assigned to the end-device instead of DevEUI, AppEUI and AppKey. Also, these parameters are stored in the server. When an end-device is trying to communicate with the server, it will send messages directly. These messages are encrypted and signed. Supposedly, only the network server with corresponding parameters can read the plaintext.

3.2. KEY MANAGEMENT

There are 3 different kinds of keys in a LoRaWAN network. One root key: AppKey, and two session keys: NwkSKey and AppSKey. Table 3.4 briefly introduces these 3 keys including their name, type, length, how keys are generate and their usages.

3.2.1. KEY GENERATION

Section 3.1 introduces how AppKey, AppSKey and NwkSkey generate in join procedure. AppKey is a 16-byte device-unique key. It is assigned by application owners to end-devices. The generation of NwkSKey and AppSKey is different in OTAA and ABP. For OTAA, these 2 keys are generated by AppKey using AppNonce from server side and DevNonce from end-device side. Every time the end-device resets or rejoins, these 2 keys will be regenerated using new nonces. In ABP, NwkSKey and AppSKey are also unique for each end-device. These 2 keys are directly assigned and stored in the end-device before transmission. They are static keys and will not change after resets.

3.2.2. KEY EXCHANGE

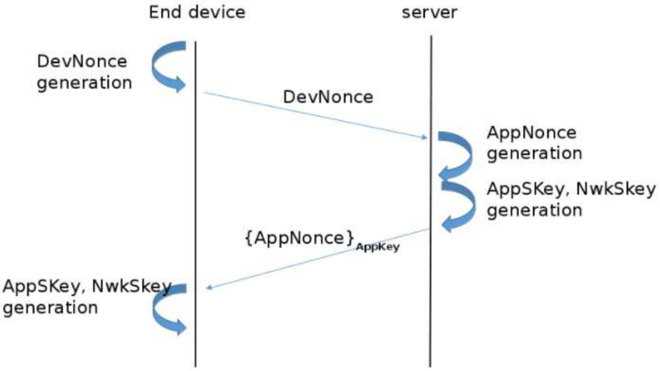


Figure 3.1: OTAA session keys exchange

Key exchange describes how keys or other information exchange so malicious parties will not get a copy. LoRa protocol shows in OTAA, AppKey is assigned to both end-device and server before communication. How AppKey exchange is out of the scope of LoRaWAN. However, how 2 session keys are exchanged is described.

Figure 3.1 shows the exchange process of NwkSkey and AppSKey. First, to join the server, the end device send a “join request”, which includes some identifiers and a “DevNonce”. On the server side, the server will respond to the join request message with a join accept if the end device is allowed to join the network. Join accept includes another “AppNonce”. In this case, both sides can generate AppSKey and NwkSKey with these 2 nonces.

LoRaWAN uses symmetric keys in a smart way. Unlike traditional symmetric key exchange, keys in LoRaWAN are not transmitted over the air. Instead, nonces are transmitted. Only when AppKey, DevNonce and AppNonce (which is encrypted) are all obtained, the third party can derive session keys. In this case, the difficulty of obtaining keys and compromising the network is increased.

3.2.3. KEY USAGE

As mentioned before, AppKey is used only in OTAA to generate NwkSKey and AppSKey. The generation process is as follows:

$NwkSKey = AES128\_encrypt(AppKey, 0\times01|AppNonce|NetID|DevNonce|pad\_16)$

$AppSKey = AES128\_encrypt(AppKey, 0\times02|AppNonce|NetID|DevNonce|pad\_16)$

It can be observed that these 2 session keys are generated by AppKey using a nonce from the end device and a nonce from the server.

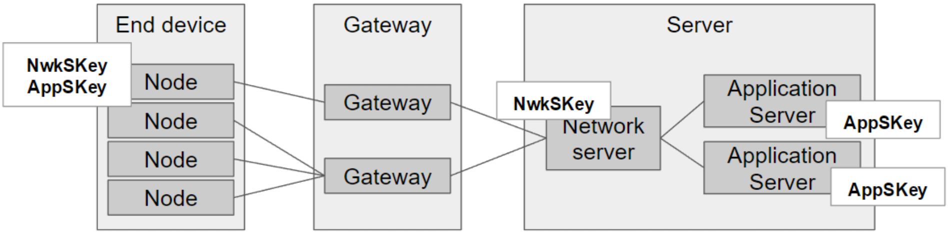


Figure 3.2: Key usage in a LoRaWAN network

Figure 3.2 shows the usage of NwkSKey and AppSKey. NwkSKey is used in network server (normally from the network operator) for encryption and decryption of command-only messages, and also in signing and sign checking of data messages. AppSKey is used in the application server for encryption and decryption for other messages. The main point of building 2 servers and 2 keys is to prevent the network operator from eavesdropping application data.

3.3. CRYPTOGRAPHY

3.3.1. ENCRYPTION AND DECRYPTION

DATA MESSAGE ENCRYPTION AND DECRYPTION

During communications, frame payload is encrypted first. If the frame payload contains only MAC commands, NwkSKey is used for encryption. Otherwise, AppSKey is used. The encryption process is as follows:

- Define blocks $Ai, i = 1...k, k = ceil(length(frame\_payload)/16)$.

- $Si = AES128\_encrypt(K, Ai)$ for $i = 1..k, k = NwkSKey$ or $AppSKey$

Ai	0×01	4×0 × 00	Dir	DevAddr	Fcnt	0×00	i
Size (bytes)	1	4	1	4	4	1	1

Table 3.5: Encryption block of a message in LoRaWAN network [4]

$S = S1|S2|..Sk$

- Truncate $(frame\_payload|pad16)\ xor\ S$ to the first $length(frame\_payload)$ octets.

This encryption method is Advanced Encryption Standard (AES), which is a symmetric encryption algorithm. It supports a block length of 128 bits and key lengths of 128, 192, and 256 bit [10]. AES has been adopted by the U.S. government for securing sensitive but unclassified material, and now it is used worldwide [30]. We can say that AES128 is secure enough for this case.

The block cipher mode of operation here is very similar to Counter (CTR) mode. Figure 3.3 shows the comparison between LoRaWAN block cipher mode and CTR mode for one block. It can be observed that for CTR mode, there is a nonce and a block counter in each block [21]. However for LoRaWAN, the nonce is changed to FCntUp or FCntDown, which is the message counter and is continuously incremented for each message. If the message counter never repeats, then this mode and CTR mode are identical.

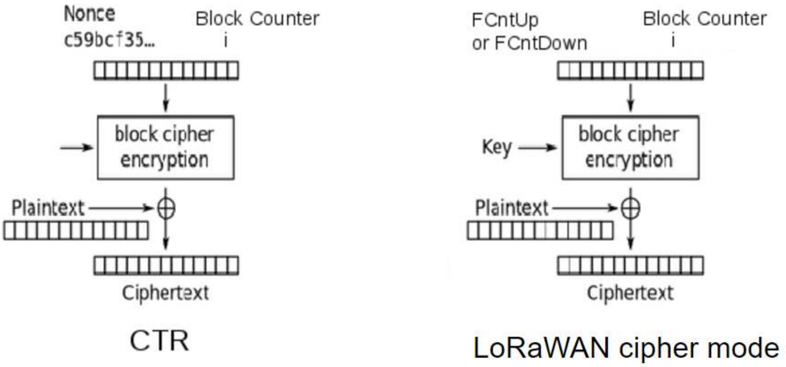


Figure 3.3: Comparison between LoRaWAN encryption mode and CTR mode for one block

JOIN MESSAGES ENCRYPTION AND DECRYPTION

Section 3.1 introduces the join procedure. It is shown that join request messages are not encrypted and join accept messages are encrypted after being signed. So here we only discuss encryption and decryption for join accept messages.

For a signed join accept message, it is encrypted using AES128 ECB mode as follows:

$AES128\_decrypt(AppKey, join\_accept|MIC)$

Here it can be noticed that a decryption operation is used for encryption. In this case, the end-device can use AES encryption to decrypt messages. The operation complexity can be decreased.

The Electronic Codebook (ECB) mode is used in this encryption. Each block is encrypted separately. However, the disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks; thus, it does not hide data patterns well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all [32]. 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.

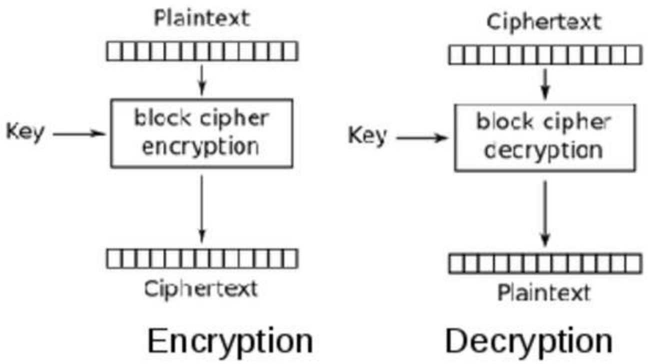


Figure 3.4: ECB cipher block mode [12]

3.3.2. MESSAGE SIGNING

Message Integrity Code (MIC) is used in LoRaWAN to provide integrity check.

DATA MESSAGE SIGNING

The MIC for data message is calculated as follows:

- Define block B_0 in table 3.6.
- $cmac = AES128\_cmac(NwkSKey, B0|MHDR|FHDR|FPort|FRMPayload)$

  $MIC = cmac[0..3]$

B_0	0×49	4×0×00	Dir	DevAddr	Fcnt	0×00	len (msg)
Size (bytes)	1	4	1	4	4	1	1

Table 3.6: Signature block of a message in LoRaWAN network [4]

Notice that here the FRMPayload has already been encrypted.

NwkSkey is used to sign messages. When uplink messages arrived at the network server, the server will first check message integrity, then transfer the well-checked message to application server.

JOIN MESSAGES SIGNING

For join request messages, MIC is generated as follows:

$cmac = aes128\_cmac(AppKey, MHDR|AppEUI|DevEUI|DevNonce)$

$MIC = cmac[0..3]$

For join accept messages, the signing process is as follows:

$cmac = aes128\_cmac(AppKey, MHDR|AppNonce|NetID|DevAddr|DLSettings|$
$RxDelay|CFList)$

$MIC = cmac[0..3]$

3.3.3. SIGN THEN ENCRYPT OR ENCRYPT THEN SIGN?

Messages in LoRaWAN networks should be signed and encrypted. 2 different kinds of messages are discussed in this part to show the differences in signing and encryption. These 2 messages are join accept message and normal data messages.

For join accept messages, “sign then encrypt” is used. For other data messages, instead, “encrypt then sign” is used. The main difference between these 2 methods is whether the integrity of ciphertext is provided. “Sign then encrypt” does not provide ciphertext integrity, since only after decryption will we know whether the integrity is maintained or not. Without ciphertext integrity, it is possible that attackers can create different ciphertext that decrypt correctly.

“Encrypt then sign” provides ciphertext integrity. However, in our case, since decryption and sign checking are operated in 2 servers for data messages, it is possible to modify data between sign checking and decryption.

3.4. COUNTER MANAGEMENT

3.4.1. COUNTER INTRODUCTION

For each end-device, there are two frame counters named FCntUp and FCntDown. FCntUp is counting uplink messages in the end-device, while FCntDown is counting downlink messages in the network server. In order to keep uplink and downlink messages in sync, there is a maximum limitation value MAX_FCNT_GAP. If the difference between uplink and downlink message number is larger than MAX_FCNT_GAP, subsequent remains will be discarded. Counter mechanism in the LoRaWAN protocol can help to prevent packet loss. Counter values are used in both encryption and signing, and the same value should not be used. Figure 3.5 shows an example of frame counter working process.

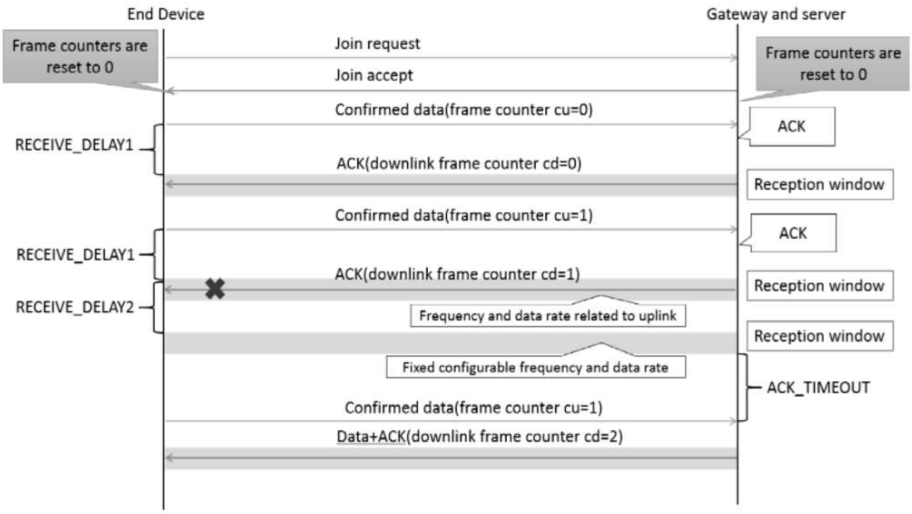

End Device
Gateway and server
Frame counters are reset to 0
Join request
Join accept
Frame counters are reset to 0
Confirmed data(frame counter cu=0)
ACK
RECEIVE_DELAY1
ACK(downlink frame counter cd=0)
Reception window
Confirmed data(frame counter cu=1)
ACK
RECEIVE_DELAY1
ACK(downlink frame counter cd=1)
Reception window
RECEIVE_DELAY2
Frequency and data rate related to uplink
Reception window
Fixed configurable frequency and data rate
ACK_TIMEOUT
Confirmed data(frame counter cu=1)
Data+ACK(downlink frame counter cd=2)

Figure 3.5: An example of frame counter working process

3.4.2. COUNTER OVERFLOW AND RESET

Both 16-bits and 32-bits frame counters are allowed in LoRaWAN. If the counter is overflowed, the counter value will be started from 0 again. For example, for a RN2483 end device, the default setting of uplink transmission is around 50 s/msg. In this case, the overflow will happen every $2^{16}/50$ s, which is approximately 38 days.

For end devices, according to the LoRaWAN specification, the counter value will be set to zero after resetting.

