Exploiting WPA3 Networks: New Vulnerabilities and Defenses

Mathy Vanhoef

POC2021 conference, Nov 11-12, 2021

Let’s start with some history

1971 ALOHANet: the 1st wireless packet data network

1997 Initial release of 802.11

- Later called Wi-Fi

1999 Wired Equivalent Privacy (WEP)

- Horribly **broken**

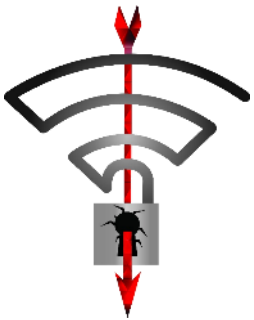
Advancements in Wi-Fi security

Early 2000 Wi-Fi Protected Access (WPA and WPA2)

- Vulnerable to offline **dictionary attacks**
- ~2009: minor attack against WPA1
- ~2016: privacy concerns about tracking
- Overall, long period with few major advancements

Advancements in Wi-Fi security

2017 Key reinstallation attacks (KRACK)



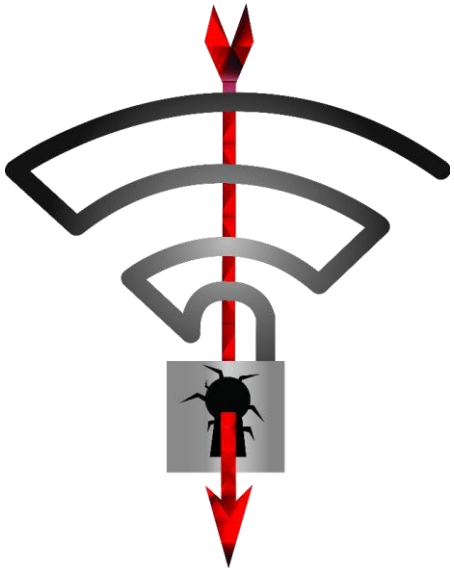
- Flaw in the standard → all devices affected
- Motivated standard bodies to improve Wi-Fi security

Advancements in Wi-Fi security

2018 Wi-Fi Protected Access 3 (WPA3)

Added handshake to prevent dictionary attacks

- Internally converts password to crypto element
- Conversions takes variable number of iterations
- Now updated with constant-time conversion

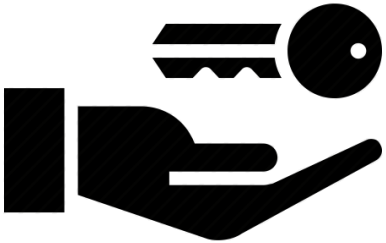


Key reinstallations against WPA2

WPA2: 4-way handshake

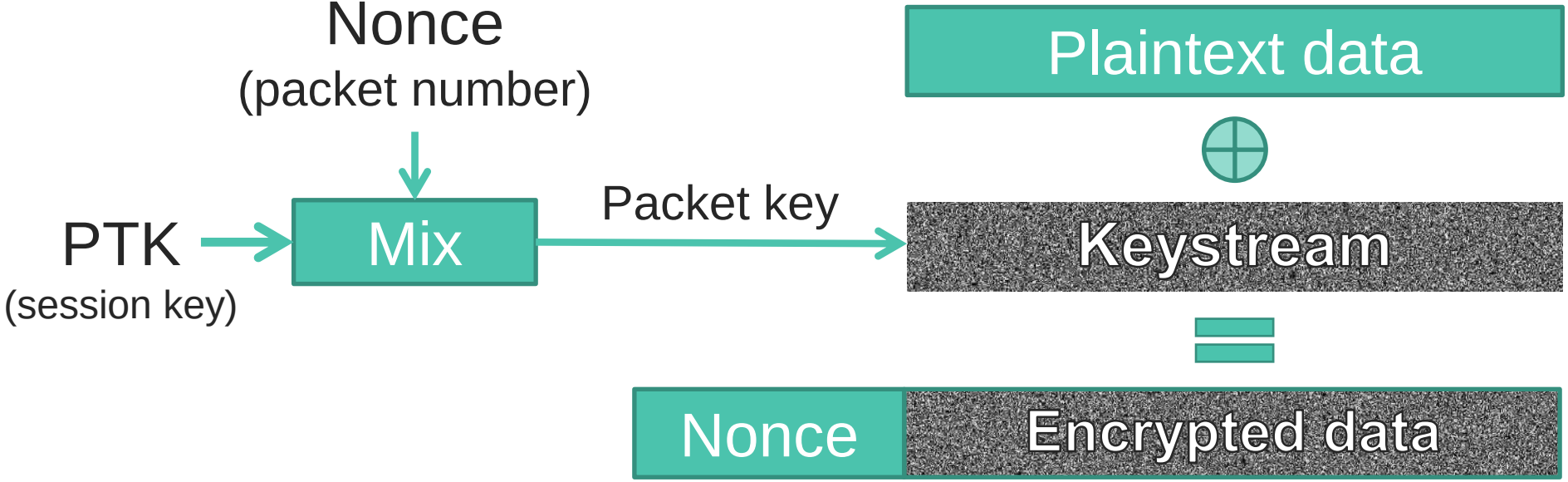
Used to connect to any protected Wi-Fi network

Mutual authentication



Negotiates fresh PTK:
pairwise transient key

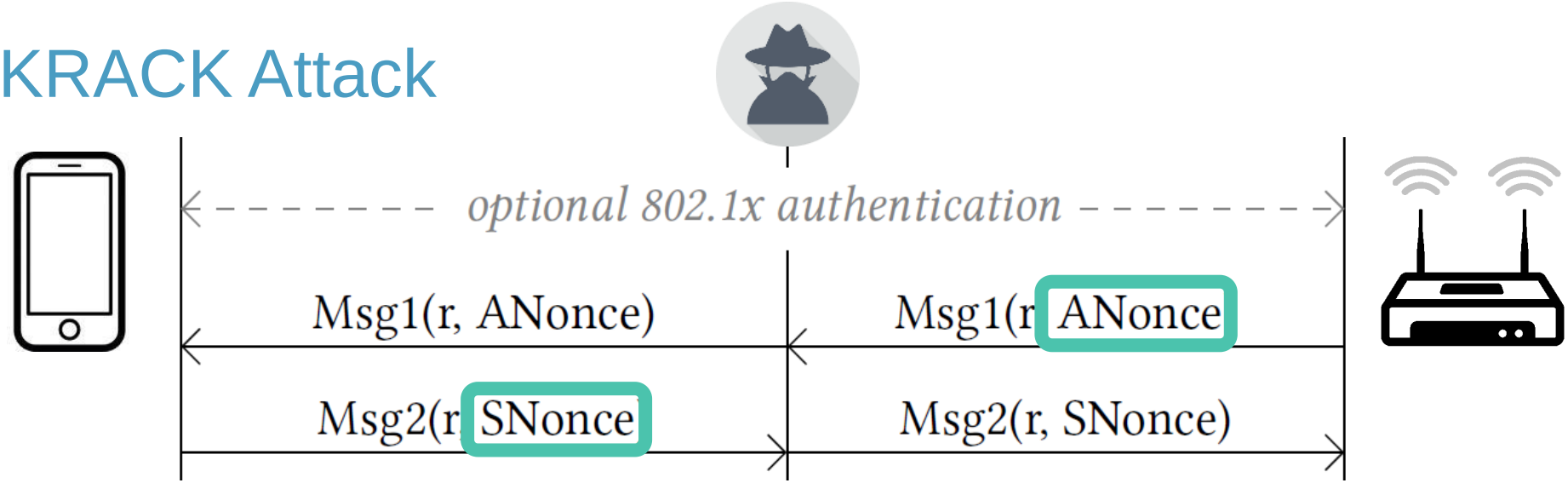
WPA2: Encryption algorithm



→ Nonce reuse implies keystream reuse (in all WPA2 ciphers)

