



Research-fueled Security Services



\ WHITE PAPER \

LoRaWAN Networks Susceptible to Hacking: Common Cyber Security Problems, How to Detect and Prevent Them

Cesar Cerrudo, CTO, IOActive

Esteban Martinez Fayo, Director of Database Security, IOActive

Matias Sequeira, Security Researcher

January 2020

Abstract

LoRaWAN is fast becoming the most popular wireless, low-power WAN protocol. It is used around the world for smart cities, industrial IoT, smart homes, etc., with millions of devices already connected.

The LoRaWAN protocol is advertised as having “built-in encryption” making it “secure by default.” As a result, users are blindly trusting LoRaWAN networks and not paying attention to cyber security; however, implementation issues and weaknesses can make these networks easy to hack.

Currently, cyber security vulnerabilities in LoRaWAN networks are not well known, and there are no existing tools for testing LoRaWAN networks or for detecting cyber attacks, which makes LoRaWAN deployments an easy target for attackers.

In this paper, we describe LoRaWAN network cyber security vulnerabilities and possible cyber attacks, and provide useful techniques for detecting them with the help of our open-source tools.

Contents

Introduction	1
LoRaWAN Applications	2
Architecture	3
Security in LoRaWAN	5
Counters	5
Device Activation	5
Confidentiality in LoRaWAN	7
Integrity in LoRaWAN	7
Security Improvements in LoRaWAN v1.1	8
Cyber Security Risks and Threats	9
Reverse Engineering Devices	9
Device Tags	9
Hardcoded Keys in Open Source Code	10
Easy-to-guess Keys	10
Network Servers with Default or Weak Credentials	10
Servers with Security Vulnerabilities	10
Compromised Device Manufacturers	10
Device/Infrastructure Deployment Technicians	10
File Disclosure	11
Service Provider Breach	11
Offline Key Cracking	12
A JoinRequest Message and a JoinAccept Message or Two JoinRequest/JoinAccept Messages	12
A JoinAccept Message and a Data Message	13
Data Packets	14
Other Issues with Keys	15
Key Cracking in LoRaWAN 1.1	15
Legacy Versions	15
Compromised Keys and Cyber Attacks	16
Denial of Service to Devices and Networks	16
Sending Valid Messages	16
Regenerating Session Keys	16
Sending Valid MAC Commands	16
Sending Fake Data	17
Cyber Attack Scenarios	18

Utilities and Smart Meters	18
Smart Industry	18
Smart Cities	18
Smart Home	19
Auditing Insecure Networks and Detecting Cyber Attacks	20
Message Replay	20
Fake Messages and Denial of Service (Simultaneous Sessions)	20
ABP Devices	21
Well-known or Non-random Keys	21
Recommendations	22
Key Protection	22
Prevent, Detect, and Monitor	22

Introduction

The long range wide area networking (LoRaWAN) protocol is designed to allow low-powered devices to communicate with Internet-connected applications over long range (LoRa) wireless connections. It is a MAC layer protocol built on top of LoRa, which is the physical layer (PHY) or the wireless modulation protocol.

As previously mentioned, one of the biggest advantages of LoRaWAN is its long range capability: a single gateway (antenna) can cover an entire city or hundreds of square miles, although it heavily depends on the environment and obstructions in a given location. Furthermore, the LoRaWAN stack does not require a licensed spectrum to transmit messages but rather the opposite, making it a low-cost technology when compared to licensed spectrum solutions.

At the time of publication, LoRaWAN's latest version is 1.1, although it is expected that end device manufacturers will not fully implement this version for a couple of years. In fact, most deployed versions are 1.0.2 and 1.0.3, the ones covered in this paper. Nonetheless, most of the scenarios presented in this paper would apply to version 1.1 as well.

LoRaWAN Applications

Since there were no existing cellular low power wide area network (LPWAN) options for IoT projects, and cellular technologies were expensive to implement or did not fit specific use-cases, LoRaWAN has become one of the main, most deployed, non-cellular LPWAN solutions. There are numerous scenarios where this protocol fits perfectly; however, for simplicity, we have grouped them as follows:¹

- Smart City (i.e. parking, lighting, traffic management, metering, weather monitoring)
- Industry (i.e. asset tracking, climate control)
- Security (i.e. panic buttons, gunshot detection, flood monitoring)
- Smart Home (i.e. alarms systems, home automation)
- Smart Agriculture
- Smart Healthcare

As a growing technology, there are many important current deployments and successful use-cases for LoRaWAN around the globe. In France, smart water meters are being massively deployed, targeting three million users in just a few years.² In Brazil, they are targeting over two million LoRaWAN devices by the end of 2019.³

It is also worth mentioning that many well-known cellular carriers are reacting to LoRaWAN's growing popularity by offering LoRa nationwide coverage as a service, such as in the Netherlands (KPN⁴), France (Orange⁵), and South Korea (Telekom⁶). These companies claim that their networks offer a low price point: on average, a tenth of LTE-based services.

One final highlight is the growth of LoRaWAN community networks, such as The Things Network,⁷ where people can connect their own gateways and let others route messages through them.

Based on information from the LoRa Alliance,⁸ there are 142 countries with LoRaWAN deployments and 121 Network Operators in 58 countries, which are expanding constantly.

¹ <https://www.semtech.com/lora/lora-applications>

² https://lora-alliance.org/sites/default/files/2019-07/birdzveolia_berlin_2019_0.pdf

³ <https://www.semtech.com/company/press/semtech-supports-deployment-of-brazilian-lorawan-based-network>

⁴ <http://www.kpn.com/>

⁵ <https://www.orange.com/>

⁶ <http://www.sktelecom.com/>

⁷ <https://www.thethingsnetwork.org/>

⁸ <https://lora-alliance.org/>

At this point there are more than 100 million LoRaWAN-connected devices with projections forecasting 730 million or more by 2023.⁹

Architecture

In version 1.0.* of the protocol, there are four key elements that shape a LoRaWAN implementation.

