applied sciences

MDPI

Article

Testbed for LoRaWAN Security: Design and Validation through Man-in-the-Middle Attacks Study

Ondrej Pospisil [1,†], Radek Fujdiak [1,*,†], Konstantin Mikhaylov [2,†], Henri Ruotsalainen [3,†] and Jiri Misurec [1,†]

[1] Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technicka 12, 61600 Brno, Czech Republic; xpospi89@vut.cz (O.P.); misurec@vut.cz (J.M.)
[2] Centre for Wireless Communications, University of Oulu, Erkki Koiso-Kanttilan katu 3, 90014 Oulu, Finland; konstantin.mikhaylov@oulu.fi
[3] Institute of IT Security Research, St. Pölten University of Applied Sciences, Campus-Platz 1, 3100 St. Pölten, Austria; henri.ruotsalainen@fhstp.ac.at
* Correspondence: fujdiak@vut.cz; Tel.: +420-541146955
† These authors contributed equally to this work.

Abstract: The low-power wide-area (LPWA) technologies, which enable cost and energy-efficient wireless connectivity for massive deployments of autonomous machines, have enabled and boosted the development of many new Internet of things (IoT) applications; however, the security of LPWA technologies in general, and specifically those operating in the license-free frequency bands, have received somewhat limited attention so far. This paper focuses specifically on the security and privacy aspects of one of the most popular license-free-band LPWA technologies, which is named LoRaWAN. The paper's key contributions are the details of the design and experimental validation of a security-focused testbed, based on the combination of software-defined radio (SDR) and GNU Radio software with a standalone LoRaWAN transceiver. By implementing the two practical man-in-the-middle attacks (i.e., the replay and bit-flipping attacks through intercepting the over-the-air activation procedure by an external to the network attacker device), we demonstrate that the developed testbed enables practical experiments for on-air security in real-life conditions. This makes the designed testbed perspective for validating the novel security solutions and approaches and draws attention to some of the relevant security challenges extant in LoRaWAN.

Keywords: LoRa; LoRaWAN; security; encryption; testbed; SDR; IoT; LPWAN

Citation: Pospisil, O.; Fujdiak, R.; Mikhaylov, K.; Ruotsalainen, H.; Misurec, J. Testbed for LoRaWAN Security: Design and Validation through Man-in-the-Middle Attacks Study. *Appl. Sci.* **2021**, *11*, 7642. https://doi.org/10.3390/app11167642

Academic Editor: Dan García Carrillo

Received: 28 June 2021
Accepted: 17 August 2021
Published: 20 August 2021

1. Introduction

The Internet of things (IoT) has gained momentum in the past few years, resulting in many devices taking their place all around us, thus opening the road for many versatile applications across different verticals. The low-power wide-area network (LPWAN) technologies, which is an umbrella term for the radio access technologies (RATs) characterized by low energy consumption, broad coverage and good scalability, are considered among the critical enablers for the massive machine type connectivity (mMTC). For these reasons, LPWANs today are actively being rolled out commercially all around the globe [1]. Among the LPWAN RATs available today, the LoRaWAN technology is among the most widely spread non-3GPP technologies (about 191 million LoRa end node devices [2]) both as a part of public and private networks; however, due to the fast pace of the development and commercialization of these technologies and the design compromises required to reduce the cost and energy consumption of the devices, the LPWAN technologies have some shortcomings. One of them is related to the security and privacy of data transfers [3–5], especially whilst maintaining backwards compatibility with the already-deployed commercial networks. Figure 1 illustrates the number of papers found in the Google Scholar database dealing with LPWANs and LPWAN security (prognosis is used for years beyond

2020 and was obtained by computing the conservative linear growth prediction for the following five years based on previous data). The trend shows the increase in the interest of the scientific community in this topic. Investigating the shape of the curve, one can see an exponential growth in the first few years, and then a gradual change to a linear curve (since 2017); however, it is worth noting that even though a significant share of the articles mention security, only a small portion focuses on security in depth.

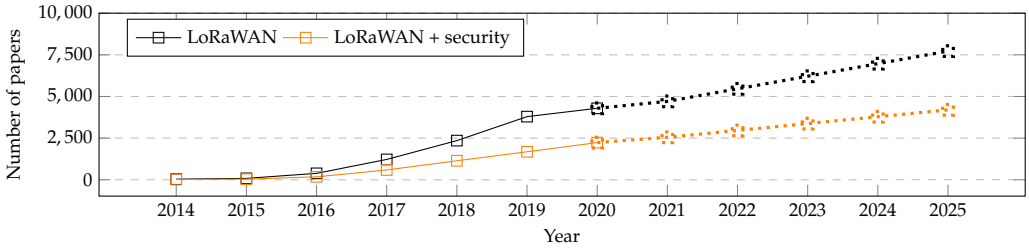

Figure 1. Results of the keyword search for the terms "LoRaWAN" and "LoRaWAN" + "Security" in Google Scholar for selected years 2014–2020 (predicted values are dotted). The correlation trend is $R = 0.893$ with a square root of $R^2 = 79.84\%$.

In this context, the contribution of the current paper is threefold:

- We provide an overview of the state-of-the-art for cyber security for LoRaWAN technology.
- We detail the design of an software-defined radio (SDR)-based testbed for trialing the LoRaWAN security in practice.
- We demonstrate the operation of the designed testbed through investigating the two practical LoRaWAN attacks, study their effects and discuss the possible ways to mitigate them. Specifically, we focus on replay and bit-flipping (i.e., change of message content) attacks. These two attacks have been selected due to their potential to have a devastating effect on infrastructure monitoring applications, which are among the core use case of LoRaWAN.

The paper is structured as follows. We start by discussing the relevant background on LoRaWAN technology, security solutions in LoRaWAN and the results of the state-of-the-art studies focusing on LoRaWAN cyber security. In Section 3, we detail the design of our security-focused testbed, our experimental environment. The experimental results demonstrating the operation of the developed testbed, including the trials of the two man-in-the-middle class attacks, are discussed in Section 4. This is followed by the summary of our results and identification of the potential future research directions in Section 5.

2. Background

2.1. LoRa and LoRaWAN Basics

The LoRaWAN technology consists of the two main elements: the physical (PHY) layer solution based on the proprietary LoRa modulation, and the link (LL) and network layer (NWK) specification. The latter is described in the open standard governed and developed by the LoRa Alliance [6].

The LoRa modulation is a variant of a frequency-chirp-spread-spectrum M-ary digital modulation, in which the instantaneous frequency is linearly increased and then wrapped to the minimum frequency when it reaches the maximum frequency of the occupied band [7]. One of the critical parameters of the LoRa modulation is the spreading factor (SF), which corresponds to the number of chips per symbol and is inversely proportional to the modulation rate of the chirp [8]. By increasing the SF a transmitter increases the time on-air, thus increasing the energy consumption and reducing the data rate, but boosts the maximum possible communication range. Notably, the signals with different SFs are quasi-orthogonal, allowing transmissions with different SFs to be correctly received simultaneously.

Following the LoRaWAN specification, a network is composed of a single network server (NS), one or multiple gateways (GWs) and end devices (EDs). In addition to these,

a network may include a dedicated join server (JS) and provide interfaces to application servers (ASs). An ED is typically represented by a sensor equipped with a radio transceiver, allowing it to communicate with a GW. Depending on the implemented class (denoted A, B or C in LoRaWAN), the media access procedure and the capability of an ED obtaining the data in downlink somewhat differs. The most basic LoRaWAN class, i.e., class A, implies Aloha channel access for uplink with a random selection of a frequency channel among all supported by the network. The two obligatory receive windows are opened by the ED after such an uplink transmission at scheduled times and using pre-specified frequency channel and SF configuration. A GW routes all the received packets to the NS through an Internet protocol (IP) backbone network connection (Ethernet, LTE, etc.). Notably, unlike the traditional cellular systems, in LoRaWAN an ED is not associated with a specific GW. Instead, all GWs forward all correctly decoded packets to the NS. The NS also features an interface for the AS, which serves as a user interface for management and data presentation and acquisition purposes. From the point of view of cyber security, the most susceptible in this architecture is the wireless connection between an ED and a GW, where a number of attacks, including, e.g., a man-in-the-middle (MitM) attack, can be carried out. The on-air LoRaWAN security is based on encryption and authentication procedures, which we detail in the following subsections.

