

# Advanced Wi-Fi Attacks Using Commodity Hardware

Mathy Vanhoef — @vanhoefm

BruCON, Belgium, 3 October 2018

# Background

- › Wi-Fi assumes each stations behaves fairly



- › With special hardware we don't have to 😊
  - ›› Continuous jamming: channel unusable
  - ›› Selective jamming: block specific packets

# Background

- › Wi-Fi assumes each stations behaves fairly



- › With special hardware we don't have to 😊
  - ›› Continuous jamming: channel unusable
  - ›› **Selective jamming**: block specific packets

# Research: use cheap hardware?



Small 15\$ USB sufficient to:

- › Testing selfish behavior in practice
- › Continuous & selective jamming
- › Enables reliable manipulation of encrypted traffic

# Research: use cheap hardware?



Attacks are cheaper than expected!

- › We should be able to **detect** them.

# Selfish Behavior

Impact of selfish behavior?

**Implement & Test!**

# Selfish Behavior

Steps taken to transmit a frame:

**In use**

# Selfish Behavior

Steps taken to transmit a frame:



1. SIFS: let hardware process the frame



# Selfish Behavior

Steps taken to transmit a frame:



1. SIFS: let hardware process the frame
2. AIFSN: depends on priority of the frame

# Selfish Behavior

Steps taken to transmit a frame:



1. SIFS: let hardware process the frame
2. AIFSN: depends on priority of the frame
3. Random backoff: avoid collisions

# Selfish Behavior

Steps taken to transmit a frame:



1. SIFS: let hardware process the frame
2. AIFSN: depends on priority of the frame
3. Random backoff: avoid collisions
4. Send the packet

# Selfish Behavior

Steps taken to transmit a frame:

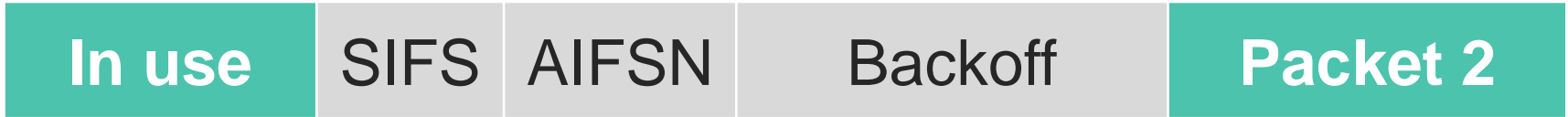


Manipulate by modifying Atheros firmware:

- › Disable backoff
- › Reducing AIFSN
- › Reducing SIFS

# Selfish Behavior

Steps taken to transmit a frame:



Manipulate by modifying Atheros firmware:

- › **Disable backoff**
  - › **Reducing AIFSN**
  - › Reducing SIFS → Reduces throughput
- Optimal strategy**  
From 14 to 37 Mbps

# How to control radio chip?

## Using memory mapped registers

- › Disable backoff:

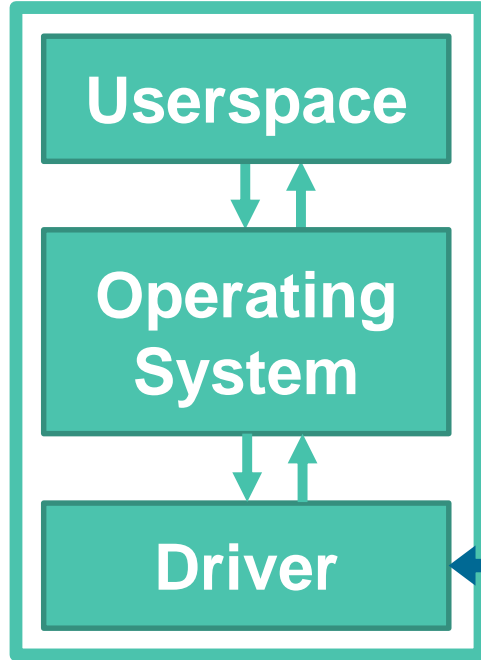
```
int *GBL_IFS_MISC = (int*)0x10F0;  
*GBL_IFS_MISC |= IGNORE_BACKOFF;
```

- › Reset AIFSN and SIFS:

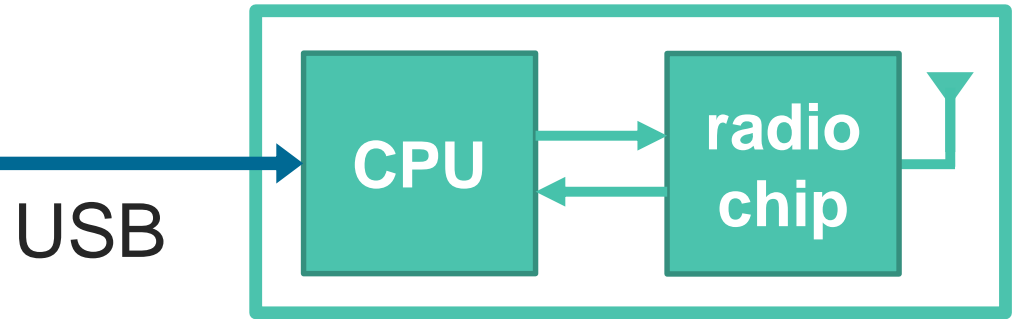
```
int *AR_DLCL_IFS = (int*)0x1040;  
*AR_DLCL_IFS = 0;
```

# We can't we just modify the driver?

## Main machine



## WiFi Dongle



Code runs on CPU of dongle  
→ **Firmware control needed**

# Countermeasures



**DOMINO defense  
system reliably detects  
this selfish behavior [1].**



# Selfish Behavior

What if there are multiple selfish stations?

- › In a collision, both frames are lost

# Selfish Behavior

What if there are multiple selfish stations?

- ~~In a collision, both frames are lost~~
- **Capture effect**: in a collision, frame with the best signal and lowest bitrate is decoded

## Similar to FM radio

Demo: The Queen station generally “wins” the collision with other stations.

# FM Radio Demo



# Selfish Behavior

Attack can abuse capture effect

- › Selfish clients will **lower** their bitrate to beat other selfish stations!
- › Until this gives no more advantage

To **increase** throughput, bitrate is **lowered**!

