Security Analysis of Zigbee Networks with Zigator and GNU Radio

Dimitrios-Georgios Akestoridis, Madhumitha Harishankar, Michael Weber, and Patrick Tague

Carnegie Mellon University

GNU Radio Conference 2020

Introduction

- The **Zigbee** protocol enables low-rate wireless mesh networking:
  - It is based on the IEEE 802.15.4 standard
  - It is utilized by numerous smart home devices
  - It supports two security models: distributed and centralized

Introduction

- The **Zigbee** protocol enables low-rate wireless mesh networking:
  - It is based on the IEEE 802.15.4 standard
  - It is utilized by numerous smart home devices
  - It supports two security models: distributed and centralized
- The **physical security of smart home residents** can be affected by the security of their Zigbee network

Introduction

- The **Zigbee** protocol enables low-rate wireless mesh networking:
  - It is based on the IEEE 802.15.4 standard
  - It is utilized by numerous smart home devices
  - It supports two security models: distributed and centralized
- The **physical security of smart home residents** can be affected by the security of their Zigbee network
- We recently studied the security consequences of the design choice to disable **MAC-layer security** in centralized Zigbee networks[1]

[1] D.-G. Akestoridis, M. Harishankar, M. Weber, and P. Tague, "Zigator: Analyzing the security of Zigbee-enabled smart homes," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2020, pp. 77–88. DOI: 10.1145/3395351.3399363

Introduction

- The **Zigbee** protocol enables low-rate wireless mesh networking:
  - It is based on the IEEE 802.15.4 standard
  - It is utilized by numerous smart home devices
  - It supports two security models: distributed and centralized
- The **physical security of smart home residents** can be affected by the security of their Zigbee network
- We recently studied the security consequences of the design choice to disable **MAC-layer security** in centralized Zigbee networks[1]
- The primary focus of this talk is on the **design of our testbed**

[1] D.-G. Akestoridis, M. Harishankar, M. Weber, and P. Tague, “Zigator: Analyzing the security of Zigbee-enabled smart homes,” in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2020, pp. 77–88. DOI: 10.1145/3395351.3399363

Packet Sniffing Options

ATUSB (top) and RZUSBSTICK (bottom)

USRP N210 with SBX daughterboard

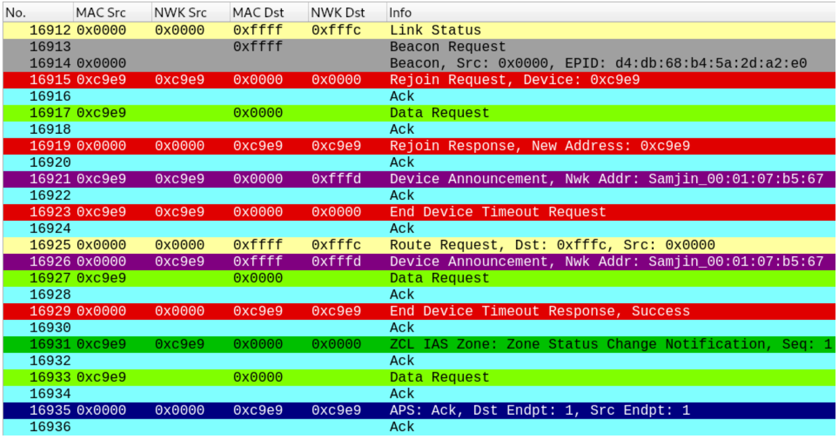
Packet Sniffing Options

ATUSB (top) and RZUSBSTICK (bottom)

