Enrollment and Security Assessment of LoRaWAN Networks

Fraunhofer AISEC - Hardware Security Department

Florian Jakobsmeier
florian.jakobsmeier@aisec.fraunhofer.de

Munich, March 20, 2019

About Me

- Student at Technical University of Munich
- Master Informatics - 4th semester
- HiWi at Fraunhofer AISEC
  - Hardware Security Department
- LoRaWAN evaluation as research project

Introduction

- What?
- Why?
- How?

Introduction

- What?
  - Sensor networks for private and professional usage
  - LoRaWAN as one popular sensor network protocol
  - Networks send data that needs protection
- Why?
- How?

Introduction

- What?
  - Sensor networks for private and professional usage
  - LoRaWAN as one popular sensor network protocol
  - Networks send data that needs protection
- Why?
  - Wireless network $\rightarrow$ multiple attack vectors
  - Network enrollment $\rightarrow$ introduce attack vectors
  - Public interest in LoRaWAN: Linux Kernel, Stadtwerke München, ...
  - How to secure a network against powerful attacker?
- How?

Introduction

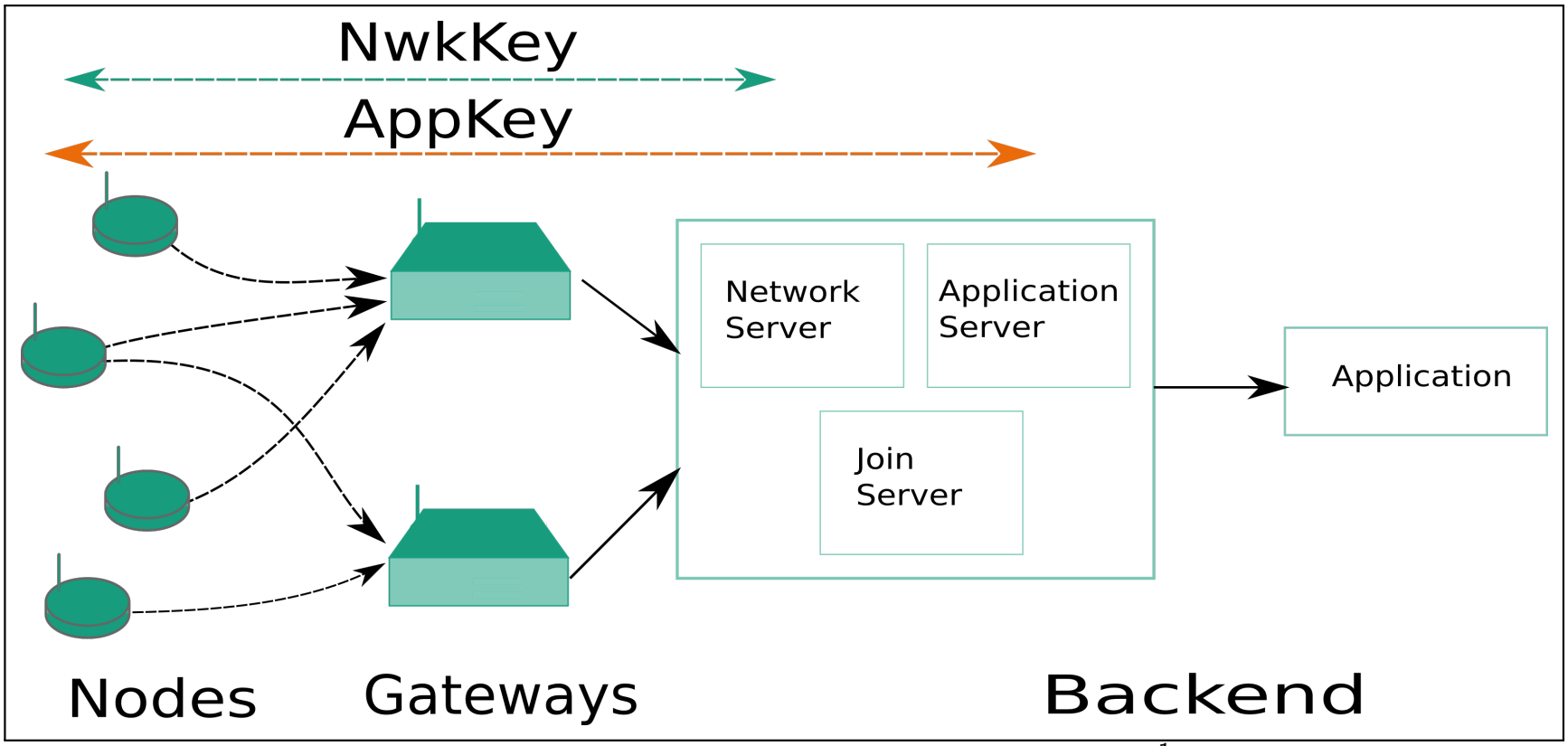
- What?
  - Sensor networks for private and professional usage
  - LoRaWAN as one popular sensor network protocol
  - Networks send data that needs protection
- Why?
  - Wireless network $\rightarrow$ multiple attack vectors
  - Network enrollment $\rightarrow$ introduce attack vectors
  - Public interest in LoRaWAN: Linux Kernel, Stadtwerke München, ...
  - How to secure a network against powerful attacker?
- How?