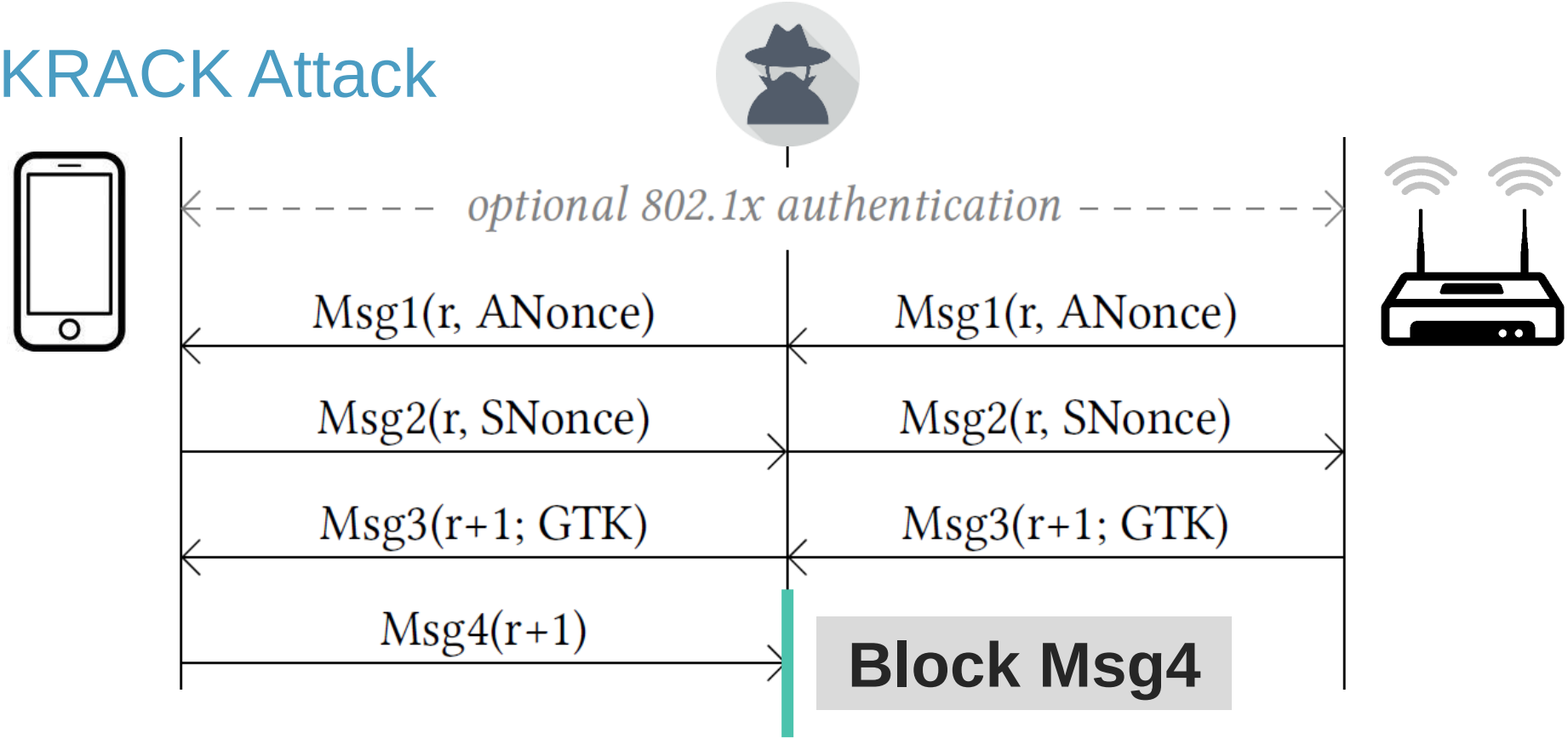
KRACK Attack

KRACK Attack

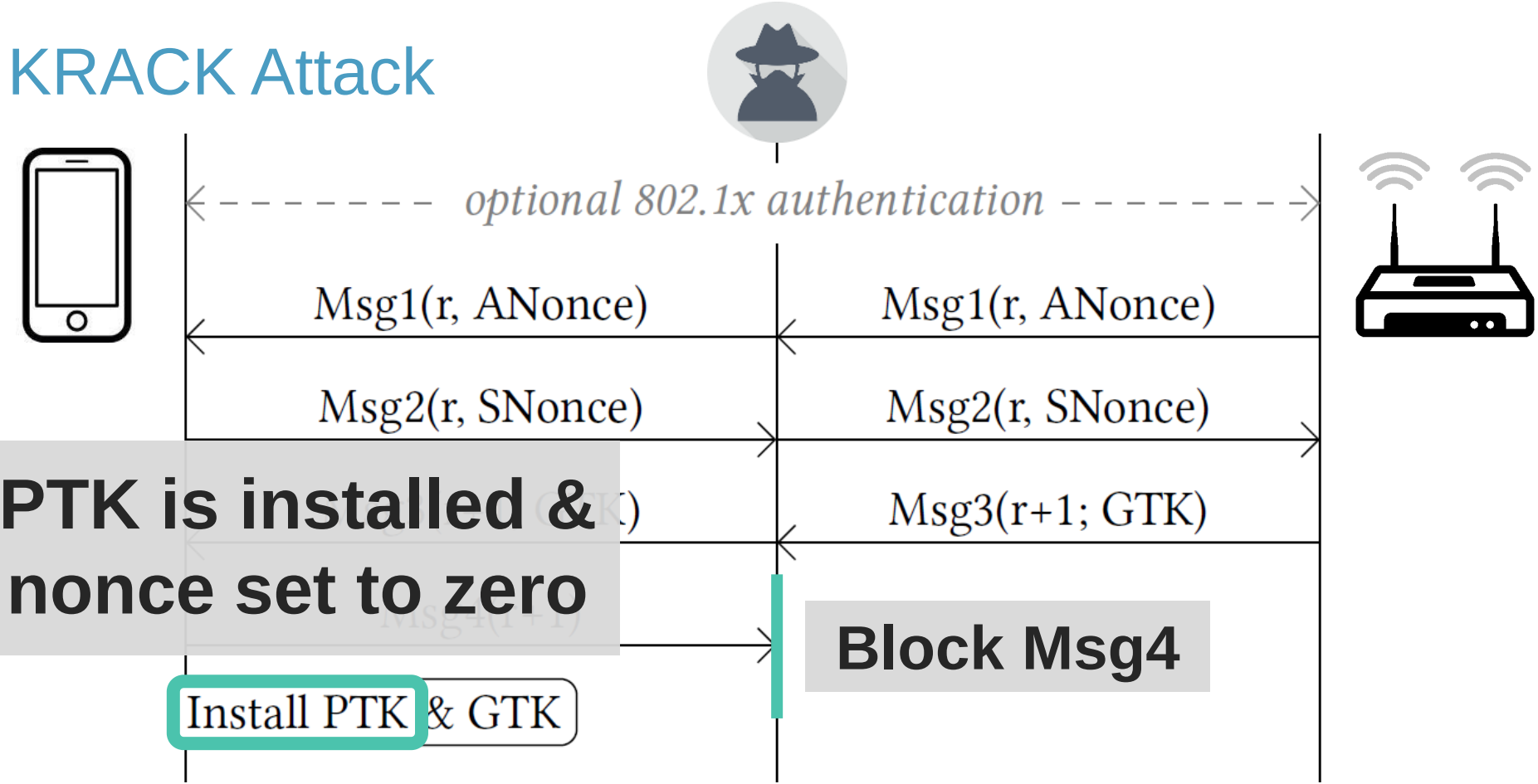


PTK = Combine(shared secret, ANonce, SNonce)