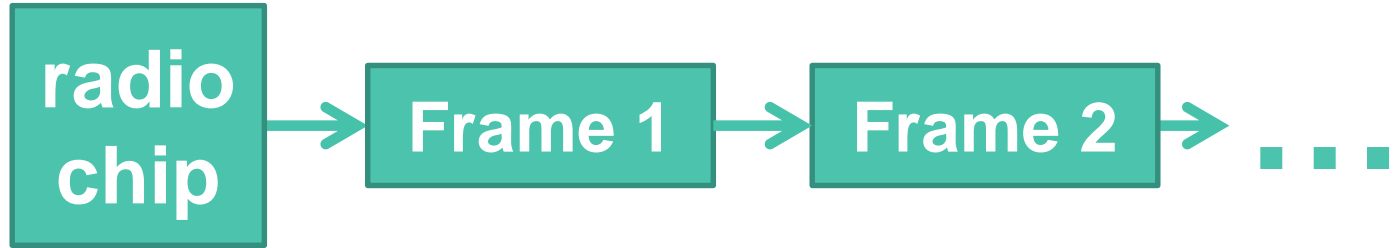
→ Other station = background noise

# Continuous jammer

Want to build a continuous jammer

- › Instant transmit: disable carrier sense
- › No interruptions: queue infinite #packets

Frames to be transmitted are in a linked list:



# Continuous jammer

Want to build a continuous jammer

- › Instant transmit: disable carrier sense
- › No interruptions: queue infinite #packets

Frames to be transmitted are in a linked list:



# Continuous Jammer

## Experiments

- › Only first packet visible in monitor mode!
- › Other devices are **silenced**.



Default antenna gives range of ~80 meters



Amplifier gives range of ~120 meters

Demo: continuous jammer

**Demo: continuous jammer**



# Raspberry Pi Supported!



# Practical Implications

Devices in 2.4 and 5 GHz band?



- › Home automation
- › Industrial control
- › Internet of Things
- › ...



Can all easily be jammed!

# Practical Implications

Devices in 2.4 and 5 GHz band?



# Practical Implications

Devices in 2.4 and 5 GHz band?



# Not just wild speculation ...



\$45 Chinese jammer to prevent cars from being locked [4]

GPS jammer to disable anti-theft tracking devices in stolen cars [5]



Disable mobile phone service after cutting phone and alarm cables [6]

# Selective Jammer

Decides, based on the header,  
whether to jam the frame

# How does it work?

1. Detect and decode header



# How does it work?

1. Detect and decode header
2. Abort receiving current frame





# How does it work?

1. Detect and decode header
2. Abort receiving current frame
3. Inject dummy packet



# How does it work?

1. Detect and decode header
  2. Abort receiving current frame
  3. Inject dummy packet
- } Easy



‣ Frame check sequence: 0x664e01f2 [incorrect,  
‣ [Malformed Packet: IEEE 802.11]

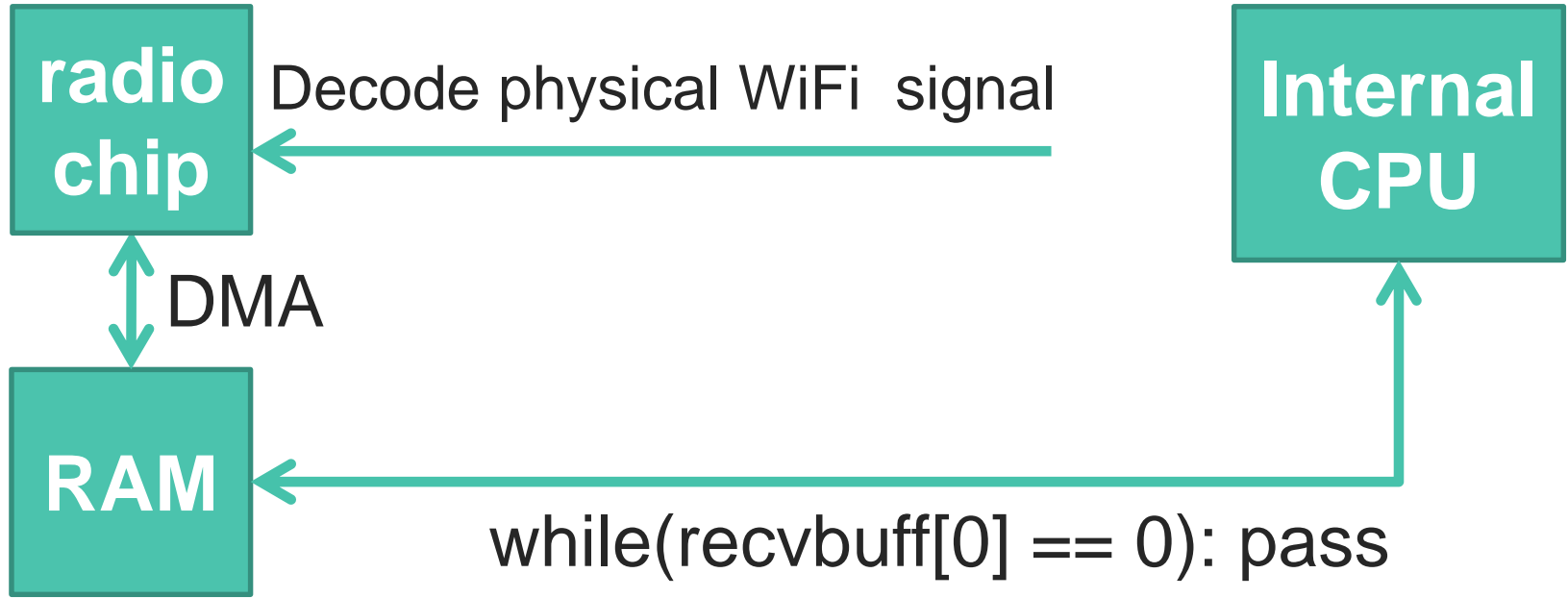
# How does it work?

1. **Detect and decode header** } **Hard**
2. Abort receiving current frame } **Easy**
3. Inject dummy packet



‣ Frame check sequence: 0x664e01f2 [incorrect,  
‣ **[Malformed Packet: IEEE 802.11]**

# Detecting frame headers?



→ Can read header of frames still in the air!