2.1.1. Encryption Algorithms

The advanced encryption standard (AES) algorithm in the electronic code book (ECB) mode is used to encrypt the on-air communication in LoRaWAN. The AES-ECB is a block cipher in which the message is divided into blocks of a fixed length—in LoRaWAN's case, 128 bits. It is characteristic of this cipher that the cryptogram depends only on the message block and the key.

The encryption in LoRaWAN proceeds as follows. Of the entire LoRaWAN message that is illustrated in Figure 2, only part of the frame payload (FRMPayload) is encrypted. This FRMPayload is encrypted with the application-specific key AppSKey in case a frame carries application data. If FRMPayload carries media access (MAC) commands, it is encrypted with the network key NwkSKey. The used encryption method can be specified from the FPort value, which signalizes the intended target of the message. For each message, a sequence of blocks A_i is constructed, where i takes values from 1 to the length of the message divided by 16 and rounded up.

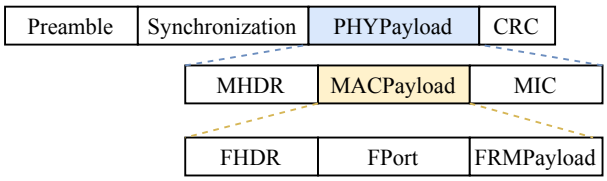

Figure 2. LoRaWAN message structures.

The structure and contents of block A_i are defined according to the LoRaWAN specification [6]. The first byte has the value $|0x01|$, the next 4 bytes are $|4 \times 0x00|$ followed by the byte for the frame direction (0 for uplink frames and 1 for downlink frames), another 4 bytes carry the device address and the frame counter. Then, a byte with value $|0x00|$ and a byte of parameter i are appended. The complete string thus appears as: $|0x01|4 \times 0x00|Dir|DevAddr|FCnt|0x00|$. This sequence of A_i blocks is encrypted using the K key, resulting in a string S. The K key is selected according to the FPort value: if the value is 0, the NwkSKey key is selected, for other values the AppSKey is used. Finally, an exclusive disjunction (XOR) operation is applied to string S to create an encrypted FRMPayload.

Subsequently, the message integrity code (MIC), allowing the verification of the integrity of the message, is generated. This code is calculated over all fields of the message, i.e., $message = |MHDR|FHDR|FPort|FRMPayload|$. Within LoRaWAN, a cipher-based message authentication code (CMAC) is used to authenticate messages. The CMAC

authentication code is based on the use of the AES block cypher in the cipher block chaining (CBC) mode. In LoRaWAN, authentication is carried out by creating a block marked B, which is then concatenated with the message ($|MHDR|FHDR|FPort|FRMPayload|$) as the key, NwkSKey is used and the CMAC cypher is executed.

2.1.2. Activation Procedure

In LoRaWAN, two options to handle the initial connection between EDs and the NS are defined. The former one is activation by personalization (ABP), which implies that all credentials are provisioned offline and is not recommended for commercial deployments due to insufficient security. The latter (and the recommended) one is over-the-air activation (OTAA). In the current paper, unless stated otherwise, we use OTAA (as defined in the standard version 1.0.x). Figure 3 visualizes the key phases and operations composing OTAA.

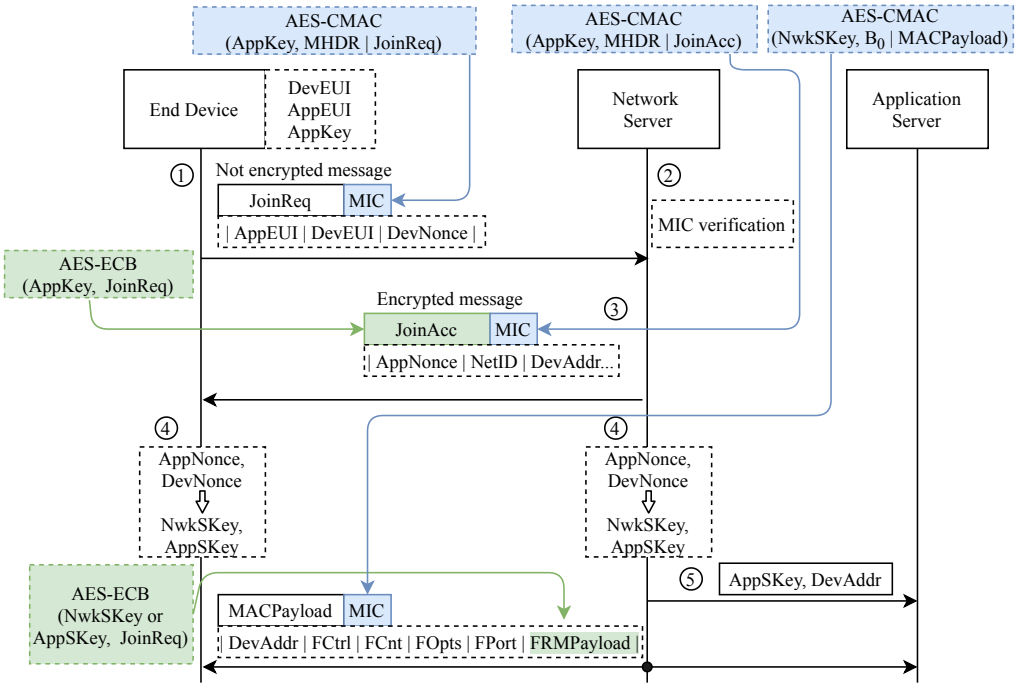

Figure 3. OTAA procedure in LoRaWAN protocol version 1.0.x.

In phase (1), the initial credentials , i.e., the DevEUI, AppEUI and AppKey, must be stored on an ED. The former represents the unique identifier of a device, while the AppEUI determines the application to handle the data and a special key associated with it. The join-request message (JoinReq) frame is composed of 8 bytes of AppEUI, 8 bytes of DevEUI and the 2 bytes of DevNonce. The message would appear as: $|AppEUI|DevEUI|DevNonce|$. The message is protected by MIC, but not encrypted. For our needs, it is essential to explain the value of DevNonce. The DevNonce value is the last 2 bytes of the JoinReq message. Because the message is not encrypted, this value can be eavesdropped. DevNonce is unique, and is usually a randomly generated value. It changes every time a device attempts to connect. Note that the NS stores DevNonce values to prevent the reuse of an old value.

In phase (2), the GW verifies the MIC, and NS checks the DevNonce value. If the checks pass, in phase (3), NS generates DevAddr, AppNonce and NetID values and shapes a join-accept (JoinAcc) message. The JoinAcc message consists of 3 bytes of AppNonce value, 3 bytes of NetID value, 4 bytes of DevAddr value, 1 byte of DL settings value, 1 byte RxDelay value and 16 bytes of an optional list of channel frequencies (CFList). The JoinAcc thus is as follows: $|AppNonce|NetID|DevAddr|DLSettings|RxDelay|CFList|$.

The value of AppNonce, which is the first 3 bytes of the encrypted JoinAcc message, is generated randomly, in the same manner as the DevNonce. For the JoinAcc message,

the MIC is generated again, and the AppKey is further used to encrypt the frame before transmission. In phase (4), the AppNonce and DevNonce are shared by the NS and the ED. NwkSKey and AppSKey values are generated from the AppNonce and DevNonce values. In phase (5), the key credentials, such as AppSKey and DevAddr, is sent to the application server. Then, the transmission of the application layer messages can begin.

2.2. Related Works

The LoRaWAN security has been considered by a number of scholars. The most up-to-date contributions can be classified into three major tracks: (i) general description of security aspects and possible vulnerabilities, (ii) new mechanisms for improving cyber security in LoRaWAN networks and (iii) preventing attacks in LoRaWAN networks. The selected works and a brief summary of their key points are summarized in Table 1.

Table 1. Summary of selected articles on LoRaWAN cyber security.