3.5. MESSAGE ACKNOWLEDGEMENT

3.5.1. ACKNOWLEDGEMENT PROCEDURE

In a LoRaWAN network, when the end device sends an uplink confirmed message, the server will check the message. If the message is acceptable, the server will respond with a frame that has the ACK bits in FCtrl. Table 3.7 shows an example of a message with ACK bits. This message is a downlink ACK message without frame payload. Note that when there is a donwlink scheduled message for a class A end device, the frame payload will be included in the message.

Physical payload	MHDR = 60	DevAddr	FCtrl = 20	FCnt	FPort	MIC
size (bytes)	1	4	1	2	1	4

Table 3.7: An example of a message with ACK bits

3.5.2. RETRANSMISSION

For an uplink confirmed message, if the end device cannot receive an ACK during its receive windows, it will retransmit the message for several times. If, after restransmissions, the end device still cannot receive an ACK, it will consider the message to be lost or rejected.

4

The vulnerability analysis method for LoRaWAN

In order to explore the vulnerabilities in a LoRaWAN network, researchers should “think like an attacker”, which means to consider and analyze the possible attacks, and find the root reasons of such attacks. This chapter introduces the method used in this research to analyze attacks. A general introduction to attacks is also given.

6 steps to analyze the vulnerabilities of a LoRaWAN system are as follows.

4.1. Step 1: Identify attacker goals

Before compromising a LoRaWAN network, the attack goal is defined in 2 aspects: compromising the network security properties and compromising network security assets.

Security properties in a network are always described with the CIA triad. The CIA triad, which includes confidentiality, integrity and availability, reflects the most popular information security requirements. In this case, in order to compromise a LoRaWAN network, it is important to consider about these 3 features.

Security assets reflect the most important parameters in a network. If these parameters are compromised, the security of the network will be compromised.

Table 4.1 shows assets in LoRaWAN networks which are protected. In the table, the classification of LoRaWAN shows the importance of such an asset. For example, NwkSKey is the primary asset, since an attack that could compromise the NwkSKey is able to di-

Asset	Primary or secondary asset	Confidentiality	Integrity
NwkSKey	P	√	√
AppSKey	P	√	√
AppKey	S	√	√
DevNonce	S	×	√
AppNonce	S	√	√
FrmPayload	P	√	√
DevAddr	S	×	√
Fcnt	S	×	√
ACK	S	×	√
MAC commands	S	√	√

Table 4.1: Assets in a LoRaWAN network

rectly compromise the network by using the key to decrypt message. However, for secondary assets such as AppKey, if an attacker get only the DevNonce, more assets are still needed to compromise the network.

Some assets in the table are confidential while others are not. For example, the confidentiality of AppNonce is protected while the DevNonce is not. This is because the derivation of two session keys needs the use of both of these nonces. If they both are not protected, session keys can be derived once the attacker gets AppKey. Protecting both is not necessary and would increase of computing complexity.

The integrity of all the assets are protected. LoRaWAN uses a strong integrity protection method (introduced in chapter 2), and always checks the integrity of whole physical payload.

The explanation of assets in table 4.1 are as follows:

- NwkSKey: NwkSKey is used in the network server to check message signature. It is generated during node activation. If confidentiality of the NwkSKey is compromised, a third party can use the NwkSkey to generate its own LoRaWAN message, and it will pass the signature checking procedure at the network server. If integrity of NwkSkey is compromised in a network server or in an end device, the communication session of these devices will be compromised, and all the messages in this communication session will be discarded because they cannot pass the signing procedure.

- AppSkey: AppSKey is used in the application server to decrypt messages. It is generated during node activation. If confidentiality of AppSKey is compromised, the attacker will be able to decrypt all the messages. The confidentiality of the whole

LoRaWAN network will be compromised. If the integrity of AppSKey is compromised, the application server or the end device will not be able to decrypt the messages properly. The data received then cannot be trusted.

- AppKey: AppKey is used in node activation for OTAA activated devices to derive AppSKey and NwkSKey. It is assigned by the application owner to both the end device and the server before activation. If the confidentiality of AppKey is compromised, the attacker will be able to operate join request replay attack, to let a malicious end device join the network. If the integrity of AppKey is compromised, the end device will not be able to join a LoRaWAN network through OTAA.

- DevNonce: DevNonce is a nonce generated by end device. In OTAA activation, DevNonce will be transmitted over the air from an end device to a gateway and a network server, and then DevNonce and AppNonce will be encrytpted by AppKey to generate NwkSKey and AppSKey. DevNonce can be transmitted in plaintext since without AppSkey, the attacker cannot achieve any attacks. The integrity of DevNonce should be protected. Without integrity protection, session keys generated by the end device and by the server will be different. The communication session will be invalid.

- AppNonce: AppNonce is a nonce or some form of unique ID by network server. It is used to generated session keys with DevNonce and AppKey. If the confidentiality of AppNonce is not protected, once the attacker get AppKey, it can easily compute NwkSKey and AppSkey, and damage the security of the whole LoRaWAN system. If the integrity of AppNonce is compromised, session keys generated by the end device and by the server will be different. The communication session will be invalid.

- FrmPayload: FrmPayload consists of important data that are transmitted. For example, temperature sensor data will be transmitted in FrmPayload. If the attacker can compromise the confidentiality of FrmPayload, the sensor data will be known to attackers, and it will cause privacy issues. If the attacker can compromise the integrity of FrmPayload, the sensor data received by servers will be invalid and should not be trusted.

- DevAddr: DevAddr is the end device identifier. It is in plaintext. If its integrity is compromised, the communication between the end device and the server will be interrupted.

- Fcnt: FCnt is the counter value in both end device and the server. It is in plaintext. If the integrity of FCnt is compromised, it will be difficult for the server and the end device to keep in synchronization. Also, modifying counter value can make it easy for attackers to achieve replay attack.

- ACK: ACK is a special message parameter, which is used to acknowledge received messages. It is in plaintext. If the integrity of ACK is compromised, it is possible that an ACK message will be modified to a normal message, and the function of ACK will be disabled.

- MAC commands: MAC commands can be sent in FOpts or in FrmPayload. Some commands, like radio parameter settings, should be confidential. Otherwise, the attacker can easily know the radio status of end device. Integrity of MAC commands should also be protected.

4.2. STEP 2: DEFINE ATTACKER CAPABILITIES

In our approach, the definition of attacker capabilities is based on Dolev-Yao Model [8], in which the attacker has complete control over the network, and can intercept, corrupt and send messages to any participant. In a LoRaWAN network, it can be defined that the attacker has the ability to:

- Have the knowledge of LoRa network and devices

- Capture and send messages over the air

- Process and store data

- Conduct encryption and decryption if keys are known

- Physical approach

4.3. STEP 3: DEFINE LORAWAN SYSTEM FEATURES

Based on the LoRaWAN specification 1.0.2, a LoRaWAN system has security features as mentioned in chapter 2. Besides security features, it also has physical features that can be influence factors for an attacker to perform attacks.

- Wireless transmission

  Wireless transmission has the nature of broadcast, giving attackers more opportunities to attack the network. Also, in a wireless sensor network, packet-based routing is connectionless, and cannot be seen as reliable if no other security mechanism is used.

- Latency

  For Class A and B in a LoRaWAN network, there is transmission latency in downlink transmission, and the synchronization issues can be critical when critical event reports or security issues are transmitted via the downlink.

- Limited Memory and Storage

  In wireless sensor networks, the memory and storage of end devices are normally small. In this case, the security mechanism for such kind of network should be simple, and the code size should also be small. From another perspective, these devices can be vulnerable to attackers that consume huge amounts of memory, because it will be easy to crash the system.

- Limited duty cycle

  Duty cycle is the ratio of period that the signal or the system is active. With a limited duty cycle, attacks like jamming can be reduced.

4.4. Step 4: Model attacks

In this step, an attack tree is built to show the possible attacks towards the LoRaWAN protocol. The attack tree is in figure 4.1.

The root node represents the final goal: compromising a LoRaWAN network. The second level nodes represent the security properties that the attacker aims to violate. The remaining levels represent the attacks that can achieve the goals in the first and second level. “And” in the figure indicates the corresponding nodes are different steps to achieve a goal.

Since the main goal of this thesis is to find vulnerabilities of the protocol, and prove these flaws really exist through conduct attacks, we only focus on attacks that are related to the vulnerabilities of the protocol. Physical attacks and attacks irrelevant to the protocol are not in our scope.

In the next chapter, the description of these 4 attacks will be given.

4.5. Step 5: Generate attack scenarios

In this step, the scenarios and applications in reality will be produced for these 4 attacks. This step can give a distinct impression for how these attacks influence the LoRaWAN network.

4.6. STEP 6: EXPERIMENTAL VALIDATION

The private LoRaWAN network is built according to LoRaWAN specification 1.0.2. The network is using LoRa (R) Technology Evaluation Kit - 800 (DV164140-1) from Microchip. This kit is in 868 MHz and includes a 6-channel gateway, two motes with RN2483 module and an example server. However, during the test, the server cannot pass the downlink confirmed message transmission test. Therefore, another server based on the work of Orne Brocaar is developed and used.

The experiment procedure and results are introduced in chapter 5.

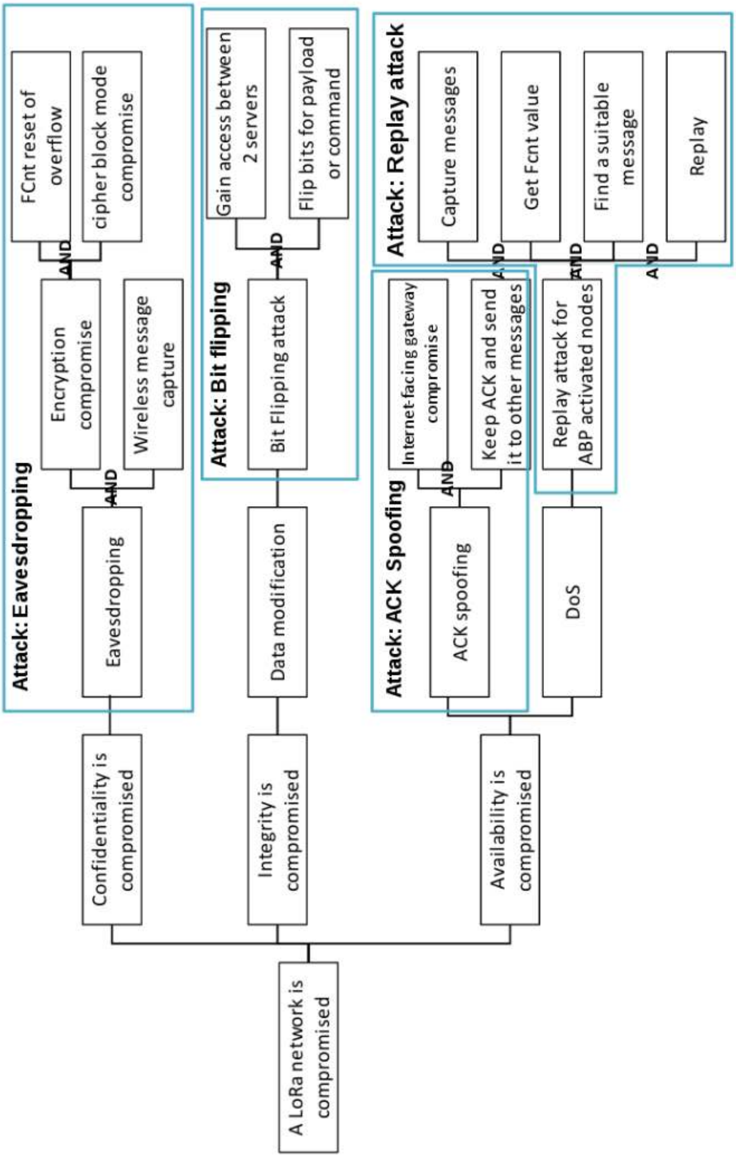


Figure 4.1: The attack tree for a LoRaWAN network. Note that the attack tree only shows 4 attacks that are found in this thesis. More attacks are still possible

5

Attacks against LoRaWAN networks

In this chapter, 4 attacks against the LoRaWAN protocol are given. These attacks are presented and analyzed according to the vulnerability analysis steps mentioned in chapter 4.

5.1. Attack 1: Replay attack for ABP activated nodes

5.1.1. Attack goal

This attack is designed to achieve spoofing and DoS.

For the server, the attack goal is to achieve spoofing. After the attack, it will accept a malicious replayed message from the attacker's end device, and the server 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.

5.1.2. Attacker capabilities

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, it will not take a long time for an attacker to wait for an overflow. However, if the attacker is performing attacks in a relatively small network, it is better if the attack is able to reset the victim end device to reduce the waiting time.

5.1.3. PROTOCOL VULNERABILITIES

This part introduces the system features that lead to this attack.

- ABP activation method has security flaws. For ABP activated end-devices, they are using 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, as long as it meets requirements as follow ,it can be accepted by LoRaWAN network server.

  - Session keys are the same as one accepted end device
  - DevAddr is the same as one accepted end device
  - 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.

- Counters are not used in a secure way. In the protocol specification, it is said 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.”
  >
  > – LoRaWAN specification 1.0.2

  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.

  No matter the end device is activated by ABP or OTAA, the replay attack is possible. Besides resetting, another method to restart the counter is 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, the attacker can also replay previous messages to cut off the communication between the end device and the server.

The main point of achieving replay attack is to make the counter value repeat. Therefore, for OTAA activated end devices, in order to achieve this attack, the attacker should wait till the counter value of the end device becomes maximum and then restart from 0. For ABP activated end devices, the attacker can also wait till the counter overflow, or the attacker can reset the end devices and make counter value start from 0. Attacking an ABP activated end device will cost much less time than an OTAA activated end device, as long as the attacker has the ability to reset end devices.