KRACK Attack



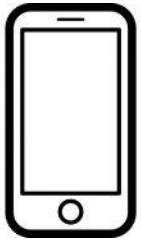
KRACK Attack

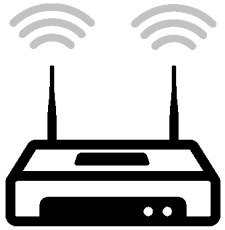


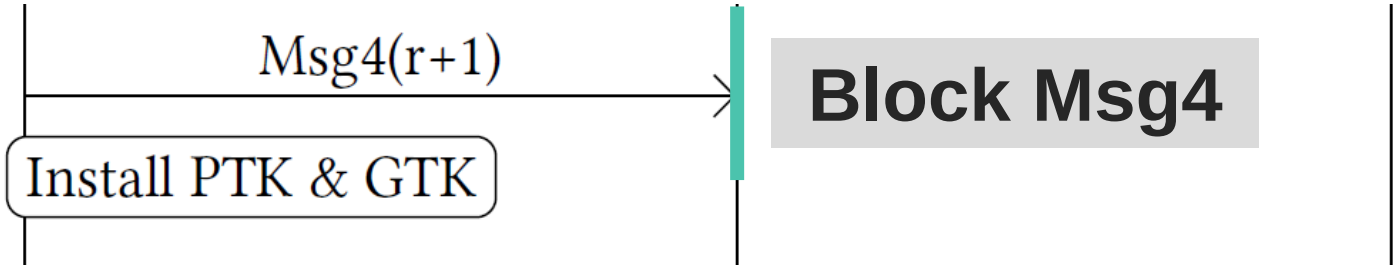
PTK is installed & nonce set to zero

Block Msg4

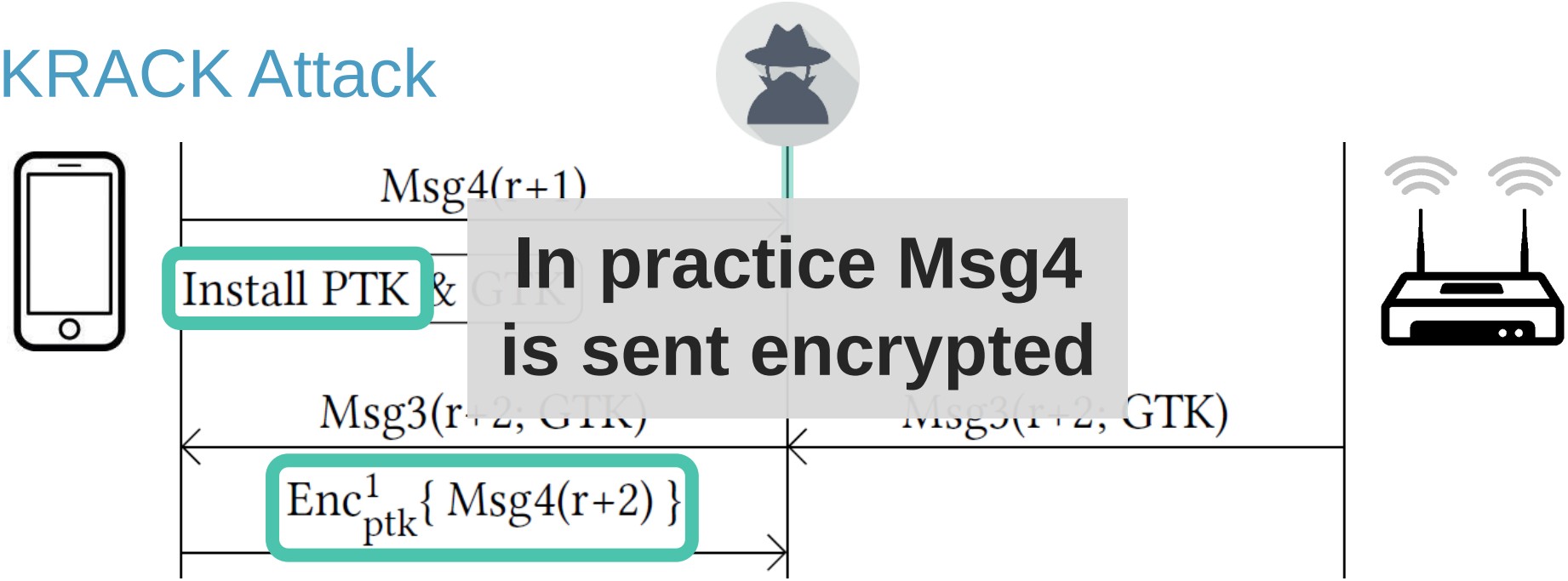
KRACK Attack





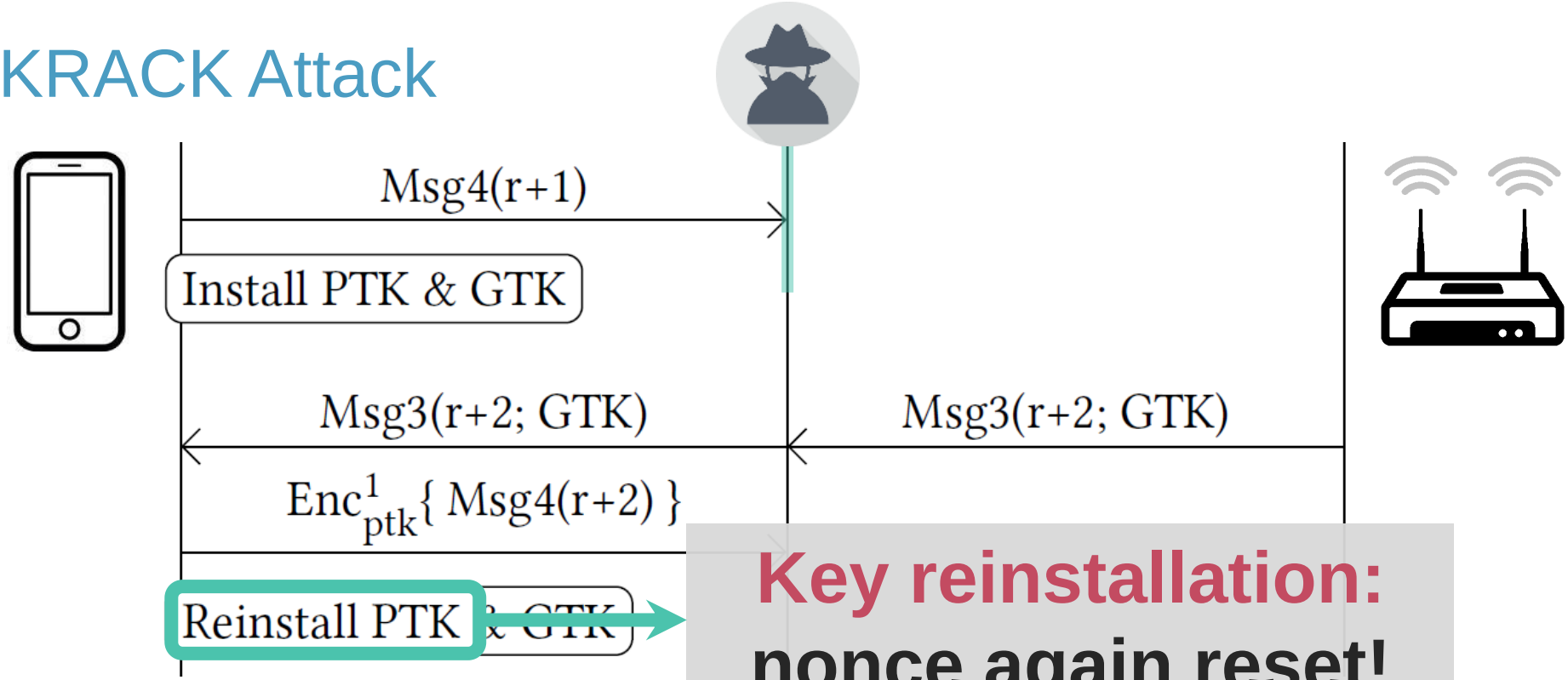


KRACK Attack



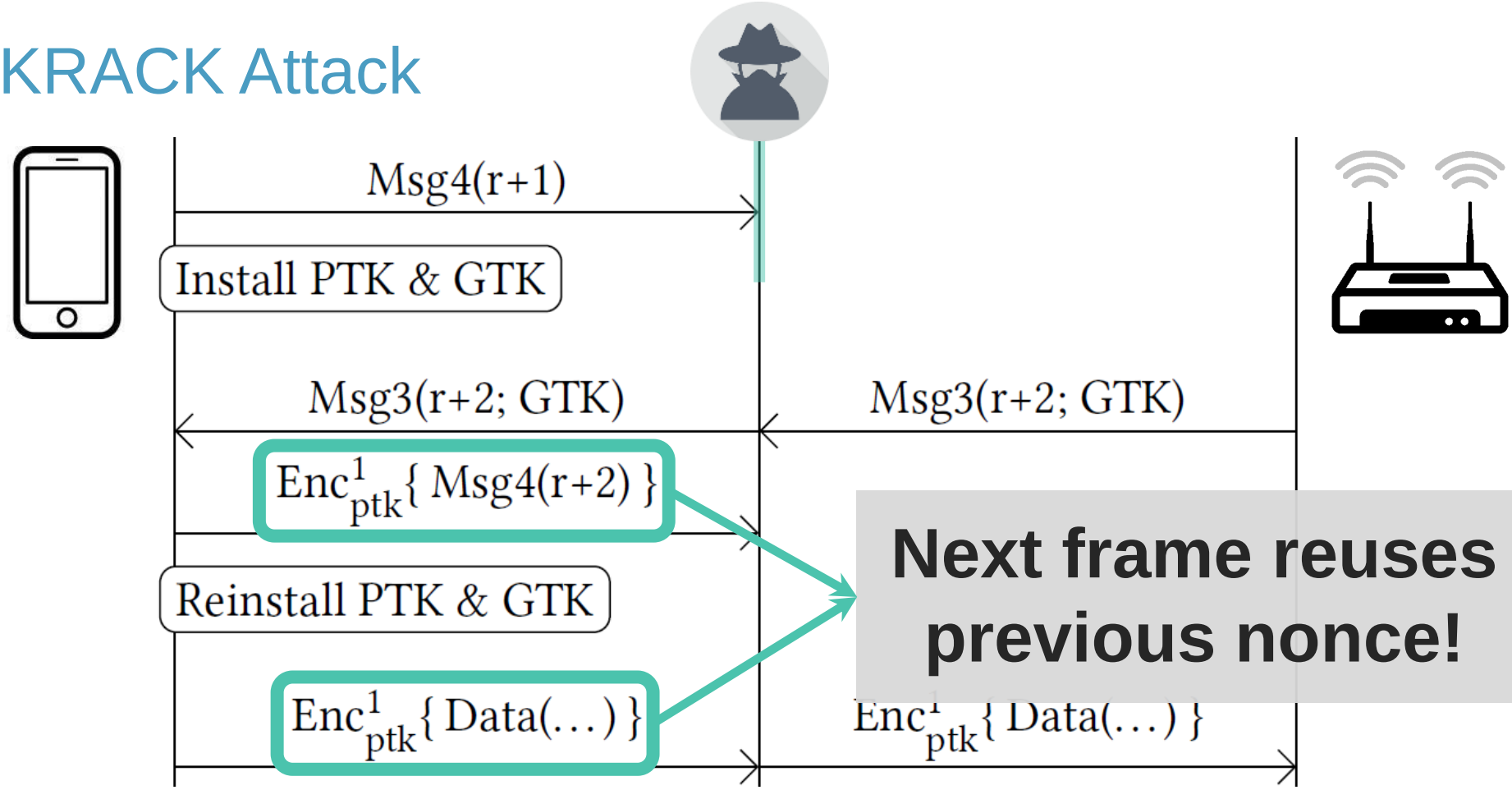
In practice Msg4 is sent encrypted

KRACK Attack

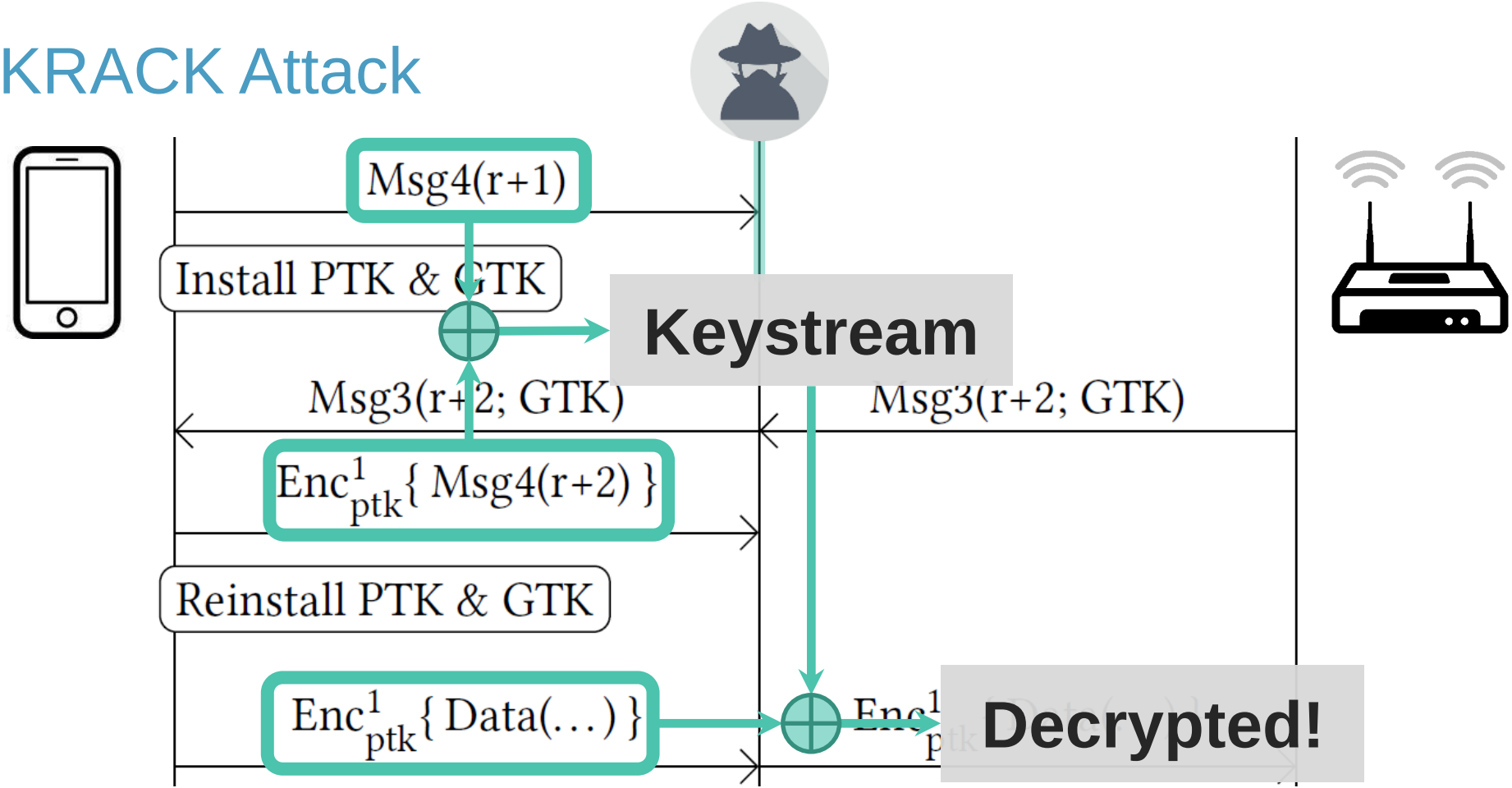


Key reinstallation:
nonce again reset!