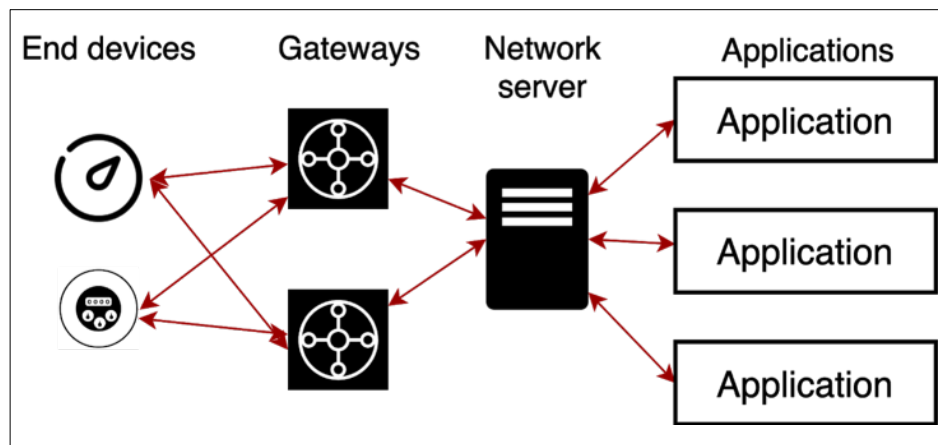


Figure 1. LoRaWAN v1.0. Architecture*

From left to right, these elements are:

- **End Devices:** The devices serve different applications and usually have sensors attached whose information is sent to the network server. These devices communicate with gateways through the LoRa protocol.
- **Gateway:** The gateway is a bridge between the LoRa wireless network and the IP stack. In other words, gateways receive broadcast messages and send data back and forth between end devices and the network server. In a LoRaWAN network, nodes/devices are not associated with a specific gateway; instead, data transmitted by a node is typically received by multiple gateways.
- **Network Server:** This is typically software that routes messages from end devices to the correct application and back. The most important function is to provide authentication and authorization of devices,¹⁰ as well as management and optimization of the network.

⁹ https://lora-alliance.org/sites/default/files/2019-07/ihsmarket_berlin_2019_0.pdf

¹⁰ In LoRaWAN v1.0.3. In version 1.1, there is a Join Server for this purpose

-
- **Application Server:** This is the destination for device application data sent as a payload in LoRaWAN messages. It is intended to be implemented by the final user, according to the user's purposes.

Putting these elements all together: devices exchange messages directly with the gateway, using the LoRa physical layer (wireless) and LoRaWAN, while the gateway exchanges messages with the network server using the TCP/IP or UDP/IP protocol, depending on the implementation. Traffic from devices to the server is called uplink, while traffic from the server to the devices is called downlink.

Security in LoRaWAN

LoRaWAN defines two layers of security: one at the network level and another at the application level. The network-level security ensures the authenticity of the node (device) in the network, providing integrity between the device and the network server. The application-layer security ensures confidentiality with end-to-end encryption between the device and the application server, preventing third parties from accessing the application data being transmitted.

To accomplish their functions, each layer makes use of a secret (in version 1.0.*), the Network Session Key (NwkSKey) and the Application Session Key (AppSKey), both 128 bits long. These security features are summarized in Figure 2. It is important to remark that data integrity between the network server and the application server is the responsibility of the service provider and not defined by the protocol.

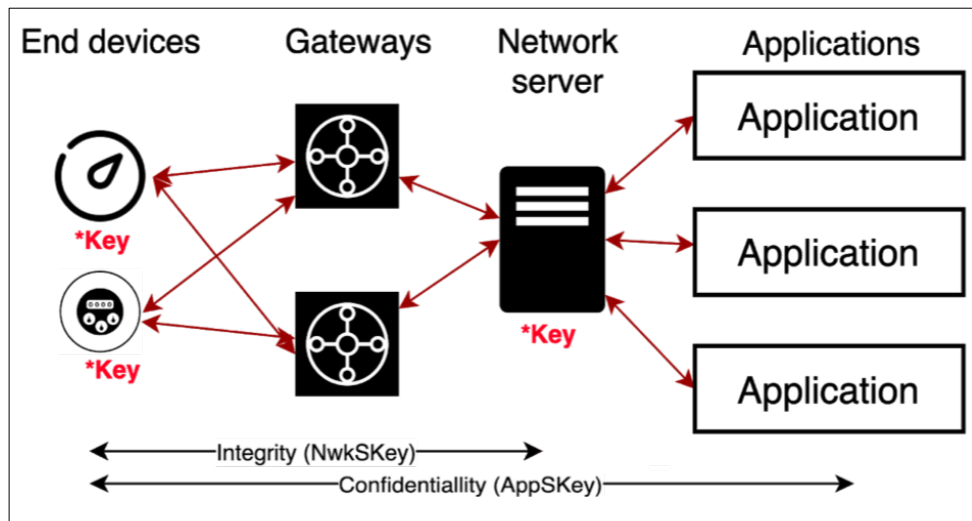


Figure 2. Session Keys and Functions in LoRaWAN v1.0.3

Counters

The protocol defines 16-bit uplink and downlink counters. One of the most important security functions of these counters is to prevent the replay of previously recorded messages (replay attacks). The protocol requires that the network server and device must both reject messages that contain a Frame Counter (FCnt) that is lower than the expected FCnt. The protocol also specifies a mechanism to acknowledge messages, although its use is not mandatory in an implementation.

Device Activation

LoRaWAN provides two methods to allow initial device activation and communication with the network server: activation-by-personalization (ABP) and over-the-air activation (OTAA). The former implies that both the session keys and other device identification information are hardcoded in the firmware and will not change throughout the device's

lifecycle; this represents a security risk that will be discussed later. The latter requires an AppKey be set, which is an AES-128 root key specific to the device, as well as the identification information in both the device and network server, which are used in the Join mechanism to derive the session keys.

