A LoRaWAN Security Assessment Test Bench

T. Claverie, J. Lopes Esteves
Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)
Paris, France

Abstract

LoRaWAN is a recent protocol and despite having been already studied from a security perspective, several attacks have not been reproduced in practice mostly due to a lack of details regarding the test benches used. After presenting previous work on the LoRaWAN protocol and the various platforms described, we present an environment based on hardware, software and SDR to study the radio layer of the protocol. The efficiency of this architecture is demonstrated by reproducing theoretical attacks on LoRaWAN 1.0.

1 Introduction

LoRaWAN is a protocol dedicated to Low Power IoT devices; its current version (1.1) was released in October, 2017 [5]. It is built upon the LoRa (Long Range) modulation patented by Semtech.

In [4] a *replay and decrypt* attack is detailed on LoRaWAN 1.0 and the authors provide a very accurate and complete overview of the state of the art up to late 2017. A formal verification of the security of the LoRaWAN handshake has been proposed in [8]. In [19], the biasing of a random number generator in presence of electromagnetic interferences is demonstrated. Session cryptographic material desynchronisation between a device and the network has been discussed in various articles [4, 19, 14, 11, 12]. The possibility of spoofing LoRaWAN gateways in order to provide fake time and position references appeared in [12, 20]. Message modification and forgery in order to attack the network has been suggested in [15, 10].

Several LoRaWAN security testing environments have already been mentioned in previous work. However, none have been described precisely enough to allow reproductibility of the results and some do not involve instrumentation of the radio layers.

In [17, 18], the author describes using a LoRaWAN gateway, two test end devices and an open-source LoRaWAN server to emulate a complete infrastructure. However, these tests concerned only the communication in the core network, beyond the gateway. A testing platform is mentioned in [12]; it is able to capture and analyze messages, however no information on its exact capabilities and building blocks are provided. In [11], hardware attacks on devices and radio attacks are mentioned. In particular, eavesdropping of LoRaWAN communications, replay attacks, and various denial of service by flooding an object with messages are described. Again, the test infrastructure description does not allow reproducing the setup.

Even when specific radio testing has been investigated, the information regarding the test setup were left aside: no details on the jamming methods [3, 6], nor on how a LoRaWAN device is perturbated by electromagnetic interference [7] were given.

2 Experimental setup

In this section, the different components used for our test bench are introduced.

LoRaWAN Evaluation kit In order to rapidly deploy a LoRaWAN 1.0 network without having to develop the core network services, a Microchip LoRaWAN development kit [13] has been set up. This development kit includes two LoRa Mote (based on the SX1276 chipset) and a LoRa gateway (SX1301) which can be connected via ethernet to core network services packaged in a ready-to-use docker container. The administration interface allows to manage gateways, applications, devices, encryption keys and some radio parameters (frequency, modulation). It constitutes a ready-to-use LoRaWAN infrastructure but is intended only for high level interaction with the protocol and thus lacks low layer flexibility.

LoRa programmable dongle A PyCom FiPy board has been used to act as a malicious LoRa node. It is a MicroPython development board supporting LoRa modulation and including a LoRaWAN stack. With this tool one can send and receive LoRa MAC frames with a better timing accuracy and lower latency than the LoRa Mote and act as both a gateway and a device.

GNU Radio In order to also have physical layer instrumentation, the GNU Radio framework was used along with a software defined radio (SDR) USB dongle based on a Realtek RTL2832U and a R820T2 tuner. Several SDR-based LoRa receivers are available, with varying completion rates [16, 2, 1].

The gr-lora block from [9] provided very good results along with detailed explanations. However significant latency is induced by the decoding layer and it only supports demodulating uplink or downlink transmissions at once. A minor change has been made to enable decoding both uplink and downlink messages within a single LoRa decoder block.

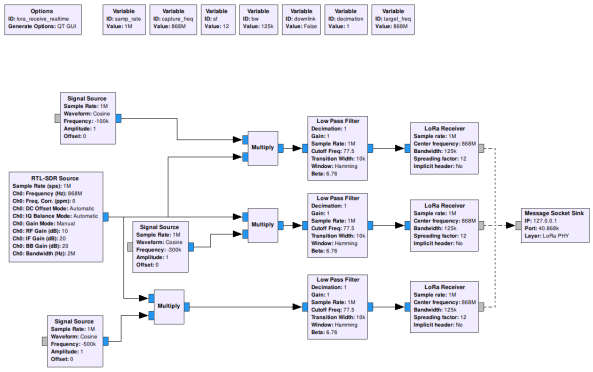

Figure 1: GNU Radio flowgraph for capturing LoRaWAN communications on 3 channels

Using this block, we implemented a GNU Radio LoRa decoder which listens on several channels and forwards the decoded frames to a UDP port (Fig. 1). This strategy allowed receiving all 21 LoRa channels in parallel. Along with the traditional waterfall visualization of the spectrum, the use of GNU Radio allows debugging the physical layer of aformentioned radio transmitters.

3 Reproducing the replay or decrypt attack

The LoRaWAN infrastucture from the development kit was used as a test network. Being easily programmable, the FiPy was used to attack. The RTL-SDR dongle with LoRa decoder block was used to monitor communications, understand the behavior of the devices and investigate in case of problems.

Attack implementation This setup reimplements various radio tests on a LoRaWAN network. In particular, it was possible to reimplement the *replay or decrypt* attack on the LoRaWAN protocol 1.0 described in [4].