USRP N210 with SBX daughterboard

We used a **USRP N210** so that we can also analyze packet jamming attacks

Wireshark Profile for Zigbee Traffic

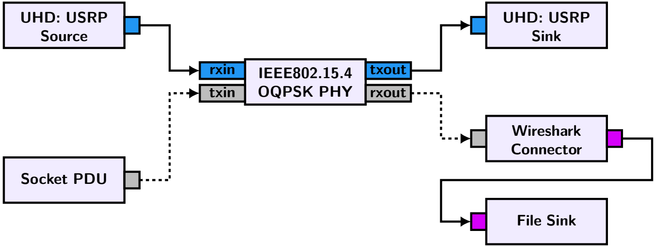


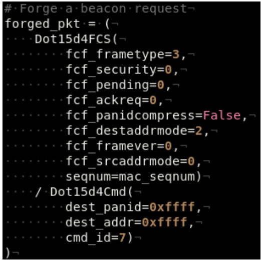
No.	MAC Src	NWK Src	MAC Dst	NWK Dst	Info
16912	0x0000	0x0000	0xffff	0xfffc	Link Status
16913			0xffff		Beacon Request
16914	0x0000				Beacon, Src: 0x0000, EPID: d4:db:68:b4:5a:2d:a2:e0
16915	0xc9e9	0xc9e9	0x0000	0x0000	Rejoin Request, Device: 0xc9e9
16916					Ack
16917	0xc9e9		0x0000		Data Request
16918					Ack
16919	0x0000	0x0000	0xc9e9	0xc9e9	Rejoin Response, New Address: 0xc9e9
16920					Ack
16921	0xc9e9	0xc9e9	0x0000	0xfffd	Device Announcement, Nwk Addr: Samjin_00:01:07:b5:67
16922					Ack
16923	0xc9e9	0xc9e9	0x0000	0x0000	End Device Timeout Request
16924					Ack
16925	0x0000	0x0000	0xffff	0xfffc	Route Request, Dst: 0xfffc, Src: 0x0000
16926	0x0000	0xc9e9	0xffff	0xfffd	Device Announcement, Nwk Addr: Samjin_00:01:07:b5:67
16927	0xc9e9		0x0000		Data Request
16928					Ack
16929	0x0000	0x0000	0xc9e9	0xc9e9	End Device Timeout Response, Success
16930					Ack
16931	0xc9e9	0xc9e9	0x0000	0x0000	ZCL IAS Zone: Zone Status Change Notification, Seq: 1
16932					Ack
16933	0xc9e9		0x0000		Data Request
16934					Ack
16935	0x0000	0x0000	0xc9e9	0xc9e9	APS: Ack, Dst Endpt: 1, Src Endpt: 1
16936					Ack

Profile available at https://github.com/akestoridis/wireshark-zigbee-profile

Packet Injection with GNU Radio and Scapy

- We can use the **gr-ieee802-15-4**[2] and **gr-foo**[3] modules to inject forged Zigbee packets over UDP and store captured Zigbee packets in PCAP format





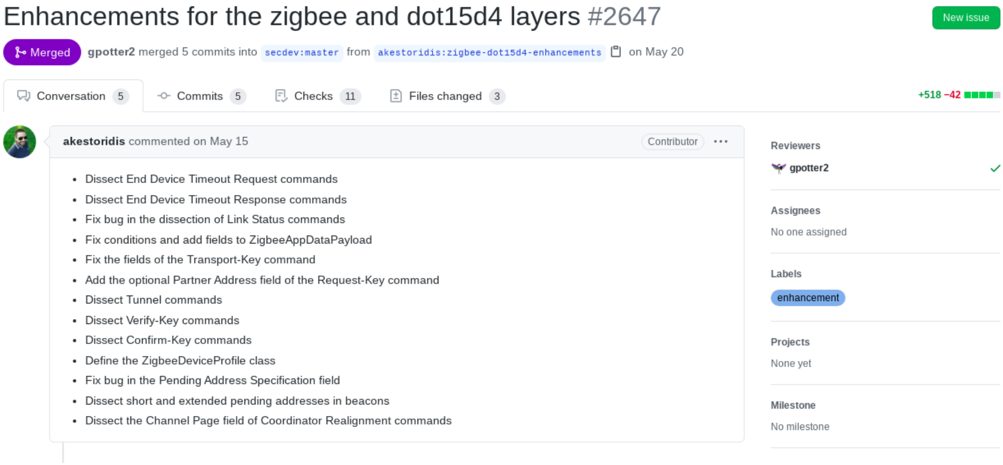
```
# Forge a beacon request
forged_pkt = (
    Dot15d4FCS(
        fcf_frametype=3,
        fcf_security=0,
        fcf_pending=0,
        fcf_ackreq=0,
        fcf_panidcompress=False,
        fcf_destaddrmode=2,
        fcf_framever=0,
        fcf_srcaddrmode=0,
        seqnum=mac_seqnum)
    / Dot15d4Cmd(
        dest_panid=0xffff,
        dest_addr=0xffff,
        cmd_id=7)
)
```

GRC flow graphs available at https://github.com/akestoridis/grc-ieee802154

[2] B. Bloessl. (2020), gr-ieee802-15-4, [Online]. Available: https://github.com/bastibl/gr-ieee802-15-4.