5.1.4. ATTACK DESCRIPTION

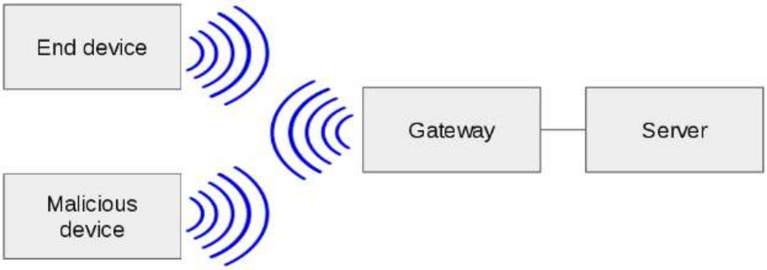


Figure 5.1: The LoRaWAN network setup for replay attack

Figure 5.1 shows the basic attack setup for such an attack. To operate the attack, these steps should be followed:

- Capture messages. Use a device to capture uplink messages of an ABP activated node, and save them into the attacker's database
- Get FCnt value. Read the uplink counter value from these messages since counter values are not encrypted.
- Wait till the end device resets or counter overflows.
- Find a suitable message. Select a captured message with suitable counter value from attacker's database.

  Here, 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.

  - If $C_m - C_e <= Gap$: Malicious message will be accepted. Messages from end device with the counter value in [Ce, Cm] will be ignored.
  - If $C_m - C_e > Gap$: Malicious message will be ignored.

  The most harmful attack is to select the counter value $C_m = Gap + C_e$, since it will take the longest time to wait till it is recovered.

- Replay. Resend the message to the gateway.

Figure 5.2 shows an example of a replay attack. Here the maximum counter gap is 16384, and it is the same as the default parameter value. The malicious message is the message in the last session with 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 smaller counter value from 25 to 70 will be ignored.

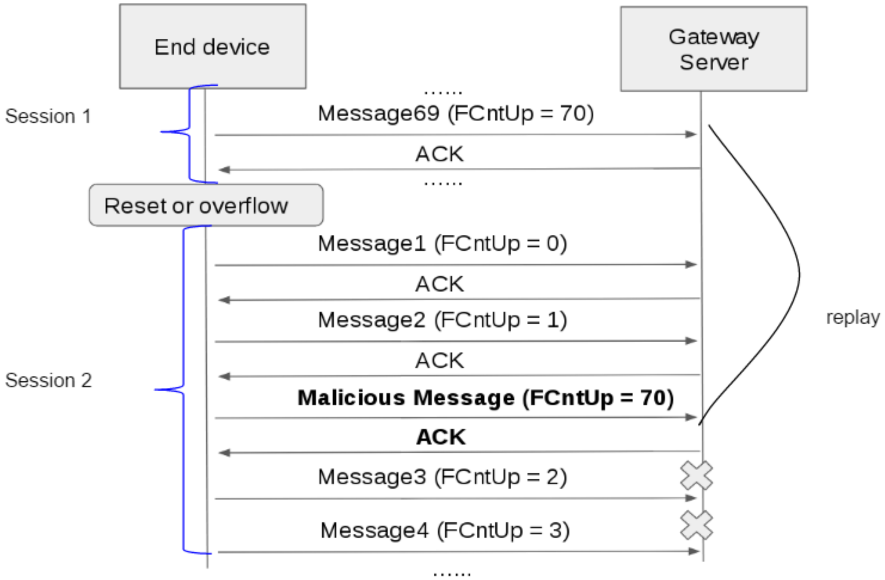


Figure 5.2: An example of Replay attack for ABP activated network

5.1.5. GENERATE ATTACK SCENARIOS

In this attack, the attacker can use a traffic sniffer to sniff LoRaWAN traffic, and use a LoRa transmitter to replay messages.

This attack can be extremely harmful for ABP activated end devices in a large LoRaWAN network. In a small LoRaWAN network with only a few end devices, the attacker may need to wait a long time for a counter overflow. However, in a large LoRaWAN network with multiple end devices, the waiting time for any one of the end devices to be overflowed is highly decreased. Once the attacker gets the largest possible counter value for one end device, it can periodically replay this message, to make the end device be rejected permanently. Unless the session keys of the end device are changed, the end device cannot be functioning again.

In addition, if the attacker can find a way to reset the end device (e.g. power outage), then there is no need for the attacker to wait for counter overflowing. By resetting the end device, and replaying the message with the largest counter value, messages from the victim end device will be rejected.

5.1.6. EXPERIMENT VALIDATION

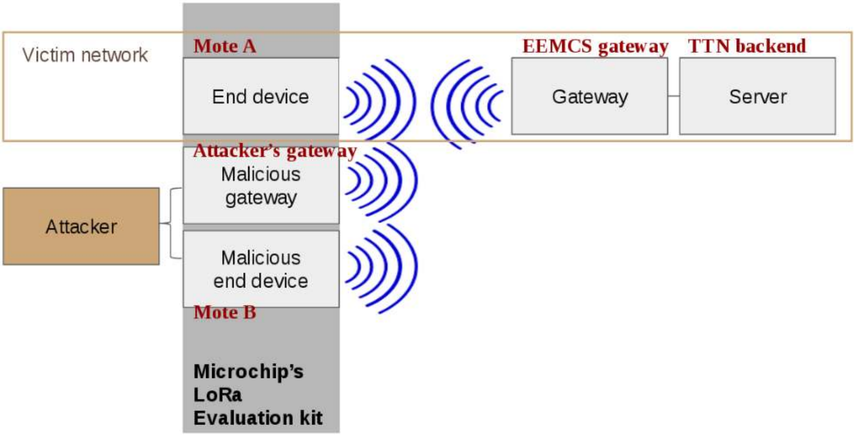
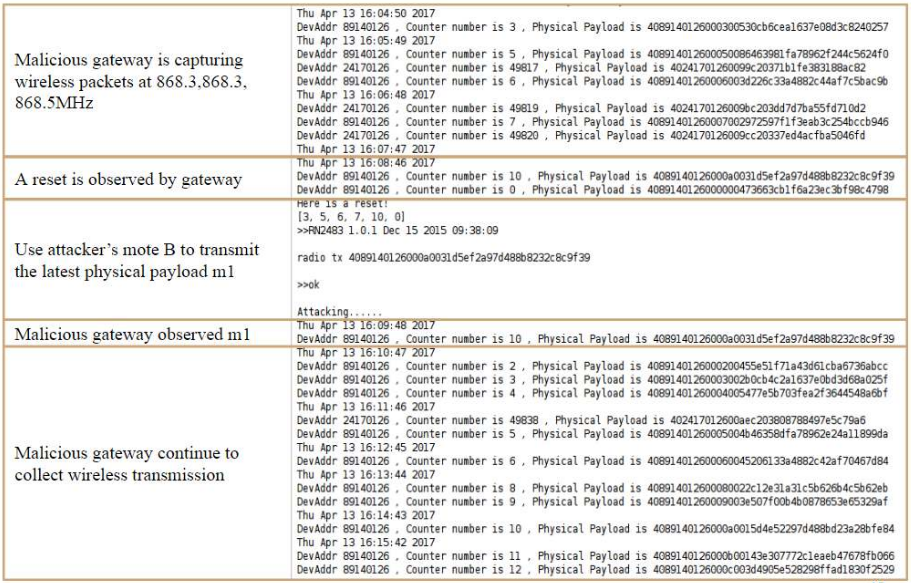


Figure 5.3: Setup for LoRaWAN replay attack

EXPERIMENT ENVIRONMENT

The attacker can use any LoRa transceiver to achieve the attack as long as the device is able to transmit and receive LoRa wireless messages. In this experiment, a LoRa gateway is used as a receiver and a LoRa mote is used as a transmitter.

In this case, the setup of this attack is shown in figure 5.3. On the one hand, the malicious attacker owns a gateway and an end device named Mote A, which belong to the Microchip's evaluation kit. On the other hand, the victim's network consists of a mote named Mote A (from the evaluation kit), a gateway and a server backend (from The



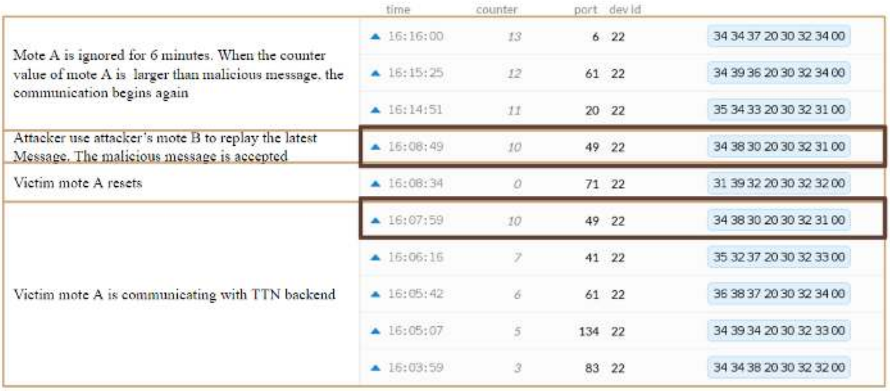
Malicious gateway is capturing wireless packets at 868.3,868.3, 868.5MHz	Thu Apr 13 16:04:50 2017 DevAddr 89140126 , Counter number is 3 , Physical Payload is 4089140126000300530cb6cea1637e08d3c8240257 Thu Apr 13 16:05:49 2017 DevAddr 89140126 , Counter number is 5 , Physical Payload is 40891401260005008646398 1fa78962f244c5624f0 DevAddr 24170126 , Counter number is 49817 , Physical Payload is 40241701260099c20371b1fe383188ac82 DevAddr 89140126 , Counter number is 6 , Physical Payload is 4089140126000 6003d226c33a4882c44af7c5bac9b 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 4089140126000700 2972597f1f3eab3c254bccb946 DevAddr 24170126 , Counter number is 49820 , Physical Payload is 4024170126009cc20337ed4acfba5046fd Thu Apr 13 16:07:47 2017
A reset is observed by gateway	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
Use attacker's mote B to transmit the latest physical payload m1	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......
Malicious gateway observed m1	Thu Apr 13 16:09:48 2017 DevAddr 89140126 , Counter number is 10 , Physical Payload is 4089140126000a0031d5ef2a97d488b8232c8c9f39
Malicious gateway continue to collect wireless transmission	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 408914012600060045206133a4882c42af70467d84 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 5.4: Log file of malicious gateway

Things Network). The victim's network is working properly, and the victim's end device is activated by personalisation.

EXPERIMENT PROCEDURE

- The victim's end device is communicating with its server via the gateway.
- The attacker tunes the gateway to frequency 868 MHz to listen to any LoRa messages in this frequency band.
- The attacker uses malicious gateway to keep capturing messages from Mote A to the TTN gateway. The collected messages are physical payloads in Hex. Devaddr, counter value are in plaintext. These physical payloads are stored in the database.
- The victim's end device performs a reset. The keys are not changed and the counter values are reset to zero.
- As long as the malicious gateway observes one reset, the attacker needs to find the message, named M, with the same devaddr and the largest counter value in the data base. The reset is defined that for the same devaddr, the counter value of the message is smaller than the last one it received.



	time	counter	port	dev id	
Mote A is ignored for 6 minutes. When the counter value of mote A is larger than malicious message, the communication begins again	▲ 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
Attacker use attacker's mote B to replay the latest Message. The malicious message is accepted	▲ 16:08:49	10	49	22	34 38 30 20 30 32 31 00
Victim mote A resets	▲ 16:08:34	0	71	22	31 39 32 20 30 32 32 00
Victim mote A is communicating with TTN backend	▲ 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 5.5: Log file of victim's server

- Use the malicious mote B as a LoRa radio transmitter to resend the physical payload it just found in database. The message is sent in the same frequency band.

- If the victim can observe the message M from its server, the replay attack is successful.

EXPERIMENT RESULT

Figure 5.4 shows the log file from the malicious gateway. The first row shows the malicious gateway is sniffing LoRa wireless packets at 868.1, 868.3, 868.5 MHz, and the gateway can read the message contents of DevAddr, counter value and physical payload. The second row shows a reset is observed by the gateway. If the current counter value is smaller than or equal to the previous counter value, the gateway will recognize as a reset happens. The third row shows the transmission of malicious messages. After a reset is observed, the attacker's mote will resend the latest physical payload. The forth row shows the malicious gateway observed the malicious message. The last row shows that the attack is finished, and the malicious gateway continues to collect messages, and wait for next reset.

Figure 5.5 shows the log file of the victim's server. It can be seen that the message received at 16:07:59 is replayed at 16:08:46, making the victim's server stop accepting messages from the same end device for 6 minutes. From figure 5.4, these rejected messages are still sniffed by the malicious gateway.

This experiment can be seen as a proof of the vulnerabilities of counter management and static keys exist in LoRaWAN protocol.

5.2. ATTACK 2: EAVESDROPPING

5.2.1. ATTACK GOAL

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 ciphertext.

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.

5.2.2. ATTACKER CAPABILITIES

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.

In order to increase the accuracy of the decryption results, it is better if the attacker also has the ability to reset the end devices.

5.2.3. PROTOCOL VULNERABILITY

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. There is also another vulnerability in this case: the cipher block mode is not secure.

From chapter 3, it is known that LoRaWAN data messages are using a block cipher mode similar to CTR. Instead of using a nonce in the block, 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.

Since LoRaWAN will provide attackers a way to get messages with same key stream, the cryptography can be compromised as follows:

If we have 2 messages with same key stream, then

$$Plaintex1 \oplus Keystream = Ciphertext1$$
$$Plaintext2 \oplus Keystream = Ciphertext2$$

Then we have:

$$Plaintext1 \oplus Plaintext2 = Ciphertext1 \oplus Ciphertext2$$

Since ciphertexts are known, in order to get the plaintexts, we first guess a part of content in plaintext 1, then derive the part of plaintext 2 in corresponding position. If all the plaintexts are readable, the guess is possible to be true. In this way, the possibilities of plaintext can be highly decreased. In order to make the guessing, there are some regular patterns in different cases. In addition, the more resets, the higher possibilities to recover the messages. This method is also called as “crib dragging” [15].

5.2.4. ATTACK DESCRIPTION

Figure 5.6 shows the setup for this kind of attack. Besides the target network, a malicious gateway and server is built by the attacker to capture wireless packets from the target network.

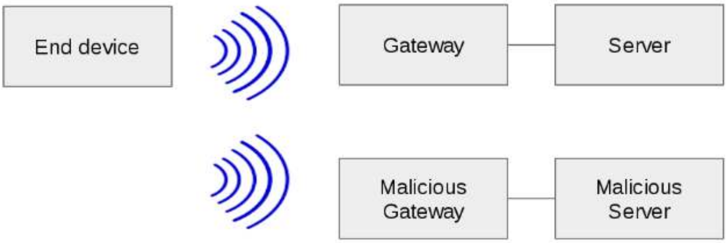


Figure 5.6: The LoRaWAN network setup for eavesdropping

The attack can be operated in following steps:

- The attacker captures and stores LoRaWAN wireless packets, and logs basic information.
- After resetting, continue to collect packets. Compare packets before and after resetting. Pair packets with same counter value.
- Coding with method crib dragging, see the result.