# In practice

1. **Detect and decode header**
2. Abort receiving current frame
3. Inject dummy packet

Poll memory until data is being written:

```
while (elapsed < msec && buff[15] == 0xF1) {  
    prev = update_elapsed(prev, freq, &elapsed);  
}
```

Timeout

Detect incoming packet

# In practice

1. **Detect and decode header**
2. Abort receiving current frame
3. Inject dummy packet

Probe request or beacon?

```
if ( (buff[0] == 0x80 || buff[0] == 0x50)
    && ((source[0] & 1) || A_MEMCMP(source, buff + 10, 6) == 0) )
{
```

buff + 10: sender of packet  
source : target MAC address

# In practice

1. Detect and decode header
2. **Abort receiving current frame**
3. Inject dummy packet

```
// Abort Rx
```

```
*((a_uint32_t *) (WLAN_BASE_ADDRESS + AR_DIAG_SW)) |= AR_DIAG_RX_ABORT;
```



Set specific bit in register

# In practice

1. Detect and decode header
2. Abort receiving current frame
3. **Inject dummy packet**

```
// Jam the packet
```

```
*((a_uint32_t *) (WLAN_BASE_ADDRESS + AR_QTXDP(TXQUEUE))) = (a_uint32_t) txads;
```

```
*((a_uint32_t *) (WLAN_BASE_ADDRESS + AR_Q_TXE)) = 1 << TXQUEUE;
```

Pointer to dummy packet



TXE: Transmit (TX) enable (E)



# Selective Jammer: Reliability

Jammed beacons with many devices/positions

How fast can it react?

- › Position of first mangled byte?
- › 1 Mbps beacon in 2.4 GHz: position 52
- › 6 Mbps beacon in 5 GHz: position 88

Context: MAC header is 34 bytes

# Selective Jammer: Reliability

Jammed beacons with many devices/positions

## Conclusion

- › 100% reliable jammer not possible
- › Medium to large packets can be jammed
- › Surprising this is possible with a limited API!

Demo: selective jammer

**Demo: jammin' beacons**

Code is online (and got updates)

**Virtual Machine:**


[github.com/vanhoefm/modwifi](https://github.com/vanhoefm/modwifi)

# Using your mobile phone

Schulz & co: jamming using mobile phones [9]



Nexus 5

+ nexmon  
= 

[github.com/seemoo-lab/wisec2017\\_nexmon\\_jammer](https://github.com/seemoo-lab/wisec2017_nexmon_jammer)

# Impact on higher-layers



What if we could  
reliably manipulate  
encrypted traffic?

We could attack WPA-TKIP

## Impact on higher-layers

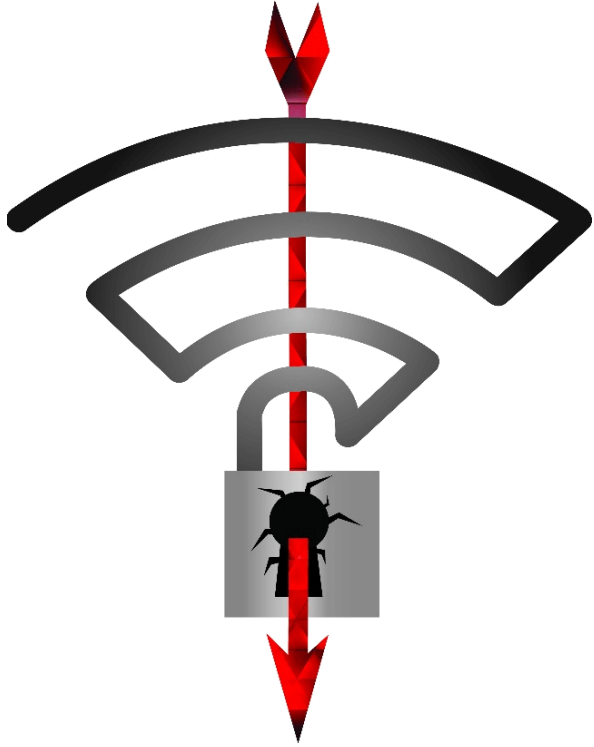


What if we could  
reliably manipulate  
encrypted traffic?

~~We could attack WPA-TKIP~~

**We can break WPA2**

# Breaking WPA2



## Key Reinstallation Attacks (KRACKs)

- › **Block & delay** handshake frames
- › Jammers can block packets!
- › Or help with getting a MitM

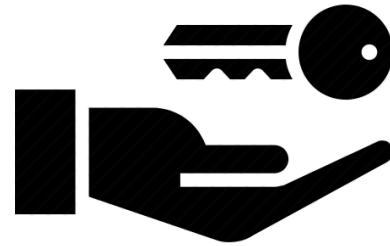


# WPA2 uses a 4-way handshake

Used to connect to any protected Wi-Fi network

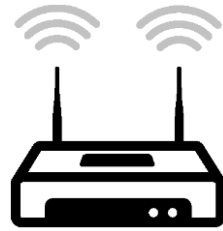
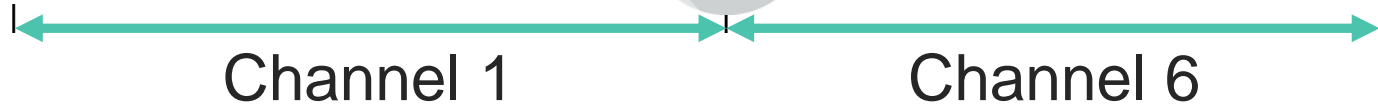
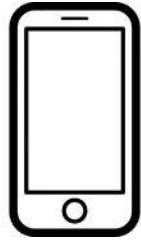


Mutual authentication



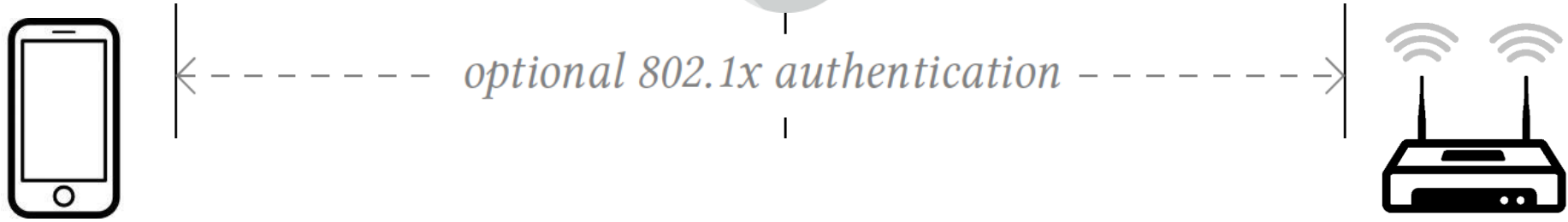
Negotiates fresh PTK:  
pairwise transient key