Work	Description
Category: General description of security and possible vulnerabilities	
Millere 2016 [9]	Possible vulnerabilities and attacks in LoRaWAN 1.0.x network.
Aras et al., 2017 [10]	Susceptibility of LoRaWAN to jamming, replay attack and wormhole.
Oniga et al., 2017 [11]	Analysis of security aspects of LoRaWAN and discussion of security options based on certificates.
Butun et al., 2018 [12]	Summary of security threats in LoRaWAN versions 1.0 and 1.1.
Category: Improving security of LoRaWAN	
Naoui et al., 2017 [13]	A solution that improves the security of the LoRaWAN 1.0 network by making better use of the relational key between the ED and NS.
Kim et al., 2017 [14]	Description of security gaps in key generation, and design of a new activation scheme based on a dual key.
Oniga et al., 2017 [15]	Security analysis of the LoRaWAN protocol and suggestion of a public key infrastructure.
Lin et al., 2017 [16]	Design of an open, trusted decentralized tamper-resistant system within LoRaWAN using blockchain technology.
Sanchez-Iborra et al., 2018 [17]	Security risk assessment for key management within LoRaWAN and design of a key management method based on ephemeral Diffie–Hellman over COSE.
Navarro-Ortiz et al., 2019 [18]	Hardware improvement of LoRaWAN security using USIM cards as cryptographic chips.
Ribeiro et al., 2020 [19]	Improved key management within the LoRaWAN architecture using Blockchain technology.
Tsai et al., 2020 [20]	Establishing relation using elliptic curves and AES algorithms to boost the security of S2KG communication between servers.
Category: Attack prevention	
Kim et al., 2017 [21]	Design of a prevention scheme for replay attack.
Sung et al., 2018 [22]	Protection against replay attack using RSSI and handshaking.
Gao et al., 2019 [23]	Design of SPT model to detect and protect against replay attack.
Thomas et al., 2020 [24]	Man-in-the-middle attack mitigation based on cryptographic Galois counter mode.

In what follows we detail the results of several studies, which deal with the replay and bit-flipping attacks, and thus are relevant to the scenarios emulated by us. The interested readers can find a more comprehensive overview of the other attack types and challenges in, e.g., [25].

Yang et al. [26] identify five major LoRaWAN vulnerabilities, including the replay attack and the bit-flipping attack. These attacks are also demonstrated by practical labora-

tory demonstrations. The authors approached the replay attack and the bit-flipping attack differently than we had. Specifically, they implied ABP activation and focused on uplink data transmissions carrying application data, while we consider the OTAA activation and demonstrate the potential consequences of its eavesdropping.

Kim et al. [21] deal with the prevention of replay attacks using the duplication of the DevNonce value. As part of their work, they created a prevention scheme, thanks to which the probability of duplication of the DevNonce value is reduced by 60–89% in comparison to conventional LoRaWAN. The article deals with the prevention of replay attacks, but not with the very possibility of performing an attack and eavesdropping. It also does not account for or cover the bit-flipping attack.

The prevention of replay attacks is also addressed by Sung et al. [22]. The authors describe how an attack can be detected from the received signal strength indication (RSSI) value. If the variation of RSSI value for an ED known to be static is detected, the security mechanisms are activated. The article deals mainly with sniffing and spoofing, with the communication being eavesdropped on by the device that is already admitted to the network, which is rather a strong implication. In this study, we consider eavesdropping on the communication by an external device that is not in the network and demonstrate how data captured during OTAA activation could be used for reconstruction of the keys.

Tomasin et al. [27] describe a replay attack focused on the DevNonce value. They detail the generation of a random DevNonce value and show that this value can be generated using a predictable value jammer. This results in reducing the entropy of this information, thus allowing a replay attack to be performed. Albeit discussing the join procedure in detail, the study does not deal with eavesdropping and individual attacks.

The empirical studies, which are dealing with LoRaWAN security, are relatively rare. Among these, the work of Hessel, Almon and Álvarez presented in [28] should be noted. In this study, the authors present the ChirpOTLE framework allowing practical evaluation of LoRaWAN security and report its use for investigating the potential of ADR and beacon spoofing in the context of denial-of-service attacks. This is worth mentioning that the ChirpOTLE is based on common off-the-shelf hardware, including LoRa transceivers; however, the focus of this study is on denial-of-service attacks rather than replay and bit-flipping attacks, which we deal with.

In contrast to the previously discussed articles, in the current work, we report the design of a platform allowing empirical validation of the LoRaWAN security, implying the use of SDR as a tool for eavesdropping of communication and subsequent reconstruction of keys. This allows the implementation of versatile attacks, including, as we demonstrate in what follows, the ones belonging to the MitM class. Note that MiTM attacks are among the most widespread for other IoT-grade wireless communication technologies and, thus, are likely to be also used one day against LoRaWAN devices and networks. Further, the authors of the previous studies mainly dealt with replay attacks and sniffing for data communication, implying that the attacker is already admitted to the network. In this study, we relax this assumption and demonstrate the possibility of launching an MitM attack by intercepting the OTAA procedure by an external device, which is not a part of an existing LoRaWAN network.

3. Testbed Design and Test Cases

3.1. Testbed Design

The structural diagram of the testbed network developed by us for studying the security aspects of LoRaWAN is illustrated in Figure 4. The network consists of an NS together with an AS, GW and EDs. The testbed includes the second ED and a terminal (laptop) with an RTL-SDR sniffer connected, which can be used to launch an attack. The RTL-SDR passes information to the GNU Radio software. Note that the designed testbed primarily focuses on investigating the attacks carried in the radio channel; however, after some further modifications, it can also be extended to investigate the attacks on the servers and via a backbone connection.

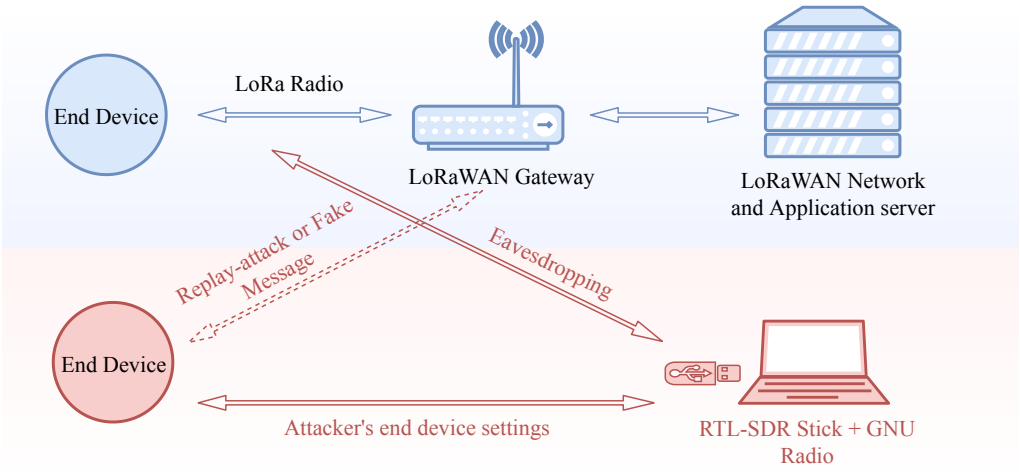

Figure 4. LoRaWAN security test bed structural diagram.

The individual elements and their configuration are described below. The whole network works currently following the version 1.0.2 of LoRaWAN specification (the upgrade to newer protocol versions is planned). The open-source Chirpstack solution was used for NS and AS [29].

3.1.1. End Devices

The two EDs used in our testbed are:

- A device built around the LoRaWAN module RHF PS01509 [30] acting as an authorized user (i.e., the "victim") has been used in our experiments. Note that this device can be replaced by any other LoRaWAN-compatible transducer or commercial product.
- The device built around a Murata [31] transceiver working in conjunction with I-CUBE-LRWAN [32] implementing the control interface and connected to a computer has been used to emulate an attack (i.e., "attacker").

3.1.2. Gateway

The LoRaWAN GW was implemented using a single-board computer Raspberry Pi 3 B [33] interfaced to the ic880a LoRaWAN concentrator [34]. The concentrator is equipped with a dipole antenna and operates in the 868 MHz band. To ensure that our work with the GW was easy and safe, we connected a reduction plate [35] between these components. The packet-forwarder application has been deployed on the GW to implement packet forwarding to/from the Chirpstack server [29]. The gateway can be seen in Figure 5.