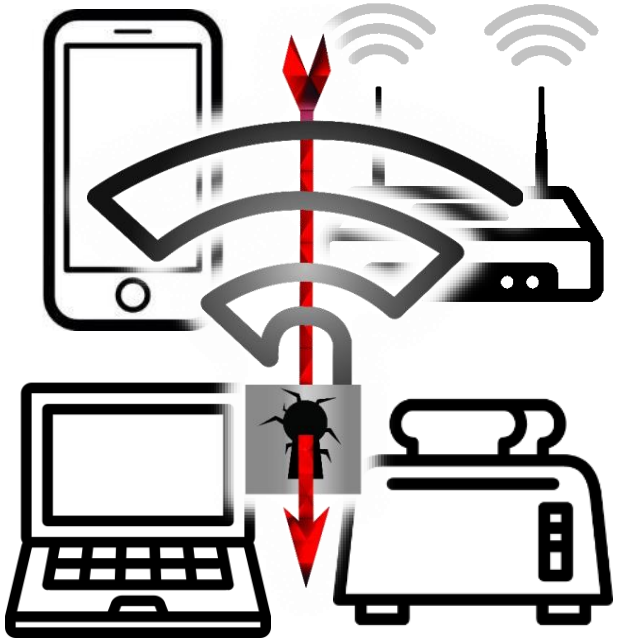
KRACK Attack



KRACK Attack



Security impact



Decrypt frames sent by victim

Replay frames towards victim

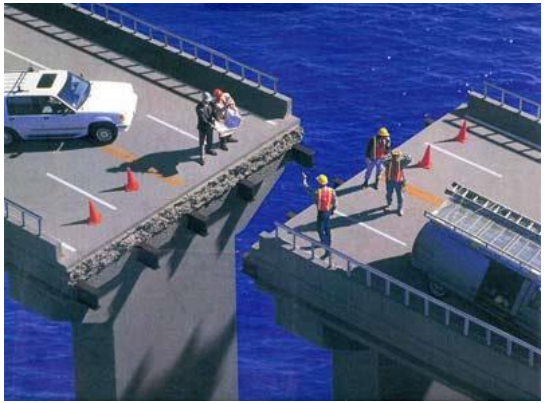
Forge frames towards/from victim

(only against TKIP/GCMP)

Root cause

- 4-way handshake proven secure
- Encryption protocol proven secure

} Combined in a state machine



State machine was not proven secure!

World-wide impact

- Affects all Wi-Fi devices that support encryption
- Caused several **updates to the IEEE 802.11 standard**
- After our findings, Wi-Fi Alliance released **WPA3**
  - **WPA3 still uses the 4-way handshake** (after the new handshake)
  - This means WPA3 implementations can still be vulnerable to KRACK

Fragmention & Aggregation Attacks

Design flaws

Implementation Flaws

Aggregation

Mixed key

Fragment cache

Implementation Flaws

Background

Sending small frames causes high overhead:

This can be avoided by **aggregating frames**:

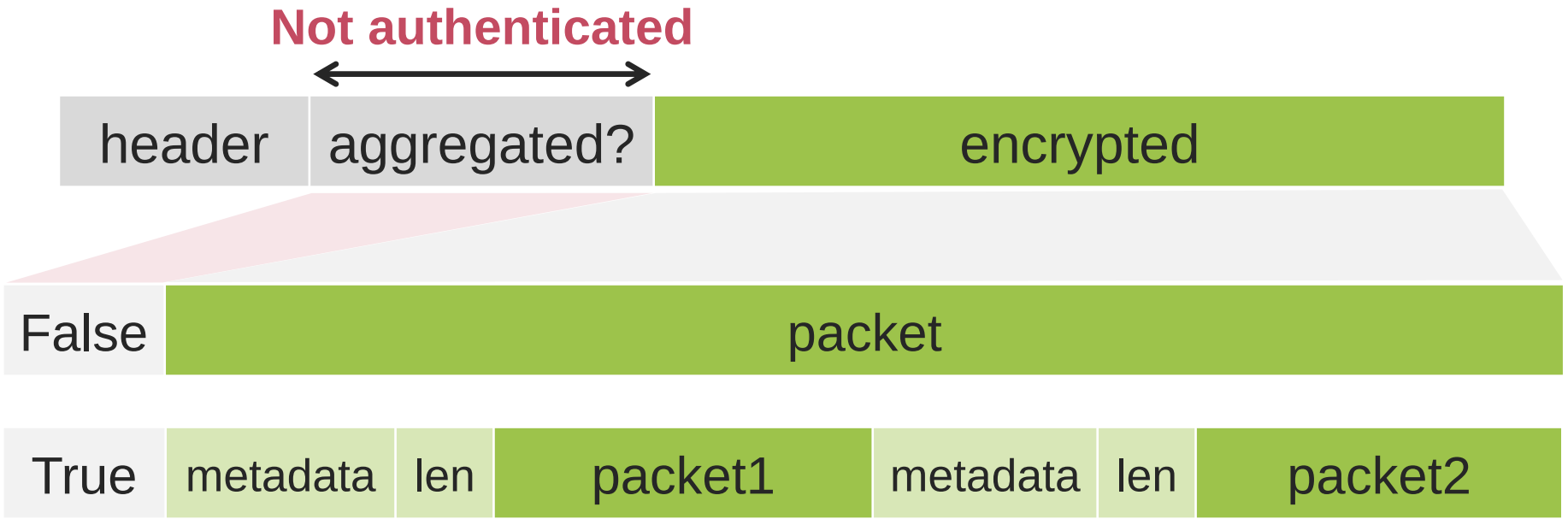
Background

Sending small frames causes high overhead:

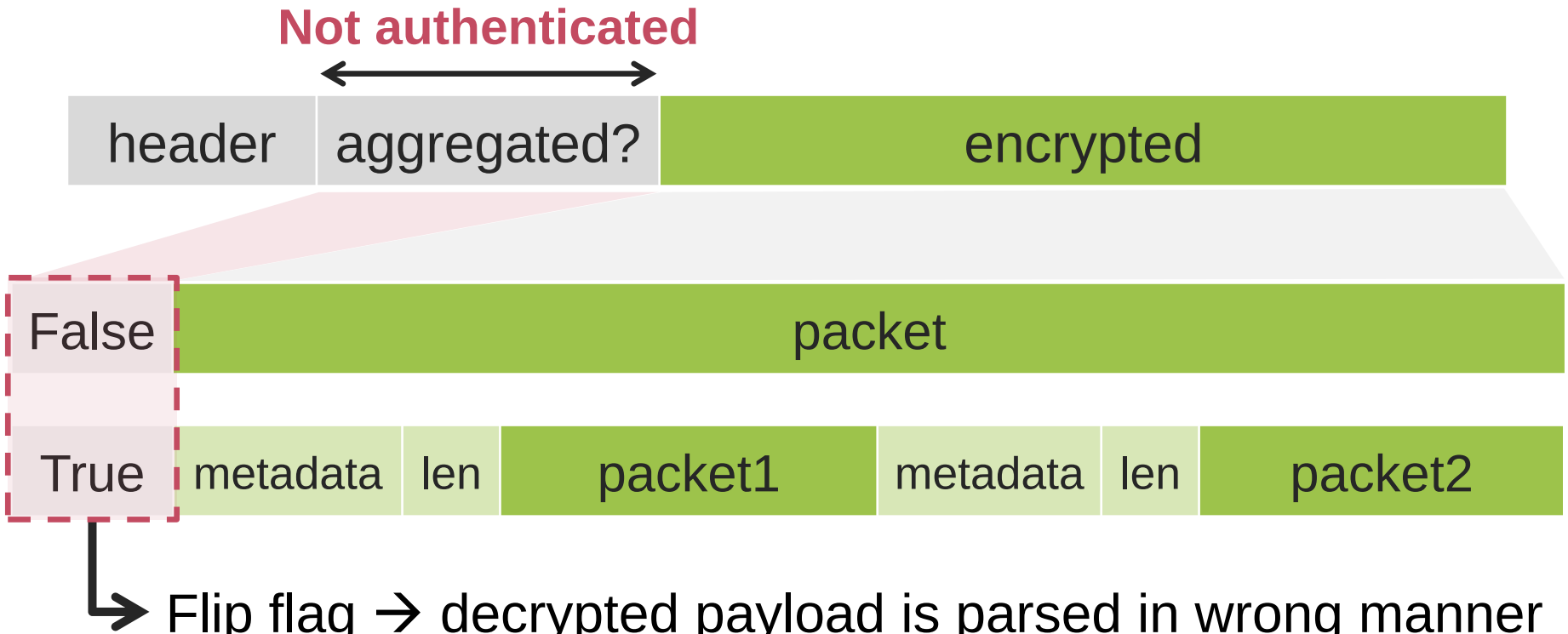
This can be avoided by **aggregating frames**:

Problem: how to recognize aggregated frames?