# KRACK Attack

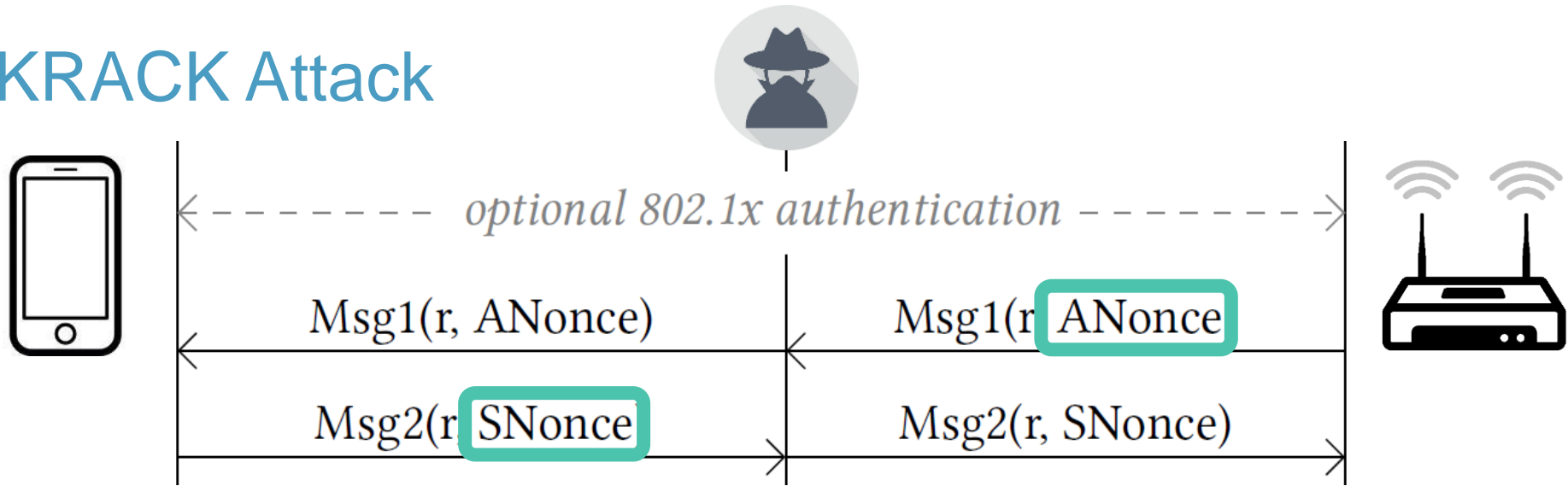


**Jam AP on channel 6  
→ victim will use channel 1**

# KRACK Attack

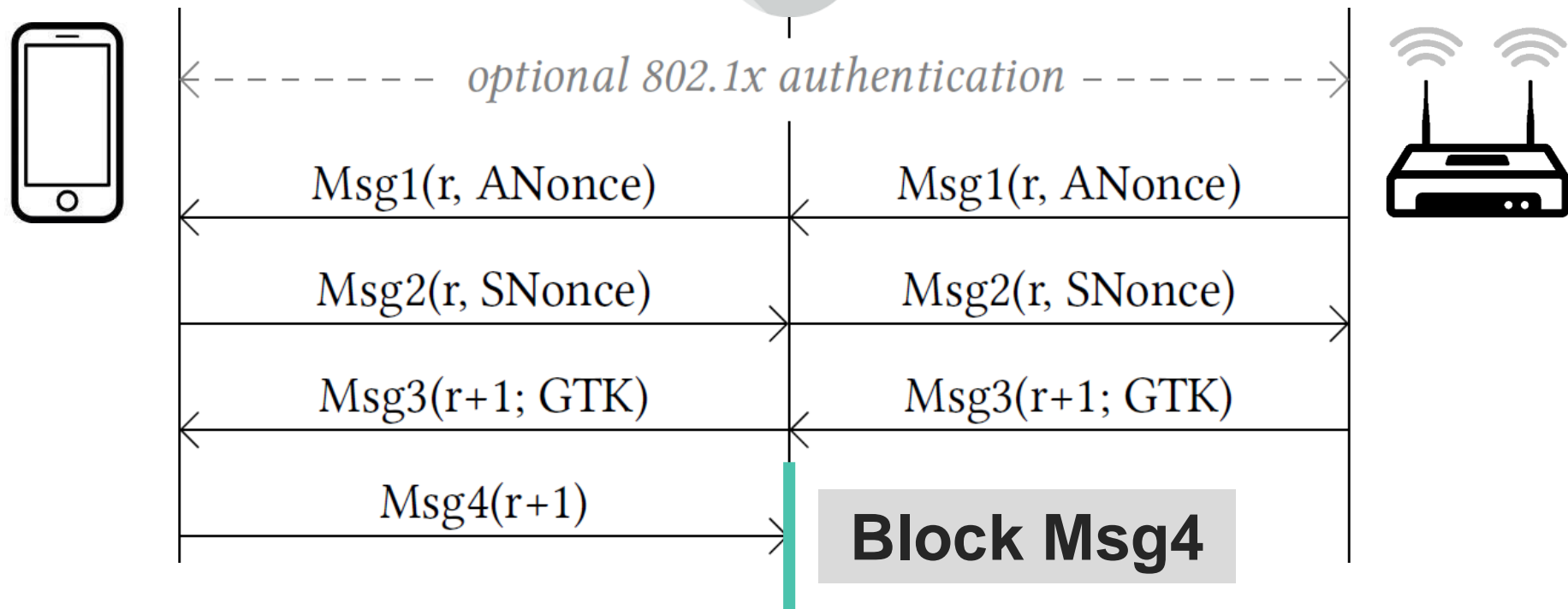


# KRACK Attack

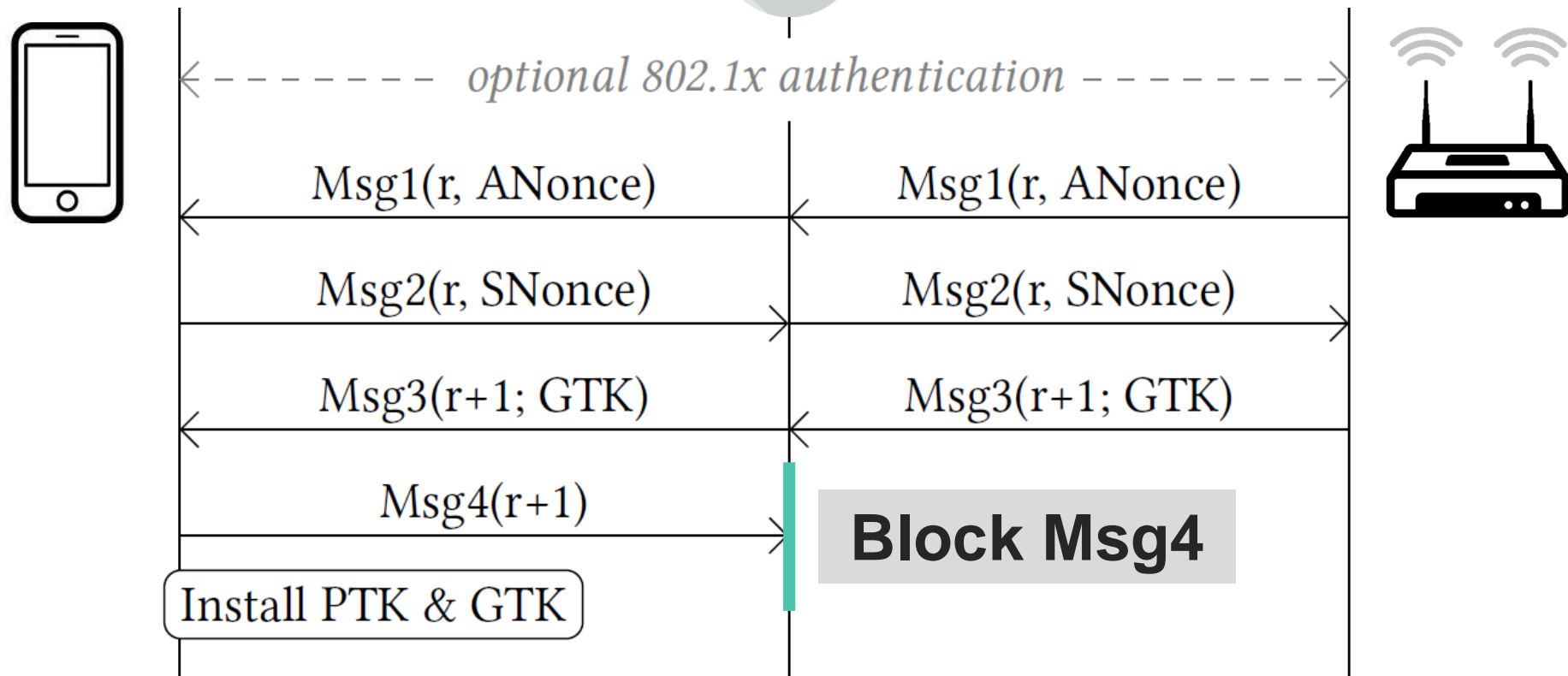