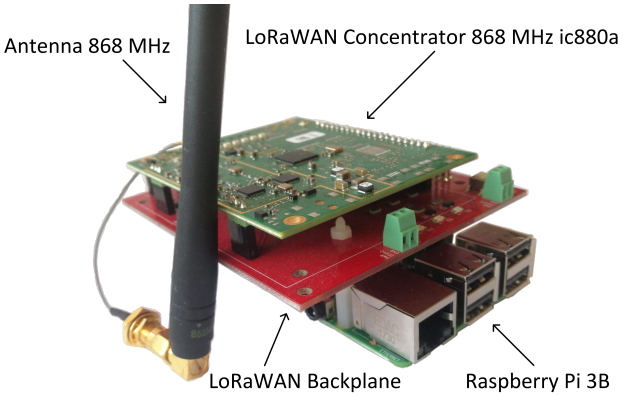

Figure 5. LoRaWAN GW comprising Raspberry Pi 3 and ic880a.

3.1.3. SDR Environment

We use an SDR connected to a laptop to implement eavesdropping, running the free GNU Radio toolkit. We chose GNU Radio due to the comprehensive support of the LoRa modulation libraries by this platform. Specifically, we use the Gr-lora library [36] to capture on-air communication. As the SDR, we chose the RTL-SDR USB device [37], which, albeit being a relatively low-cost solution, covers the frequency band from 24 MHz up to 1766 MHz. The SDR has been equipped with an omnidirectional antenna with a gain of 3 dBi.

3.2. Emulated Attack Scenarios

To evaluate the developed testbed and obtain a deeper insight into the security of LoRaWAN we have trialed the two different types of MitM attacks, namely:

- **The replay attack.** This trial is carried out by an attacker intercepting the transmission between ED and GW. Specifically, the attacker eavesdrops a message from the ED and sends it via its own malicious device to the GW. The success of this attack depends on whether the frame counter on NS is activated or not. In case the frame counter is activated, the first step is to jam the ED before it can deliver the eavesdropped message to the GW [21] and then proceed with the attack. When the frame counter is not activated (which is common in many commercial networks to enable ABP devices re-joining the network after a power-down or reboot), the attack can be carried out without any jamming. The time sequence of this attack can be seen in Figure 6.
- **The bit-flipping attack.** The attacker intercepts the message and decrypts it, modifies it, encrypts it again and sends it to the GW. This attack allows an attacker to change all the information in the message. The time sequence of this attack can be seen in Figure 7.

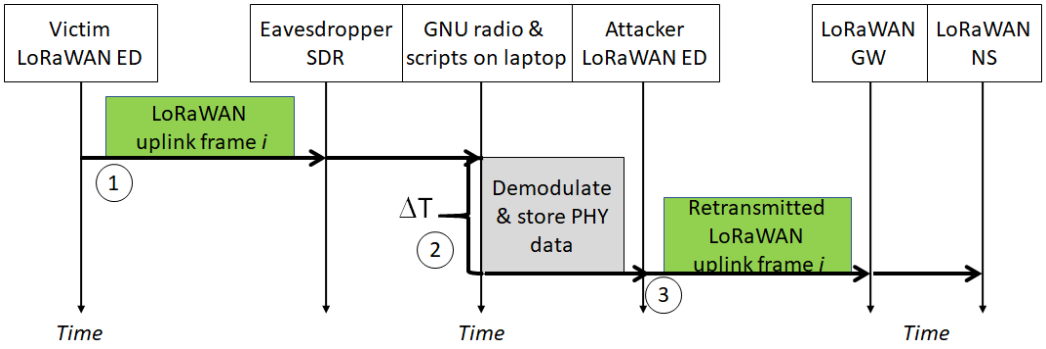

Figure 6. Replay attack time sequence and phases.

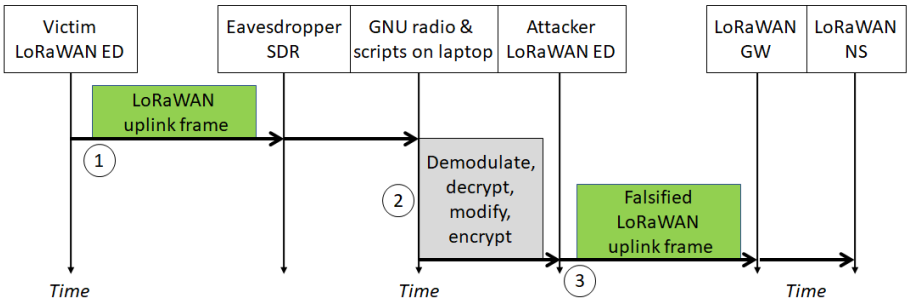

Figure 7. Bit-flipping attack time sequence and phases.

4. Experimental Results

In the following subsections, we detail the phases of the testbed development and testing and highlight the most notable experimental results.

4.1. LoRa PHY Intercepting and Decoding

We started by investigating how the LoRa-encoded data transfer can potentially be intercepted by a GNU Radio. The experimentally found configuration (blocks and the respective settings) allowing reception of the LoRa-modulated signals using the Gr-lora library is depicted in Figure 8.

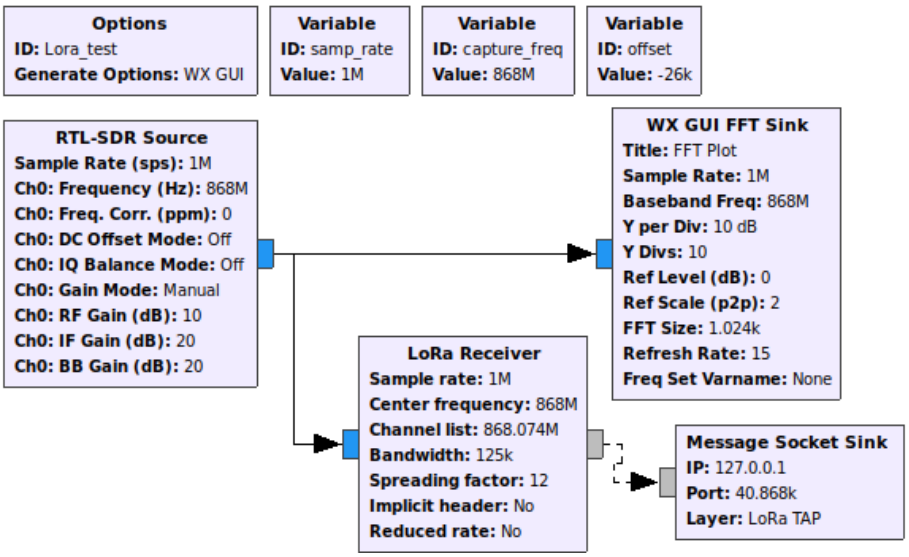

Figure 8. GNU Radio configuration for eavesdropping a LoRa-encoded packet.

To validate the correctness of the system's operation and these settings, we disabled the encryption on the victim device and configured it to transmit a pre-defined hexadecimal text ("NESIFROVANY_TEXT" encoded in ASCII, to be specific) as a PHY layer payload. The decoded by the GNU Radio packet and its structure are shown in Figure 9 (the hexadecimal values are in red font) and Figure 10, respectively. Note that the RSSI of the received radio signal, depicted in Figure 9, may provide some insight into the location of both the ED and, in case of downlink packets, the GW. Similarly, the commercial LoRaWAN networks often reserve one frequency channel (i.e., the g3 band, 869.40–869.65 MHz, allowing for 10% duty cycle) for the second receive window—RX2. The eavesdropping of packets sent in this band may allow an attacker to obtain information about the locations of the GWs even if these data are not publicly available.

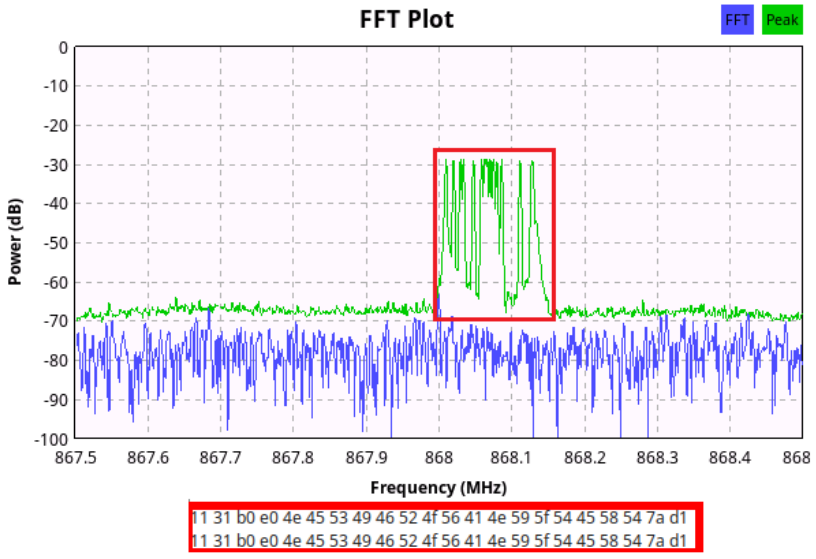

Figure 9. Results of message capture by the GNU-Radio.