Figure 5.7 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, we can use crib ragging to derive the plaintext. In different cases, the implementation of crib dragging can be different. For example, if the plaintexts are sentences in English, it is easy to guess. If the plaintexts are numbers, we need to find the regular patterns behind the numbers first.

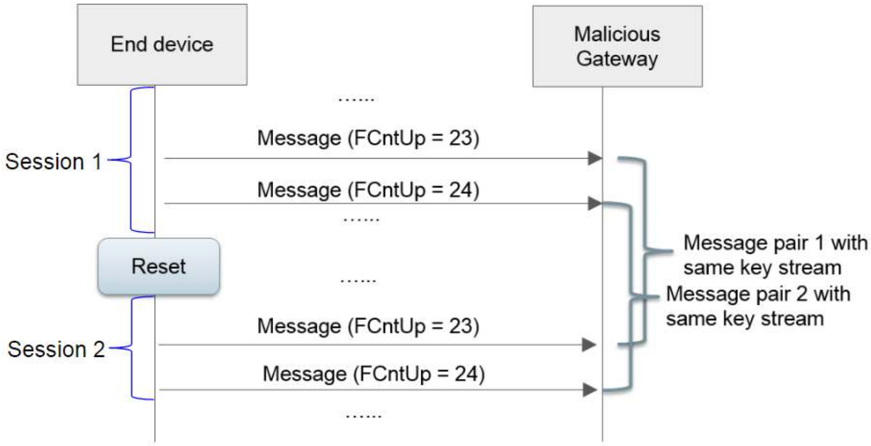


Figure 5.7: An example for Eavesdropping attack

5.2.5. GENERATE ATTACK SCENARIOS

Based on the encryption flaws in LoRaWAN, for ABP activated devices, the eavesdropping attack aims to compromise the privacy of transmitted data. The attacker needs a LoRaWAN receiver to receive wireless messages, and then perform decryption.

The method to collect a message with the same counter value is to wait for overflowing, or reset the end device manually. When the attacker is able to reset the end device, this attack can be easily achieved. The attacker just needs to operate reset for 3 or more times, then multiple messages can be decrypted.

This attack is harmful in a LoRaWAN system when LoRa is used to transmit critical data. It is needed for an attacker to get multiple messages with the same counter value. For a LoRaWAN network that is working for a long time, the attacker may need to sniff the traffic for a long time (e.g. several months), or reset the end device for several times, to get enough messages. However, the efficiency of this attack is better in the long run. After the attacker can decrypt received messages, every time a new message is sniffed, the attacker will be able to decrypt it. Unless the session keys are changed, the privacy of data cannot be guaranteed.

5.2.6. EXPERIMENT VALIDATION

EXPERIMENT ENVIRONMENT

In this expreiment, a mote is configured and is sending data messages periodically. The gateway in the kit is used as the malicious gateway, and it keeps tracking the uplink

MHDR	DevAddr	FCtrl	FCnt	FPort	FrmPayload	MIC
40	99999999	00	6700	3D	0295178267571592	007B3A8A

Table 5.1: Physical payload format of a given message

messages from the mote. From the server side, even if the mote does not join in the network, the physical payload is visible.

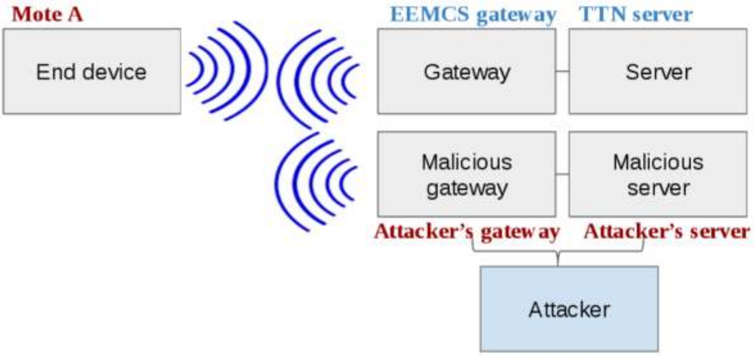


Figure 5.8: The setup for eavesdropping

EXPERIMENT PROCEDURE

LoRa can be used to transmit both numbers and alphanumeric strings. The attack methods are different for these 2 kinds of message content.

- Assumption 1: Plaintext consists of numbers
  - Define message format

    An attacker with basic knowledge of LoRaWAN devices can figure out the message format in this network. In our case, the physical payload format of a data message is as follows: Here is an example physical payload we received: "409999999900670003D0295178267571592007B3A8A".

    The message format for it is in table 5.1 .

    The parameters such as counter value and device address are known. Based on these parameters, after resetting, the messages with same key stream can be paired.
  - Define regular pattern for the implemented LoRaWAN network.

In this implementation, the default frame payload for data messages in these devices is 16 bytes, consisting of one light measure value and one temperature value. With basic knowledge of these devices, the regular patterns for frame payload can be derived as follows:

  - ◇ The plaintext only consists of numbers, space and placeholder. The length of the plaintext is 8. There are only 12 possibilities for one digit, 10 numbers from 0 to 9, one space, and one placeholder.
  - ◇ A message is divided to 2 parts: light and temperature. There must be one and only one space in between.
  - ◇ The 1st part of the message is light value. There are at least 2 digits. If the light value is only 1 digit, there will be a placeholder to fill in the space after the space between light and temperature. Also, the light value will not start with a “0”.
  - ◇ The 2nd part of the message is temperature value. There are 3 digits for temperature. If the temperature value is 2 digits, then it will start with a “0”.
  - ◇ If the whole length is shorter than 8, there will be one or two placeholders at the end.

- Build the prediction model with python.

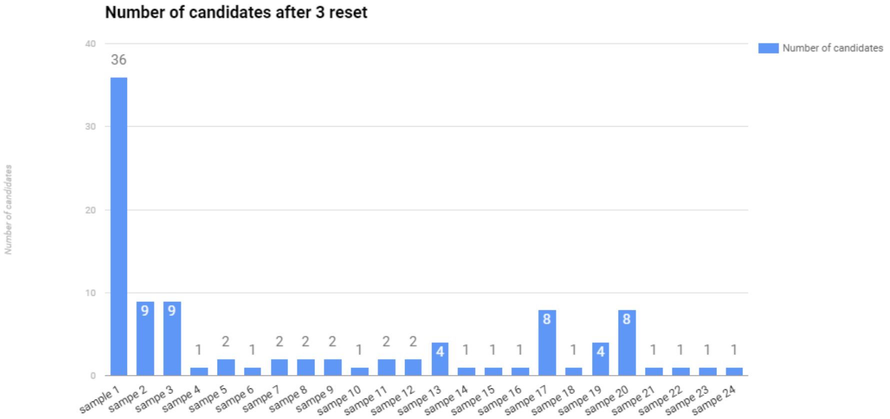


Figure 5.9: The number of possible candidates after calculation

        - Choose plaintext 1 as the victim, and plaintext 2 as the reference object. Both these plaintexts are unknown.
        - Traverse 12 options for each digit for plaintext 1, and check the corresponding plaintext 2 to see whether it is readable. Pick up all the readable options.
        - Use the regular patterns as constraints to decrease the number of possible options for plaintexts.
        - Increase times of resetting, using more plaintexts to derive the victim plaintext.
    - Increase times of experiment to see the statistics.

- Assumption 2: Plaintext consists of alphanumeric strings.

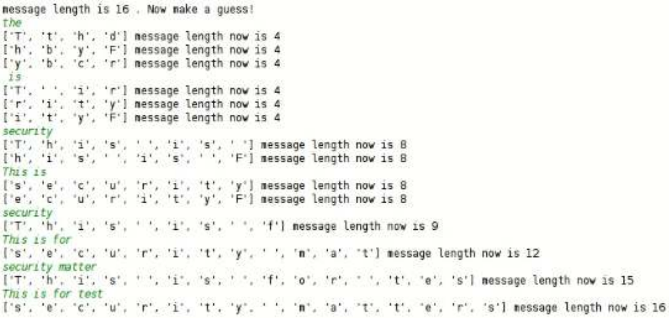
  Crib dragging method is used to decrypt alphanumeric-string messages. The attack begins with guessing one possible word, and the derive the corresponding word. An example is given in follow:

    - The victim is sending alphanumeric-string messages to the server. Counter reset happens.
    - The attacker can sniff the wireless traffic, and store the ciphertext data. After the counter reset is sniffed by the attacker, the attacker will pair messages with the same counter value. For example, a pair of ciphertexts are: "8b8049fc22d1246b7d79564e02f0d376" and "ac8d43fa70d123323b7b451a02f0d271".
    - The attacker knows the corresponding relationship between a pair of messages. Then, the attacker will start guessing. In this case, the attacker may start with the common words such as "the".
    - After xor calculation of two ciphertexts and one plaintext "the" in different positions, we found there is no readable words in the result, which is the corresponding plaintext. Then, it is possible that "the" is not in the plaintext.
    - The attacker guesses another word "is". After calculation, a possible result is got: "rity". Afterwards, the attacker may think "rity" is a part of "security", and then, "security" will be used in guessing. If the attacker is lucky, the message content can be derived after several times of guessing.

  Figure 5.10 shows the crib dragging process in this example.

EXPERIMENT RESULTS

- If only numbers are transmitted:



```
message length is 16 . Now make a guess!
the
['T', 't', 'h', 'd'] message length now is 4
['h', 'b', 'y', 'F'] message length now is 4
['y', 'b', 'c', 'r'] message length now is 4
 is
['T', ' ', 'i', 'r'] message length now is 4
['r', 'i', 't', 'y'] message length now is 4
['i', 't', 'y', 'F'] message length now is 4
security
['T', 'h', 'i', 's', ' ', 'i', 's', ' '] message length now is 8
['h', 'i', 's', ' ', 'i', 's', ' ', 'F'] message length now is 8
This is
['s', 'e', 'c', 'u', 'r', 'i', 't', 'y'] message length now is 8
['e', 'c', 'u', 'r', 'i', 't', 'y', 'F'] message length now is 8
security
['T', 'h', 'i', 's', ' ', 'i', 's', ' ', 'f'] message length now is 9
This is for
['s', 'e', 'c', 'u', 'r', 'i', 't', 'y', ' ', 'm', 'a', 't'] message length now is 12
security matter
['T', 'h', 'i', 's', ' ', 'i', 's', ' ', 'f', 'o', 'r', ' ', 't', 'e', 's'] message length now is 15
This is for test
['s', 'e', 'c', 'u', 'r', 'i', 't', 'y', ' ', 'm', 'a', 't', 't', 'e', 'r', 's'] message length now is 16
```

Figure 5.10: An example of crib dragging

The experiment result is shown in figure 5.9. After 3 resets, the number of possible candidates are calculated.

In these 24 valid samples, 45.8% samples can have only one candidate, which is the final result of original data. For the others, the number of possible candidates is highly decreased (from 12^8 to 4,8,9,36...)

This attack is based on the assumption that the attacker is able to conduct reset and affect the light sensor value.

If the attacker is not able to affect the sensor, the number of resets needed will be larger.

- if only alphanumeric strings are transmitted:

  This experiment is objective since it is based on guessing. This experiment shows the corresponding relationship between 2 ciphertexts with the same counter value. It can be seen that reusing the same counter value can decrease the confidentiality of a LoRaWAN system.

5.3. ATTACK 3: BIT FLIPPING ATTACK

5.3.1. ATTACK GOAL

The goal of this attack is to prove that the integrity between the network server and the application server is not protected. If the attacker has the ability to compromise the transmission in between, there is no way for an application server to recognize whether the message comes from the attacker or the network server.

This attack is to flip a bit between the network server and the application server, to see if the application server can decrypt the message properly.

5.3.2. ATTACKER CAPABILITIES

In order to achieve this attack, the attacker should have the ability to:

- perform man-in-the-middle attack between the network server and the application server.
- have the basic knowledge of physical payload format.
- have the basic knowledge of the message type from the end device.

5.3.3. PROTOCOL VULNERABILITY

The integrity between application server and network server is not checked. Uplink messages are encrypted then signed. After they are received by the network server, the network server will use NwkSKey to check the signature of the message. After this, encrypted messages are accepted in the network server and then handled to application server. Between the network server and the application server, data can be modified during the handling, because when messages arrive in the application server, the integrity of ciphertext will not be checked anymore.

5.3.4. ATTACK DESCRIPTION

Figure 5.11 shows the setup for bit flipping attack in a LoRaWAN network. The attack can be conducted as follows:

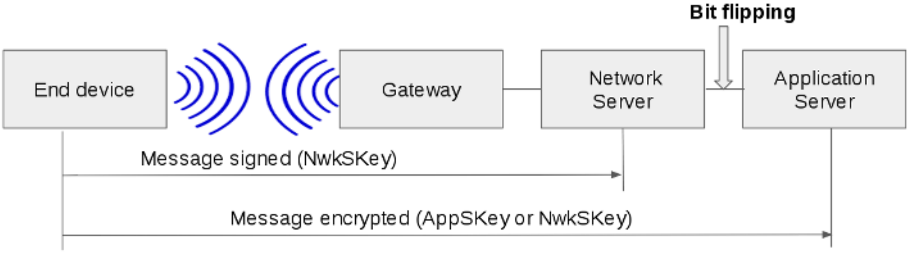


Figure 5.11: The LoRaWAN network setup for bit flipping

- If the malicious attacker has access to the network server, or has the ability to conduct a man-in-the-middle attack between the network server and the application server, it is able to conduct a bit flipping attack.
- We know that:

$$plaintext \oplus keystream = ciphertext$$

$$ciphertext \oplus keystream = plaintext$$

- The position of plaintext is corresponding to the same position of ciphertext. Based on the attacker's goal, the attacker can modify the ciphertext to affect the plaintext.
- Header, routing information, commands can also be modified.