**PTK** = Combine(shared secret,  
ANonce, SNonce)

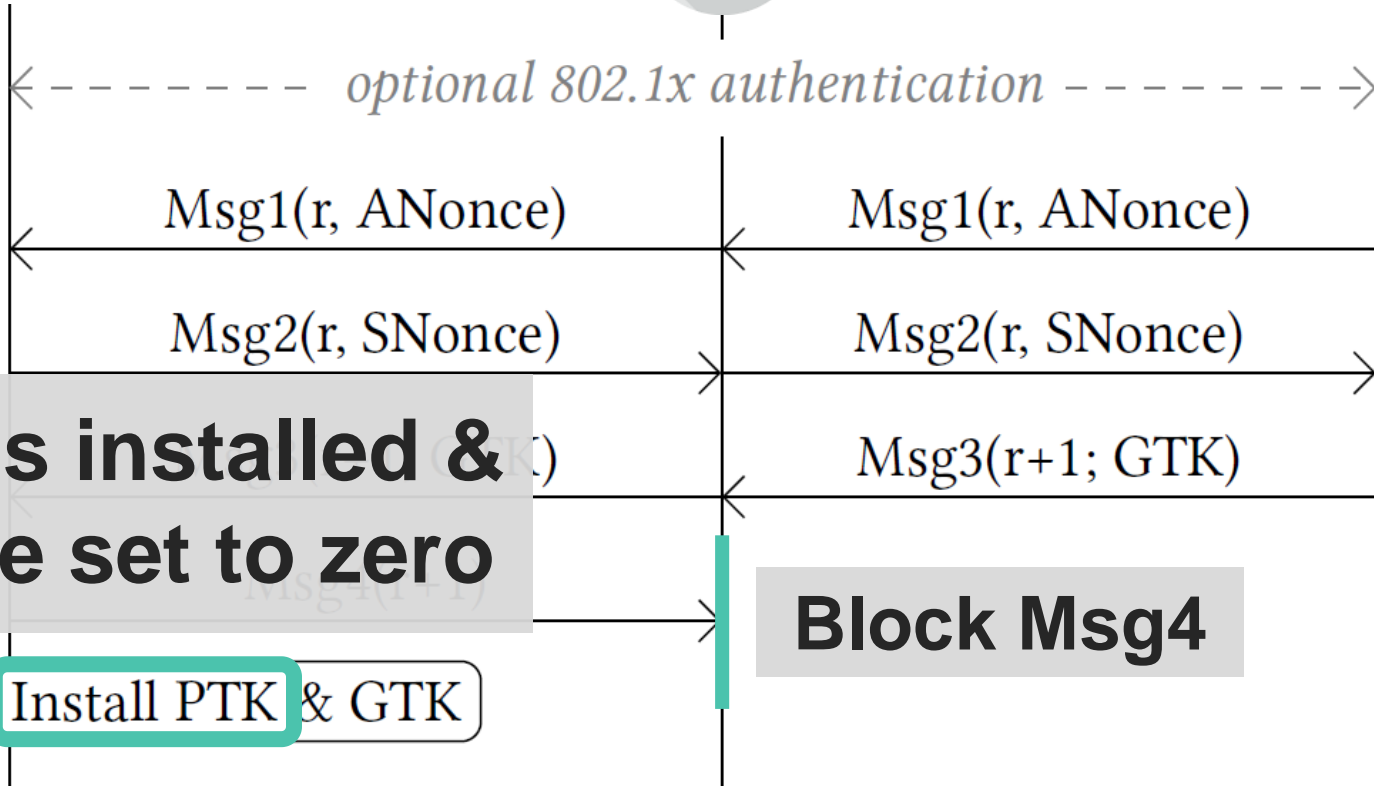
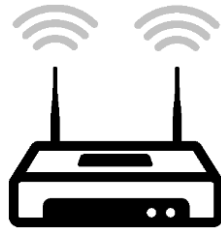
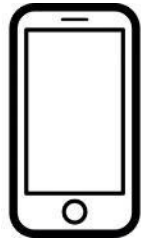
# KRACK Attack