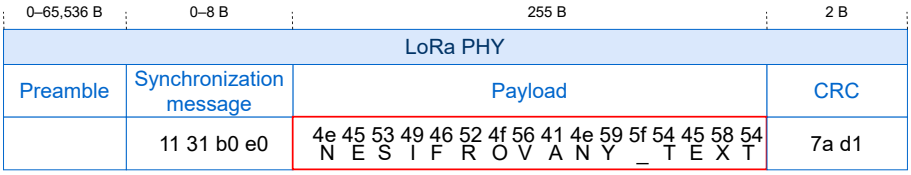

Figure 10. Decoding of the captured LoRa packet.

4.2. LoRaWAN MAC Interception and Decryption

After confirming that the GNU Radio can decode LoRa-modulated radio transmissions, we focused on making it understand and decode the MAC-layer formatting and encoding. For this, we re-enabled the LoRaWAN support and payload encryption on the victim device. Recall that the part of the PHYPayload (FRMPayload) is encrypted either with the AppSKey key (if the message carries application data) or with the NwkSKey key (if the message carries a MAC command); therefore, an attacker needs to possess these keys for decrypting the traffic of a specific ED. There are different ways to obtain these keys. For example, an insider might extract them from the database at AS or intercept the transmission and reconstruct them from the DevNonce, AppNonce and NetID. These can be extracted from the OTAA activation procedure (see Figure 3) by capturing the JoinReq (DevNonce) and the JoinAcc (AppNonce, NetID) as discussed in Section 2.1.2. The JoinReq is not encrypted and the DevNonce value (the last 2 bytes) might be extracted and used straight for the key reconstruction; however, the JoinAcc is already encrypted by using the AES 128 [38] in ECB mode with AppKey [6]. Often, the AppKey is stored in the memory of an ED in raw format and can be extracted by obtaining physical access to the serial interface. Once the AppKey is obtained, the JoinAcc can be decrypted to obtain the AppNonce and NetID, and start the reconstruction process. The sequences showing how this can be accomplished are depicted in Figure 11. The first byte indicates whether the sequence will be used for NwkSKey $|0x01|$ or AppSKey $|0x02|$ and it is followed by the AppNonce (3 B), NetID (3 B), DevNonce (2 B) and Padding (7 B, i.e., zeros are appended until data length becomes a multiple of sixteen).

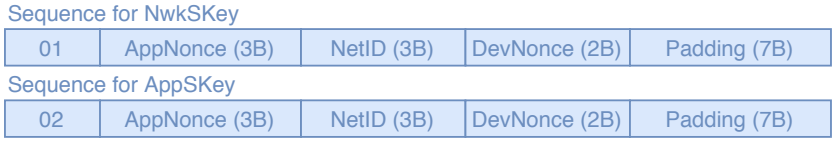

Figure 11. String for deriving session keys.

Using these sequences, it is possible to obtain the NwkSKey and/or AppSKey:

$$NwkSKey = AES_{128}(AppKey, |0x01|AppNonce|NetID|DevNonce|pad_{16}|), \quad (1)$$

$$AppSKey = AES_{128}(AppKey, |0x02|AppNonce|NetID|DevNonce|pad_{16}|). \quad (2)$$

Based on the theoretical background above, we carried out a practical trial, summarized in Figure 12 and successfully demonstrated the possibility of decrypting the messages within the LoRaWAN session. First, we obtained the AppKey by physically accessing the victim's serial interface. Then, we captured the PHY layer radio packet by eavesdropping the target device (RHF PS01509 described in Section 3.1.1) by SDR (HW, RTL-SDR Stick described in Section 3.1.3) and GNURadio (SW, described in Section 4.1). Recall that only a part of the message is encrypted (FRMPayload), and the rest of the message is not encrypted (Synchronization message, CRC, MHDR, MIC, DevAddr, FCtrl, FCnt, FOpts and FPort). Gradually, we parsed the message into parts until we were left with only the part of the FRMPayload. When decrypting the FRMPayload, it is essential to determine which key was used for encryption (AppSKey or NwkSKey). This can be found using the FPort value; therefore, if the FPort field contains 0, it is reserved only for MAC commands (e.g., ADR, link check . . .), and it is encrypted with NwkSKey. If application data are transmitted

(FPort value 1–223), LoRaWAN MAC protocol testing (FPORT 224) or ports reserved for future standardized application extensions (FPORT 225–255), FRMPayload is encrypted by the AppSKey application key. Subsequently, we generated k of 16-bytes A_i blocks. The number of blocks (k) depends on the size of the FRMPayload. The first byte is always $0x01$. The next four bytes are equal to $0x00$, followed by a byte to determine the direction of communication ($|0x00|$ for uplink or $|0x01|$ for downlink), 4 bytes for the device address, 4 bytes are reserved for the frame counter (2 bytes of FCnt followed by 2 bytes of zero octets), and one byte is set to $0x00$. The last byte is i, which signalizes the order number of A_i blocks. The A_i block is encrypted using the AES-128 in ECB mode. In our case, we had a transmission with application data, and, therefore, the A_i block was encrypted using the AppSKey. If there are more A_i blocks, they are encrypted one by one:

$$S_i = AES_{128,ECB}(AppSKey, A_i), \tag{3}$$

and then chained:

$$S = |S_1|S_2|\ldots|S_k|. \tag{4}$$

Finally, an exclusive disjunction (XOR) operation is carried on block S and the FRMPayload, which is padded with zeros to have the same size as block S. Specifically, for the illustrated case, the ASCII-encoded "Ahoj" message was transmitted by the ED, captured by SDR and successfully decrypted, as shown in the Figure 12.

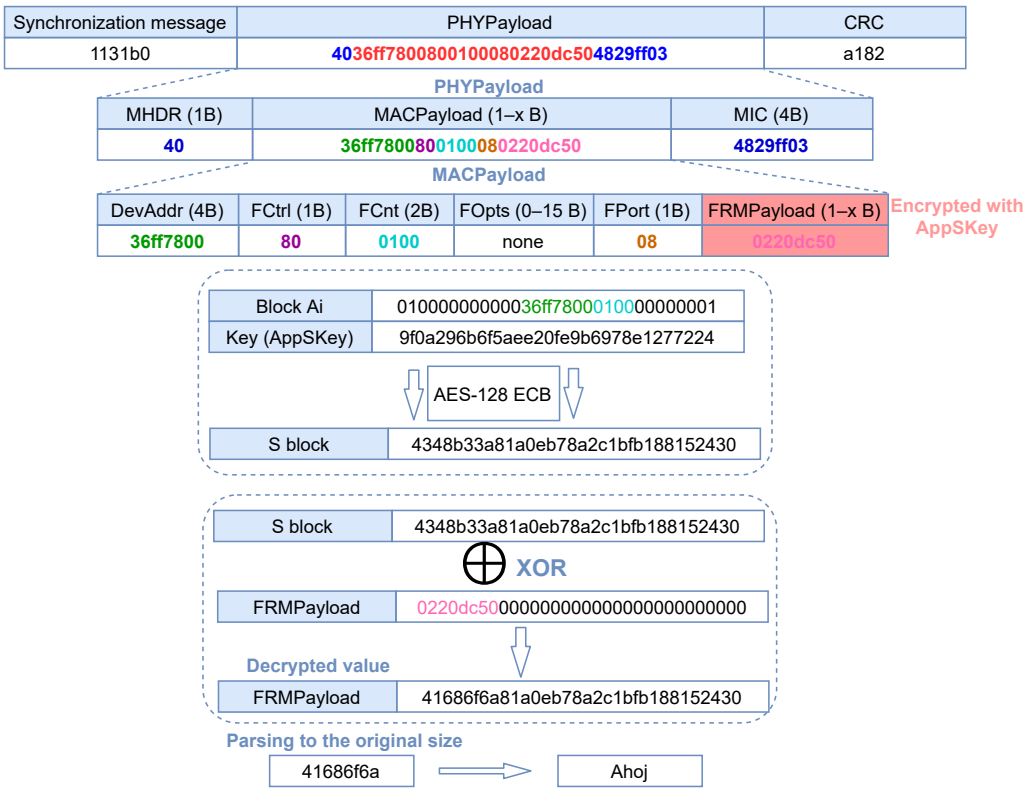

Figure 12. LoRaWAN message decrypting.

