Device re-identification in LoRaWAN through messages linkage

March 2022

Samuel Pélissier, Mathieu Cunche, Vincent Roca, Didier Donsez

INSA - CITI, Inria - Privatics, LIG

Introduction

Growing numbers

- 10 billion of IoT devices
- 225 million of LoRaWAN devices

Countless applications

- Smart home
- Medical and healthcare
- Transportation
- Agriculture
- Energy management
- ...

Evergreen privacy concerns

- Identity
- Location
- Activity

A temperature sensor (Comfort by Adeunis)

Motivations

Assessing privacy protections: linking identifiers using network traces

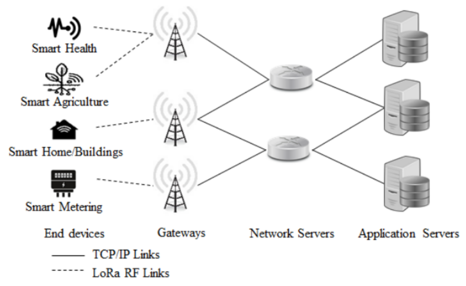
- Tracking users positions using BLE/WiFi [1, 2, 3]
- Fingerprinting devices using the PHY layer [4]
- Inferring activity through metadata [5]

In LoRaWAN

- Linking identifiers: information about the end-device / its application
- Map it to an already known identity, activity, or location
- Passive collection:
  - Cheap for an attacker (100/300$)
  - Easy

LPWAN (Low-Power Wide-Area Network)

- Long range
- Low bit rate
- Low energy consumption
- Low cost

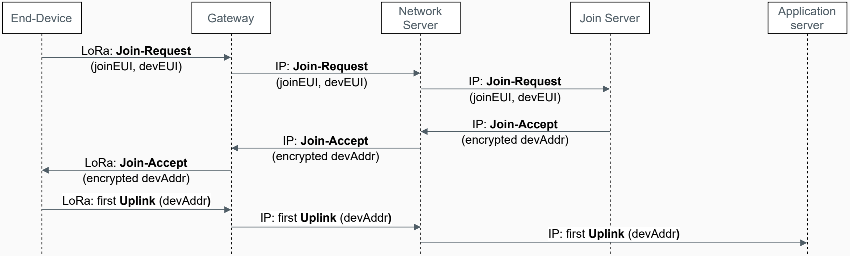


A typical LoRaWAN network architecture (Sundaram et al., 2019)

Background (2) identifiers and activation

Two relevant identifiers

- **DevEUI**: unique for the lifetime of the end-device (MAC address)
  - Only exposed in the Join Request
- **DevAddr**: randomly generated for each session (pseudonym)
  - Only exposed in the Uplink messages



A passive observer has no way to link back the two identifiers

Threat model

- End-device left unmodified
- Encrypted payload
- The attacker is **passive**:
    - Does not inject or alter messages
    - Eavesdrops only the physical link (ED <-> GW)
    - Controls several gateways

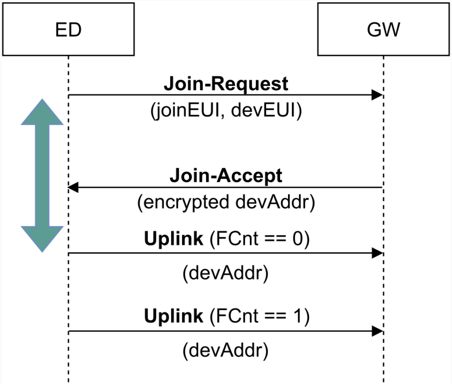
Note: the long range of transmission increases the attack surface but does not change the threat model.

Linking join requests and uplink messages (1)

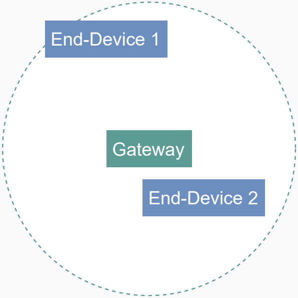
Finding the corresponding following message in a set of uplinks:

time

Type	Identifier	Feature
Join Request	**DevEUI 1**	
Uplink	DevAddr A	Frame Counter: 42
Uplink	**DevAddr B**	**Frame Counter: 0**



Linking join requests and uplink messages (2)



Example of a physical architecture for two end-devices and one gateway

Using distances to compare uplink messages:

time ↓

Type	Identifier	Feature
Join Request	**DevEUI 1**	**RSSI: -42**
Uplink	DevAddr A	RSSI: -100
Uplink	**DevAddr B**	**RSSI: -44**

Linking join requests and uplink messages (3)