5.3.5. GENERATE ATTACK SCENARIOS

This attack is based on the assumption that the attacker is able to compromise the communication between the network server and the application server. In a LoRaWAN network, the communication type between the network server and the application server can be Ethernet, WiFi, 3G, etc. If the attacker is able to find a method to achieve man-in-the-middle attack between 2 servers, the attacker will be able to modify any LoRaWAN packets. If the FrmPayload is modified, the data application server receives will be false. If the DevAddr of the message is changed, the application server will consider the data is from another end device. If the counter value is modified, and the message satisfies the conditions that the message's counter value is smaller than the counter value in the server, the application server will reject and discard the message.

5.3.6. EXPERIMENT VALIDATION

EXPERIMENT ENVIRONMENT

The setup of bit flipping attack is shown in figure 5.12. A network server and an application server are simulated and built. In order to simplify the procedure, these 2 servers only have the basic functions. The network server has the ability to check the signature while the application server has the ability to perform decryption.

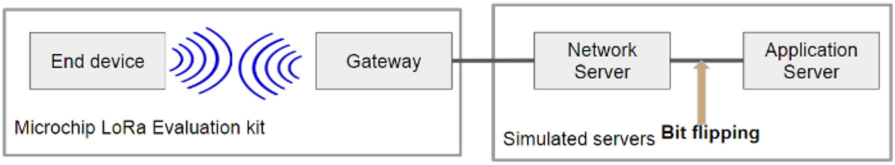
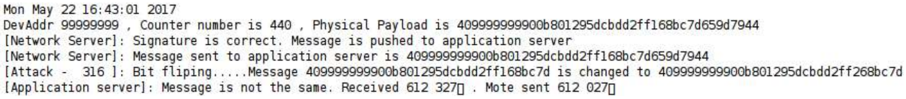


Figure 5.12: The setup of bit flipping attack

EXPERIMENT PROCEDURE

- The end device is sending messages, and the gateway is receiving and transferring these messages to the network server. The network server has the NwkSKey, and



```
Mon May 22 16:43:01 2017
DevAddr 99999999 , Counter number is 440 , Physical Payload is 409999999900b801295dcbdd2ff168bc7d659d7944
[Network Server]: Signature is correct. Message is pushed to application server
[Network Server]: Message sent to application server is 409999999900b801295dcbdd2ff168bc7d659d7944
[Attack -  316 ]: Bit fliping.....Message 409999999900b801295dcbdd2ff168bc7d is changed to 409999999900b801295dcbdd2ff268bc7d
[Application server]: Message is not the same. Received 612 327▯ . Mote sent 612 027▯
```

Figure 5.13: An example of bit flipping attack result

it will calculate the signature and check whether the messages can be accepted or not.

- An attacker conducts a man-in-the-middle attack between the network server and the application server. Based on different transmission protocols between these 2 servers, the compromise method can be different. After the compromise, the attacker has the ability to modify data between the 2 servers.
- Based on this vulnerability that between the 2 servers, the integrity of messages is not guaranteed, the attacker is able to flip a bit to change the original meaning of a message. Based on different attack goals, the attacker can perform different attacks.
  - Flip bits of FrmPayload. After flipping bits of FrmPayload, the sensor data will be modified. This attack is able to compromise the integrity of the original data.
  - Flip bits of FCnt value. The attacker can increase the counter value till it is larger than the FCnt in the server, then this message will not be accepted in the application server.
  - Flip bits of DevAddr. After modifying DevAddr, the application will consider the message to be a message from another end device.

EXPERIMENT RESULTS

An attack towards data message is performed. The result of this attack is shown in figure 5.13. After changing 1 bit, the temperature result is changed from "027" to "327".

5.4. ATTACK 4: ACK SPOOFING

5.4.1. ATTACK GOAL

This attack is designed to prove the flaw of ACK design in LoRaWAN. In order to prove that the ACK used in LoRaWAN can also be used to acknowledge other messages from the same end device, after the attack, the end device should be able to accept an ACK for an uplink confirmed message. This ACK is sent by the attacker, and it is not the ACK for this message.

5.4.2. ATTACKER CAPABILITIES

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

- having the control of the gateway.
- recognizing ACK messages and cutting of donwlink transmission of ACK from the gateway to the end device when needed.
- reading ACK messages and choosing ACK with suitable DevAddr and FCnt value.
- sending chosen ACK messages from the gateway to the end device.

5.4.3. PROTOCOL VULNERABILITY

In most of the cases, the gateway is internet-facing, making the LoRaWAN system more vulnerable. Also, building a malicious gateway is feasible. Through attacks such as UDP spoofing, a malicious gateway can be added into a LoRaWAN system. A protocol flaw is that the ACK for uplink message doesn't indicate which message it actually confirmed, it only confirms the last message it receives. So it is possible that the malicious or hacked gateway can keep the confirmation and use it for future messages.

Table 5.2 shows a physical payload example of an downlink ACK message. It can be observed that there is no FrmPayload in the message, and FCtrl "20" indicates the message is used for acknowledging. Only DevAddr shows the ACK is for end device "99999999". FCnt is a constraint. For the end device, if FCnt in the end device is larger than FCnt downlink, this ACK cannot be accepted at the end device. If FCnt in the end device is smaller than FCnt in this message, the ACK can be accepted.

5.4.4. ATTACK DESCRIPTION

There are 2 conditions for this attack:

- The ACK should satisfy the requirements of counter. Since there is a FCnt in an ACK message, in order to make the ACK to be accepted by the end device, the downlink FCnt should be larger than dnctr value in the end device. In this case, the ACK cannot just be replayed. It is also necessary to make sure the same ACK is not received by the end device before.

MHDR	DevAddr	FCtrl	FCnt	MIC
60	88889999	20	0B00	BAE1557A

Table 5.2: Physical payload format of an ACK message

- The gateway is compromised, or the gateway is malicious. Since compromising a gateway is not in the scope of this research, it is assumed that the gateway has already be compromised.

The attack setup is shown in figure 5.14. Note that the gateway is already compromised, and the attacker has the full ability to control and program the gateway. The attack can be conducted as follows:

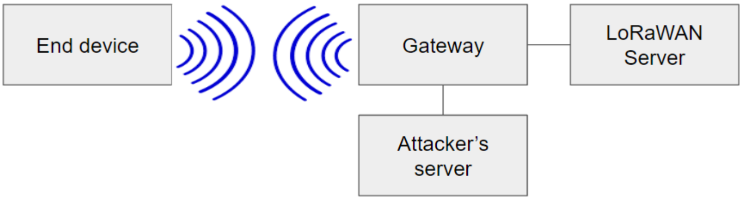


Figure 5.14: The LoRaWAN network setup for ACK spoofing

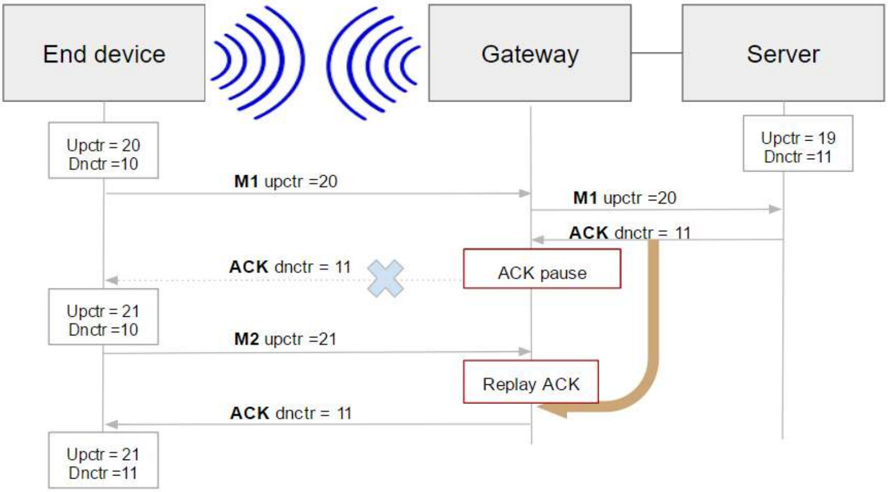


Figure 5.15: An example of ACK spoofing

- When the end device and the LoRaWAN server are communicating, the gateway should be used to transmit messages in between. When a downlink ACK *K* is received by the gateway, the attacker will control the gateway to hold the ACK *K* and pause the ACK transmission, and the end device will not receive the ACK as it supposed to. After a few times of trying, the end device will consider the uplink confirmed message to be lost.

- When the next uplink confirmed message from the same end device is received by the gateway, the attacker will control the gateway to replay the ACK message K to the end device before the gateway transmits the original message to the LoRaWAN server. Since there is no way for an end device to know whether the ACK K is for this message or messages before, the end device will accept the ACK K and consider the message to be acknowledged.

Figure 5.15 shows an example of ACK spoofing attack. At the beginning, the end device has the $upctr = 20$, $dnctr = 10$ while the LoRaWAN server has the $upctr = 19$, $dnctr = 11$. An uplink confirmed message M1 is sent by the end device to gateway and the server. When the gateway receives an ACK from gateway, instead of transmitting it downlink, the gateway will hold it and save it into its database. After a few times of retransmission, the end device will give up sending M1, and consider it to be lost or rejected. Since the end device sent an uplink message and did not get the reply, now for the end device, $upctr = 21$, $dnctr = 10$. Then, after a while the end device decide to send another uplink confirmed message M2. After the malicious gateway receives the M2, it will directly replay with the ACK with $dnctr = 11$. After checking the signature, device address and counter value, the end device will accept the ACK, and consider M2 to be confirmed and acknowledged. Now for the end device, it has the $upctr = 21$, $dnctr = 11$.

5.4.5. Generate attack scenarios

This attack is based on the assumption that the gateway is malicious, or the attacker has performed gateway spoofing. In this case, the attacker has fully control of the gateway, and it can achieve ACK spoofing.

In theory, the LoRaWAN gateway only has the function of transferring massages. If an attacker has fully control of a gateway, it can only operate physical attacks to harm the LoRaWAN network. However, with the ACK design flaw, the gateways become also a vulnerable point in a LoRa network.

5.4.6. Experiment validation

Experiment environment

Figure 5.16 shows the setup for ACK spoofing attacker. The end device and the gateway are from the Microchip LoRa evaluation kit. In order to replay the same ACK, the same ACK producing server is built. For the end device, it has the default retransmission time 7, which means for an uplink confirmed message from the end device, if it cannot receive an ACK, it will retransmit the message for at most 7 times till the ACK is received.

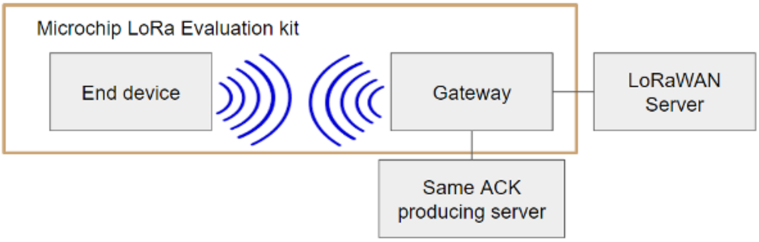


Figure 5.16: The setup of ACK spoofing attack

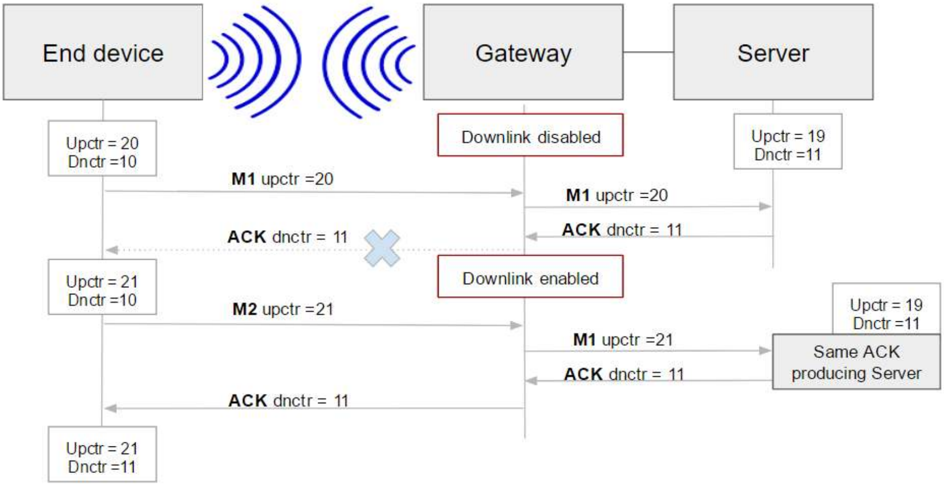


Figure 5.17: A example of ACK spoofing attack

EXPERIMENT PROCEDURE

Figure 5.17 shows the procedure of the ACK spoofing attack experiment.

- Disable the gateway donwlink transmission by setting a wrong donwlink port number. After setting, the gateway is disabled to send any downlink packets including ACK messages, but it can still receive packets from the LoRaWAN server.
- Set the end device counter value as: $upctr = 20$, $dnctr = 10$.

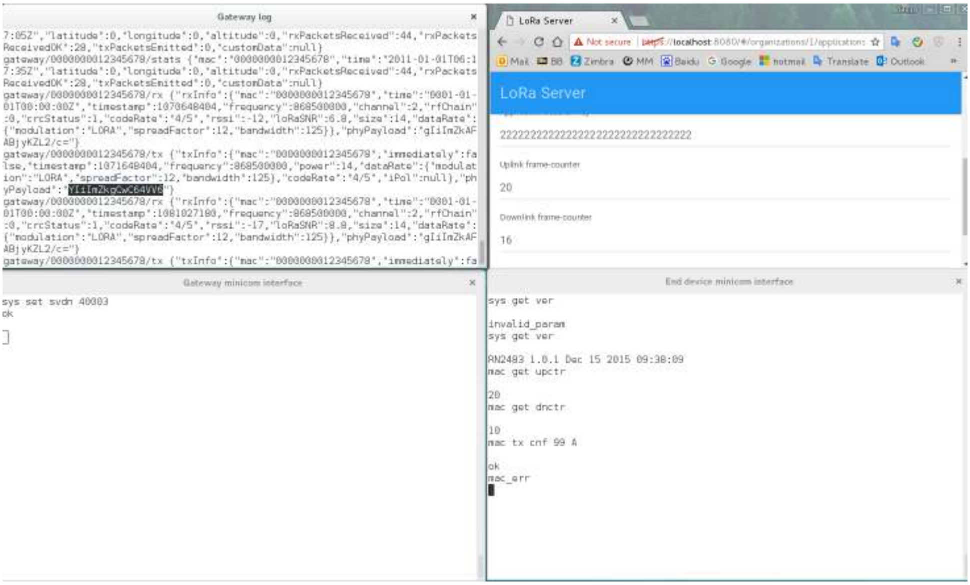
- Set the server counter value as: $upctr = 19$, $dnctr = 11$. Since the upctr value is smaller than the value in the end device, the message from end device with upctr 20 will be accepted.