As shown in Figure 3, the Join mechanism is performed with a JoinRequest message whose data is signed with the AppKey and sent by the device to the network server. The network server then replies with a JoinAccept message, which is signed and encrypted with the AppKey.

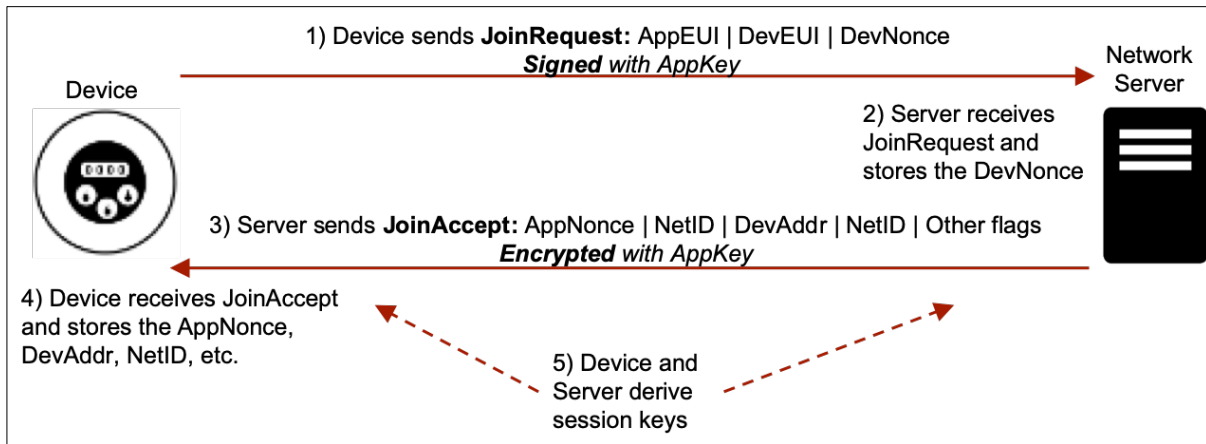


Figure 3. Session Key Generation in LoRaWAN v1.0.*

Looking at Figure 3, the most important data fields exchanged in the Join procedure are:

- In the JoinRequest, which is sent in plaintext:
 - **Application Identifier (AppEUI)**: This is a global application ID in IEEE EUI64 address space that uniquely identifies the entity able to process the JoinRequest.
 - **End Device Identifier (DevEUI)**: This is a global end device ID in IEEE EUI64 address space that uniquely identifies the end device. Nonetheless, we have seen that this uniqueness is not usually respected in implementations, and this value can easily be spoofed.
- In the JoinAccept, which is encrypted with the AppKey:
 - **End Device Address (DevAddr)**: This is a 32-bit identifier for the end device within the current network. In our experience, the DevAddr is like a session ID for devices and usually changes from session to session (after a Join procedure is performed), depending on the implementation of the network server being used. In ABP devices, this field is unchanged throughout the entire lifespan of the device.
 - **Network Identifier (NetID)**: This is a value shared by all of the devices across the same LoRaWAN network.

- **DevNonce** (JoinRequest) and **AppNonce** (JoinAccept) are random numbers/nonces used in session key generation to avoid replay attacks.

After these messages are exchanged by the device and network server, they are both able to generate session keys using the exchanged values. Figure 4 summarizes the data that must be encrypted with the AppKey in order to obtain session keys.

$\text{AES}(\text{AppKey}, 0x1 + \text{AppNonce} + \text{NetID} + \text{DevNonce}) = \text{AppSKey}$ $\text{AES}(\text{AppKey}, 0x2 + \text{AppNonce} + \text{NetID} + \text{DevNonce}) = \text{NwkSKey}$

Figure 4. Data Required for Session Key Derivation

Confidentiality in LoRaWAN

In the LoRaWAN protocol, the confidentiality of messages is achieved by encrypting only the data payload (FRMPayload in Figure 5), which is the data exchanged between devices and the server. The rest of the headers shown in Figure 5, such as the MAC header (MHDR), Frame Header (FHDR) and its data fields, and the Message Integrity Code (MIC), are sent in plaintext. As mentioned before, the secret used for confidentiality is the AppSKey.

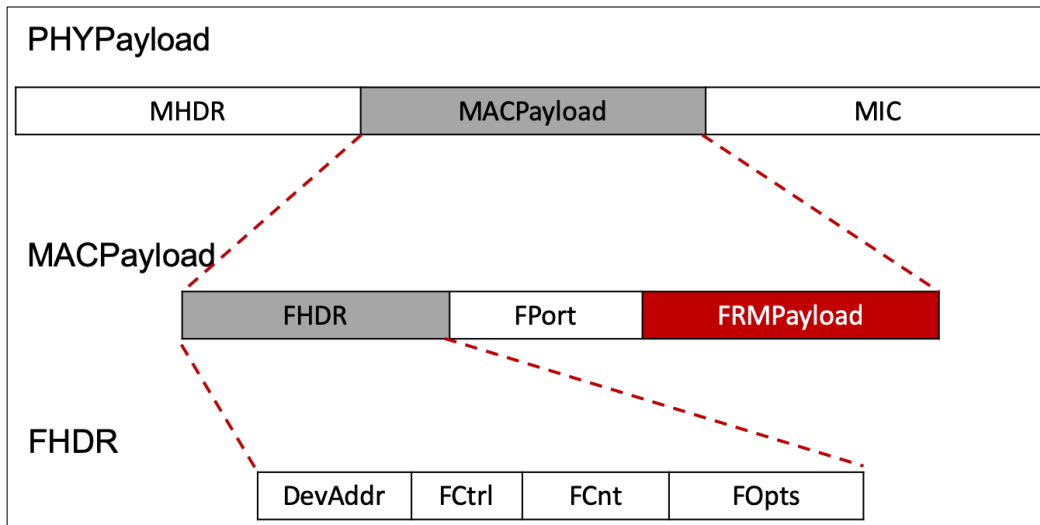


Figure 5. LoRaWAN PHYPayload Structure

Integrity in LoRaWAN

As previously mentioned, the secret used for message integrity is the NwkSKey. To protect the integrity of LoRaWAN messages, a MIC is used. The MIC depends on the entire LoRaWAN message and is appended to the end of each message. This MIC is four bytes long and is formed by the first four octets of the CMAC operation, as shown in Figure 6. Note that the CMAC operation is performed on the entire LoRaWAN message (MHDR | MACPayload) and must be performed after the encryption operation.