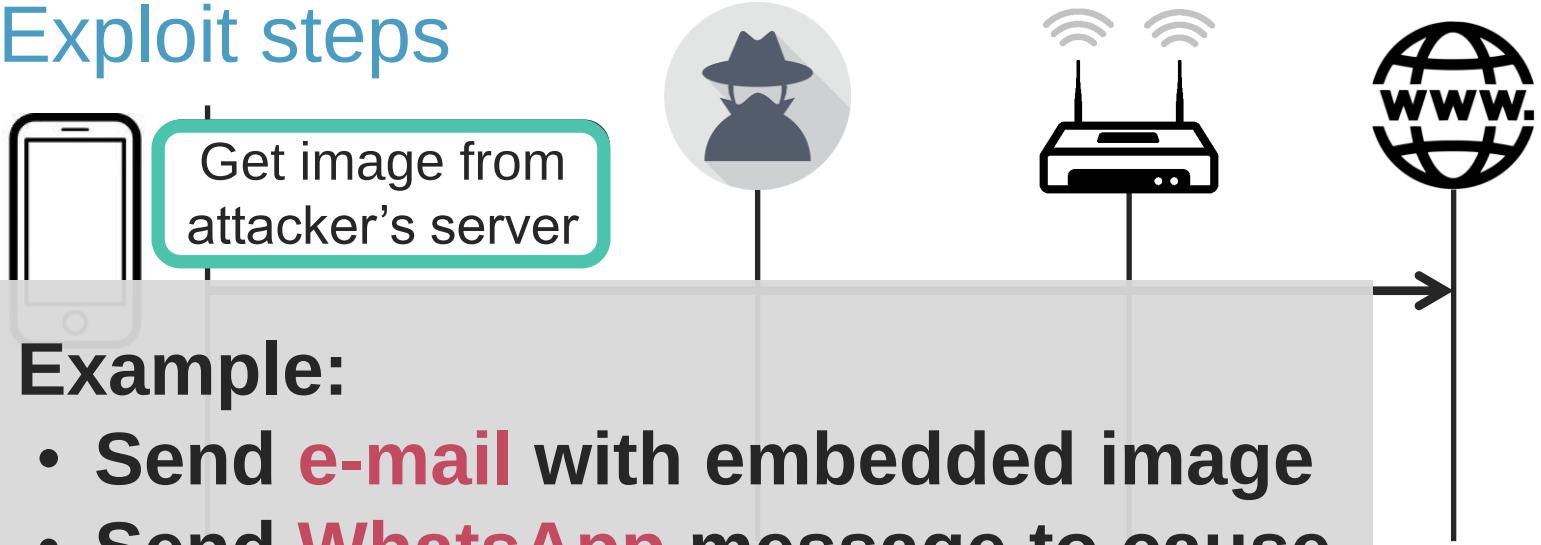
Aggregation design flaw



Aggregation design flaw



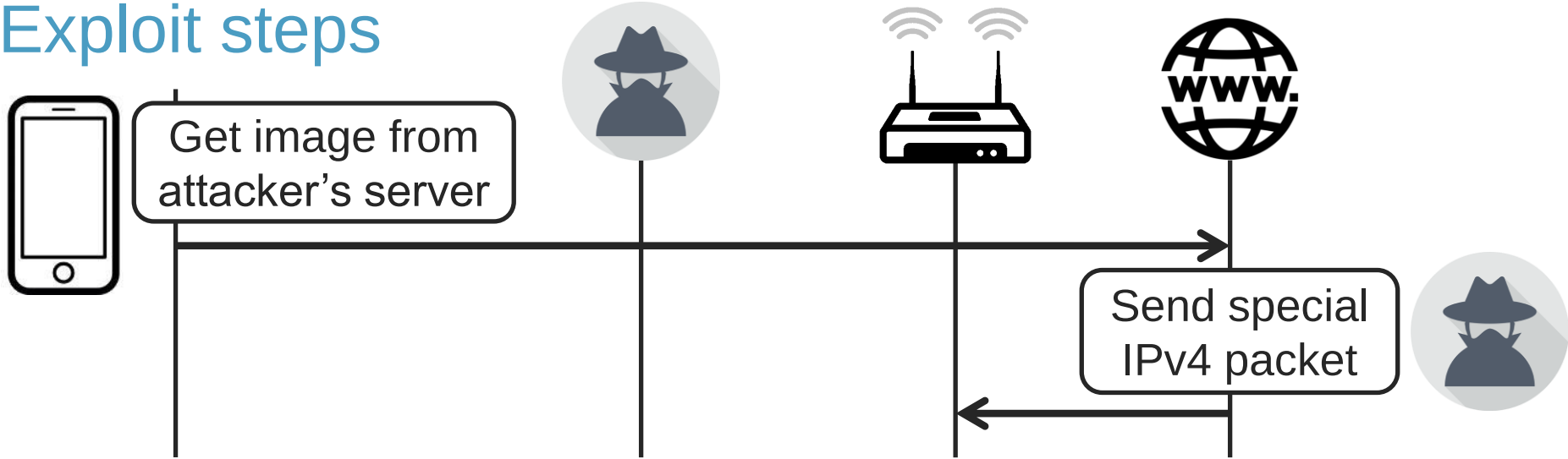
Exploit steps



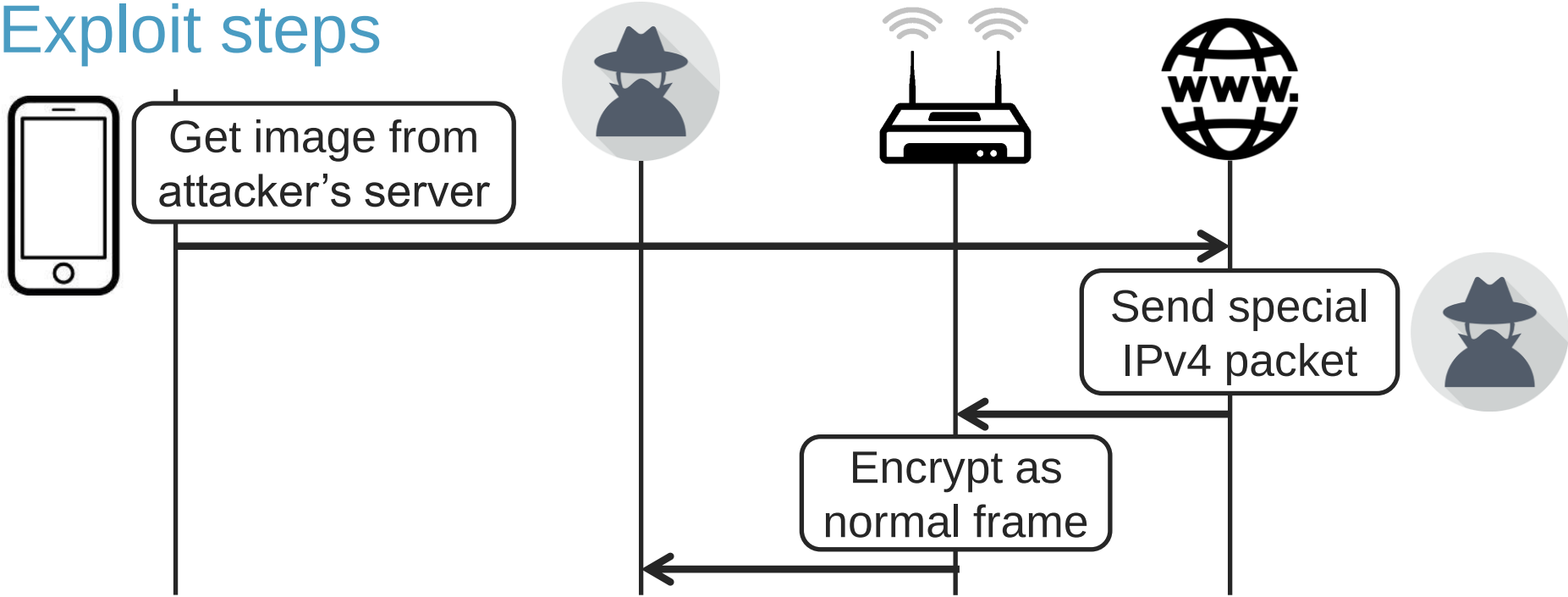
Example:

- **Send e-mail with embedded image**
- **Send WhatsApp message to cause link/image preview**