[3] B. Bloessl. (2020), gr-foo, [Online]. Available: https://github.com/bastibl/gr-foo.

Scapy Enhancements



Enhancements for the zigbee and dot15d4 layers #2647

Merged gpotter2 merged 5 commits into `secdev:master` from `akestoridis:zigbee-dot15d4-enhancements` on May 20

akestoridis commented on May 15

- Dissect End Device Timeout Request commands
- Dissect End Device Timeout Response commands
- Fix bug in the dissection of Link Status commands
- Fix conditions and add fields to ZigbeeAppDataPayload
- Fix the fields of the Transport-Key command
- Add the optional Partner Address field of the Request-Key command
- Dissect Tunnel commands
- Dissect Verify-Key commands
- Dissect Confirm-Key commands
- Define the ZigbeeDeviceProfile class
- Fix bug in the Pending Address Specification field
- Dissect short and extended pending addresses in beacons
- Dissect the Channel Page field of Coordinator Realignment commands

Reviewers
gpotter2 ✓

Assignees
No one assigned

Labels
enhancement

Projects
None yet

Milestone
No milestone

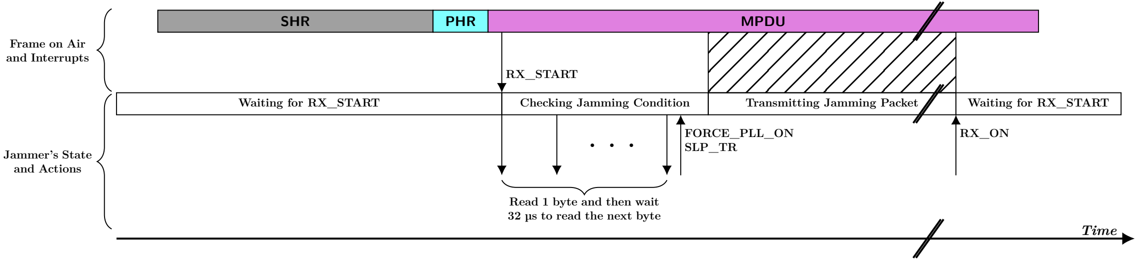
Source: https://github.com/secdev/scapy/pull/2647

Launching Attacks with an ATUSB

- We modified the firmware of an ATUSB in order to enable:
  1. The injection of **time-critical** Zigbee packets
  2. The **selective jamming** of Zigbee packets

Launching Attacks with an ATUSB

- We modified the firmware of an ATUSB in order to enable:
  1. The injection of **time-critical** Zigbee packets
  2. The **selective jamming** of Zigbee packets
- High-level description of our implementation of a selective jammer:



Frame on Air and Interrupts
SHR
PHR
MPDU
RX_START
Jammer's State and Actions
Waiting for RX_START
Checking Jamming Condition
Transmitting Jamming Packet
Waiting for RX_START
FORCE_PLL_ON
SLP_TR
RX_ON
. . .
Read 1 byte and then wait 32 µs to read the next byte
Time