Radio

- Estimated Signal Power euclidean distance
- Received Signal Strength Indication euclidean distance
- Signal to Noise Ratio euclidean distance
- Euclidean distance based on gateways receiving the messages

LoRa

- Datarate
- Spreading Factor

LoRaWAN

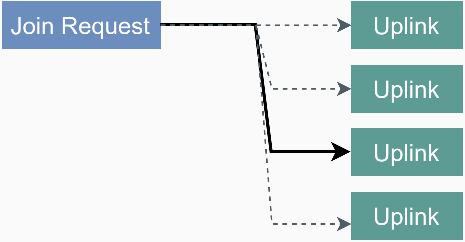
- **Frame Counter**
- Payload length
- OUI extracted from the DevEUI

Application

- Time of arrival difference between Join-Request and the studied Uplink
- Timestamps euclidean distance
- Time of arrival difference between two Uplink messages
  with identical DevAddr

Methodology

Using machine learning for binary classification:



Classifiers

- Decision Tree (DT)
- Naive Bayes (NB)
- Logistic regression (LR)
- K-Nearest Neighbours (kNN)
- Random Forest (RF)
- AdaBoost (AB)
- LightBGM (LBGM)

Experimental results

Classifier	TPR	FPR
RF	**0.7939**	**0.0010**
DT	0.8074	0.0012
AB	0.7973	0.0014
LGBM	0.8074	0.0016
kNN	0.6318	0.0015
NB	0.9595	0.2418

With the frame counter

Classifier	TPR	FPR
RF	0.4493	0.0007
DT	0.5912	0.0028
AB	0.4865	0.0017
LGBM	**0.6453**	**0.0010**
kNN	0.5777	0.0020
NB	0.1115	0.0193

Without the frame counter

- Multiple classifiers provide good performances using the frame counter.
  - 0.8 TPR and 0.001 FPR for the Random Forest classifier.
- Removing the Frame Counter reduces performance.
  - This can be a counter measure.

TPR: True Positive Rate; FPR: False Positive Rate

Counter measures

Obfuscating the frame counter

Hiding the frame counter reduces the attack's performances.

- Encrypting a part of the header containing the frame counter.
- Using a random offset, eg: exchanging the first value of the frame counter during the join procedure.
- Not backward compatible.

Introducing randomness

- Radio-based features (randomly changing the emission power: may loose some messages);
- Time-based features (random delay after receiving the Join Accept);
- Payload length (padding);
- Multiple first uplink messages (decoys [5]).
- Reduces performance.

Obfuscating device identifiers

- Resolvable addresses (eg: BLE);
- Shared DevAddr for multiple end-devices (use NetworkSessionKey for identification).
- Not backward compatible.

Conclusion

- Reliably re-identifying end-devices is possible.
- The Frame Counter is greatly responsible for the attack's performance.
- Counter measures often require to change the LoRaWAN specification.

Merci!

References:

[1] *Tracking Anonymized Bluetooth Devices*. Becker et al.
`https://doi.org/10.2478/popets-2019-0036`

[2] *Linking Bluetooth LE & Classic and Implications for Privacy-Preserving Bluetooth-Based Protocols*. Ludant et al.
`https://doi.org/10.1109/SP40001.2021.00102`

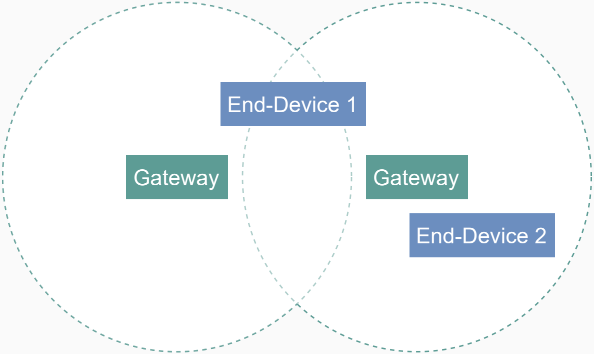
[3] *Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms*. Vanhoef et al.
`https://doi.org/10.1145/2897845.2897883`

[4] *Physical-Layer Fingerprinting of LoRa Devices Using Supervised and Zero-Shot Learning*. Robyns et al. `https://doi.org/10.1145/3098243.3098267`

[5] *I Send, Therefore I Leak: Information Leakage in Low-Power Wide Area Networks*. Leu et al. `https://doi.org/10.1145/3212480.3212508`

[6] *Discovery Privacy Threats via Device De-Anonymization in LoRaWAN*. Spadaccino et al. `https://doi.org/10.1109/MedComNet52149.2021.9501247`

Multiple gateways architecture



Machine learning process