  $\rightarrow$ Enroll and evaluate LoRaWAN network

Project Goal

- Evaluate LoRaWAN regarding its security aspects
  - Setup a LoRaWAN network
  - Evaluate security of:
    - Protocol
    - Software
    - Hardware
    - Enrollment process

LoRaWAN Basics



LoRaWAN network overview [adapted from: ARM MBed OS[1]]

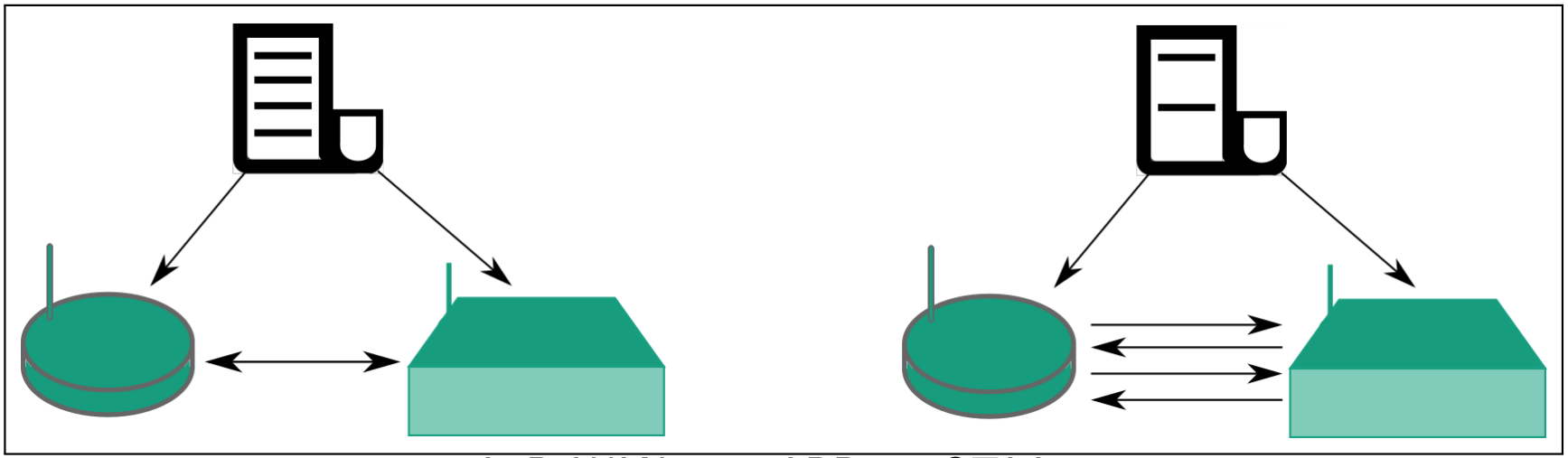
[1] Building your own private LoRa network. https://os.mbed.com/docs/v5.8/reference/building-your-own-private-lora-network.html

LoRaWAN Basics

- Different versions: IDP with v1.1
- Established encryption algorithms: AES-128-{CTR|CMAC}
- Nested security with distinct root keys
  - Network key: integrity protected network data
  - Application key: encrypted application data
- Restricted downlink connection
  - Dependens on node class
  - Class A: listen after send, Class B/C: listen regularly

LoRaWAN Network Join

- Differentiate: static $\leftrightarrow$ dynamic join
  - Activation by personalization (ABP): all security credentials stored on device
  - Over the air activation (OTAA): root keys on device, everything else established dynamically
- Re-Join
  - Node re-joins the network $\rightarrow$ new keys, new counter, ...