This attack leverages a nonce reuse (*DevNonce*) in the handshake protocol between a device and the network: The device sends a *JoinRequest* message and the network replies with a *JoinAccept* message. After that, they both have shared cryptographic material and can start sending data frames to each other. The attack scenario is the following: the attacker listens and waits until he captures a full join handshake from the targeted device. He captures messages in this first session.

When the device initiates a new join procedure, the attacker spoofs the gateway and forces a reinitiation until the *DevNonce* sent by the device is the same as in the captured session; the attacker then responds with the captured *JoinAccept*.

Once the join procedure succeeds, the device starts sending data frames. Due to the nonce reuse, they will be encrypted with the same keystream as the captured session, which allows an attacker to partially break the confidentiality of the messages.

Results Our implementation of the *replay or decrypt* attack took about three days to complete, with around 2^{15} *DevNonce* tested and a ten-second delay between two trials. With this setup, it is also possible to implement several desynchronisation attacks described in [4, 19, 14, 11, 12].

More generally, this combination of ready-to-use infrastructure and SDR proved invaluable for security testing the LoRaWAN protocol. This setup can also be used to monitor the LoRaWAN communications and develop detection heuristics for these attacks. An unusual spectral occupation infringes the specifications and may indicate a problem or an attack.

4 Conclusion

In this paper we described a low cost security assessment platform for the LoRaWAN protocol. We provided a precise description of the building blocks in order to guarantee accurate reproducibility of the test conditions.

Furthermore, we validated our strategy by very quickly implementing a theoretical attack on LoRaWAN 1.0 (which has been fixed in LoRaWAN 1.1). The theoretical results have been reproduced and the estimations of the attack cost and complexity were confirmed.

The hybrid radio approach involving a ready-to-use development kit, a progammable radio dongle and a software defined radio has shown interesting benefits in this case, combining the advantages of each platform while compensating for their drawbacks.

In particular, it provides an interesting framework to analyze the impacts of attacks on all layers, such as jamming or intentional electromagnetic interference, and to test protocol stacks.

References

[1] Lora-sdr, 2016. https://github.com/myriadrf/LoRa-SDR.

[2] rtl-sdrangelove, 2016. https://github.com/hexameron/rtl-sdrangelove.

[3] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stéphane Delbruel, Wouter Joosen, and Danny Hughes. Selective Jamming of LoRaWAN Using Commodity Hardware. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous 2017, pages 363–372, New York, NY, USA, 2017. ACM. event-place: Melbourne, VIC, Australia.

[4] Gildas Avoine and Loïc Ferreira. Rescuing LoRaWAN 1.0. Technical Report 651, IACR, 2017.

[5] LoRa Alliance Technical Committee. LoRaWAN 1.1 Specification, 2017.

[6] S. M. Danish, A. Nasir, H. K. Qureshi, A. B. Ashfaq, S. Mumtaz, and J. Rodriguez. Network Intrusion Detection System for Jamming Attack in LoRaWAN Join Procedure. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2018.

[7] S. M. Danish, H. K. Qureshi, and S. Jangsher. Jamming Attack Analysis of Wireless Power Transfer on LoRaWAN Join Procedure. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, December 2018.

[8] Mohamed Eldefrawy, Ismail Butun, Nuno Pereira, and Mikael Gidlund. Formal security analysis of lorawan. *Computer Networks*, 148:328 – 339, 2019.

[9] Matt Knight. gr-lora, 2016. https://github.com/BastilleResearch/gr-lora.

[10] JungWoon Lee, DongYeop Hwang, JiHong Park, and Ki-Hyung Kim. Risk analysis and countermeasure for bit-flipping attack in LoRaWAN. In *2017 International Conference on Information Networking (ICOIN)*, pages 549–551, Da Nang, January 2017.

[11] Franck L'Hereec and Nicolas Joulain. Sécurité LoRaWAN. In *Computer & Electronics Security applications Rendez-vous (C&ESAR) 2016*, pages 92–108, Rennes, France, 2016.

[12] Renaud Lifchitz. Security review of LoRaWAN networks. In *Hardwear.io*, The Hague, Netherlands, 2016.

[13] Microchip. Lora technology evaluation kit, 2018. https://www.microchip.com/DevelopmentTools/ProductDetails/DV164140-1.

[14] Robert Miller. LoRa Security - Building a Secure LoRa Solution. Whitepaper, MWR Labs, 2016.

[15] Robert Miller. LoRa the explorer: Attacking and Defending LoRa Systems. In *Syscan 360 Singapore*, Singapore, 2016.

[16] Pieter Robyns. gr-lora, 2018. https://github.com/rpp0/gr-lora.

[17] Sébastien Roy. Lorawan: Déploiement d'une infrastructure de test - partie 1/2. Guide technique, MISC Mag, 2018. https://www.miscmag.com/lorawan-deploiement-dune-infrastructure-de-test-partie-1-2/.

[18] Sébastien Roy. Lorawan: Déploiement d'une infrastructure de test - partie 2/2. Guide technique, MISC Mag, 2018. https://www.miscmag.com/lorawan-deploiement-dune-infrastructure-de-test-partie-2-2/.

[19] S. Tomasin, S. Zulian, and L. Vangelista. Security Analysis of LoRaWAN Join Procedure for Internet of Things Networks. In *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6, March 2017.

[20] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers. Security Vulnerabilities in LoRaWAN. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 129–140, April 2018.