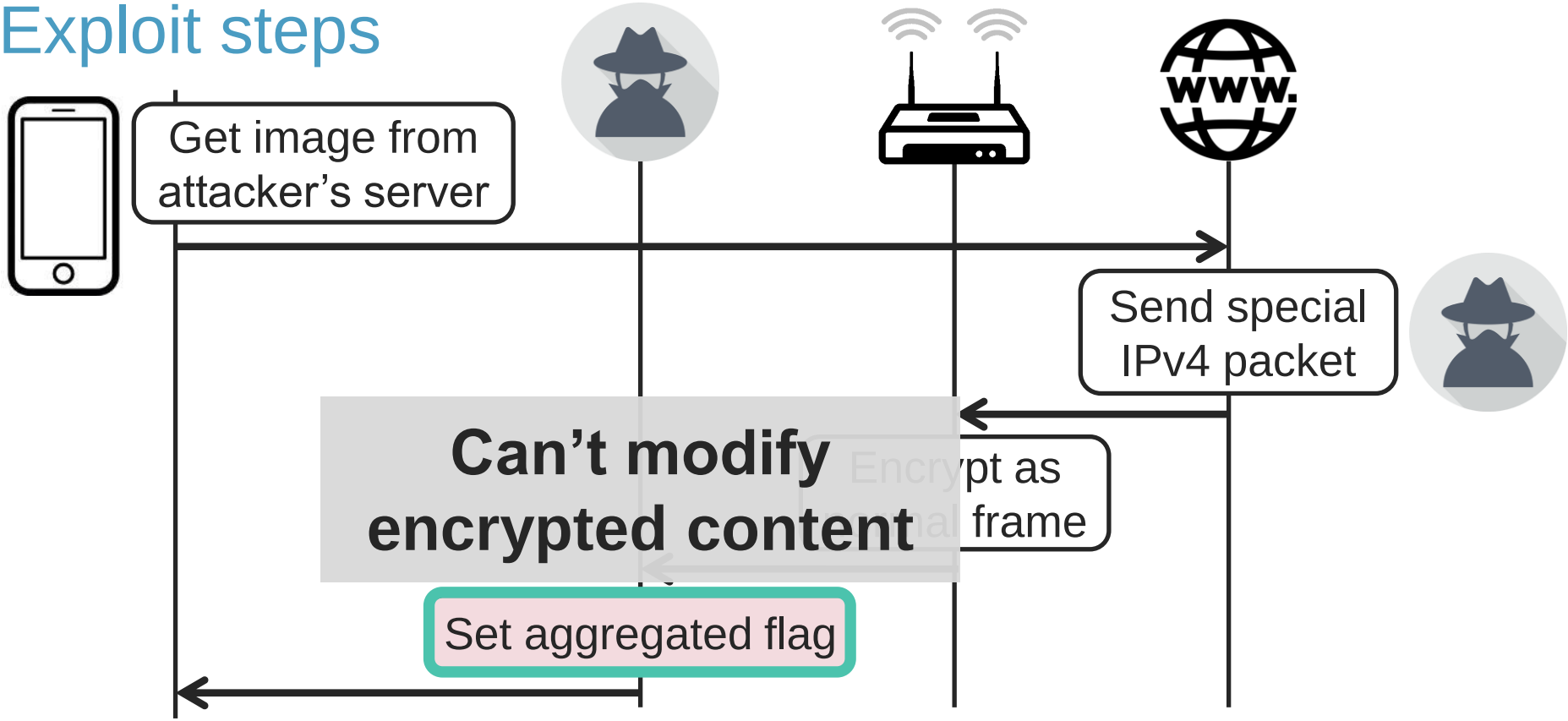
Exploit steps



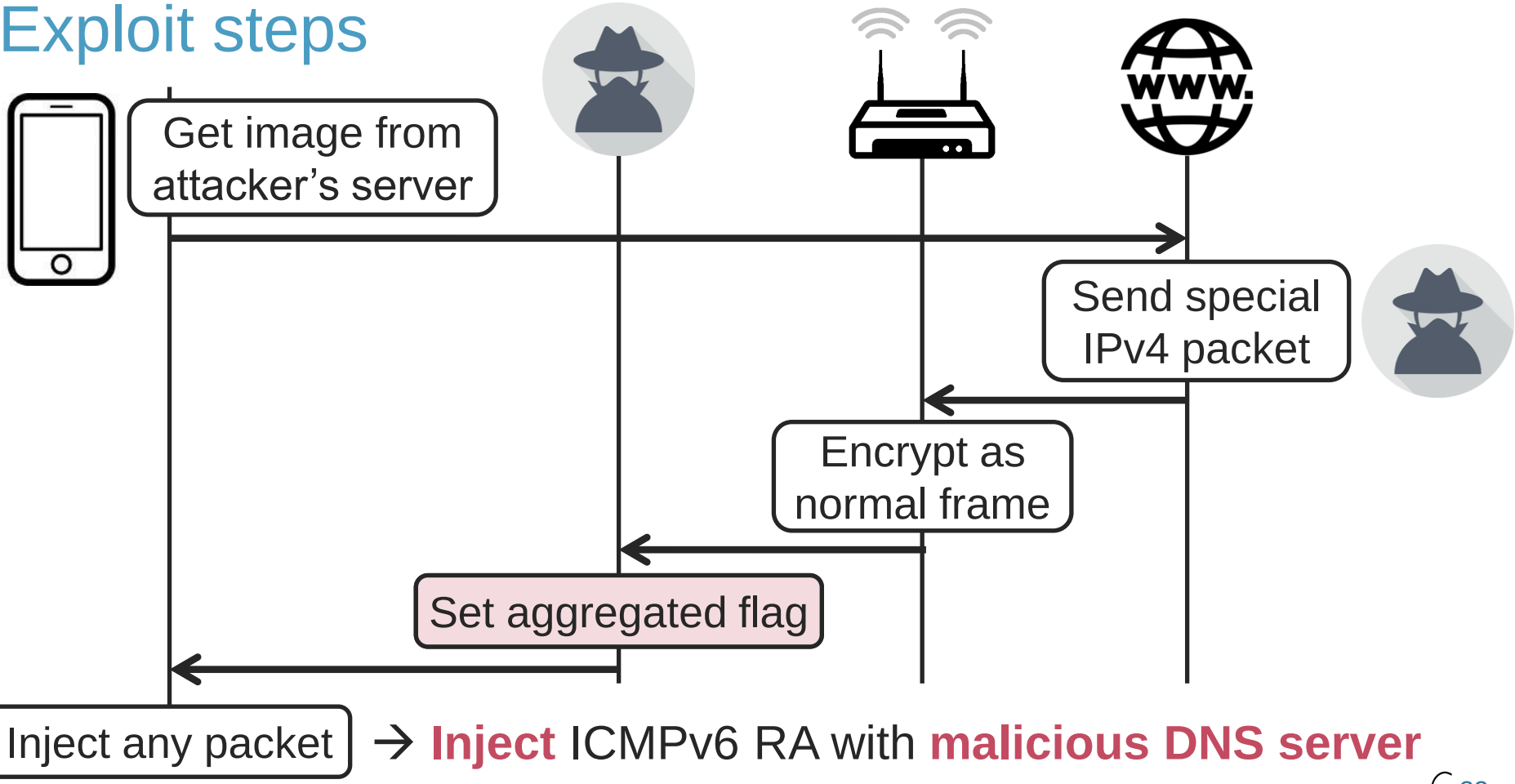
Exploit steps



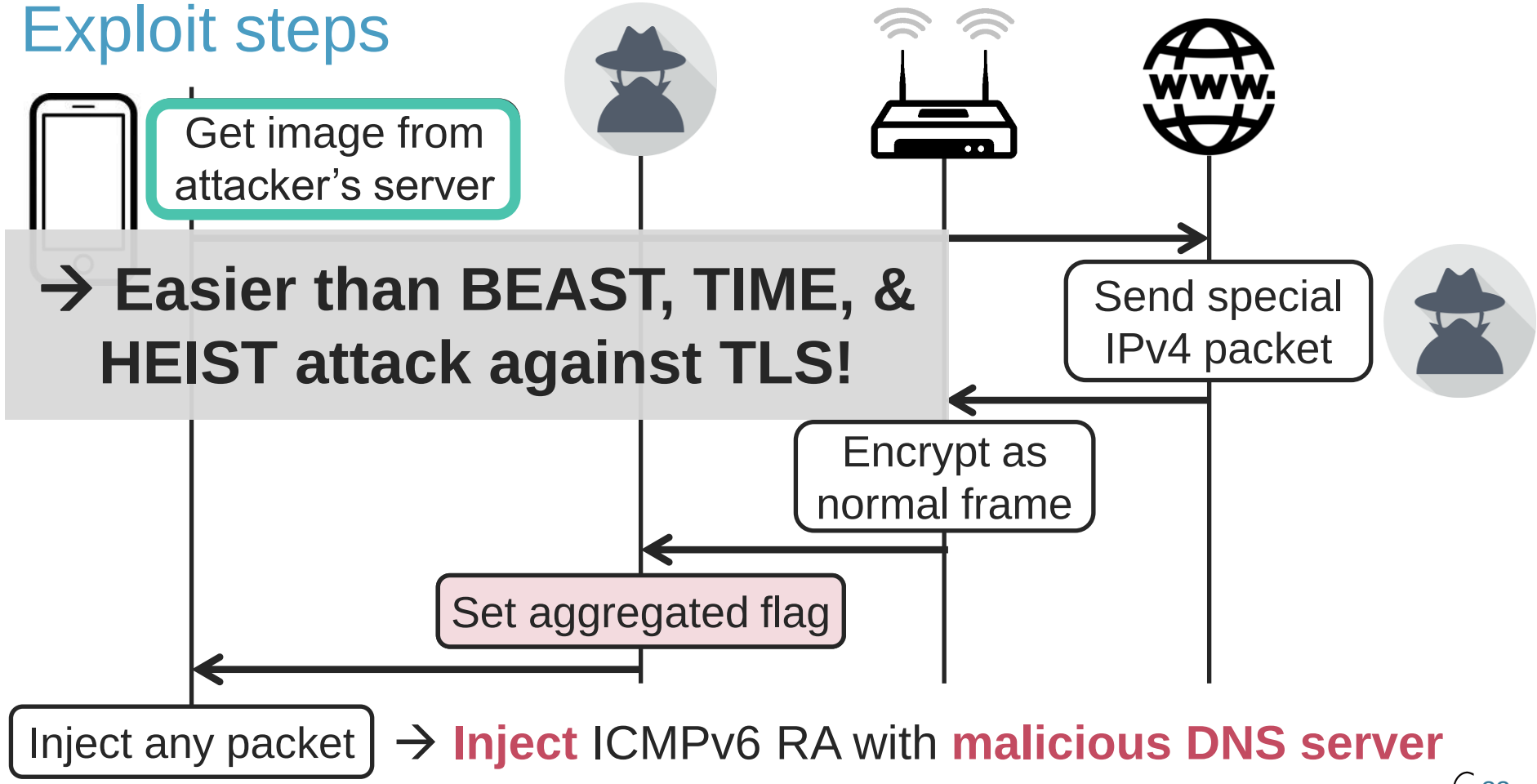
Exploit steps



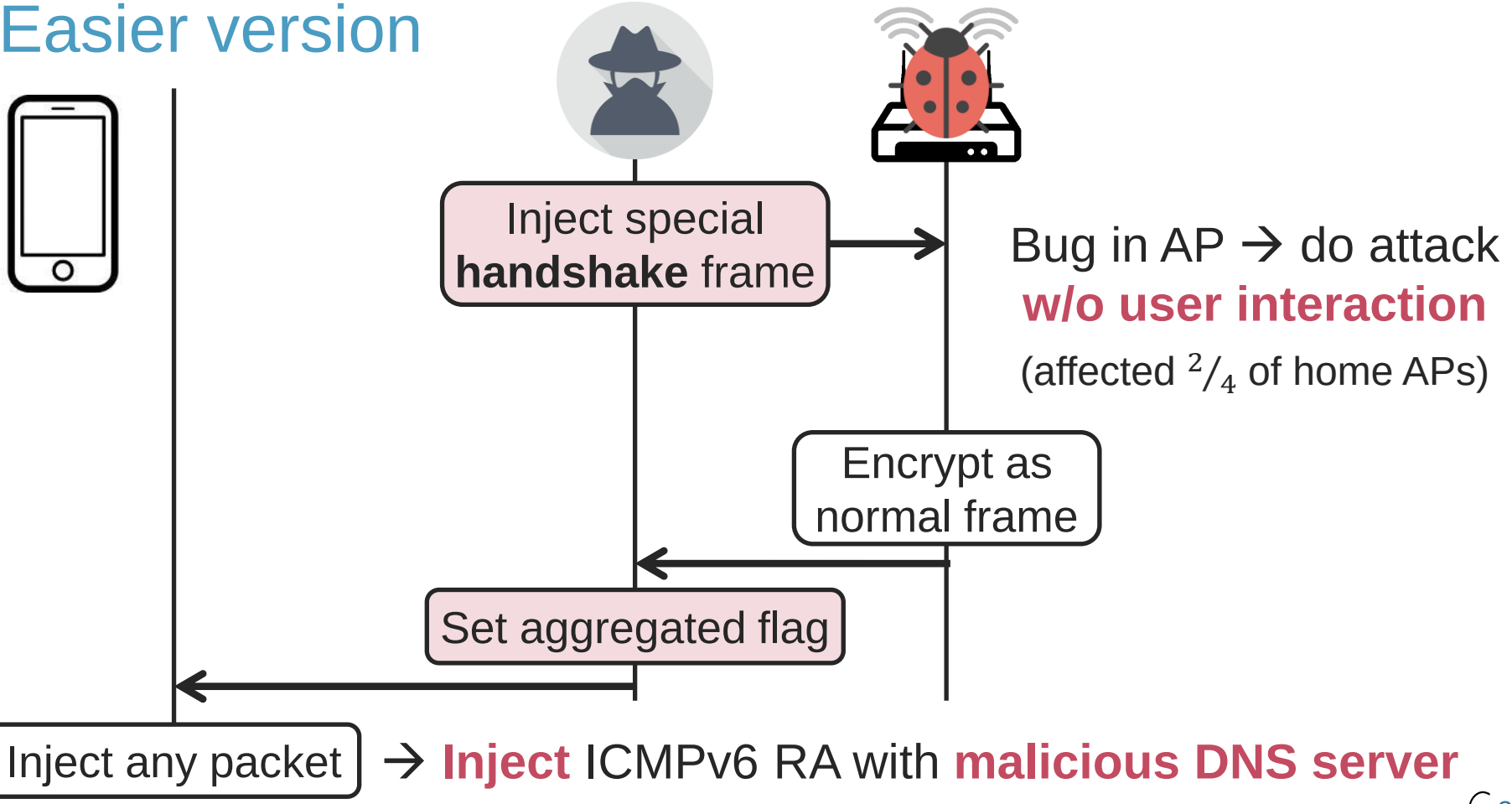
Exploit steps



Exploit steps



Easier version



Design flaws

Implementation Flaws

Design flaws

- Plaintext frames
- Broadcast fragments
- Cloacked A-MSDUs
- Mixed fragments
- EAPOL forwarding
- Out of order fragments
- Out of order frag

Trivial frame injection

Plaintext frames wrongly accepted:

- Depending if **fragmented**, **broadcasted,** or while **connecting**

Trivial frame injection

Plaintext frames wrongly accepted:

- Depending if **fragmented**, **broadcasted,** or while **connecting**
- Examples: Apple and some Android devices, some Windows dongles, home and professional APs, and many others!

→ Can trivially **inject frames**

Trivial frame injection

DEMO!

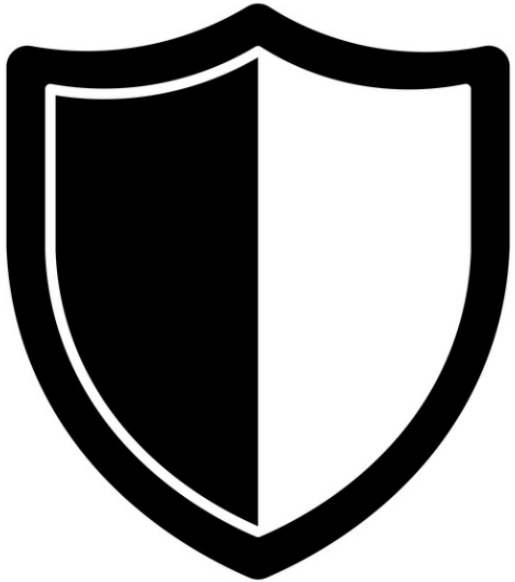
New OnePlus 6 vulnerabilities

Two new implementation vulnerabilities:

- CVE-2021-36197: the OnePlus 6 accepts any **plaintext frames during the 4-way handshake**
- CVE-2021-36196: the OnePlus 6 accept **plaintext broadcast frames while connected** to the network

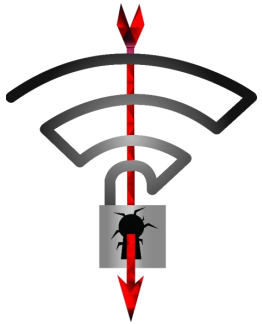
Various attacks possible:

- Example: make the OnePlus using a malicious DNS server
- Reported to OnePlus but **not acknowledged**



Channel Validation

Background: recent attacks require MitM



Most Key Reinstallation Attacks (KRACKs)

- **Block & delay** handshake frames
- E.g. 4-way & group handshake

Fragmentation & Aggregation Attacks

- Aggregation attack: **modify** header
- Fragmentation attacks: **block** frames

Background: recent attacks require MitM

Traffic Analysis

- **Capture all** encrypted frames
- **Block** certain encrypted frames

Attacking broadcast TKIP

- **Block** MIC failures
- **Modify** encrypted frames

ChopChop

Attacks used special multi-channel MitM

AP is cloned on **different channel**

Handshake succeeds &
can reliably manipulate frames!

Preventing multi-channel MitM

Verify operating channel when connecting to a network

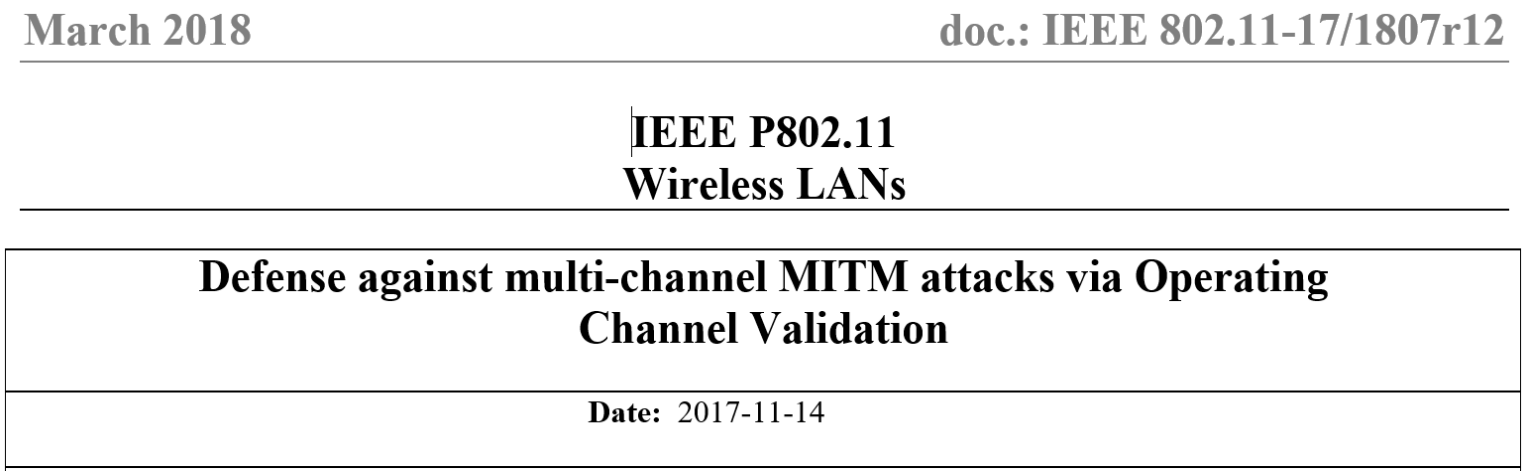
Also need to handle some edge cases

- After the clients wakes up from sleep mode
- When the network switches channel due to radar detection