- The end device sends a confirmed message M1 to gateway with $upctr = 20$. The message is accepted in the server and the server will respond an ACK with $dnctr =$ 11.

- The gateway received the ACK. Since downlink transmission is disabled, no messages are sent to the end device.

- Since there is no ACK received by the end device, the end device retransmits M1 for 7 times, and then considers M1 to be lost or rejected. For each time of retransmission, the gateway receives a new ACK from the server with dnctr from 11 to 18. After retransmission, the counter value for end device becomes $upctr = 21$, $dnctr = 10$.

- Enable the gateway donwlink transmission, and activate the same ACK producing server. In an ideal situation without packet loss, the gateway receives 7 ACKs from $dnctr = 11$ to $dnctr 18$, all these ACKs can be used in replay.


Gateway log
LoRa Server
Not secure
https://localhost:8080/#/organizations/1/applications
Mail
BB
Zimbra
MM
Baidu
Google
hotmail
Translate
Outlook
LoRa Server
Gateway minicom interface
End device minicom interface

```
7:05Z","latitude":0,"longitude":0,"altitude":0,"rxPacketsReceived":44,"rxPackets
ReceivedOK":28,"txPacketsEmitted":0,"customData":null}
gateway/0000000012345678/stats {"mac":"0000000012345678","time":"2011-01-01T06:1
7:35Z","latitude":0,"longitude":0,"altitude":0,"rxPacketsReceived":44,"rxPackets
ReceivedOK":28,"txPacketsEmitted":0,"customData":null}
gateway/0000000012345678/rx {"rxInfo":{"mac":"0000000012345678","time":"0001-01-
01T00:00:00Z","timestamp":1070648404,"frequency":868500000,"channel":2,"rfChain"
:0,"crcStatus":1,"codeRate":"4/5","rssi":-12,"loRaSNR":6.8,"size":14,"dataRate":
{"modulation":"LORA","spreadFactor":12,"bandwidth":125}},"phyPayload":"gIiImZkAF
ABjyKZL2/c="}
gateway/0000000012345678/tx {"txInfo":{"mac":"0000000012345678","immediately":fa
lse,"timestamp":1071648404,"frequency":868500000,"power":14,"dataRate":{"modulat
ion":"LORA","spreadFactor":12,"bandwidth":125},"codeRate":"4/5","iPol":null},"ph
yPayload":"YIiImZkgCwC64VV6"}
gateway/0000000012345678/rx {"rxInfo":{"mac":"0000000012345678","time":"0001-01-
01T00:00:00Z","timestamp":1081027100,"frequency":868500000,"channel":2,"rfChain"
:0,"crcStatus":1,"codeRate":"4/5","rssi":-17,"loRaSNR":8.8,"size":14,"dataRate":
{"modulation":"LORA","spreadFactor":12,"bandwidth":125}},"phyPayload":"gIiImZkAF
ABjyKZL2/c="}
gateway/0000000012345678/tx {"txInfo":{"mac":"0000000012345678","immediately":fa
```

22222222222222222222222222222222

Uplink frame-counter

20

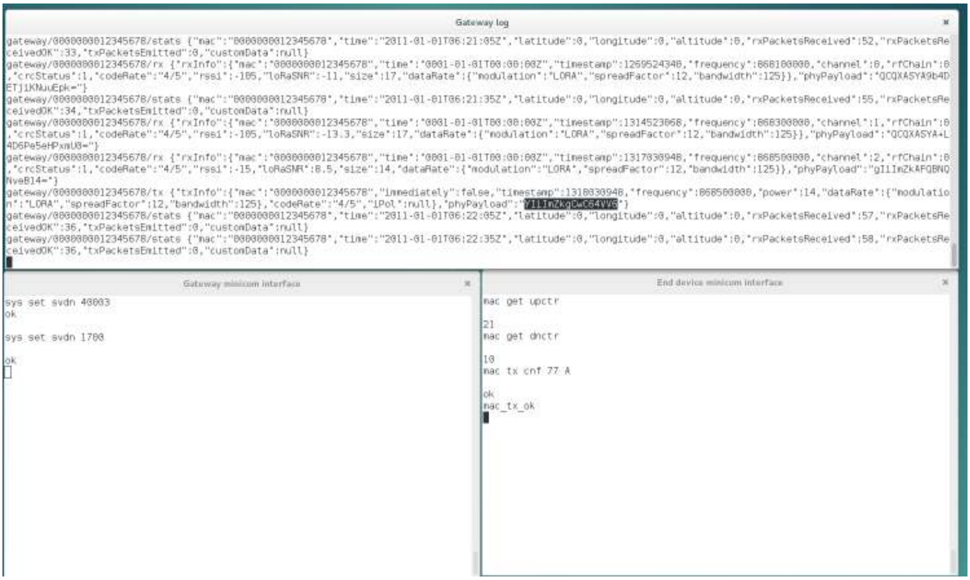
Downlink frame-counter

16

```
sys set svdn 40003
ok
```

```
sys get ver

invalid_param
sys get ver

RN2483 1.0.1 Dec 15 2015 09:38:09
mac get upctr

20
mac get dnctr

10
mac tx cnf 99 A

ok
mac_err
```

Figure 5.18: The example of ACK spoofing attack (1)



```
gateway/0000000012345678/stats {"mac":"0000000012345678","time":"2011-01-01T06:21:05Z","latitude":0,"longitude":0,"altitude":0,"rxPacketsReceived":52,"rxPacketsRe
ceivedOK":33,"txPacketsEmitted":0,"customData":null}
gateway/0000000012345678/rx {"rxInfo":{"mac":"0000000012345678","time":"0001-01-01T00:00:00Z","timestamp":1269524340,"frequency":868100000,"channel":0,"rfChain":0
,"crcStatus":1,"codeRate":"4/5","rssi":-105,"loRaSNR":-11,"size":17,"dataRate":{"modulation":"LORA","spreadFactor":12,"bandwidth":125}},"phyPayload":"QCQXA5YA9b4D
ETj1KNuuEpk="}
gateway/0000000012345678/stats {"mac":"0000000012345678","time":"2011-01-01T06:21:35Z","latitude":0,"longitude":0,"altitude":0,"rxPacketsReceived":55,"rxPacketsRe
ceivedOK":34,"txPacketsEmitted":0,"customData":null}
gateway/0000000012345678/rx {"rxInfo":{"mac":"0000000012345678","time":"0001-01-01T00:00:00Z","timestamp":1314523068,"frequency":868300000,"channel":1,"rfChain":0
,"crcStatus":1,"codeRate":"4/5","rssi":-105,"loRaSNR":-13.3,"size":17,"dataRate":{"modulation":"LORA","spreadFactor":12,"bandwidth":125}},"phyPayload":"QCQXA5YA+L
4D6Pe5eHPxmU0="}
gateway/0000000012345678/rx {"rxInfo":{"mac":"0000000012345678","time":"0001-01-01T00:00:00Z","timestamp":1317030948,"frequency":868500000,"channel":2,"rfChain":0
,"crcStatus":1,"codeRate":"4/5","rssi":-15,"loRaSNR":8.5,"size":14,"dataRate":{"modulation":"LORA","spreadFactor":12,"bandwidth":125}},"phyPayload":"gIiImZkAFQBNQ
NveB14="}
gateway/0000000012345678/tx {"txInfo":{"mac":"0000000012345678","immediately":false,"timestamp":1318030948,"frequency":868500000,"power":14,"dataRate":{"modulatio
n":"LORA","spreadFactor":12,"bandwidth":125},"codeRate":"4/5","iPol":null},"phyPayload":"YIiImZkgCwC64VV6"}
gateway/0000000012345678/stats {"mac":"0000000012345678","time":"2011-01-01T06:22:05Z","latitude":0,"longitude":0,"altitude":0,"rxPacketsReceived":57,"rxPacketsRe
ceivedOK":36,"txPacketsEmitted":0,"customData":null}
gateway/0000000012345678/stats {"mac":"0000000012345678","time":"2011-01-01T06:22:35Z","latitude":0,"longitude":0,"altitude":0,"rxPacketsReceived":58,"rxPacketsRe
ceivedOK":36,"txPacketsEmitted":0,"customData":null}
```

```
sys set svdn 40003
ok

sys set svdn 1700

ok
```

```
mac get upctr

21
mac get dnctr

10
mac tx cnf 77 A

ok
mac_tx_ok
```

Figure 5.19: The example of ACK spoofing attack (2)

- The end device sends a confirmed message M2 to gateway with $upctr = 21$, $dnctr = 10$. The same ACK producing server will reply the ACK with $dnctr = 11$. And the end device will accept the ACK.

EXPERIMENT RESULTS

Figure 5.18 and 5.19 show the result of ACK spoofing attack.

Figure 5.18 shows that the gateway downlink communication is disabled. When the end device sends a confirmed message which should be acknowledged by the server, it cannot receive the ACK, and it gets a "mac_err". However, the ACK is received by the gateway, and the physical payload of this ACK message is

"phyPayload":"YIiImZkgCwC64VV6".

Figure 5.19 shows that the attacker uses a same ACK producing server to produce the same ACK, and sends it to the end device. The ACK message is

"phyPayload":"YIiImZkgCwC64VV6"

It is the same as the previous one. This time, the end device gets a "mac tx ok", which means the message is confirmed by the attacker.

5.5. DISCUSSION OF ATTACKS TOWARD LORA CLASS B

In this research, attacks towards LoRa class A networks are mainly discussed. In class B and class C networks, there may also be protocol vulnerabilities. This section introduces flaws in LoRa class B networks.

5.5.1. BACKGROUND OF LORA CLASS B NETWORKS

LoRa class B is created to balance the power consumption and the donwlink transmission. It is an optimal solution for battery-powered LoRa end devices.

In a LoRa class B network, besides receiving windows after transmission, there are extra receiving windows at scheduled time for end devices. These windows open periodically, and are activated by beacons sent by the gateway. In order to open receiving windows at fixed times, gateways should synchronously broadcast a beacon to give time reference to end devices [5].

The beacon has the physical format as shown in table 5.3. For EU 863-870MHz ISM band, it has the BCNPayload as shown in table 5.4.

PHY	Preamble	BCNPayload

Table 5.3: Physical format of a beacon [5]

BCNPayload	NetID	Time	CRC	GwSpecific	CRC
Size (bytes)	3	4	1	7	2

Table 5.4: BCNPayload format [5]

5.5.2. PROTOCOL VULNERABILITY

The vulnerability of LoRa class B networks is that the beacons are not encrypted. Since there is no encryption, all the information that the beacon contains is in plaintext. If there is any crucial data transmitted, the attacker is able to read it. In addition, though it is claimed that CRC is used to protect the integrity of the beacon's common part (Time and NetID), CRC depends on physical layer parameters, and it can also be calculated by the attacker. If the attackers have the basic knowledge of BCNPayload, the attackers can build and send their own malicious beacon with malicious parameters, and these beacon will be received and processed by the end devices.

5.5.3. ATTACK DESCRIPTION

- Attack 1: Finding the location of a LoRa gateway

Table 5.5 shows the content of GwSpecific field in BCNPayload. InfoDesc is the parameter that indicates the kinds of gateway antennas, and Info for InfoDesc 0-2 shows the GPS coordinate od gateway's antennas.

GwSpecific	InfoDesc	Info
Size (bytes)	1	6

Table 5.5: The content of GwSpecific field [5]

When InfoDesc = 0,1 or 2, the content of the Info field are the GPS coordinates of the antenna which are broadcasting the beacon. Table 5.6 shows the content of Info field. Lat represents for latitude and Lng represents for longitude. Since there is no encryption, the beacon contains information of gateway GPS coordinate in plaintext. As long as the attacker can receive and read the beacon, the attacker is able to find the location of the gateway. If the attacker can find the gateway location, it is possible that the attacker can perform attacks such as physical attack and ACK spoofing.

Info	Lat	Lng
Size (bytes)	3	3

Table 5.6: The content of Info field [5]

- Attack 2: Beacon spoofing

  Since there is no signature in beacon messages, the integrity of the beacon is not protected. If the attacker has the ability to build his/her own beacon, the end device will receive and process it. An attacker can achieve beacon spoofing in different ways.

  - Time spoofing. If the attacker is able to build his/her own beacons, it is possible that the attacker can build a beacon with a random time value. In this case, the end device will open the beacon window based on the time value in beacon, and this will lead to the result that the window the end device opens does not fit the beacon transmission time. Another similar attack is when the attacker can send the beacon randomly. In this way, the end device will also open the receiving window at an inappropriate time.
  - GPS information spoofing. It is possible to fake the GwSpecific if the attacker is able to build malicious beacons. Since the GwSpecific is used for end devices to notice the network server when they are moving to another cell, fak-

ing the GwSpecific will cause the end devices to send wrong notices, and make the network server have wrong knowledge of the end devices' location.

- Power consuming. As long as the beacon can be accepted by the end device, the end device will open receiving windows according to the time that the beacon indicates. In this case, if the attacker is able to send multiple of beacons, the end device will open receiving windows for multiple times. In this way, huge amounts of power will be consumed. For battery-powered network, this attack will cause end devices to get disabled.

6

Attack mitigation and security suggestions

6.1. Overview

In this chapter, attack mitigation and security suggestions will be given towards 4 attacks: replay attack, eavesdropping, bit flipping and ACK spoofing. Attack mitigation sections include practical tips and notices to protect LoRaWAN networks from being attacked, while security suggestion sections give suggestions towards the protocol to manage risks and vulnerabilities.