4.3. Replay Attack

The results presented in the two previous subsections demonstrate the feasibility of receiving and decoding the LoRaWAN packets by the developed testbed in unencrypted and encrypted modes; therefore, we proceeded to investigate the possibility of launching and the potential effect of the attacks. First, we trialed the replay attack. For this, we configured the victim ED to periodically send a valid data packet. The packet has been captured by the SDR, handed to the GNU Radio application (see Figure 13), decoded and then retransmitted by the attacker ED to the GW. Using the I-CUBE-LRWAN software

extension, we gained complete control over the Murata module serving as the attacker. By modifying the code, we forced the Murata module to broadcast the required captured sequence. On Listing 1 one can see a part of the code to send the captured message from the Murata device.

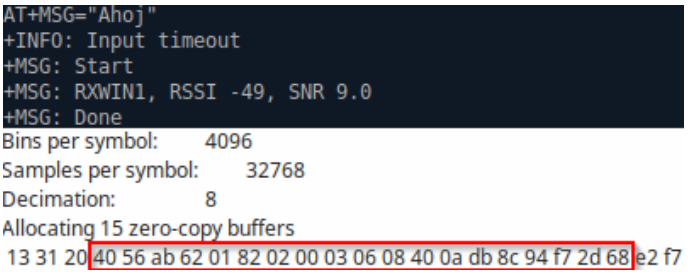

Figure 13. Packet sent from the victim ED (top, black background) and the packet captured by GNU Radio (bottom, white background).

Listing 1: Packet sent by the attacker

```
Radio.SetTxConfig( MODEM_LORA, 14, 0, 0, 12, 1, 8, false,
true, 0, 0, false, 3000 );
Radio.SetChannel( 868100000 );
uint8_t buf[100];

buf[0] = 0x40; buf[1] = 0x56; buf[2] = 0xab; buf[3] = 0x62;
buf[4] = 0x01; buf[5] = 0x82; buf[6] = 0x02; buf[7] = 0x00;
buf[8] = 0x03; buf[9] = 0x06; buf[10] = 0x08; buf[11] = 0x40;
buf[12] = 0x0a; buf[13] = 0xdb; buf[14] = 0x8c; buf[15] = 0x94;
buf[16] = 0xf7; buf[17] = 0x2d; buf[18] = 0x68;

Radio.Send( buf, 19 );
```

The results of the attack on the NS in the case when no strict frame counter is enabled are demonstrated in Figure 14. The message transmitted by the attacker is successfully received and injected into the database. Note that even if the strict counter is enabled (resulting in NS dropping the message with repeating counter) the attack can be launched if the original transmission of the victim is jammed. It is also important to note that we have not implied the knowledge of NwkSKey, AppSKey or any other key for this attack.

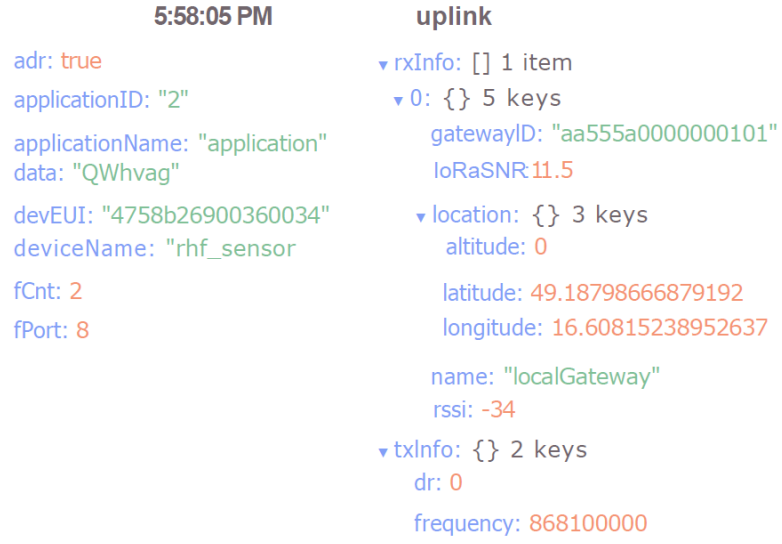

Figure 14. A replay message in the NS database.

4.4. Bit-Flipping Attack

After confirming the feasibility of executing a replay attack, we investigated the possibility of injecting a fake message. Specifically, the procedures for decryption, modification and re-encryption are detailed in Figure 15.

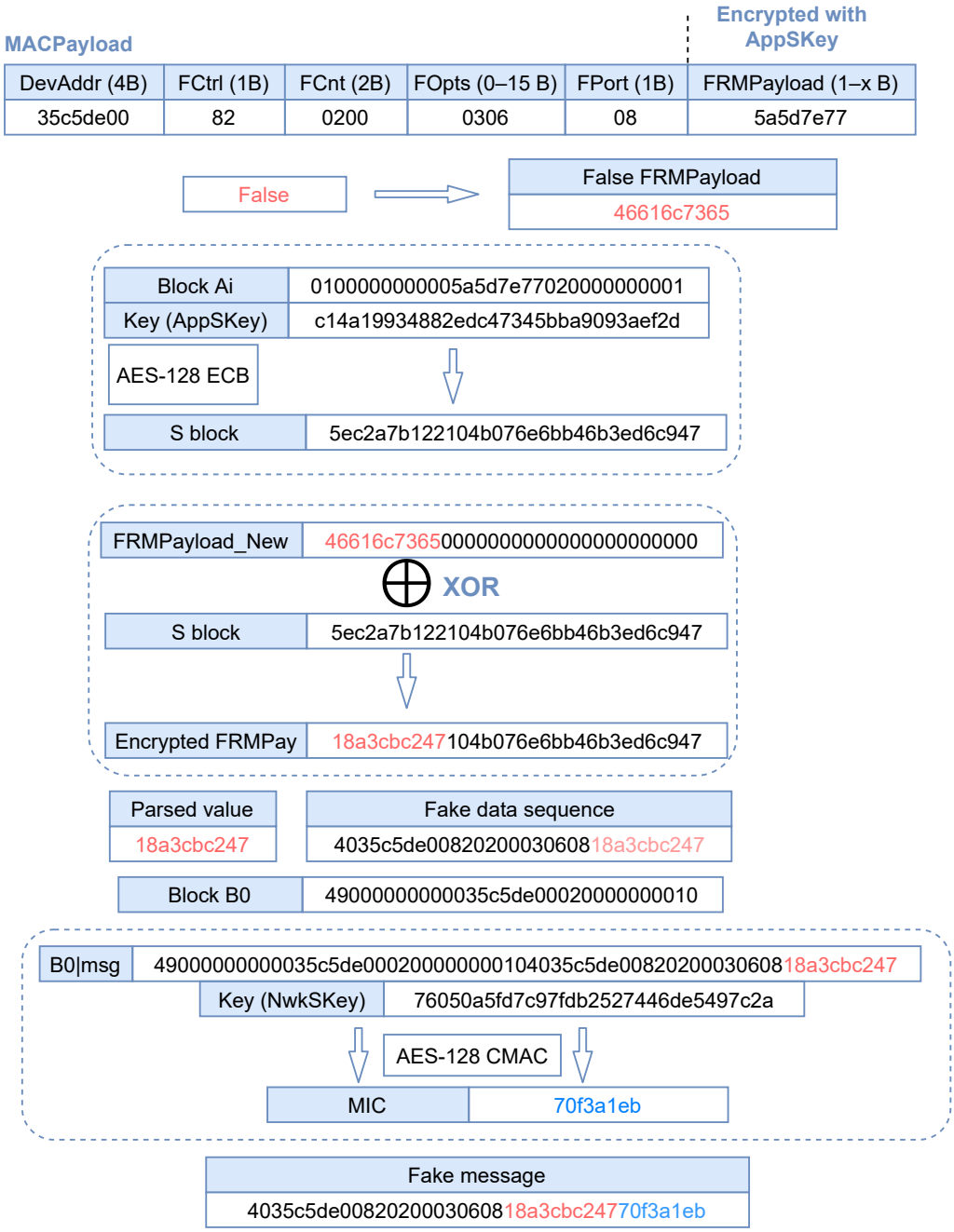

Figure 15. Bit-flipping attack: the procedure for creating a fake message.