→ Implemented on Linux in wpa_supplicant and hostapd

Specification

- Collaborated with industry to standardize defense (with Broadcom and Intel)
- **Now part of the 2020 update to the IEEE 802.11 standard**



March 2018 doc.: IEEE 802.11-17/1807r12

IEEE P802.11
Wireless LANs

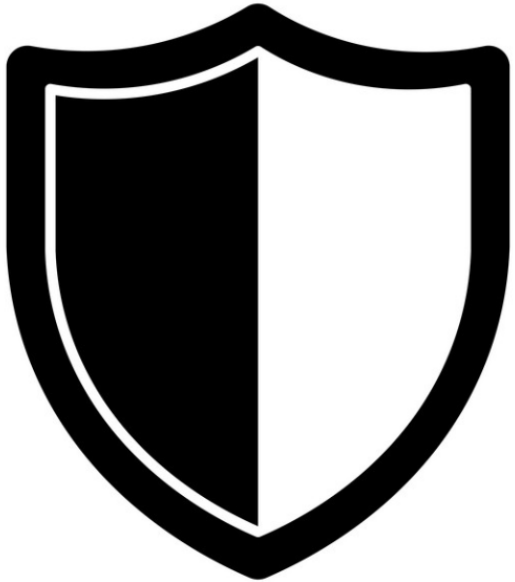
Defense against multi-channel MITM attacks via Operating Channel Validation

Date: 2017-11-14

Specification

- Collaborated with industry to standardize defense (with Broadcom and Intel)
- **Now part of the 2020 update to the IEEE 802.11 standard**

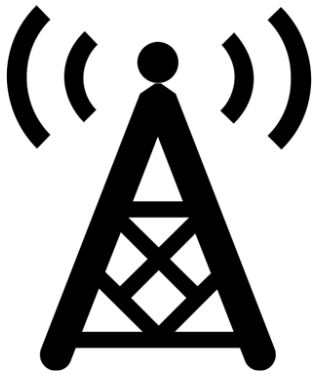
- Recognized as an **optional feature of WPA3**
- Good initial step, hopefully becomes mandatory in future



Beacon Protection

Background: beacons

- Wi-Fi networks use beacons to announce their presence
- They are sent every ~100 ms by an Access Point



Contains properties of the network:

- Name of the network
- Supported bitrates (e.g. 11n or 11ac)
- Regulatory constraints (e.g. transmission power)
- ...

Beacons are not protected

```
› Tag: SSID parameter set: cisco
› Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
› Tag: DS Parameter set: Current Channel: 1
› Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
› Tag: Country Information: Country Code GB, Environment Unknown (0x04)
› Tag: Power Constraint: 3
› Tag: ERP Information
› Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
› Tag: QBSS Load Element 802.11e CCA Version
› Tag: RM Enabled Capabilities (5 octets)
› Tag: HT Capabilities (802.11n D1.10)
› Tag: RSN Information
› Tag: Mobility Domain
› Tag: HT Information (802.11n D1.10)
› Tag: Extended Capabilities (10 octets)
› Ext Tag: HE Capabilities (IEEE Std 802.11ax/D3.0)
› Ext Tag: HE Operation (IEEE Std 802.11ax/D3.0)
› Ext Tag: Spatial Reuse Parameter Set
```

- WPA version & channel: verified when connecting
- All other fields can be spoofed by an adversary

Other attacks & findings

Spoof medium access parameters

› **Reduces the bandwidth** of clients

Battery depletion attacks

› Spoof beacons to **make clients stay awake**

Partial ma**chine-in-the-middle** attack

› Might bypasses channel operating validation

Designing a defense

Conclusion: we need to authenticate beacons!