Modified firmware available at https://github.com/akestoridis/atusb-attacks

Packet Analysis with Zigator

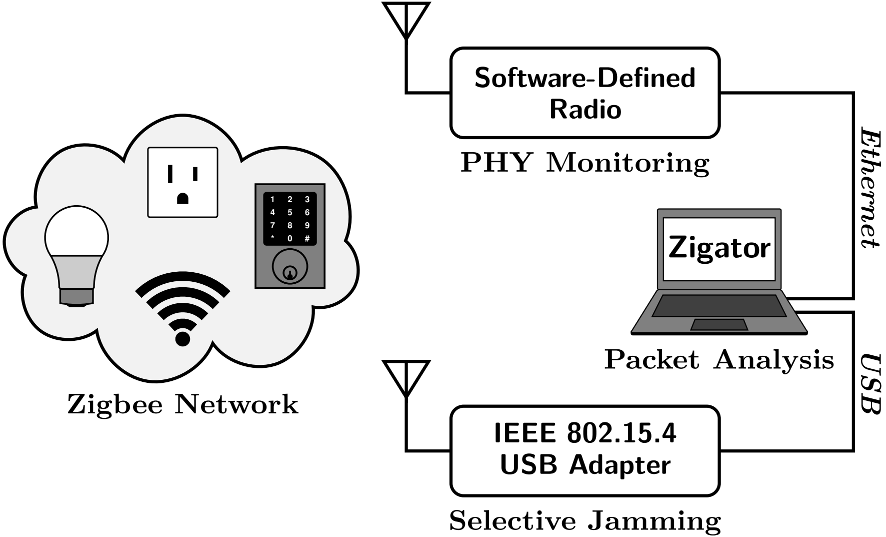
- Selected dependencies of Zigator:
  - **Scapy** ⇒ Parsing and forging of Zigbee packets
  - **PyCryptodome** ⇒ Implementation of the AES cipher
  - **Scikit-learn** ⇒ Training of decision tree classifiers

Packet Analysis with Zigator

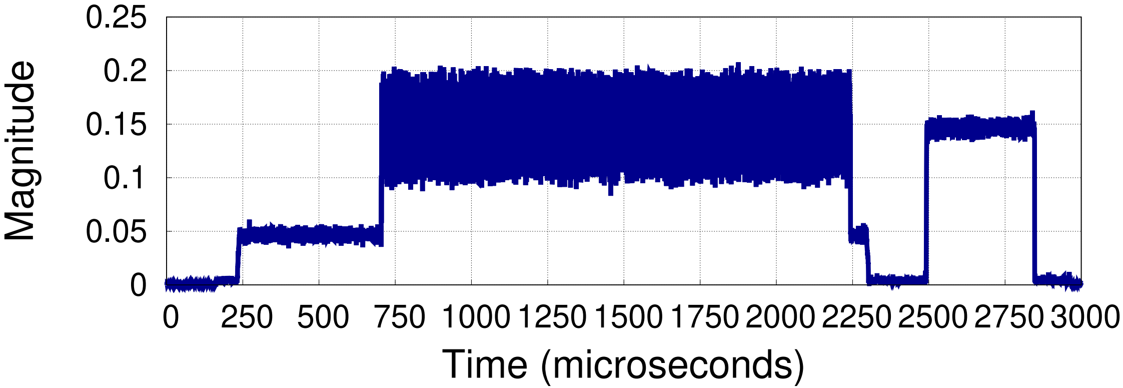
- Selected dependencies of Zigator:
  - **Scapy** ⇒ Parsing and forging of Zigbee packets
  - **PyCryptodome** ⇒ Implementation of the AES cipher
  - **Scikit-learn** ⇒ Training of decision tree classifiers
- Selected **features** of Zigator:
  - Derive preconfigured Trust Center link keys from install codes
  - Decrypt and verify Zigbee packets
  - Encrypt and authenticate Zigbee packets
  - Infer information from captured Zigbee packets
  - Inject forged packets over UDP
  - Launch selective jamming and spoofing attacks with an ATUSB