$$B0 = 0x49 \mid 4 * 0x00 \mid Dir \mid DevAddr \mid FCntUp \text{ or } FCntDown \mid 0x00 \mid len(MHDR \mid MACPayload)$$

$$cmac = aes128_cmac(NwkSKey, B0 \mid MHDR \mid MACPayload)$$

$$MIC = cmac[0..3]$$

Figure 6. MIC Generation

B0 is a byte array composed of a fixed pool of bytes and variable bytes, such as *Dir*, which is the direction of the LoRaWAN message (0 for uplink frames and 1 for downlink frames), the *DevAddr*, and the *FCnt*, which could be the uplink or downlink value.

Security Improvements in LoRaWAN v1.1

The newer version of the protocol brought many security enhancements, which were intended to solve some of the weaknesses discovered in previous versions.¹¹ These improvements included:

- Adding one more server to the LoRaWAN infrastructure, the Join Server. This server is in charge of deriving the session keys instead of the network server. This way, the network server never handles the AppSKey.
- Using two root keys instead of one: the AppKey and the NwkKey.
- Using five session keys instead of two (to cipher MAC commands separately, to compute the MIC in parts, to cipher the application payload, etc.).
- Implementing two independent counters for the network and application layers, and making the counters 32 bits instead of 16 bits.

Despite these important enhancements, the security posture of LoRaWAN implementations is still a matter of concern, regardless of the version. Hence, most of the cyber security risks and threats presented in this paper are still valid in the newer version of the protocol.

¹¹ Security Vulnerabilities in LoRaWAN. Xueying Yang; Evgenios Karampatzakis; Christian Doerr; Fernando Kuipers: <https://ieeexplore.ieee.org/author/37086385138>

Cyber Security Risks and Threats

Since LoRaWAN is advertised as a secure protocol, users and developers have embraced it, relying on the protocol's constant security revisions and well-designed mechanisms to transmit data and generate session keys in a secure manner. While it provides tangible benefits, such as reduced cost, easy installation and maintenance, and long-range connectivity, LoRaWAN has known weaknesses¹² and comes with great risks. Our intention is to spur discussion about the security of LoRaWAN implementations. A malicious hacker with the proper equipment and knowledge can capture LoRaWAN wireless traffic in order to analyze it and perform cyber attacks from miles away. This is why it is very important to ensure that LoRaWAN networks are well secured. In order to do this, we must be able to identify cyber security issues and detect possible cyber attacks.

Common problems that face LoRaWAN implementations are related to the keys and their management. Once the keys are compromised, the LoRaWAN network becomes vulnerable, as the keys are the source of the network's only security mechanism, encryption. After reviewing vendor documentation, one may quickly realize that it is not difficult to obtain credentials for devices that are physically accessible.

There are many methods to obtain keys, as described in the following sections.

Reverse Engineering Devices

Keys can be extracted from devices. It is possible to sniff or spoof the communication between the microcontroller unit (MCU) and the LoRa radio module, which is over an SPI or UART interface.¹³ Moreover, it is possible to copy or clone the device's firmware if flash security was not enabled. Firmware is also available online or can be obtained from the vendor in some way. It is worth noting that the use of a Secure Element (SE) does not guarantee that devices will not be reverse engineered.

Device Tags

Many devices come with a tag displaying a QR code and/or text with the device's DevEUI, AppKey, and more, which is intended to be used in the commissioning process. If these tags are not removed before placing the device in its final location (and the values were not changed when commissioning the device), an attacker with physical access to a device can use the information on this tag to generate valid session keys.

¹² Security Vulnerabilities in LoRaWAN. Xueying Yang; Evgenios Karampatzakis; Christian Doerr; Fernando Kuipers: <https://ieeexplore.ieee.org/author/37086385138>

¹³ <https://core.ac.uk/download/pdf/84932416.pdf>

Hardcoded Keys in Open Source Code

By performing a bit of open source intelligence, an attacker can obtain source code from open source repositories or vendors websites. Most source code includes hardcoded AppKeys (OTAA devices) and AppSKeys/NwkSKeys (ABP devices), which are meant to be replaced before deploying the device. Unfortunately, these keys are not always replaced, and devices are deployed with these hardcoded keys.

Easy-to-guess Keys

In order to simplify the commissioning of a device in the network, AppKeys without sufficient randomness, such as repeated characters or with incremental values, are used. If an attacker obtains a single device's AppKey by guessing the logic used to generate AppKeys or by brute-force, the attacker might gain access to the entire LoRaWAN network. Also, some manufacturers set easy-to-guess keys in devices, such as $\text{AppKey} = \text{DevEUI} + \text{AppEUI}$ or $\text{AppKey} = \text{AppEUI} + \text{DevEUI}$, while others use the same AppEUI and AppKey or the same AppEUI for all devices (DevEUI and AppEUI are values transmitted in cleartext in LoRaWAN messages).

Network Servers with Default or Weak Credentials

A quick search on Shodan¹⁴ for well-known LoRaWAN server web headers, results in numerous Internet-facing servers. Many of these servers use default credentials, such as admin/admin, or weak credentials that are easy to guess. Once an attacker logs in, the keys can be obtained/stolen from these servers.