6.2. Replay attack for ABP activated nodes

6.2.1. Attack mitigation

Replay attack for ABP activated nodes is based on the vulnerabilities of ABP activation and counter management. The requirement to achieve the replay attack is to reuse counter values for an end device with the same session keys. In order to prevent this attack from happening, in practice, these attack countermeasures can be performed to reduce the risk of being attacked:

- ABP should be used only in certain circumstances. ABP activated end-devices are using static keys, every time after resetting, the keys will remain the same, and the counter value will be reset to 0. In this case, counter values will be reused every time the end device resets. 5.1.3 describes that attacking an ABP activated end device will cost much less time than an OTAA activated end device as long as the attacker has the ability to reset end devices. Attacking ABP activated end devices is easier to achieve comparing to attacking OTAA activated end devices.

Therefore, it is recommended that ABP activation method should only be used in experimental environments or configuration processes.

However, using OTAA activation does not mean the end device is secure. It can only decrease the possibility for an end device to be compromised. This is because counter overflowing will still cause counter values reused. Using OTAA means the attacker should wait for a long time to perform the replay attack. But once the attacker gets the largest possible counter value for one end device, it can periodically replay this message, to make the end device be objected permanently. Comparing to the severe consequences of this attack, the long-time waiting is still worthwhile.

- Physical protection of the end devices in order to prevent them from being reset by malicious parties. Physically protecting the end device can reduce the risk of being replay-attacked. It is because resetting is an effective method in attacking ABP activated end devices. It can help the attacker to decrease the waiting time for getting messages with the same counter value. If the attacker cannot reset the end devices, the only way to achieve the attack is to wait for counter overflowing. However, this mitigation method is not very practical. In the future, there will be a huge amount of LoRa end devices. How to protect them physically is not a easy problem.

- To prevent the replay attack, the end device should change its session keys every time when the counter reaches its maximum value. If the end device is using OTAA method, it should go through the OTAA activation procedure again to obtain new session keys. If the end device is using ABP method, it should be re-configured, and session keys should be changed. In this case, though the counter values are reused, session keys will prevent the server from accepting malicious messages. Then, this attack will not be possible.

  This mitigation against the replay attack is effective but not practical. It is inconvenient to manually re-activate and configure an end device every time it overflows. Moreover, for end devices located in remote area, this mitigation will cost huge amount of resources since it should be operated manually. In this way, mitigation methods against this attack without improving the protocol itself are not sufficient.

6.2.2. SECURITY SUGGESTIONS

6.2.1 introduces mitigation methods against the replay attack. However, these mitigation methods are not sufficient enough to prevent replay attacks. In order to solve the problem from the beginning, it is better to improve the protocol to fix vulnerabilities.

- According to LoRaWAN specification 1.0.2, after the node activation or the reset, the frame counters on the end-device and the frame counters on the network server for that end-device are reset to 0. One way to increase the security level is to remain the counter value in the server after resetting. In this way, every time an ABP activated end device resets, its counter value will restart from 0 while the corresponding counter value in the server will not be changed. Then when the end device sends messages to the server, the server will not accept the messages until the counter value of the end device becomes larger the counter value in the server.

  This method prevents all the messages with reused counter value. In this way, resetting ABP activated end devices becomes useless for an attacker in the replay attack. The attacker can only achieve this attack by waiting for counter value overflowing.

  The disadvantage of this method is that it sacrifices the availability of the end device. When the end device resets, it will keep sending messages, but none of these messages will be accepted till the counter value reaches the value in the server. During this period, the server cannot receive messages from the end device. The worst situation is the end device resets when the counter value is very large, then the waiting time can be very long. In addition, the attacker can still attack the system by replaying reused messages from overflowing.

- Another method is to add a function to end devices. Every time it resets or the counter value reaches its maximum value, the end device should be triggered and then be able to re-activate automatically. No matter if the end device is activated by OTAA or ABP in the last session, it should use OTAA to rejoin the network. This means the end device should go through the "Join request - Join accept" procedure again. This procedure is possible to be passed automatically.

  For OTAA activated end devices, re-activation is easy. However for ABP activated end devices, in the original ABP configuration, the protocol should be improved such that AppKey should also be set and stored in both the end device and the server. This is because ABP activation method only needs session keys instead of AppKey, in the original ABP configuration, there is no AppKey assigned for both the end device and the server. If AppKey is not assigned, the end device cannot join the network using OTAA.

The advantage of this method is that it protects end devices from being replay-attacked. This is because session keys will be changed after join procedure of OTAA, then even though the attacker has the acceptable counter value, session keys are not the same, and the malicious messages from the attacker will not be accepted. Also, ABP activated end devices are re-activated using OTAA, the insecure activation method, ABP, is discarded. In this way, there are no messages with the same counter value and the same session keys. Replay attacks are prevented.

The disadvantage of this method is it increases the complexity of firmware design for end devices. However, in order to protect security of LoRaWAN network, it is a trade-off.

6.3. EAVESDROPPING

6.3.1. ATTACK MITIGATION

Chapter 5 introduces the eavesdropping attack. From the experiment results, it can be observed that using messages with the same counter value, the contents of the messages may to be decrypted. This attack is based on the protocol flaw that the cipher block mode LoRaWAN used is not secure enough. The key of performing this attack is to get message groups. The definition of a message group is messages from the same end device with the same session keys and same counter value. So, as long as the attacker gets message groups, it can make use of the protocol flaw, and use traversing method to compromise the confidentiality of the messages. In order to reduce the risk of being eavesdropped, these mitigation methods can be used in practice:

- ABP can only be used in certain circumstances. If ABP is used, and the attacker is able to reset the end device, message groups can be obtained easily. Experiment result shows that after a few resets, enough message groups are collected, then the original data can be easily decrypted. Also with the increase of the sessions, the decryption accuracy can be increased. In this case, ABP cannot be used when important data is transmitted.

  For OTAA activated end device, although resetting has no effect on performing the attack, the attacker can still wait for the counter overflow to achieve eavesdropping. Though it may take a long time, it can still be worthwhile when crucial data is transmitted.

- Prevent the end devices from being reset by attackers. If the attacker is able to reset the end device, it will take much less time to perform the attack. This is because eavesdropping attack needs counter value to be reused for several times. If the attacker is not able to reset the end device, the attacker will wait for several sessions, sometimes a few months or even a year to perform the attack. In this case, protecting the end device physically can highly reduce the risk of being eavesdropped. However, as mentioned in last section, it is not easy in a large scale network to protect the end devices from being reset. Attackers can simply cut off and recover the power to achieve the reset.

- Change the session keys periodically. Since this attack needs to collect several messages with the same counter value and session keys, changing session keys periodically can prevent the attacker from collocating enough messages. If every time the counter value in the end device reaches its maximum value, the end device re-activate, there will not be enough messages for the attacker to perform decryption. Without enough messages, the possibility of decrypting original message can be highly decreased. For OTAA activated messages, this process can be done by going through the "Join request- Join accept" procedure again. For ABP activated messages, the session keys should be changed manually.

  If the end device changes session keys every time the counter value of it reaches the maximum value, it is impossible for an attacker to perform eavesdropping anymore. However, it is not convenient. Especially for ABP activated end devices, the change of session keys needs the re-configuration on both end device side and the server side. In a large scale network or remote areas, it is impractical.

- Don’t use LoRa to send crucial data. Only data that is not afraid of being eavesdropped can be sent. Since mitigation methods against eavesdropping has their own disadvantages, the confidentiality of LoRaWAN networks cannot be guaranteed. In this case, it is better that crucial data should not be transmitted with LoRa technology. The eavesdropping attack will cause privacy issues. If it is used for crucial data, it may even cause criminal problems.

6.3.2. SECURITY SUGGESTIONS

Since mitigation methods against eavesdropping has their own drawbacks, it will be better if the protocol flaws can be fixed. In eavesdropping attack, the vulnerabilities that the attacker makes use of are:

- ABP is using static keys.
- Counters will be restarted from 0 in both the end device and the server for ABP activated devices.
- The cipher block mode in LoRaWAN is based on the assumption that counter value will not be reused.

In this way, security suggestions can be given to fix these vulnerabilities.

- The block cipher mode that LoRaWAN uses is similar to AES-CTR. Instead of using a nonce, it uses a counter value in the block. AES-CTR mode is secure based on the assumption that the nonce will never be reused. But in a LoRaWAN network, counter overflow and counter reset will both cause the counter reusing. In this case, the block cipher mode is not secure in LoRaWAN. However, the block cipher mode has its own advantages. Using counter value instead of a nonce can reduce the transmission payload length. Since counter value has the function of indicating message number, keeping both sides in sync, etc, it is more efficient to use counter value. Otherwise, the nonce needs to be known by both the end device and the server, and it needs to be transmitted over the air from the end device and the server. Then, the physical payload length will be longer, and increasing the physical payload will decrease the transmission efficiency. It is a trade-off between efficiency and security.

- Session keys can be changed automatically when the counter value of the end device reaches its maximum value or the end device resets. Details are explained in 6.2.2. The advantage of this method is that it prevents eavesdropping from happening. After changing session keys periodically, it is impossible for the attacker to collect messages with the same counter value and the same session keys, and the block cipher mode will be secure since with the same session keys of an end device, the counter value will never be reused. In this way, the counter value is functioning like a nonce.

  This method also has its disadvantages. It will require larger memory for computation and storage. It will also consume more power for an end device. In addition, it increases the complexity of the firmware design.

6.4. BIT FLIPPING ATTACK

6.4.1. ATTACK MITIGATION

The bit flipping attack is based on the vulnerability of the protocol that the integrity is not protected between the application server and the network server. In LoRaWAN, the server consists of a network server and an application server, and it is a federated server infrastructure. The network server in practice is normally established by a network operator, and because of the infrastructure, the network server is not able to eavesdrop the application data. The application server in practice normally belongs to application owners. The application server and the network server are cooperating in the process of join procedure as well as traffic control.

The key to achieve this attack is to attack the transmission method between 2 servers. In order to prevent the bit flipping attack, mitigation can be enforced to reduce the risk of been attacked.

- A secure transmission method between the network server and the application server should be chosen and used. Since the protocol gives vendors freedom to choose the transmission method between 2 servers, there are many choices, such as Ethernet, WiFi, 3G, etc. In this case, since LoRaWAN did not provide any protection method between the 2 servers, the security between the network server and the application server depends on the transmission method the vendor chooses. Therefore, the application owner should be familiar with the security of the transmission method, and be aware of potential threats.

- When the application server receives invalid messages, it is possible that the bit flipping attack is being performed. Thus, the transmission between 2 servers should be checked. Under this attack, the attacker is able to flip data bits of encrypted messages. An attacker with basic knowledge of LoRa message format is able to modify the data. However, since there is no way for an attacker to predict the decryption result after the modification, the attacker cannot modify the data into a certain result. Instead, the attacker can only modify the ciphertext in a specific location. Therefore, in the application server, after decryption, the result is not always readable.

  Based on different attack goals, the attacker can perform different attacks. When the FrmPayload is modified between 2 servers, the result can be readable. However, if the DevAddr or the FCnt is modified, the result must be unreadable. The reason is the encryption and decryption in LoRaWAN uses FCnt and DevAddr on both end device and the server. If the attacker modifies these 2 parameters, which means the end device uses original parameters for encryption, while the application server uses modified parameters for decryption, there is no way for the application server to get the same plaintext as it in the end device.

  Since the bit flipping attack often occurs with invalid or unreadable messages, these messages can be seen as an alert for bit flipping attack. If the application owner can observe these messages, it is possible that this network is compromised, and the connection between the network server and the application server should be checked.

6.4.2. SECURITY SUGGESTIONS

Considering the integrity protection, it is better if the protocol can provide end-to-end encryption. Thus the security between the application server and the network server can be independent from the transmission method. Otherwise, if the transmission method is not secure, the LoRaWAN network is not secure anymore.

One method to protect the integrity between the network server and the application server is check the MIC again when the message reaches the application server. From the LoRaWAN specification, the MIC is checked in the network server to make sure the messages are not modified before the network server. After the checking, the message is transmitted to the application server, and the application server will not check the MIC again. It is suggested for the application server to check the MIC with NwkSKey to see whether the message is changed or not between 2 servers. Before, the NwkSKey is owned by the network server, which is the network operator in practice. It will be better if the application server, which is the application owner, also has the NwkSKey, then it can calculate MIC to check the signature.

It is discussed that the application server has the NwkSKey is not appropriate, but since the calculation of NwkSKey and the AppSKey are similar, the application server does not need extra information to calculate the NwkSKey. It is acquiesced that the application owner is able to get the NwkSKey if he/she wants.

6.5. ACK SPOOFING

6.5.1. ATTACK MITIGATION

ACK spoofing is based on the protocol flaw that the ACK message does not indicate which message it confirmed. ACK messages are messages used for confirming uplink messages and indicate these uplink messages are received by the server. However, since the ACK message does not include any parameter to indicate which uplink message it actually confirms, the ACK message can be used to confirm any uplink messages as long as the counter value is suitable. The key point to achieve ACK spoofing is to control the gateway to transmit the ACK messages as the attacker wants to. In order to prevent this attack from happening, mitigation methods as follows can be performed.

- Gateway should be protected. The transmission between the gateway and the server should be protected. In a LoRaWAN, the gateway has the function of transmitting messages uplink or downlink as soon as messages arrive. It will not change or know the message contents. If an attacker wants to attack a LoRa network by attacking a LoRaWAN gateway, a common method is to physically attack the gateway, such as cutting off the power of the gateway or disrupting the functionality the gateway.

MHDR	DevAddr	FCtrl	FCnt	FPort	FrmPayload	MIC
80	88889999	00	7C00	63	C8	A64BDBF7

Table 6.1: An example of an uplink confirmed message

MHDR	DevAddr	FCtrl	FCnt	MIC
60	88889999	20	0B00	BAE1557A

Table 6.2: The format of an ACK message

However, with the protocol vulnerability of ACK messages, the attacker can make use of the ACK message flaw to achieve severe attacks. If the attacker is able to compromise a gateway and has full control of it, the ACK spoofing can be achieved. In this case, the LoRa gateway should be protected physically, and also it should be protected from being controlled by malicious parties.

The disadvantage of this mitigation is that it is not easy to check the state of gateways in a large network or in a remote location, it is better if the protocol can provide secure solutions for this problem.

- Confirmed messages should be used carefully. In a LoRa network, confirmed messages have to be acknowledged by the receiver. ACK messages are messages to show the acknowledgement. In practice, the confirmed messages can be used by the end device to check the connection between the end device and the server, or can be used for transmitting crucial messages that have to be received by the server. Since the ACK spoofing attack is able to confirm unnecessary messages and ignore confirmed messages, when using the confirmed messages, it is important to check the gateway, and make sure the acknowledgement is not crucial.