This time, instead of just repeating the received packet of the victim as is, we aimed at modifying its payload. Specifically, our goal has been to replace the original application payload with the "False" message encoded in ASCII. For that, we created a new A_i block and encrypted it using the AES-128 ECB with the AppSKey (obtained as discussed in Section 4.2) and thus created a block S. We added zeros to the new payload and executed the XOR operation. This created an encrypted value of the FRMpayload. Then, we parsed the value and added it to the original header. In order to make the NS accept the message, we also had to calculate and pass the new MIC value. To calculate the MIC value, it is

necessary to create a block B_0 and attach the original message to this block. We encrypted this new message using the AES-128 in CMAC mode, using the network relational key NwkSKeyas the key. Only the first 4 byte values are parsed from the resulting sequence and attached to the message. We sent this fake message to the server, and the server received it and added it to the database, as depicted in Figure 16. Note, that the value "*RmFsc2U =*" in the Figure 16 is in Base64 format. Converting to hexadecimal it becomes *46616c7365*, which corresponds to *"False"* message encoded in ASCII.

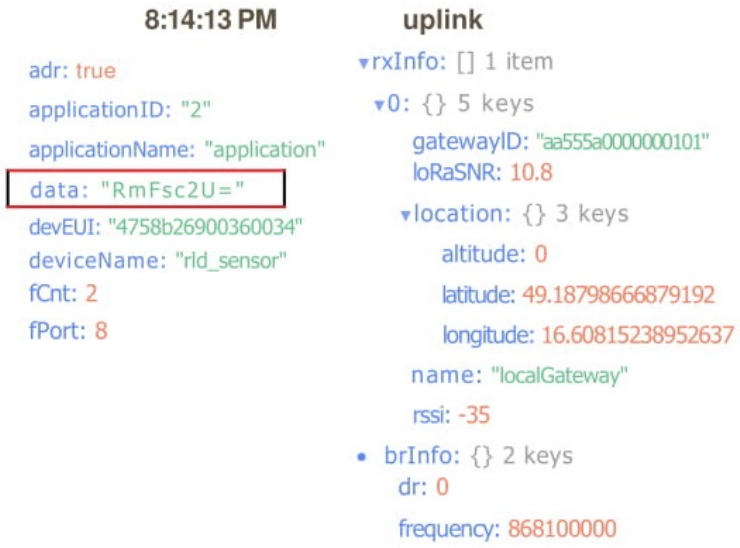

Figure 16. Fake message injected in LoRaWAN NS database.

5. Conclusions

In this paper, we have first discussed the cyber security aspects of LoRaWAN LPWAN technology, and then introduced and detailed the design of an SDR and GNU Radio-based testbed for assessing and experimenting with the on-air security in LoRaWAN. Then, we have reported the results of our experiments conducted using the designed testbed. Specifically, we have shown how LoRa packets can be eavesdropped by an SDR. Further, we have demonstrated that knowing the AppKey, the AppSKey and NwkSKey can be reconstructed by eavesdropping the OTAA join procedure. Having these keys, an intruder can launch a bit-flipping attack, resulting in false data being accepted by the NS.

Note that despite these attacks having potentially devastating consequences, state-of-the-art technology already provides means to mitigate them. First, the newer versions of the LoRaWAN protocol (i.e., LoRaWAN 1.1.x protocol) decouples the NwkSKey from AppKey; however, to use this, an upgrade to LoRaWAN 1.1.x has to be performed both on the GWs/NSs (a software update) and on the devices (e.g., the hardware or firmware). Notably, especially the latter, requires substantial efforts and costs. Second, the attacks demonstrated by us in this paper (except the replay attack) imply either unencrypted transmission (which is not supported by LoRaWAN) or the knowledge of AppKey. While the early-day EDs used to allow reading this key and other security credentials back from an ED, the LoRaWAN sensors of today usually prevent access to these data; however, given the demonstrated attack procedures in this paper, it becomes clear that these credentials should be secured at the AS as well.

All in all, the man-in-the-middle attacks in the LPWAN in general and LoRaWAN, in particular, has attained rather limited attention so far. Nonetheless, this field offers multiple challenges to be addressed and problems still to be solved:

- Development of the algorithms and tools (if needed—inclusive of the dedicated hardware devices) to detect and classify the on-air attacks in LoRaWAN. Specifically, the algorithms can be based on monitoring the re-connection patterns of the individual

devices, their traffic patterns and variation of their radio-channel parameters (e.g., RSSI and signal-to-noise ratio (SNR), as discussed by Sung et al. [22]).
- Development of algorithms and procedures enabling EDs to detect bogus GWs/NS.
- Engineering the mechanisms and procedures allowing re-connection and re-establishment of control over the hijacked sensors and the EDs suffering an attack (e.g., connected to a bogus NS).
- Addressing the novel types of attacks, specific for the IoT networks (e.g., the energy-depletion attack [39]).

We expect that these challenges will become even more critical in the coming years, with the further deployment of LoRaWAN networks. Specifically, the introduction of the non-terrestrial satellite-based LoRaWAN networks enabled by novel long range-frequency hopping spread spectrum (LR-FHSS) modulation [40] will clearly bring new security challenges. Importantly, to validate and assess the efficiency of the newly suggested security mechanisms in practice, one requires a specialized testbed. To address this need, the current study presents and details the design of a flexible, SDR-based security-oriented testbed for a LoRaWAN network; therefore, we are certain that the receipts and results reported in this paper will serve not only as a motivation but also provide a reference toolset for practical security studies dealing with LoRaWAN, and, potentially, the other LPWANs.

Author Contributions: Conceptualization, O.P., R.F., K.M., H.R. and J.M.; methodology, O.P., H.R. and R.F.; software, O.P.; validation, O.P., R.F., K.M., H.R. and J.M.; formal analysis, O.P., R.F., K.M., H.R. and J.M.; investigation, O.P., R.F. and K.M.; resources, O.P., R.F. and K.M.; data curation, O.P.; writing—original draft preparation, O.P., R.F., K.M. and J.M.; writing—review and editing, O.P., R.F., K.M., H.R. and J.M.; visualization, O.P., R.F. and K.M.; supervision, R.F.; project administration, R.F.; funding acquisition, R.F. and J.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the Technology Agency of the Czech Republic under Grant reg. No. TK02030013. The work of K.M. was also supported by the Academy of Finland 6Genesis Flagship under Grant No. 318927.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ABP	Activation By Personalization
AES	Advanced Encryption Standard
AS	Application Server
ASCII	American Standard Code for Information Interchange
CBC	Cipher Block Chaining
CMAC	Cipher Based Message Authentication Code
ECB	Electronic Code Book
ED	End Device
GNU	The GNU Project
GW	Gateway
IP	Internet Protocol
IoT	Internet of Things
JS	Join Server
LL	Link Layer
LPWA	Low Power Wide Area
LPWAN	LPWA Network
LR-FHSS	Long Range-Frequency Hopping Spread Spectrum
LTE	Long Term Evolution

MitM	Man in the Middle
MAC	Media Access Control layer
MIC	Message Integrity Code
mMTC	massive Machine Type Connectivity
NS	Network Server
NWK	Network Layer
OTAA	Over the Air Activation
PHY	Physical layer
RAT	Radio Access Technology
RSSI	Received Signal Strength Indication
SDR	Software-Defined Radio
SF	Spreading Factor
SNR	Signal-to-Noise Ratio
USIM	Universal Subscriber Identity Module