Servers with Security Vulnerabilities

The software installed on the network servers is not immune to security issues. Hackers can harness a poorly secured server or exploit a software vulnerability to gain access to the LoRaWAN network management and thus, to devices' AppKeys.

Compromised Device Manufacturers

Most of the time, manufacturers are in charge of installing the firmware on devices as well as setting the keys. Therefore, if a manufacturer's system is compromised, attackers can compromise the keys of thousands of devices that are or will be used in different LoRaWAN networks around the world.

Device/Infrastructure Deployment Technicians

Technicians usually configure devices with the help of computers, smartphone apps, special equipment, etc. The keys used during this process can remain on the computers,

¹⁴ <http://shodan.io/>

phones, or other equipment used by the technician and be exposed to possible cyber attacks.

File Disclosure

Device manufacturers usually store the keys in files and share them with their clients via email, flash storage, online, etc. These files are handled by several people, and an unauthorized party who gains access to these files can access all of the keys.

Service Provider Breach

Service providers usually offer the LoRaWAN infrastructure that routes LoRaWAN messages back and forth between devices and applications. This infrastructure includes both the gateways and network servers that manage the LoRaWAN network; these servers require the devices' AppKeys in order to join the network. Since these keys could be stored in backups, databases, etc., a data breach of a service provider could divulge all of the keys used by their clients.

Offline Key Cracking

Attackers can attempt to crack the AppKey by performing dictionary or brute-force attacks. Cracking an AppKey requires a pair of messages, which can be any of the following combinations:

A JoinRequest Message and a JoinAccept Message or Two JoinRequest/JoinAccept Messages

The process to crack an AppKey using a JoinAccept message and a JoinRequest message is summarized in Figure 7.

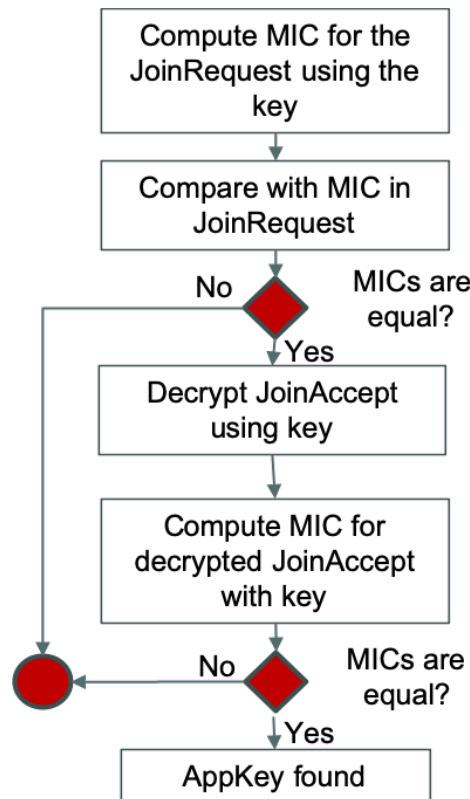


Figure 7. AppKey Cracking with JoinRequest and JoinAccept

The first step is to compute the MIC of the JoinRequest message using the key being tested. After this, the computed MIC is compared to the MIC of the original message, and if they match, the tested key is a potential AppKey. Note that we cannot be sure that this key is the actual AppKey because the MICs could have collided (remember that the MIC is a hash which is truncated to its first four bytes).

Now is when the JoinAccept message comes into play. It is used to double check the potential key. To do this, the JoinAccept message must be decrypted using the key being tested and the MIC computed. If the generated MIC matches the one from the original message, there is a high probability that we have the correct AppKey.

To crack an AppKey using two JoinRequest messages, the first step described above should be performed twice (the first time to find a possible AppKey with the first JoinRequest and the second time to double check the potential key with the second JoinRequest). The same goes for two JoinAccept messages, except the second step should be performed twice, each time using a different JoinAccept message.

A JoinAccept Message and a Data Message

Figure 8 provides a graphic summary of the process for identifying the AppKey with a JoinAccept message and a data message.

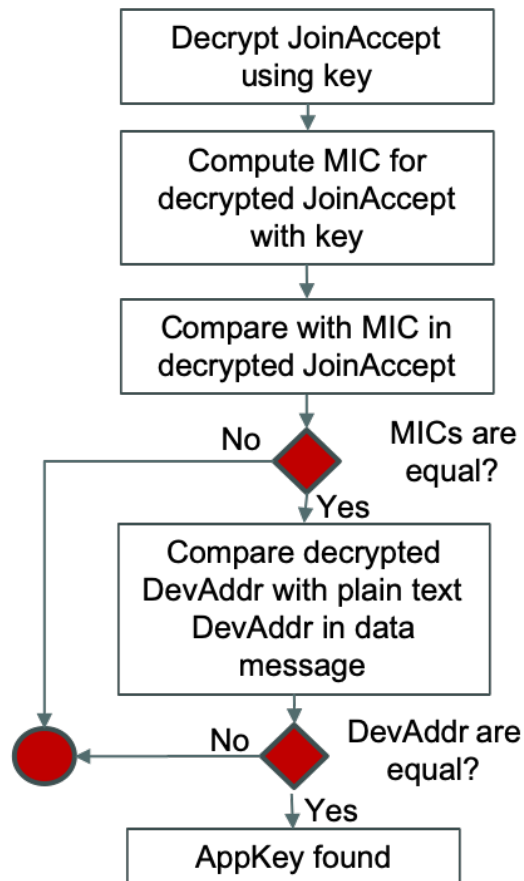


Figure 8. AppKey Cracking with JoinAccept and Data Message

The JoinAccept message is used to determine if a key can be considered a potential AppKey. In order to do so, the JoinAccept message must be decrypted with the key being tested and the MIC must be computed. If the computed MIC matches the decrypted MIC, the AppKey is now a potential AppKey. Finally, if the decrypted DevAddr from the JoinAccept message matches the plaintext DevAddr in the data packet of the same device, the AppKey was indeed correct.