  However, in practice, it is not convenient to check the acknowledgement for end devices every time. If the end device decides to send uplink confirmed messages, it is possible that the acknowledgement it receives is not correct.

6.5.2. SECURITY SUGGESTIONS

Since the mitigation methods against the ACK spoofing attack are not effective or convenient, it is better if the protocol flaw can be fixed. In this case, security suggestions towards the flaw that ACK messages do not indicate which confirmed message it acknowledged are given.

An example of an uplink confirmed message and its corresponding ACK message are shown in tables 6.1 and 6.2.

It can be observed that for ACK messages, if there is no downlink piggyback messages, there is no frame payload and port number. The suggestion is to add the counter value of the confirmed message to the ACK message. Then the physical payload of the ACK message will become as in table 6.3.

MHDR	DevAddr	FCtrl	FCnt	FCnt (uplink)	MIC
60	88889999	20	0B00	7C00	3C67AE2A

Table 6.3: The improved format of an ACK message

The improved ACK message includes 2 bytes of FCnt from the uplink confirmed message that it acknowledges. The MIC is calculated for physical payload including uplink FCnt. For uplink transmissions, if the server receives a confirmed message, it should return an ACK with uplink FCnt bits. Then the end device will check the uplink counter value of the ACK message it receives. If the gateway is malicious, and the attacker controls the gateway to hold the ACK for the next transmission, the end device will notice that the FCnt value it receives is different from the FCnt value it transmits, then it should not accept the ACK message.

The advantage of this method is that it ties the uplink confirmed message with its corresponding ACK message by checking counter value. If counter resets and overflows are not considered, there is no way to achieve ACK spoofing. However, the disadvantage of this method is that it increases the length of ACK message. The transmission may take longer time.

6.6. ATTACKS TOWARDS LORA CLASS B NETWORKS

6.6.1. ATTACK MITIGATION

The attacks toward LoRa class B networks found in this research are based on the protocol flaw that the beacon messages are not encrypted or signed. The confidentiality and integrity of beacons can be compromised. Beacons are messages that sent from the gateway to make sure the end device and the gateway are synchronized, then in multicast, end devices can open receiving windows in synchronization. However, since the beacons are not protected, the attackers are able to read beacon contents and build their own malicious beacons. In order to prevent these attacks, mitigation methods are given.

- In order to prevent the damage of eavesdropping GPS information, LoRa gateways should be protected. Since the beacon contains the location of gateways and the information is not encrypted, attackers may be able to find the gateways and perform physical attacks towards the gateway. In this way, when using LoRa class B, gateways should be protected physically. Also, in practice, LoRa class B should be used carefully. When the GPS information of the gateway is not crucial, it is fine to use the beacon.

- In time spoofing, attackers will send beacons with wrong time value to confuse the end device. In order to prevent time spoofing, it is important that the end device should check the time value in the beacon and the time value it has. If the difference is large, it is possible that the beacon is malicious. In this case, some attacks can be avoided, and the end device will avoid open unnecessary receiving windows.

- If the attacker is able to build its own malicious beacons, he/she can send the beacon randomly to make the end device open extra receiving windows. In this way, battery-powered end devices will consume much more power, and the life time will be decreased.

6.6.2. SECURITY SUGGESTIONS

In practice, some mitigation methods are given to protect LoRa networks from being compromised. However, because of the design flaw of the beacon, the mitigation methods cannot sufficiently protect the LoRa class B network. In this case, the protocol flaw needs to be fixed as follows.

- Beacons are controlled and transmitted by the gateways in synchronization. However, the gateway does not provide security mechanism for encrypting or signing beacons. In this way, attackers are able to build their own beacons.

  One method to prevent this situation is to use NwkSKey to encrypt and sign the beacons. However, in order to achieve signing, NwkSKey should be configured to gateways, making the network more vulnerable.

  Another method is to decrease the times of transmission of GPS information. This can be achieved as follows: At the beginning of the beacon transmission, the gateway should send a beacon including GPS information to the end device to keep the end device synchronized and know the location of the gateway. Afterwards, beacons with GPS information should be sent with a larger time period than beacons without GPS information. This method can benefit end devices with fixed location. However, for moving end devices, the accuracy of the end device location will be decreased. Since it will recognize its cell according to the gateway GPS information. With lower frequency of getting GPS information, there will be latency for network server to know the location of end devices.

- In order to prevent the time spoofing or location spoofing from the attackers, it will be better if the end device can compare the current beacon information with the previous beacons. If the information gap between these 2 beacons is large, then the end device should not accept the beacon.

7

Conclusion

7.1. Summary

LoRa is a proprietary spread spectrum modulation scheme, and it has been gaining popularity in recent years. It promises long range and low battery usage, and it's aiming to power the next generation of Internet of Things applications as a worldwide standard for IoT communication. LoRaWAN is a mac layer protocol for long-rang low-power communication that based on LoRa technology. Since LoRaWAN is a new protocol, the security level of it is not fully studied and thus cannot be guaranteed in practice. In order to figure out the security of LoRa, four research questions are raised in this research.

- How secure is LoRaWAN?
- What are vulnerabilities of LoRaWAN?
- What kinds of attacks are possible toward LoRa networks?
- How to produce secure solutions for LoRaWAN?

In order to answer these questions, this research introduces the security features of LoRaWAN. First, in order to build secure transmission connection between the end device and the server, LoRaWAN uses 2 activation methods, OTAA and ABP, in the connection establishment procedure. The activation methods help to set session keys for LoRaWAN and control access of end devices to the server. Second, for OTAA, keys are generated in a secure way by transmitting nonce over the air and calculating in both end device and the server. Then keys are used on both sides in encryption and signature. Third, LoRaWAN uses AES-128 in key generation, encryption and signature, and a cipher block mode which is similar to CTR is used to protect the confidentiality of the LoRa messages. Forth, in order to prevent replay attacks, counters are used in LoRaWAN to ensure only messages satisfy requirements can be accepted. Last but not least, acknowledgements are used to confirm messages. In order to decrease packet loss, retransmission method is used.

After security features of LoRaWAN are studied, to figure out the vulnerabilities of LoRaWAN and possible attacks toward the protocol, vulnerability analysis method in this research is given to provide a systematic way to analyze the protocol. In the analyzing procedure, first, the goal of attack are introduced and the attacker's capability is defined. Then the vulnerabilities of the protocol are introduced. Based on the vulnerabilities, attacks are presented. The scenarios in practice is also described. At the end, the attacks are implemented to validate the protocol vulnerabilities.

To answer the research question of "what are the vulnerabilities and attacks in LoRaWAN", vulnerabilities and attacks based on these protocol flaws are introduced. There are 4 kinds of attacks researched in this thesis. These attacks are replay attack, eavesdropping, bit flipping and ACK spoofing.

Replay attack is based on the vulnerabilities in ABP activation method and counter management. In this attack, the attacker is able to replay messages to cause the server ignore messages from end devices. Eavesdropping is based on the vulnerability in counter management and the encryption method of LoRaWAN. With the protocol flaws, the attacker is able to decrypt LoRa messages. Bit flipping is based on the vulnerability that the integrity between network server and the application server is not protected. The attacker is able to modify messages between these 2 servers. ACK spoofing is based on the vulnerability in ACK message format. The attacker can make use of this flaw and send random ACKs to confirm messages in a LoRaWAN.

After analyzing and implementing attacks, attack mitigation methods in practice and security suggestions toward the protocol are introduced to reduce the harm of the attacks.

In conclusion, the research presents a security analysis of LoRaWAN. Protocol vulnerabilities are found and analyzed. Attacks are implementation to validate the protocol flaws. The result of the research shows that the security of LoRaWAN is not well-developed, and still needs to be improved. The result of the research can be used to increase the security level of LoRaWAN protocol, and can be used for LoRa devices to reduce the risk of being compromised.

7.2. FUTURE WORK

As alluded to previously, there is ample room for improvements in LoRaWAN security study.

For vulnerability of LoRaWAN, it is possible that there are still different kinds of attacks toward LoRaWAN protocol. For example, in this research, LoRa class B is not fully studied. Though security concerns and a few possible attacks are presented, class B attacks are not implemented to validate the vulnerability theory. This is because LoRa device development is still at a very early stage, and mass production for LoRaWAN class B devices is not launched yet. In addition, the LoRa class C communications are not considered in this research. With the development of LoRaWAN and LoRa devices, the protocol flaws in class B and class C communications should also be analyzed, and attacks should be implemented to validate the findings.

Besides LoRa class B and class C communications, another possible advancement is to consider the influence of large scale LoRaWANs. In a large scale LoRaWAN, the number of end devices and gateways will be large, and different scenarios, like overlapping, collisions and gateway capacity should be considered.

Also, physical attacks are not discussed in this thesis. Since this research is mainly focusing on the protocol itself, physical attacks such as side channel analysis are not considered. In the future, the ability of LoRa devices to fend off the side channel attacks needs more evaluation.

An additional possible enhancement can be security solutions that based on the research results. Security solutions, such as a secure LoRa platform, can be studied and given to guarantee security of LoRaWAN.

Bibliography

[1] Mohamed Abomhara and GM Kien. Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security*, 4:65–88, 2015.

[2] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, and Joan Melia. Understanding the limits of lorawan. *arXiv preprint arXiv:1607.08011*, 2016.

[3] Eric Alata, Vincent Nicomette, Mohamed Kaâniche, Marc Dacier, and Matthieu Herrb. Lessons learned from the deployment of a high-interaction honeypot. In *Dependable Computing Conference, 2006. EDCC'06. Sixth European*, pages 39–46. IEEE, 2006.

[4] LoRa Alliance. Lorawan™ specification. *LoRa Alliance*, 2015.

[5] LoRa Alliance. Online available :https://www.lora-alliance.org/. In *Civilian Conservation Corps (CCC) (2016)*, 2016.

[6] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[7] Norbert Blenn and Fernando Kuipers. Lorawan in the wild: Measurements from the things network. *arXiv preprint arXiv:1706.03086*, 2017.

[8] Iliano Cervesato. The dolev-yao intruder is the most powerful attacker. In *16th Annual Symposium on Logic in Computer Science—LICS*, volume 1, 2001.

[9] Semtech Corporation. Lora modulation basics. In *AN1200.22*, May 2015.

[10] Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. 1999.

[11] Jonathan de Carvalho Silva. Lorawan-low power wan protocol for iot.

[12] Morris Dworkin. Recommendation for block cipher modes of operation. methods and techniques. Technical report, DTIC Document, 2001.

[13] Gartner. http://www.gartner.com/newsroom/id/3165317. In *Global Wireless Submit 2016*, 2016.

[14] KPN Grootzakelijk. Lora technology.

[15] http://travisdazell.blogspot.nl/2012/11/many-time-pad-attack-crib drag.html. crib dragging.

[16] Microchip Technology Inc. Lora mote user's guide. In *DS40001808B*, 2015-2016.

[17] Microchip Technology Inc. Lora® technology gateway user's guide. In *DS40001827A*, 2016.

[18] Microchip Technology Inc. Lora® technology evaluation suite user's guide. In *DS40001847A*, 2016.

[19] Terrance R Ingoldsby. Understanding risks through attack tree analysis. *Computer Security Journal*, 20(2):33–59, 2004.

[20] Ralf Küsters and Tomasz Truderung. Using proverif to analyze protocols with diffie-hellman exponentiation. In *Computer Security Foundations Symposium, 2009. CSF'09. 22nd IEEE*, pages 157–171. IEEE, 2009.

[21] Helger Lipmaa, David Wagner, and Phillip Rogaway. Comments to nist concerning aes modes of operation: Ctr-mode encryption. 2000.

[22] George Margelis, Robert Piechocki, Dritan Kaleshi, and Paul Thomas. Low throughput networks for the iot: Lessons learned from industrial implementations. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 181–186. IEEE, 2015.

[23] Sjouke Mauw and Martijn Oostdijk. Foundations of attack trees. In *International Conference on Information Security and Cryptology*, pages 186–198. Springer, 2005.

[24] Jordy Michorius. What's mine is not yours: Lora network and privacy of data on publishing devices. 2016.

[25] Konstantin Mikhaylov, Juha Petäjäjärvi, and Tuomo Hänninen. Analysis of the capacity and scalability of the lora wide area network technology.

[26] Sarra Naoui, Mohamed Elhoucine Elhdhili, and Leila Azouz Saidane. Enhancing the security of the iot lorawan architecture. In *Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), International Conference on*, pages 1–7. IEEE, 2016.

[27] MWR Labs R. Miller. Lora security - building a secure lora solution. In *[Online]. Available: https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf*, 2016.

[28] Sadiq Ur Rehman, Iqbal Uddin Khan, Muzaffar Moiz, Sarmad Hasan, et al. Security and privacy issues in iot. *International Journal of Communication Networks and Information Security*, 8(3):147, 2016.

[29] Bruce Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

[30] John Schwartz. Us selects a new encryption technique. *New York Times*, 3, 2000.

[31] Bhupjit Singh and Bipjeet Kaur. Comparative study of internet of things infrastructures & security. In *Global Wireless Submit 2016*, 2016.

[32] NIST Computer Security Division's (CSD) Security Technology Group (STG). Block cipher modes. April 12, 2013.

[33] Michael Sutton, Adam Greene, and Pedram Amini. *Fuzzing: brute force vulnerability discovery*. Pearson Education, 2007.

[34] Joël Toussaint, Nancy El Rachkidy, and Alexandre Guitton. Performance analysis of the on-the-air activation in lorawan. In *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*, pages 1–7. IEEE, 2016.

[35] Sean Whalen, Matt Bishop, and Sophie Engle. Protocol vulnerability analysis. *Department of Computer Science, University of California, Davis, USA, Technical Report CSE-2005-04*, 2005.

[36] Andrew J Wixted, Peter Kinnaird, Hadi Larijani, Alan Tait, Ali Ahmadinia, and Niall Strachan. Evaluation of lora and lorawan for wireless sensor networks. In *SENSORS, 2016 IEEE*, pages 1–3. IEEE, 2016.

[37] Simone Zulian. Security threat analysis and countermeasures for lorawan join procedure. 2016.