Zigator source code available at https://github.com/akestoridis/zigator

Testbed Overview

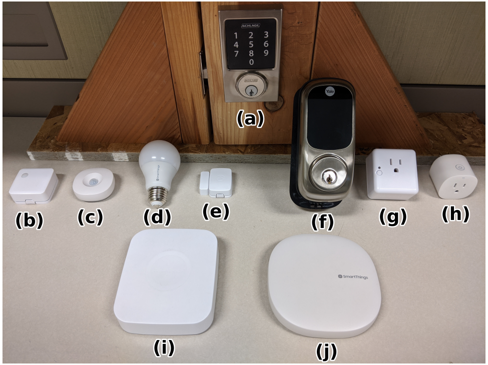


Captured I/Q Signal during an Attack

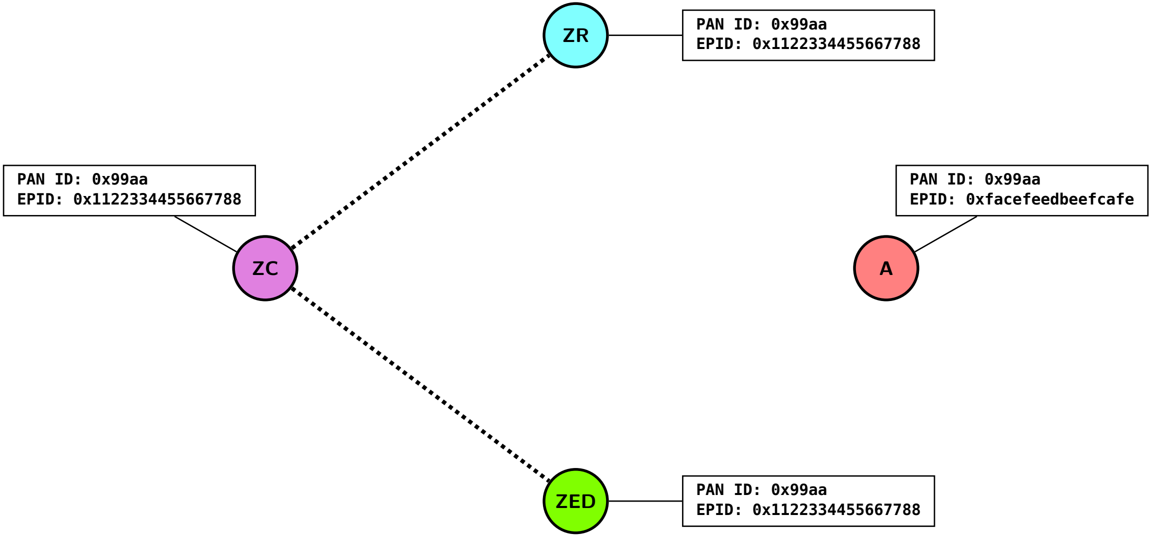


CRAWDAD dataset cmu/zigbee-smarthome

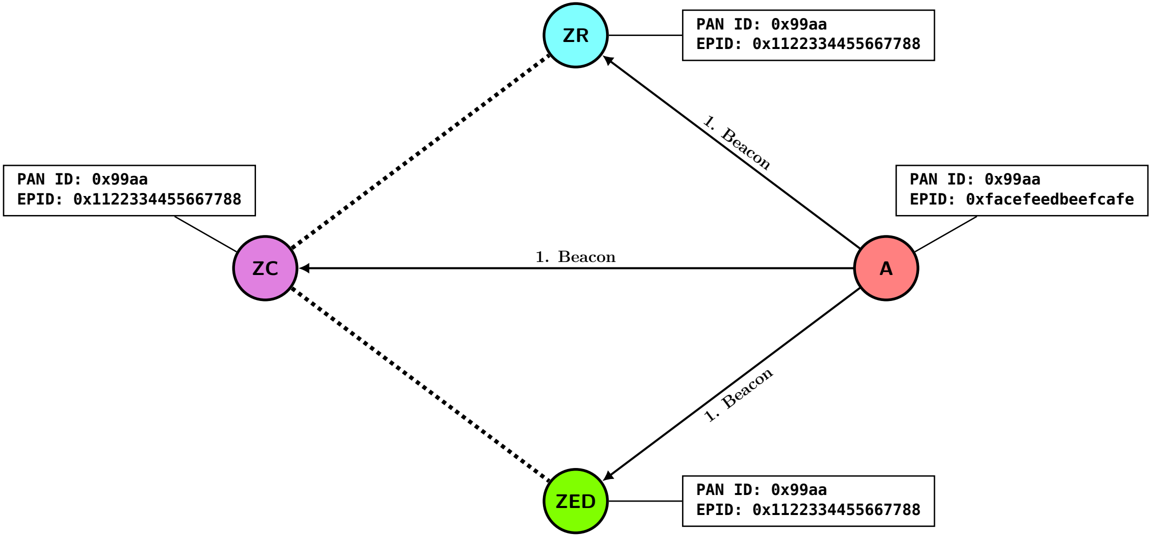
- We captured packets that were generated from **ten commercial Zigbee devices**
- Our experiments lasted about 34.644 hours in total and resulted in a dataset of **571,509 valid packets**
- Our dataset is available to download from the **CRAWDAD research data archive**:
  - https://doi.org/10.15783/c7-nvc6-4q28