# KRACK Attack



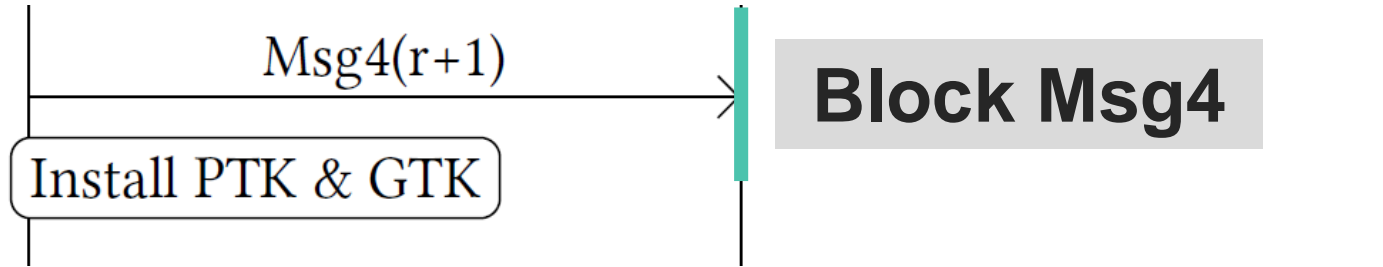
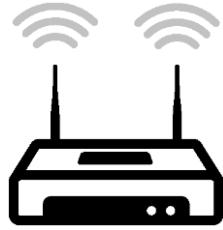
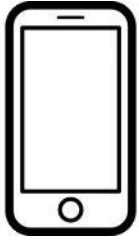
# KRACK Attack



Install PTK & GTK

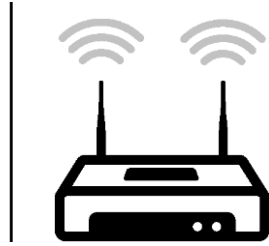
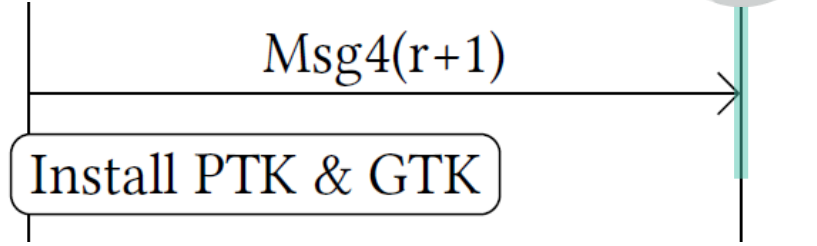
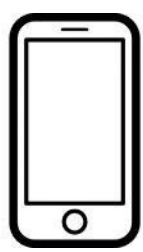
Block Msg4

