Risk Analysis document – group 9 – phases 2 and 3

For this document, the group members have created a risk analysis for the secure environment that we have developed throughout phases 2 and 3 of the project.

The impact will be determined by the following factors:

Graphical user interface, text, application

Description automatically generated

Determining the impact level per event:

Table

Description automatically generated

Determining the probability for each event:

Table

Description automatically generated

Determining the risk (probability \* impact):

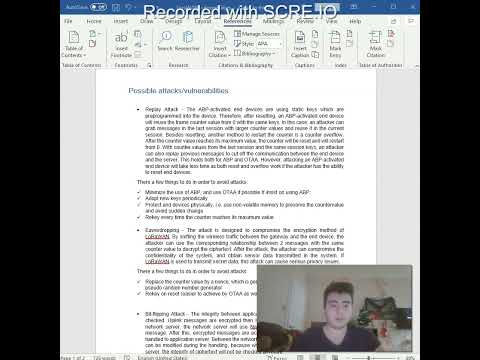
Chart, table

Description automatically generated

Text

Description automatically generatedText

Description automatically generated with low confidence

Given our lack of knowledge and experience when it comes to LoRa devices and how they work, we decided to do extensive research about it. Down below, Aleksandar explains how the research has been conducted and what it is about:[](https://www.youtube.com/embed/DUE_Y_qdMrY?feature=oembed)

The content of the research:

Possible attacks/vulnerabilities:

• Replay Attack - The ABP-activated end devices are using static keys which are preprogrammed into the device. Therefore, after resetting, an ABP-activated end device will reuse the frame counter value from 0 with the same keys. In this case, an attacker can grab messages in the last session with larger counter values and reuse it in the current session. Besides resetting, another method to restart the counter is a counter overflow. After the counter value reaches its maximum value, the counter will be reset and will restart from 0. With counter values from the last session and the same session keys, an attacker can also replay previous messages to cut off the communication between the end device and the server. This holds both for ABP and OTAA. However, attacking an ABP-activated end device will take less time as both reset and overflow work if the attacker has the ability to reset end devices.

There are a few things to do in order to avoid attacks:

* Minimize the use of ABP, and use OTAA if possible If insist on using ABP:
* Adopt new keys periodically
* Protect end devices physically, i.e., use non-volatile memory to preserve the counter value and avoid sudden change
* Rekey every time the counter reaches its maximum value

• Eavesdropping - 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.

There are a few things to do in order to avoid attacks:

* Replace the counter value by a nonce, which is generated from a cryptographically secure pseudo random number generator
* Rekey on reset (easier to achieve by OTAA as well)

• Bit-flipping Attack - 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.

There are a few things to do in order to avoid attacks:

* Run the integrity check value at the application server instead of the network server
* Repurpose protocol fields: replace the CRC by a MIC

• ACK Spoofing - 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.

There are a few things to do in order to avoid attacks:

* Recall in Bit-flipping Attack Mitigation part that we use MIC to guarantee integrity throughout the whole transmission.
* Apply MIC to both the connection to the network server and the application server

• LoRa Class B Attacks - 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 beacons will be received and processed by the end devices.

Counter measure:

* change the PHY CRC to a MIC → authenticating the beacon frames