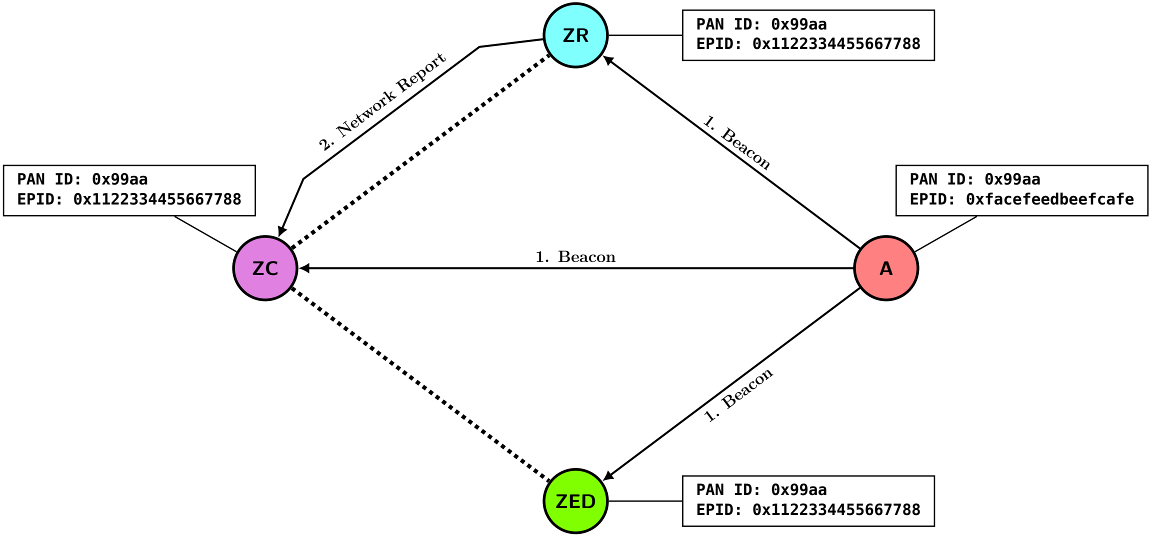
Disconnecting Zigbee Devices



Disconnecting Zigbee Devices



Disconnecting Zigbee Devices



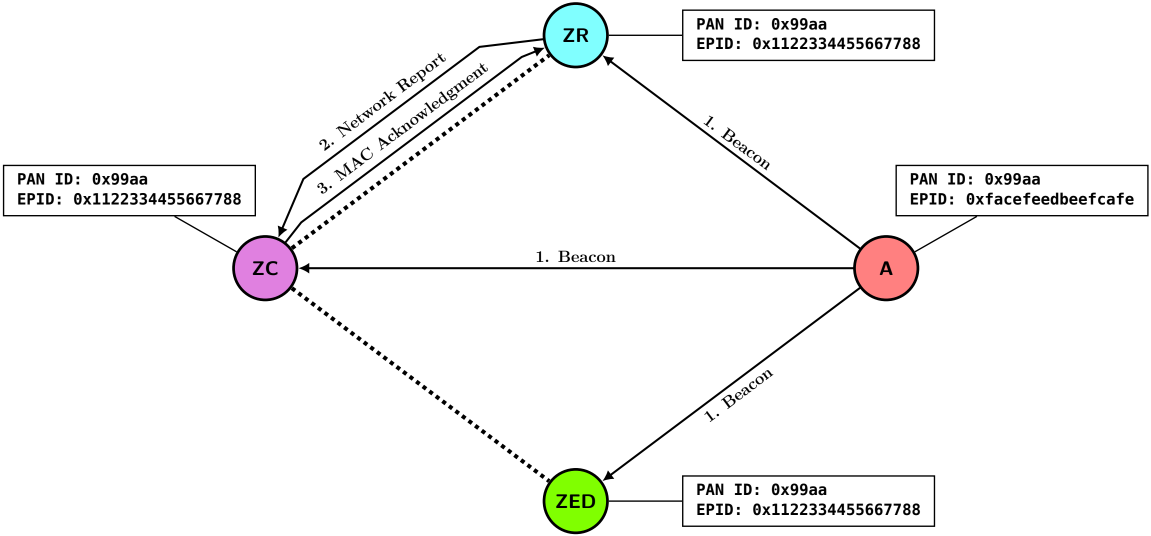
PAN ID: 0x99aa
EPID: 0x1122334455667788

PAN ID: 0x99aa
EPID: 0x1122334455667788

PAN ID: 0x99aa
EPID: 0xfacefeedbeefcafe

PAN ID: 0x99aa
EPID: 0x1122334455667788

Disconnecting Zigbee Devices



PAN ID: 0x99aa
EPID: 0x1122334455667788

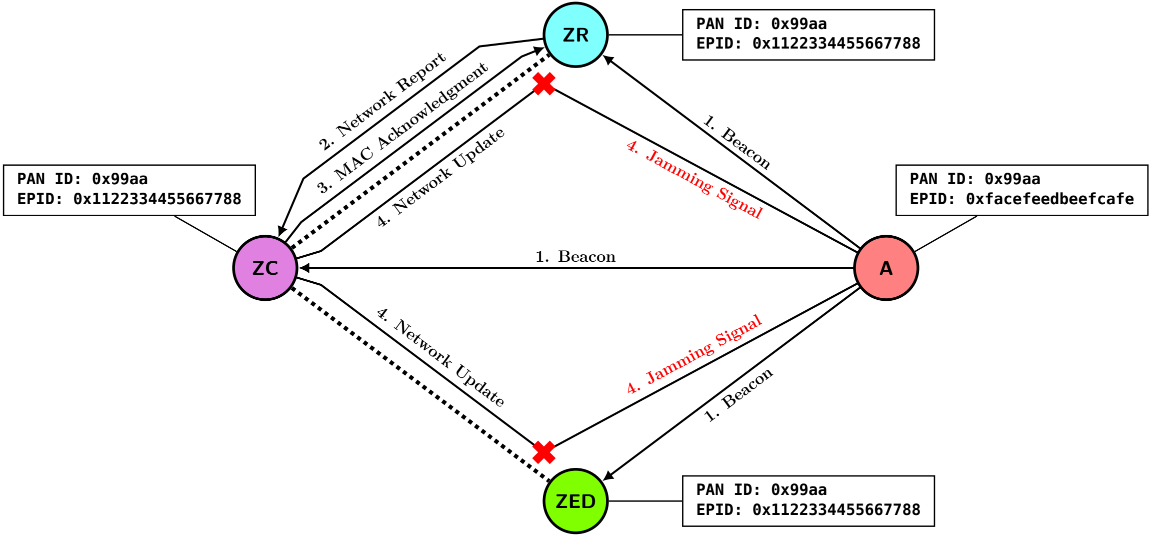
PAN ID: 0x99aa
EPID: 0x1122334455667788

PAN ID: 0xfacefeedbeefcafe

PAN ID: 0x99aa
EPID: 0xfacefeedbeefcafe

PAN ID: 0x99aa
EPID: 0x1122334455667788

Disconnecting Zigbee Devices



ZR

PAN ID: 0x99aa
EPID: 0x1122334455667788

ZC

PAN ID: 0x99aa
EPID: 0x1122334455667788

A

PAN ID: 0x99aa
EPID: 0xfacefeedbeefcafe

ZED

PAN ID: 0x99aa
EPID: 0x1122334455667788

2. Network Report

3. MAC Acknowledgment

4. Network Update

1. Beacon

4. Jamming Signal

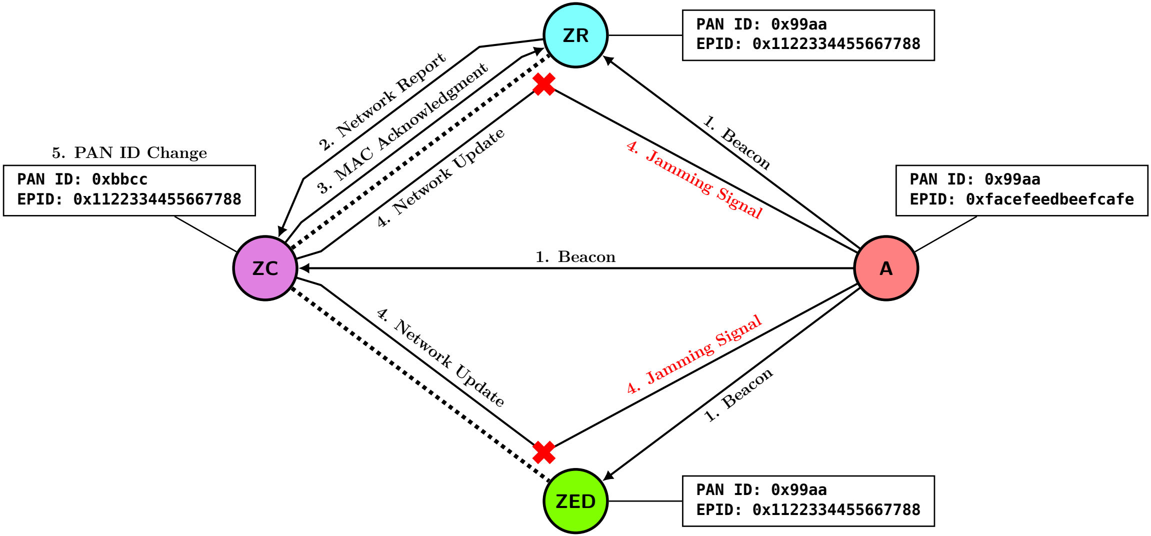
1. Beacon

4. Network Update

4. Jamming Signal

1. Beacon

Disconnecting Zigbee Devices