# KRACK Attack

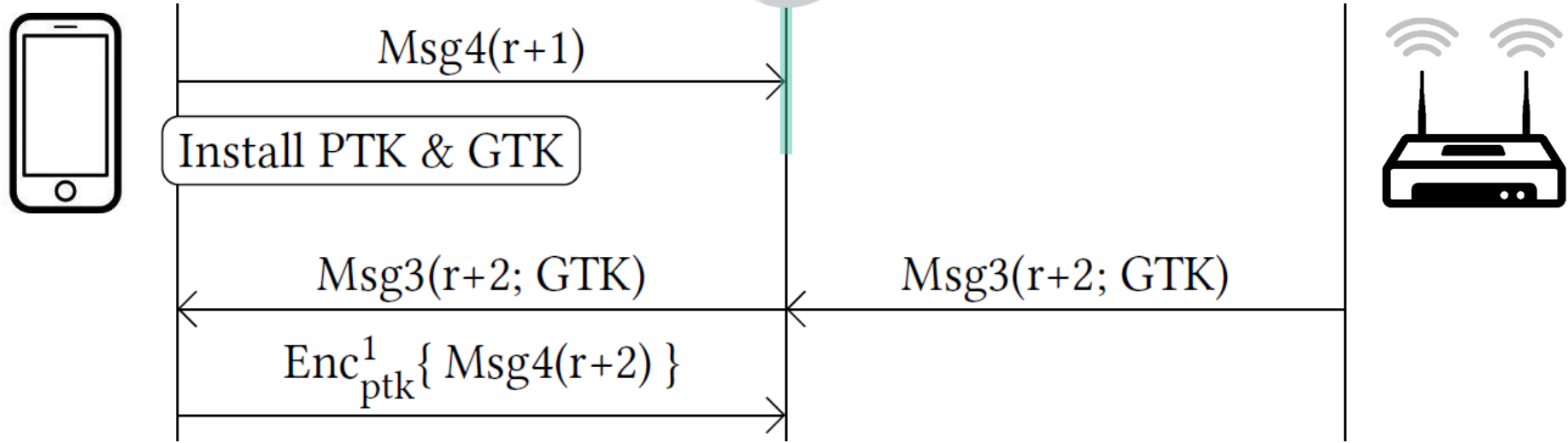




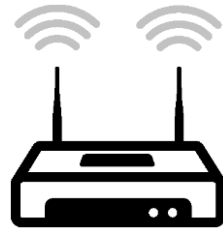
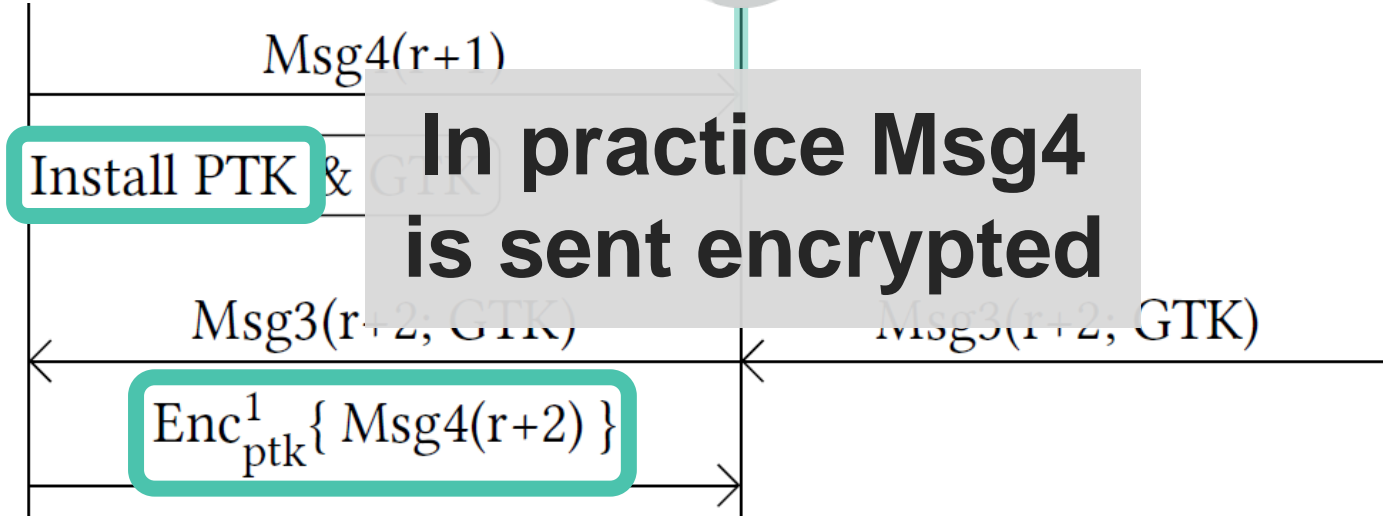
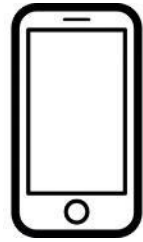
# KRACK Attack