Data Packets

Figure 9 illustrates the process for validating whether a device is using an AppKey and NetID by brute-forcing the DevNonce and AppNonce.

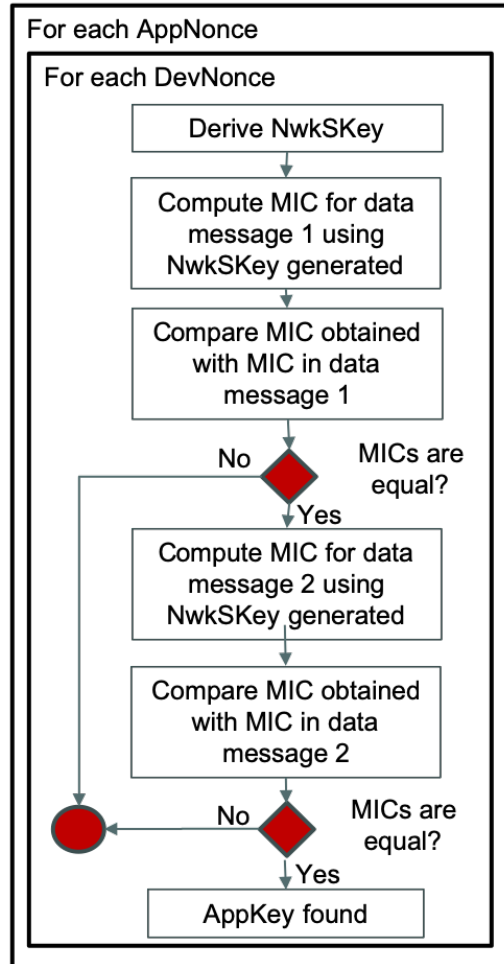


Figure 9. AppKey Cracking with Data Packets

This method is not the most efficient way to crack an AppKey, as we need to brute-force the AppKey, DevNonce, AppNonce, and NetID; however, it is suitable for testing whether a device is using an AppKey and NetID that were identified using one of the previous techniques.

Given an AppKey/NetID pair, only the DevNonce (two bytes) and the AppNonce (three bytes), a total of 40 bits, would have to be brute-forced, which is feasible in a reasonable amount of time. For each combination of DevNonce and AppNonce, the NwkSKey is generated. Using this NwkSKey, the MIC is computed and compared to the MIC from the data message. If the MICs match, the AppKey/NetID pair plus the DevNonce and AppNonce could be the correct values. To make sure these values are indeed the correct ones, another data packet should be checked using this same process.

Other Issues with Keys

Besides the possibility of cracking the AppKey, we found that:

- In many implementations, the same keys are used for a group of devices. This allows an attacker to control, spoof, or perform a DoS to many devices, or even to a whole LoRaWAN network, by cracking or guessing the key for a single device.
- As mentioned before, keys from open source repositories or keys provided by vendors/manufacturers are often not replaced at deployment. We have compiled and uploaded a dictionary of public AppKeys¹⁵ that can be used to assess the strength of an AppKey on a first approach.
- Making things worse, the keys for some devices cannot be changed. If a key is compromised, the device cannot be protected by changing its key, and it remains vulnerable.

Key Cracking in LoRaWAN 1.1

Although the security of the LoRaWAN protocol was improved in version 1.1, and one more root key (NwkKey) was added, it is possible to crack this new key using similar techniques to those used to find the AppKey in previous versions. If attackers manage to crack this key, at a minimum, a DoS could be performed against the LoRaWAN network. While it is a little bit more difficult to crack the AppKey in this version, it is still possible.

Legacy Versions

Another problem related to LoRaWAN is security revisions. Security revisions are very valuable, they drive the protocol to a secure state despite the associated complexity. Nonetheless, each time a revision is released, questions arise about what should become of deployed devices that cannot be upgraded and how well-known vulnerabilities are handled.

Based on our research, we concluded that, in most cases, vulnerabilities are not fixed as updates cannot be implemented. For example, it is usually not possible to migrate LoRaWAN 1.0.3 devices to version 1.1 due to hardware limitations. This implies that 1.0.3 devices will not receive any substantial security updates, making additional solutions necessary to secure the LoRaWAN implementation.

¹⁵ <https://github.com/IOActive/laf/blob/master/auditing/analyzers/bruteforcer/keys.txt>

Compromised Keys and Cyber Attacks

After presenting the methods that an attacker can employ to compromise the AppKey of a device, it is worth mentioning that it is practically impossible to detect the exact moment a key is compromised. There are ways to detect anomalies that may be caused by an attacker using a compromised key, which we will discuss later.

Considering how easily an attacker can compromise device keys, it is important to be aware of all of the possible attacks that can be launched against LoRaWAN networks.

Denial of Service to Devices and Networks

There are many ways an attacker can cause a DoS. It will depend on the robustness of the targeted device and its network server.

Sending Valid Messages

Due to the protocol's specification, the server will not accept a message with a smaller FCnt than the last message it received. Thus, an attacker with valid session keys could cause a DoS by sending an uplink data message with a FCnt value greater than a real device would send. Messages sent by the targeted (real) device with a lower FCnt value than the message sent by the attacker will be discarded by the network server until the device's messages surpass this counter. If the attacker keeps sending messages with larger counters, then the targeted device messages will continue to be discarded by the network server.

Regenerating Session Keys

An attacker could craft and send a valid JoinRequest, impersonating a device's DevEUI, and the network server will respond with a JoinAccept. After this, the attacker can generate new session keys and send an uplink data message to the network server, which will activate the new session on the network server and invalidate the old one. As the impersonated device will not be listening for a JoinAccept, it will not regenerate session keys; its messages will not be accepted by the network server because it will continue using the previous keys, which are no longer valid.