Summary of Findings and Developments

- Options for **keeping** Zigbee devices disconnected:
  - Spoofing of MAC acknowledgments
  - Selective jamming of Rejoin Response commands
  - Selective jamming of beacons

Summary of Findings and Developments

- Options for **keeping** Zigbee devices disconnected:
  - Spoofing of MAC acknowledgments
  - Selective jamming of Rejoin Response commands
  - Selective jamming of beacons
- We observed that some Zigbee Routers either did not initiate or significantly delayed the **rejoin process** when Network Update commands are jammed:
  - Our SmartThings Smart Bulb did not initiate that process within 38 hours
  - Our Centralite 3-Series Smart Outlet delayed that process for about 25 minutes

Summary of Findings and Developments

- Options for **keeping** Zigbee devices disconnected:
  - Spoofing of MAC acknowledgments
  - Selective jamming of Rejoin Response commands
  - Selective jamming of beacons
- We observed that some Zigbee Routers either did not initiate or significantly delayed the **rejoin process** when Network Update commands are jammed:
  - Our SmartThings Smart Bulb did not initiate that process within 38 hours
  - Our Centralite 3-Series Smart Outlet delayed that process for about 25 minutes
- We responsibly disclosed our findings to the **Zigbee Alliance**:
  - Specification changes will prevent malicious PAN ID changes
  - The firmware of SmartThings hubs was modified to ignore PAN ID conflicts[4]

[4] SmartThings Community. (2020), Hub firmware release notes - 0.31.4, [Online]. Available: https://community.smartthings.com/t/hub-firmware-release-notes-0-31-4/197941

Conclusion

- Our testbed design enables in-depth security analysis of Zigbee networks:
  - Packet Sniffing ⟹ Software-Defined Radio
  - Packet Injection ⟹ Software-Defined Radio and IEEE 802.15.4 USB Adapter
  - Packet Jamming ⟹ IEEE 802.15.4 USB Adapter
  - Packet Analysis ⟹ Zigator
- Additional resources:
  - http://mews.sv.cmu.edu/research/zigator/
- Questions?
  - `{akestoridis, mharisha, mikex, tague}@cmu.edu`