LoRaWAN join: ABP vs. OTAA

Network Setup

- Plenty of software and hardware found online
  - Mostly outdated, not supported, not guarantied to work in future
- TheThingsNetwork (TTN) community strives to push LoRaWAN usage
  - Provides: Software and Hardware
  - Most referenced resource provider
  - Most used implementation of LoRaWAN software

→ Use TTN hardware and software as reference point

Protocol

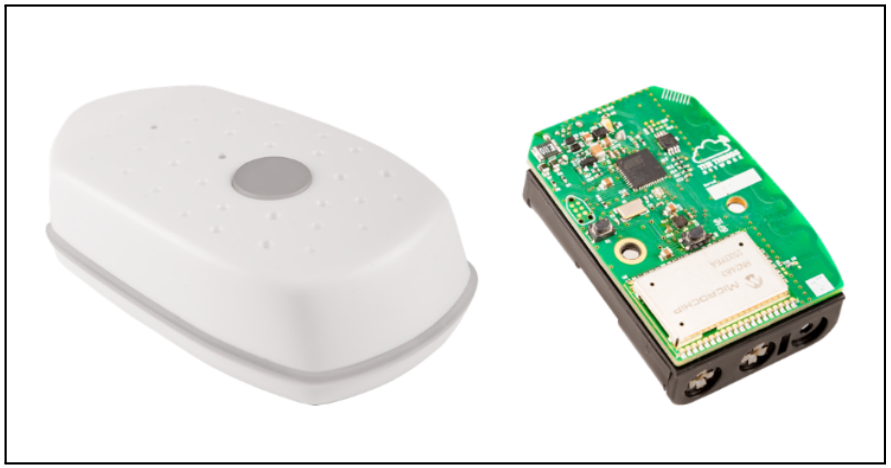
- Use of established algorithms
- Key Management recomendations
  - No key update enforced
  - Possible with Re-Join
- Security as one protocol goal
  - Split root keys from v1.0 to v1.1
  - Many recommendations, but no enforcements

→ LoRaWAN protocol well thought out

→ Security as one update focus

Node Setup

- TheThings Node
- Supports multiple sensors
  - Light-
  - Temperature-
  - Acceleration-
- Stores root and session keys
- Accessible by attacker