References

1. Chettri, L.; Bera, R. A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. *IEEE IoT J.* **2019**, *7*, 16–32. [CrossRef]
2. Semtech LoRa Technology Overview | Semtech. Available online: https://www.semtech.com/lora (accessed on 25 February 2021).
3. Shanmuga Sundaram, J.P.; Du, W.; Zhao, Z. A survey on lora networking: Research problems, current solutions, and open issues. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 371–388. [CrossRef]
4. McPherson, R.; Irvine, J. Secure decentralised deployment of LoRaWAN sensors. *IEEE Sens. J.* **2020**, *21*, 725–732. [CrossRef]
5. Fujdiak, R.; Mikhaylov, K.; Stusek, M.; Masek, P.; Ahmad, I.; Malina, L.; Porambage, P.; Voznak, M.; Pouttu, A.; Mlynek, P. Security in Low Power Wide Area Networks: State-of-the-Art and Development towards the 5G. In *LPWAN Technologies for IoT and M2M Applications*; Academic Press: Cambridge, MA, USA, 2020; pp. 373–396.
6. Sornin, N.; Luis, M.; Eirich, T.; Kramp, T.; Hersent, O. Lorawan Specification. Available online: https://lora-alliance.org/resource_hub/lorawan-specification-v1-0/ (accessed on 19 August 2021).
7. Kim, S.; Heonkook, L.; Jeon, S. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. *Sensors* **2020**, *20*, 1008. [CrossRef] [PubMed]
8. Croce, D.; Gucciardo, M.; Mangione, S.; Santaromita, G.; Tinnirello, I. Impact of LoRa imperfect orthogonality: Analysis of link-level performance. *IEEE Commun. Lett.* **2018**, *22*, 796–799. [CrossRef]
9. Miller, R. Lora security: Building a secure lora solution. In *MWR Labs Whitepaper*; F-Secure Cyber Security Limited: Basingstoke, UK, 2016; pp. 1–18.
10. Aras, E.; Ramachandran, G.S.; Lawrence, P.; Hughes, D. Exploring the security vulnerabilities of LoRa. In Proceedings of the 3rd International Conference on Cybernetics (CYBCONF), Exeter, UK, 21–23 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6. [CrossRef]
11. Oniga, B.; Dadarlat, V.; De Poorter, E.; Munteanu, A. Analysis, design and implementation of secure LoRaWAN sensor networks. In Proceedings of the 13th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 7–9 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 421–428. [CrossRef]
12. Butun, I.; Pereira, N.; Gidlund, M. Security risk analysis of LoRaWAN and future directions. *Future Internet* **2019**, *11*, 3. [CrossRef]
13. Naoui, S.; Dadarlat, V.; Elhdhili, M.E.; Munteanu, A. Enhancing the security of the IoT LoraWAN architecture. In Proceedings of the International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), Paris, France, 22–25 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–7. [CrossRef]
14. Kim, J.; Song, J. A dual key-based activation scheme for secure LoRaWAN. *Adv. Wirel. Commun. Mob. Comput. Technol. Internet Things* **2017**, *2017*, 1–12. [CrossRef]
15. Oniga, B.; Dadarlat, V.; De Poorter, E.; Saidane, L.A. A secure LoRaWAN sensor network architecture. In Proceedings of the 2017 IEEE SENSORS, Glasgow, UK, 29 October–1 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–3. [CrossRef]
16. Lin, J.; Shen, Z.; Miao, C.; Liu, S. Using blockchain to build trusted LoRaWAN sharing server. *Int. J. Crowd Sci.* **2017**, *1*, 270–280. [CrossRef]
17. Sanchez-Iborra, R.; Sánchez-Gómez, J.; Pérez, S.; Fernández, P.J.; Santa, J.; Hernández-Ramos, J.L.; Skarmeta, A.F. Enhancing lorawan security through a lightweight and authenticated key management approach. *Sensors* **2018**, *18*, 1833. [CrossRef] [PubMed]
18. Navarro-Ortiz, J.; Chinchilla-Romero, N.; Ramos-Munoz, J.J.; Munoz-Luengo, P. Improving Hardware Security for LoRaWAN. In Proceedings of the Conference on Standards for Communications and Networking (CSCN), Granada, Spain, 16 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6. [CrossRef]
19. Ribeiro, V.; Holanda, R.; Ramos, A.; Rodrigues, J.J. Enhancing Key Management in LoRaWAN with Permissioned Blockchain. *Sensors* **2020**, *20*, 3068. [CrossRef]
20. Tsai, K.; Leu, F.; Hung, L.; Ko, C. Secure Session Key Generation Method for LoRaWAN Servers. *IEEE Access* **2020**, *8*, 54631–54640. [CrossRef]

21. Kim, J.; Song, J. A simple and efficient replay attack prevention scheme for LoRaWAN. In Proceedings of the 7th International Conference on Communication and Network Security, Tokyo, Japan, 24–26 November 2017; ACM: New York, NY, USA, 2017; pp. 32–36. [CrossRef]
22. Sung, W.; Ahn, H.; Kim, J.; Choi, S. Protecting end-device from replay attack on LoRaWAN. In Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si, Gangwon-do, Korea, 11–14 February 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 167–171. [CrossRef]
23. Gao, S.; Li, X.; Ma, M. A Malicious Behavior Awareness and Defense Countermeasure Based on LoRaWAN Protocol. *Sensors* **2019**, *19*, 5122. [CrossRef] [PubMed]
24. Thomas, J.; Cherian, S.; Chandran, S.; Pavithran, V. Man in the Middle Attack Mitigation in LoRaWAN. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–28 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 353–358. [CrossRef]
25. Torres, N.; Pinto, P.; Lopes, S.I. Security Vulnerabilities in LPWANs—An Attack Vector Analysis for the IoT Ecosystem. *Appl. Sci.* **2021**, *11*, 3176. [CrossRef]
26. Yang, X.; Karampatzakis, E.; Doerr, C.; Kuipers, F.A. Security Vulnerabilities in LoRaWAN. In Proceedings of the Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA, 17–20 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 129–140. [CrossRef]
27. Tomasin, S.; Zulian, S.; Vangelista, L. Security analysis of lorawan join procedure for internet of things networks. In Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, CA, USA, 19–22 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6. [CrossRef]
28. Hessel, F.; Almon, L.; Álvarez, F. ChirpOTLE: A Framework for Practical LoRaWAN Security Evaluation. In Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'20), Linz, Austria, 8–10 July 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 306–316. [CrossRef]
29. Orne Brocaar. ChirpStack Network Server. Available online: https://github.com/brocaar/chirpstack-network-server (accessed on 25 August 2020).
30. RHF-PS01509, Lorawanclass a/cat Command Specification. Available online: https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/datasheet/module/lorawan_class_ac_at_command_specification_-_v4.4.pdf (accessed on 25 February 2021).
31. Sub-GModule Data Sheet, BP-ABZ-C. Available online: https://wireless.murata.com/pub/RFM/data/type_abz.pdf (accessed on 25 August 2020).
32. I-CUBE-LRWAN, STM32 LoRa® Software Expansion for STM32Cube. Available online: https://www.st.com/resource/en/data_brief/i-cube-lrwan.pdf (accessed on 25 August 2020).
33. Raspberry Pi 3 Model B+. Available online: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf (accessed on 25 August 2020).
34. WiMOD iC880A, Datasheet. Available online: https://webshop.ideetron.nl/Files/3/1000/1211/Attachments/Product/IB4c6A1J5Uh6Ej5D3i6cQ88q1P2D1404.pdf (accessed on 25 August 2020).
35. IC880A LoRaWAN Gateway Backplane v2.0. Available online: https://shop.coredump.ch/product/ic880a-lorawan-gateway-backplane/ (accessed on 25 August 2020).
36. Robyns, P. Gr-Lora. Available online: https://github.com/rpp0/gr-lora (accessed on 25 August 2020).
37. R820T High Performance Low Power Advanced Digital TV Silicon Tuner Datasheet. Available online: https://rtl-sdr.com/wp-content/uploads/2013/04/R820T_datasheet-Non_R-20111130_unlocked.pdf (accessed on 25 August 2020).
38. Processing Standards Publication 197, Advanced Encryption Standard (AES). Available online: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf (accessed on 25 August 2020).
39. Mikhaylov, K.; Fujdiak, R.; Pouttu, A.; Voznak, M.; Malina, L.; Mlynek, P. Energy Attack in LoRaWAN: Experimental Validation. In Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES), Canterbury, UK, 26–29 August 2019; ACM: New York, NY, USA, 2019; pp. 1–6. [CrossRef]
40. LoRaWAN Protocol Expands Network Capacity with New Long Range—Frequency Hopping Spread Spectrum Technology. 2021. Available online: https://blog.semtech.com/lorawan-protocol-expands-network-capacity-with-new-long-range-frequency-hopping-spread-spectrum-technology (accessed on 25 February 2021).