Sending Valid MAC Commands

As specified by the protocol, MAC commands are for network administration and are primarily used for radio frequency (RF) synchronization, such as channel and timing settings. These commands could be sent in plaintext in the FOpts field or in the FRMPayload, which is encrypted. An attacker with a valid NwkSKey (required to generate a valid MIC) could request the network server change the RF settings, thus desynchronizing the targeted devices (which would not have received the request to change RF settings). This attack could also be performed in the opposite direction, as the attacker could impersonate the server and send commands to the device and desynchronize the connection.

Sending Fake Data

This is the worst-case scenario. If attackers obtain the keys for a device or a group of devices, they can send fake data to the LoRaWAN network, affecting the applications using the data.

Imagine a LoRaWAN device measuring the pressure of a critical gas pipeline, which needs to be under constant monitoring. An attacker with valid session keys could craft and send LoRaWAN messages with normal behavior data for the pipeline pressure, masking any anomaly and hiding a physical attack against this pipeline. If not caught in time, such an attack could lead to an environmental, economic, or, in a worst-case scenario, lethal disaster.

Cyber Attack Scenarios

It can be difficult to define the impact of cyber attacks against LoRaWAN networks in terms of monetary losses, business impact, and how they affect people's lives. We present the following examples to help properly assess the real risks.

Utilities and Smart Meters

This scenario assumes a LoRaWAN network for a utility with several thousand LoRaWAN smart meters. The LoRaWAN network infrastructure is supplied by a third-party service provider and allows the utility to remotely manage the smart meters and collect user-consumption data. Attacks against this network could have several consequences.

A DoS attack will prevent the utility from billing end users, as it will not be able to access consumption readings. The utility will be impacted financially, because they cannot use the smart meters. Furthermore, the utility could blame the third-party service provider and potentially cause financial problems for the provider. This kind of attack could be persistent, causing increasing monetary losses as the LoRaWAN network becomes useless and the utility has to switch back to manual readings. There could be other unexpected consequences affecting regular service and the end users.

Smart Industry

Some industries rely on sensors to monitor the proper functioning of their facilities and to automate tasks; they use sensors such as CO₂ (fire), temperature, pressure, leakage, etc. In these scenarios, if the LoRaWAN network is hacked, sensor data could be faked by attackers, which, depending on how the data is being used, could cause serious issues.

For instance, if a pressure sensor is monitoring pipes, tanks, or containers with dangerous chemicals, inflammable liquids, etc., then fake data could result in unexpected actions. If the fake data indicates low pressure, the system could try to adjust and raise the pressure, which could cause a pipe to break or even explode, affecting the industrial plant and the employees.

Smart Cities

As pointed out in our previous research, *An Emerging US (and World) Threat: Cities Wide Open to Cyber Attacks*,¹⁶ smart city technology is vulnerable and can be hacked. Smart cities are a common use case for LoRaWAN networks and could be attacked by exploiting the LoRaWAN security issues highlighted in this paper.

Some of the services that could be hacked in a smart city's LoRaWAN network are smart street lighting, smart waste management, gunshot detection, flood and seismic monitors,

¹⁶ https://ioactive.com/pdfs/IOActive_HackingCitiesPaper_CesarCerrudo.pdf

public transportation signs, etc. Attackers targeting a smart city could bring down any service related to the LoRaWAN network and have a significant impact on the city's population.

Smart Home

Home automation is a growing use case for LoRaWAN, with applications such as smart lights, alarm and security systems, smart locks, smoke detectors, smart irrigation, pet trackers, smart windows shades, etc. An attacker who successfully hacks a smart home's LoRaWAN network could disable the alarm system, unlock doors, or remotely monitor the house. All of the benefits promised by smart home technology could be turned against the user and become dangerous if the LoRaWAN network is compromised.

The attack scenarios for LoRaWAN networks are nearly endless, these are just a small sample. Each attack scenario will have a different effect, some more dangerous and some less, but each with a real impact on organizations, businesses, and people.

Auditing Insecure Networks and Detecting Cyber Attacks

Currently, there is no way to know if a LoRaWAN network is under attack, if it has already been hacked, or if the keys are weak and easy to guess. Basically, there are no resources for protecting LoRaWAN networks.

Although it is not possible to detect the exact moment when a device's AppKey is compromised, there are checks that can be used to infer that a LoRaWAN network is the target of a malicious hacker, already compromised, or otherwise malfunctioning. This section describes some proposed controls, which are intended to act as a passive security layer (for detection purposes only) in a LoRaWAN network.

To assist with security testing/auditing of LoRaWAN networks and to help detect cyber attacks, IOActive has created the LoRaWAN Auditing Framework (LAF) with several tools. How to use these tools is outside of the scope of this paper, but the reader can review the tools,¹⁷ identify where the previously described checks are implemented, and see them in action by using a LoRaWAN gateway or connecting to a well-known LoRaWAN network server implementation.

Message Replay

With the help of a simple database, it is possible to check if a message was already received. The best approach for doing this is by checking the message's MIC. In case messages share an identical MIC, the rest of the fields, such as the Message Type (MType), DevAddr, and DevEUI, can be checked. This approach can quickly detect possible duplicate messages, instead of having to check field-by-field, which is not reasonable given a large number of messages.

An alternative method for detecting JoinRequest message replays, is to look for duplicate DevNonces for the same device, which should be random enough to not collide very often.

Fake Messages and Denial of Service (Simultaneous Sessions)

Once they have the AppKey, attackers can generate session keys and inject fake data in the server with one condition: the FCnt of the message must be higher than the FCnt of the last message received by the server.