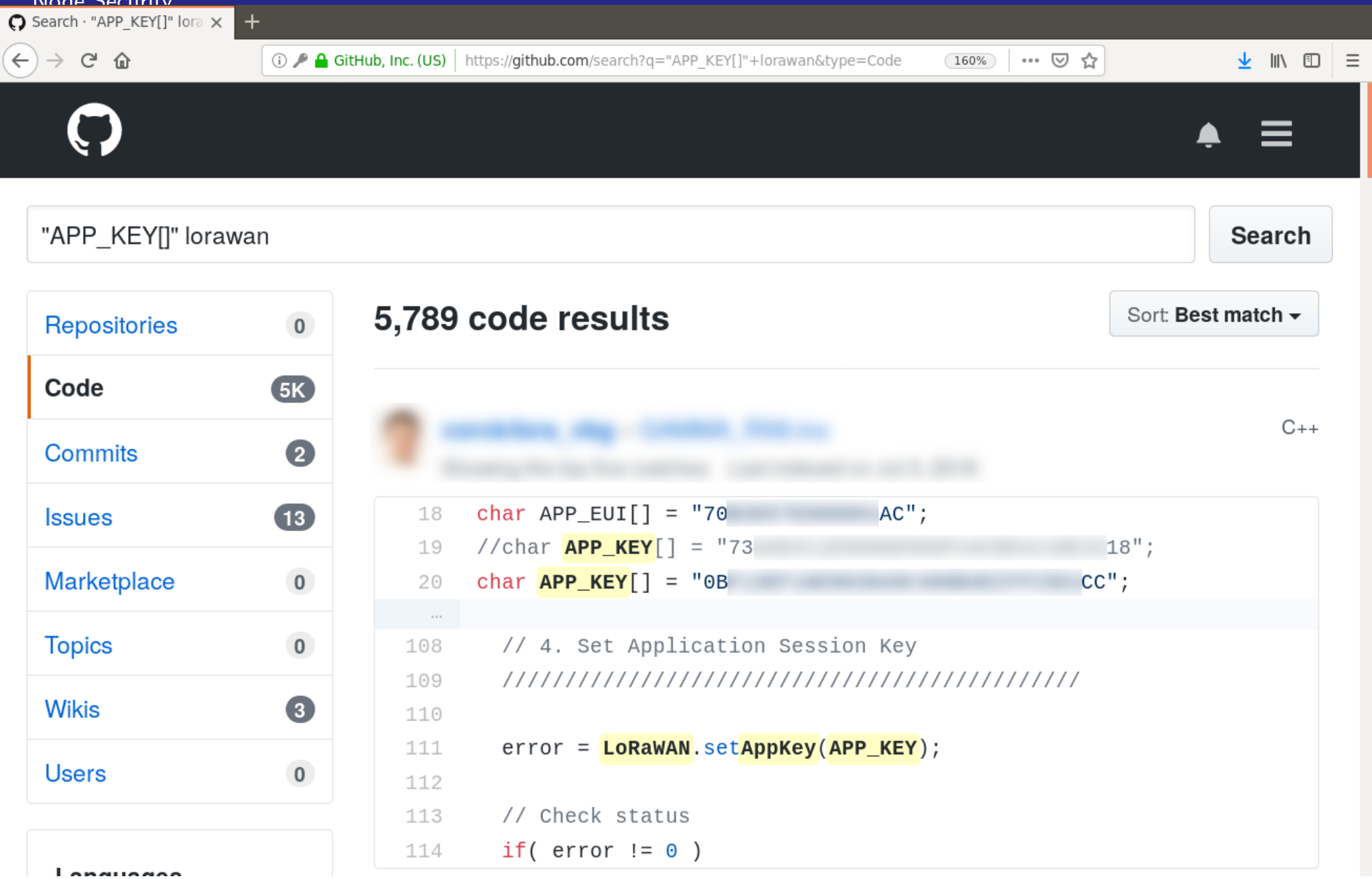


TheThings Node [TheThingsNetwork[2]]

[2] https://www.thethingsnetwork.org/docs/devices/node/

Node Security

- Secure storage is recommended
  - Might not be supported by hardware
  - Easiest and most used solution: store in binary
- Setup nodes with same keys
- Active debug interface
  - Secret credentials printed in plaintext
- Random Number Generator:
  - Suitable for cryptographic purposes?
  - Influenceable by attacker
- Mbed OS node emulator
  - Credentials storage unknown


Search · "APP_KEY[]" lora
GitHub, Inc. (US)
https://github.com/search?q="APP_KEY[]"+lorawan&type=Code
160%
"APP_KEY[]" lorawan
Search
Repositories 0
Code 5K
Commits 2
Issues 13
Marketplace 0
Topics 0
Wikis 3
Users 0
5,789 code results
Sort: Best match
C++
18 char APP_EUI[] = "70 AC";
19 //char APP_KEY[] = "73 18";
20 char APP_KEY[] = "0B CC";
108 // 4. Set Application Session Key
109 //
110
111 error = LoRaWAN.setAppKey(APP_KEY);
112
113 // Check status
114 if(error != 0)

Node Security

- Secure storage is recommended
  - Might not be supported by hardware
  - Easiest and most used solution: store in binary
- Setup nodes with same keys
- Active debug interface
  - Secret credentials printed in plaintext
- Random Number Generator:
  - Suitable for cryptographic purposes?
  - Influenceable by attacker
- Mbed OS node emulator
  - Credentials storage unknown

Gateway Setup

- TheThingsNetwork Gateway
- Plug-and-Play solution
- Register gateway in the backend
- Activation over WiFi
  - Gateway opens Access Point with default password
  - Connection secured with WPA2

TheThings Gateway [TheThingsNetwork[3]]