Design goals:

- Focus on **efficiency & simplicity** to encourage adoption
- Must be straightforward to implement

Design approach

Rely on **symmetric encryption**

› Reuse existing crypto primitives of Wi-Fi

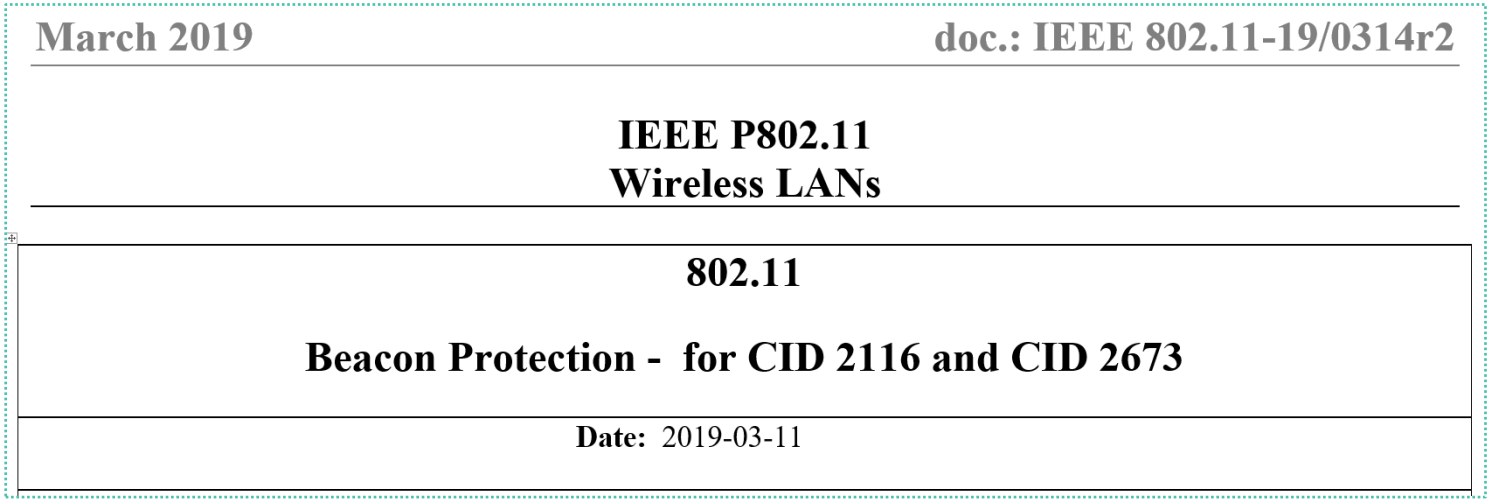
We **defend against outsider attacks**

› Adversary doesn’t possess network credentials

› Similar to protection of broadcast Wi-Fi traffic

Specification

- Collaborated with industry to standardize our defense (Intel, Broadcom, Qualcomm and Huawei)
- **Now part of the 2020 update to the IEEE 802.11 standard**



March 2019 doc.: IEEE 802.11-19/0314r2

IEEE P802.11
Wireless LANs

802.11

Beacon Protection - for CID 2116 and CID 2673

Date: 2019-03-11

Specification

- Collaborated with industry to standardize our defense (Intel, Broadcom, Qualcomm and Huawei)
- **Now part of the 2020 update to the IEEE 802.11 standard**

- Recognized as an **optional feature of WPA3**
- Good initial step, hopefully becomes mandatory in future

Practical guidelines

Show how to configure modern Wi-Fi networks?

- Enable **WPA3** in mixed mode
- Assure devices are **updated**. Pressure vendors if no updates are being provided.
- Enable **beacon protection & channel validation** once is becomes available. Ask vendors to implement it!

Enabling channel validation

Enable OCV in the .config when building hostapd & wpa_sup:

```
CONFIG_OCV=y
```

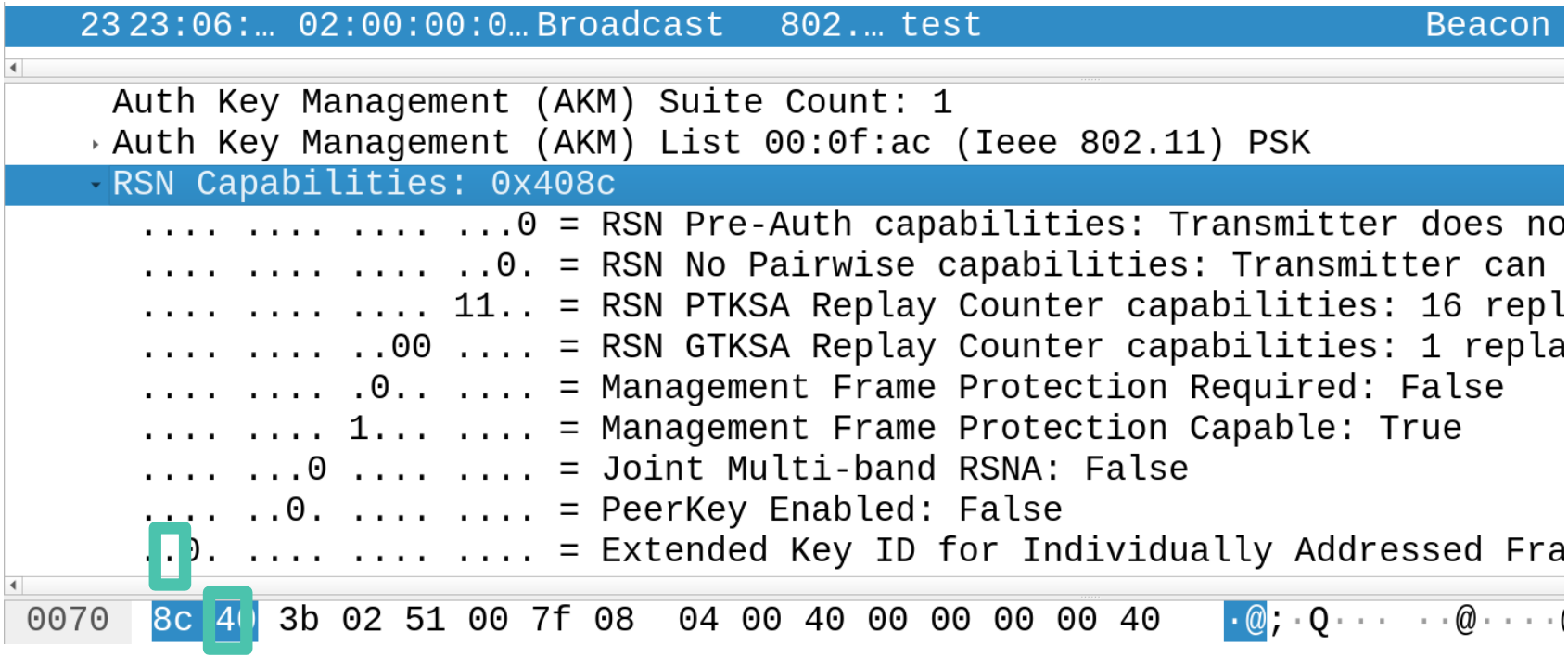
Enable OCV and 802.11w in the hostapd.conf file:

```
ieee80211w=1
ocv=1
```

Finally, enable OCV in the wpa_supplicant.conf file:

```
network={    ...
             ieee80211w=1
             ocv=1          }
```

Verifying OCV network support



Above **bit is set** when OCV is enabled (not yet recognized by wireshark) → the 6th bit of the 2nd byte **in the RSNE**

Verifying OCV usage

```
$ ./wpa_supplicant -D nl80211 -i wlan2 -c client.conf –d
...
wlan2: State: 4WAY_HANDSHAKE -> 4WAY_HANDSHAKE
wlan2: WPA: RX message 3 of 4-Way Handshake from
02:00:00:00:00:00 (ver=2)
...
WPA: OCI KDE in EAPOL-Key - hexdump(len=9): dd 07 00 0f
ac 0d 51 01 00
wlan2: WPA: Sending EAPOL-Key 4/4
```

› AP indeed sends the **Operating Channel Information (OCI)**

Enabling beacon protection

Enable beacon protection and 802.11w in hostapd.conf:

```
ieee80211w=1
beacon_prot=1
```

And do the same in the wpa_supplicant.conf file:

```
network={    ...
             ieee80211w=1
             beacon_prot=1              }
```

Only used **if the hardware supports** beacon protection…

Verifying beacon protection network support

```
15… 22:23:… 02:00:00:0… Broadcast   802.… test                    Beacon frame.
  ▸ Extended Capabilities: 0x04 (octet 1)
  ▸ Extended Capabilities: 0x00 (octet 2)
  ▸ Extended Capabilities: 0x40 (octet 3)
  ▸ Extended Capabilities: 0x00 (octet 4)
  ▸ Extended Capabilit
  ▸ Extended Capabilit
  ▸ Extended Capabilit
  ▸ Extended Capabilit
  ▸ Extended Capabilit
▸ Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
▾ Tag: Management MIC
    Tag Number: Management MIC (76)
    Tag length: 16
    KeyID: 6
    IPN: ff0100000000
    MIC: 0fd40906ea57facf
```

Every beacon includes a Management MIC!

Verifying beacon protection usage

```
$ ./wpa_supplicant -D nl80211 -i wlan2 -c client.conf –d
...
WPA: BIGTK in EAPOL-Key - hexdump(len=30): [REMOVED]
wlan2: WPA: Sending EAPOL-Key 4/4
...
wlan2: WPA: BIGTK keyid 6 pn 000000000000
WPA: BIGTK - hexdump(len=16): [REMOVED]
wpa_driver_nl80211_set_key: ifindex=5 (wlan2) alg=4 ...
...
```

› Client recieved beacon protection key (**BIGTK**) and installed it

Thank you!

Questions?