If the spoofed device keeps sending messages (functioning normally), the server would start to discard valid messages, since they would have a smaller FCnt. Hence, when the LoRaWAN server receives messages with a smaller FCnt value than expected, it is

¹⁷ <https://github.com/IOActive/laf>

possible to infer that a parallel session was established for that device and someone is conducting a DoS attack and/or sending fake messages.

ABP Devices

As a good security practice, implementing ABP devices is discouraged, since session keys stay the same for the life of the device. ABP devices are prone to attacks, such as replay or eavesdropping. Fortunately, it is possible to determine with a high degree of accuracy which devices are ABP-activated by keeping track of the resets made by the devices (when FCnt goes back to zero) and the absence of the Join process. In other words, when a device's FCnt resets to zero, and no previous Join process was detected, it can be inferred that the device is ABP-activated. This could allow a security team to identify ABP-activated devices and flag them to be replaced.

Well-known or Non-random Keys

As stated in the previous section, easy-to-guess keys are a real problem. This means that attempting to crack an AppKey, or many of them, can reveal insecure networks. This can be done using any of the methods presented in the previous section of this paper with the help of IOActive's LAF open-source tools.

Recommendations

Key Protection

Device keys should be properly protected, since once a key is compromised, the LoRaWAN network can be hacked.

Simple best practices include:

1. Replace the keys provided by vendors with random keys.
2. Use different keys for different devices.
3. Audit (crack) the root keys used in order to detect weak keys.
4. Make sure service providers follow security best practices and have a secure infrastructure.

It is very important to remark that devices should have a unique and random AppKey, and the key should not be shared with any other device in a LoRaWAN network. Keys should be regularly audited to detect weak keys, as LoRaWAN network devices can be deployed constantly and new devices added with weak keys.

Even when following best practices, keys can still be compromised via the techniques described in this paper. Therefore for sensitive deployments, such as smart metering, we recommend using a SE on the devices and a Hardware Security Module (HSM) on the infrastructure. This way keys are never exposed, as they are stored in secure hardware and cannot be read.

Prevent, Detect, and Monitor

The best approach to preventing attacks is holistic, where the complete LoRaWAN ecosystem is secured. This can only be achieved if all of the technology that is part of the ecosystem (devices, gateways, network servers, join servers, application servers, and applications) is properly security audited. This way, possible security problems are identified and fixed. This should be done at least twice a year, as the ecosystem is not static. LoRaWAN networks are very dynamic with new components being added regularly.

As is the case with any other network, LoRaWAN networks are exposed to constant cyber attacks and can be hacked. Sometimes the best defenses fail, and prevention is not enough. Then, it is important to constantly monitor LoRaWAN networks in order to detect and react to attacks. This could be done using third-party software that can monitor LoRaWAN traffic, analyze it, and detect possible security problems while providing ways to prevent, stop, or mitigate attacks.

The LAF open-source tools referenced earlier can be used for monitoring, detecting, and preventing cyber attacks.

About Cesar Cerrudo

Cesar is a professional hacker, cyber security futurist and entrepreneur. Cesar Cerrudo is Chief Technology Officer for IOActive Labs, where he leads the team in producing ongoing, cutting-edge research in areas including Industrial Control Systems/SCADA, Smart Cities, the Internet of Things, Robots, Blockchain, and software and mobile device security. Cesar is a world-renowned cyber security researcher with more than 15 years of experience. Throughout his career, Cesar is credited with discovering and helping to eliminate dozens of vulnerabilities in leading applications, including Microsoft SQL Server, Oracle Database Server, Microsoft Windows, and Twitter, to name a few. He has a record of finding more than 50 vulnerabilities in Microsoft products including 20 in Microsoft Windows operating systems.

Based on his unique research, Cesar has authored whitepapers about cyber security problems, attacks and exploitation techniques in different widely used technology. He has presented at a variety of company events and conferences around the world, including Microsoft, Intel, Black Hat, Bellua, CanSecWest, EuSecWest, WebSec, HITB, Microsoft BlueHat, EkoParty, FRHACK, H2HC, Infiltrate, 8.8, Hackito Ergo Sum, NcN, Segurinfo, RSA, and DEF CON.

He started Securing Smart Cities (<http://www.securingsmartcities.org>), a nonprofit initiative to make cities around the world safer, after he found that most Smart City technologies are vulnerable to cyber attacks.

Cesar collaborates with and is regularly quoted in print and online publications. His research has been covered by Wired, Bloomberg Businessweek, TIME, The Guardian, CNN, NBC, BBC, Fox News, The New York Times, New Scientist, Washington Post, Financial Times, The Wall Street Journal, and so on.

About Esteban Martinez Fayo

Esteban is the Director of Database Security at IOActive with more than 15 years of experience in the information security field. He has discovered and helped to fix multiple security vulnerabilities in key enterprise software from major vendors like Oracle, Microsoft, and IBM. As part of his research, he has developed and presented novel database attack techniques at international conferences such as Black Hat, DEF CON, EkoParty, WebSec, and NcN. Throughout his career, Esteban has performed dozens of penetration tests and provided security advice for companies across a variety of industries.

About Matías Sequeira

Matías started his career in the cyber security field as an information security consultant, where he worked for clients in the financial and medical software fields. Later, he started to research ransomware and defense measures against it as part of the AntiRansomware Team. Currently, his research interests focus on IoT security.

About IOActive

IOActive is a comprehensive, high-end information security services firm with a long and established pedigree in delivering elite security services to its customers. Our world-renowned consulting and research teams deliver a portfolio of specialist security services ranging from penetration testing and application code assessment through to semiconductor reverse engineering. Global 500 companies across every industry continue to trust IOActive with their most critical and sensitive security issues. Founded in 1998, IOActive is headquartered in Seattle, USA, with global operations through the Americas, EMEA and Asia Pac regions. Visit <https://ioactive.com/> for more information. Read the IOActive Labs Research Blog: <http://blog.ioactive.com>. Follow IOActive on Twitter: <http://twitter.com/ioactive>.