[3] https://www.thethingsnetwork.org/docs/gateways/gateway/

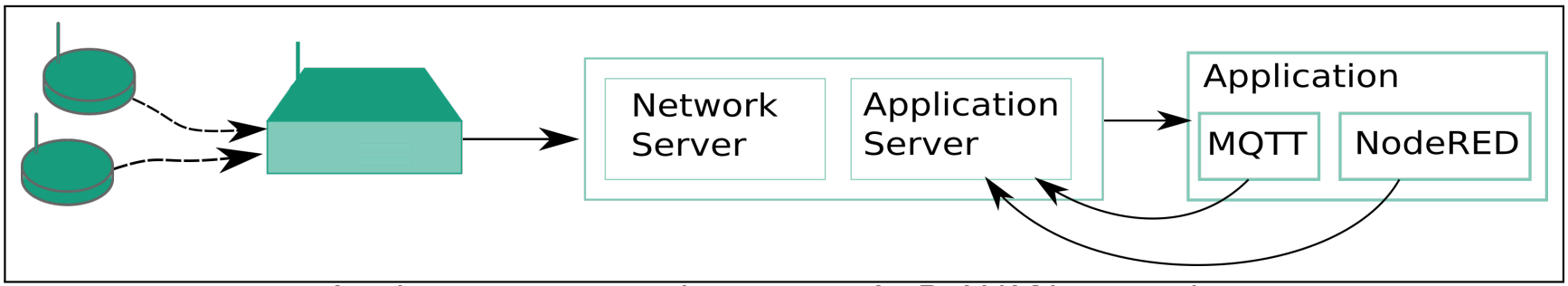
Gateway Security

- Stores gateway key
  - Hardware supported key storage?
  - Key storage on TTN Gateway in binary
- Active debug interface
- Gateway activation using WiFi
  - Not changeable default password
  - Send data not secured otherwise

  → Attack on WPA2 with known PW possible

Backend Setup

- TTN backend gets recommended
- Node-,Gateway- and Application management
- Key and UID generation for all devices
- Hosted on TTN server
- Data access with TTN access token
  - Message Queue Telemetry Transport (MQTT) Protocol
  - NodeRed



Application type and access in LoRaWAN network

Backend Security

- Hosted in cloud
  - Security unknown
  - Access to security credentials unknown
  - Open Source Software $\rightarrow$ build and host backend on own server
- Connection NS $\leftrightarrow$ AS assumed trusted
  - Not necessarily on same device
  - Worst Case: distinct devices, unsecured connection
  - Security on TTN servers not known
- RNG source not known
  - No information about nonce quality
- Backend data access
  - NodeRed: default access via network not secured
  - MQTT: backend credentials in source code

Attacks on LoRaWAN

- Replay attack
  - No *freshness* check for *Join Accept* message
- Jamming
  - Simple: jam frequency
  - Elaborated: e.g. selective jamming
- Key extraction
  - Credentials stored in plaintext in unprotected memory
- RNG
  - Source: traffic on different frequencies
  - Jamming attack: influence nonce values
- Downgrade attacks
  - LoRaWAN specifies backwards compatibility
  - End-Device falls back to lower version
  - Old attack vectors are valid again

Impact on your Network

- Unsafe key storage
  - Stored in: source code or unsafe memory
  - Key extractable $\rightarrow$ attacker gains controll over network
- Active debug interfaces
  - Simple key extraction
- Shared secret key
  - One compromised node $\rightarrow$ effect on whole network
- Jamming (simple and elaborated)
  - Network operation prevented
- Unsafe backend connection
  - (Secrect) Data extractable from traffic
- Random source not suitable or influenced by attacker
  - Encryption can be broken easier

Future Work

- Secure Firmware-over-the-Air (FOTA) support
- Secure key storage on device
  - LoRaWAN recommends secure storage
  - Often keys stored in binary
  - E.g.: Esp32 Flash Encrypt
- Further security checks on working network
  - WiFi access point open $\rightarrow$ entrypoint for an attacker?

Conclusion

- LoRaWAN as one example of sensor network protocols
  - Protocol well thought out
  - LoRaWAN assumed cryptographically secure
- Evaluated reference network
  - Few design issues
  - User errors can enable attack
- Keep common security issues in mind
  - Key management is hard
  - Physical attacks are a threat
  - Select hardware with security mechanisms
  - Consult experts for security concerns