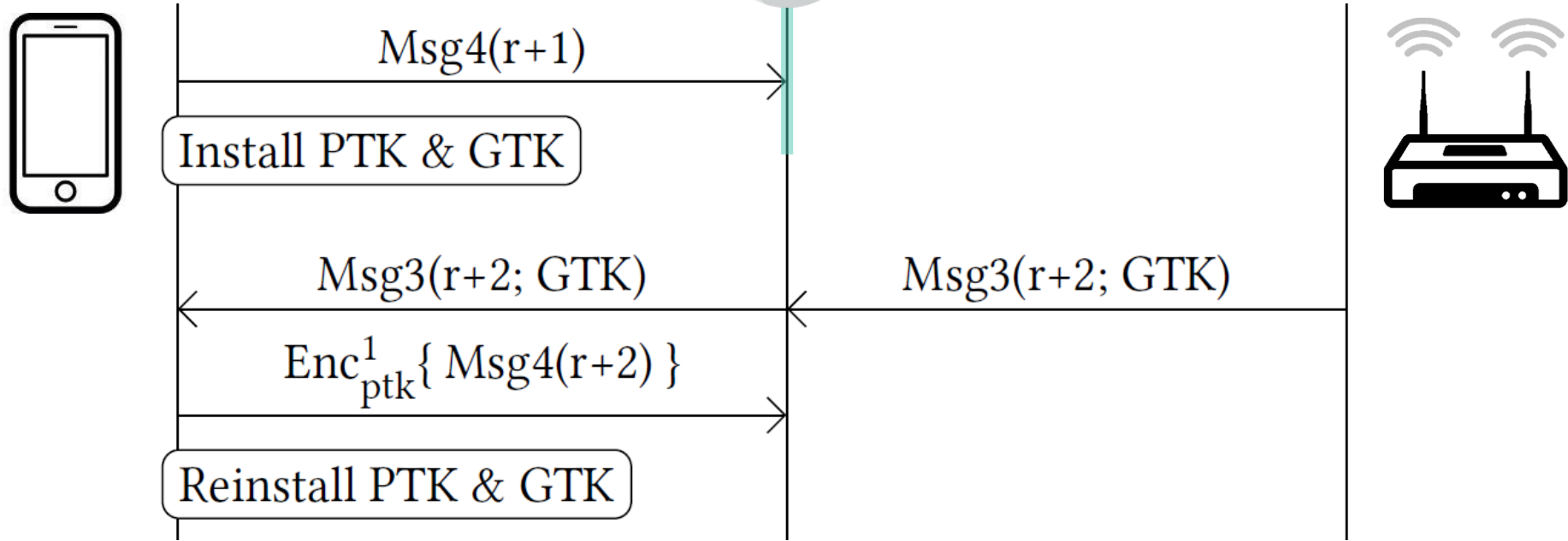
# KRACK Attack



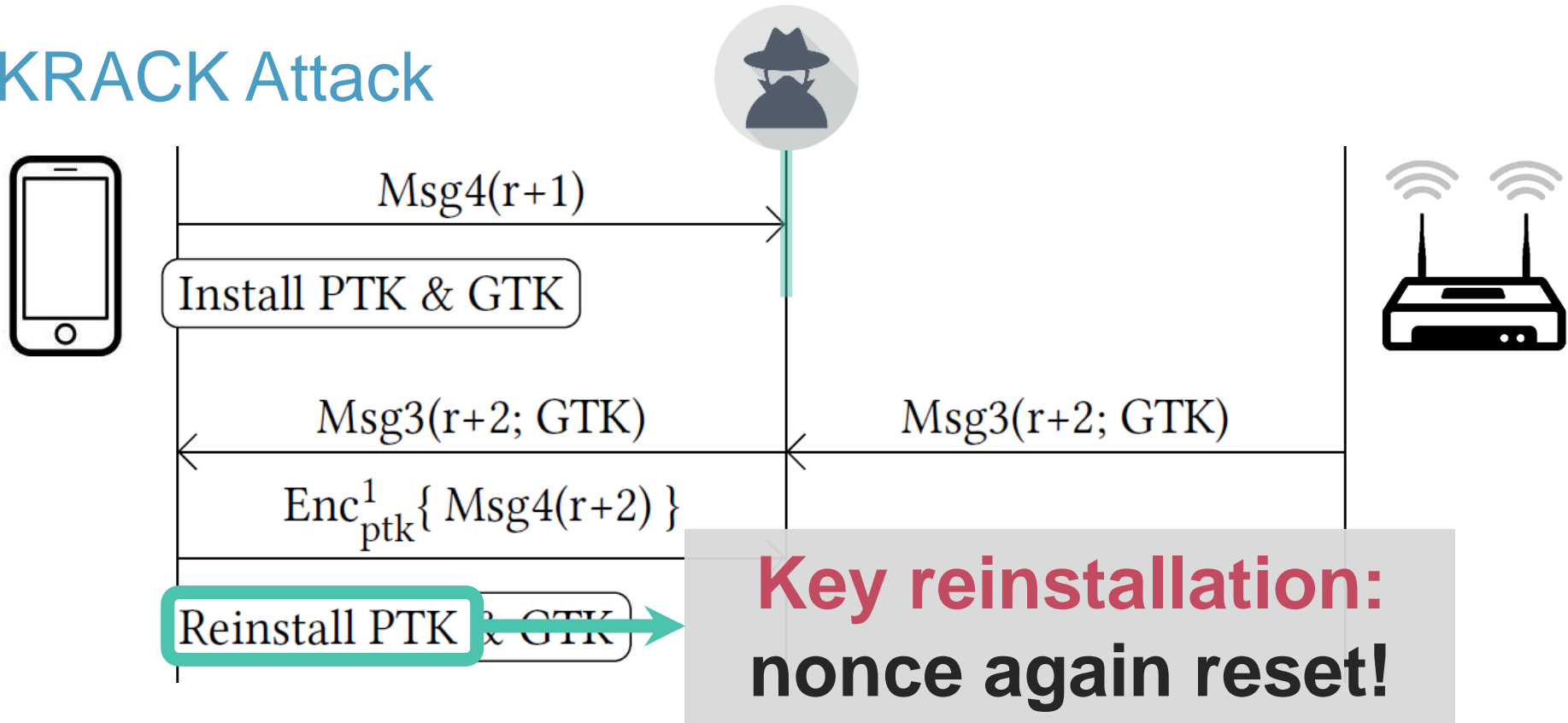
# KRACK Attack



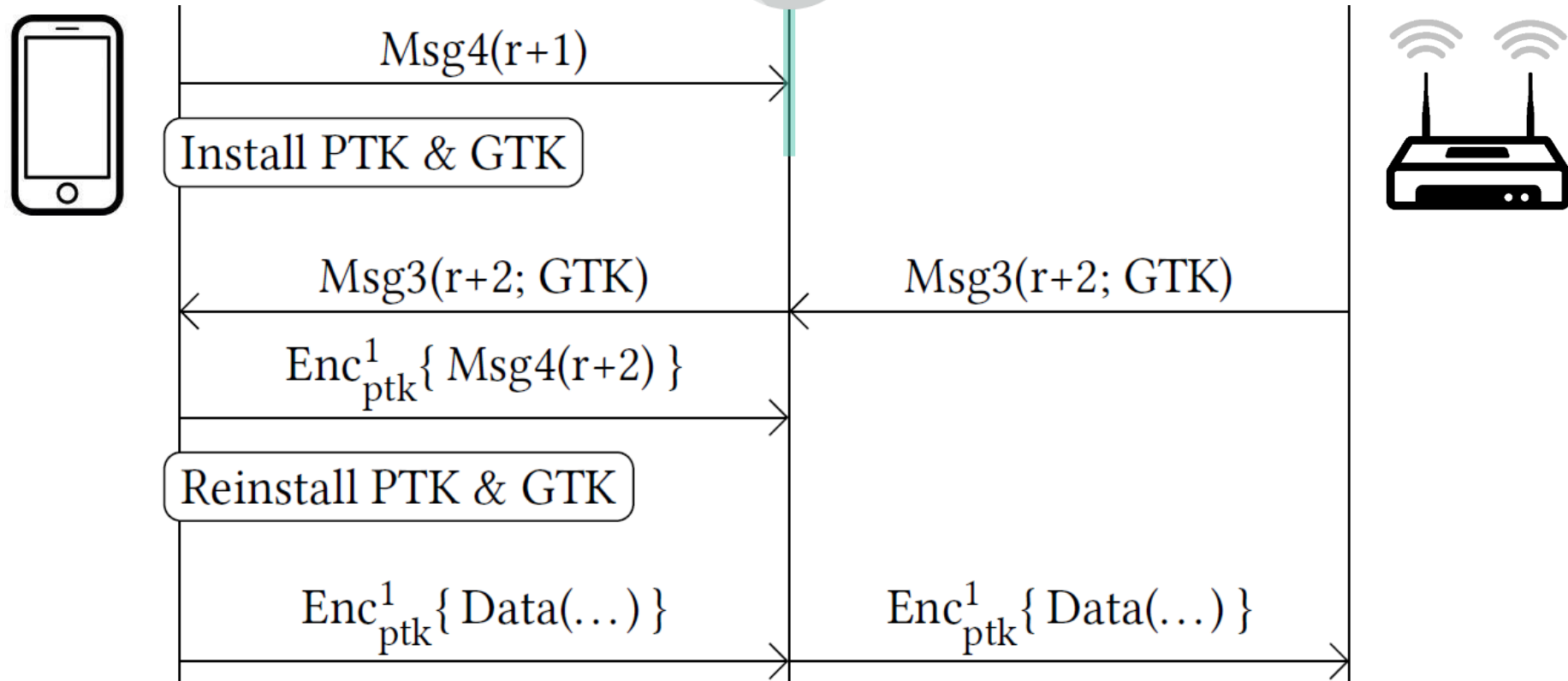
# KRACK Attack



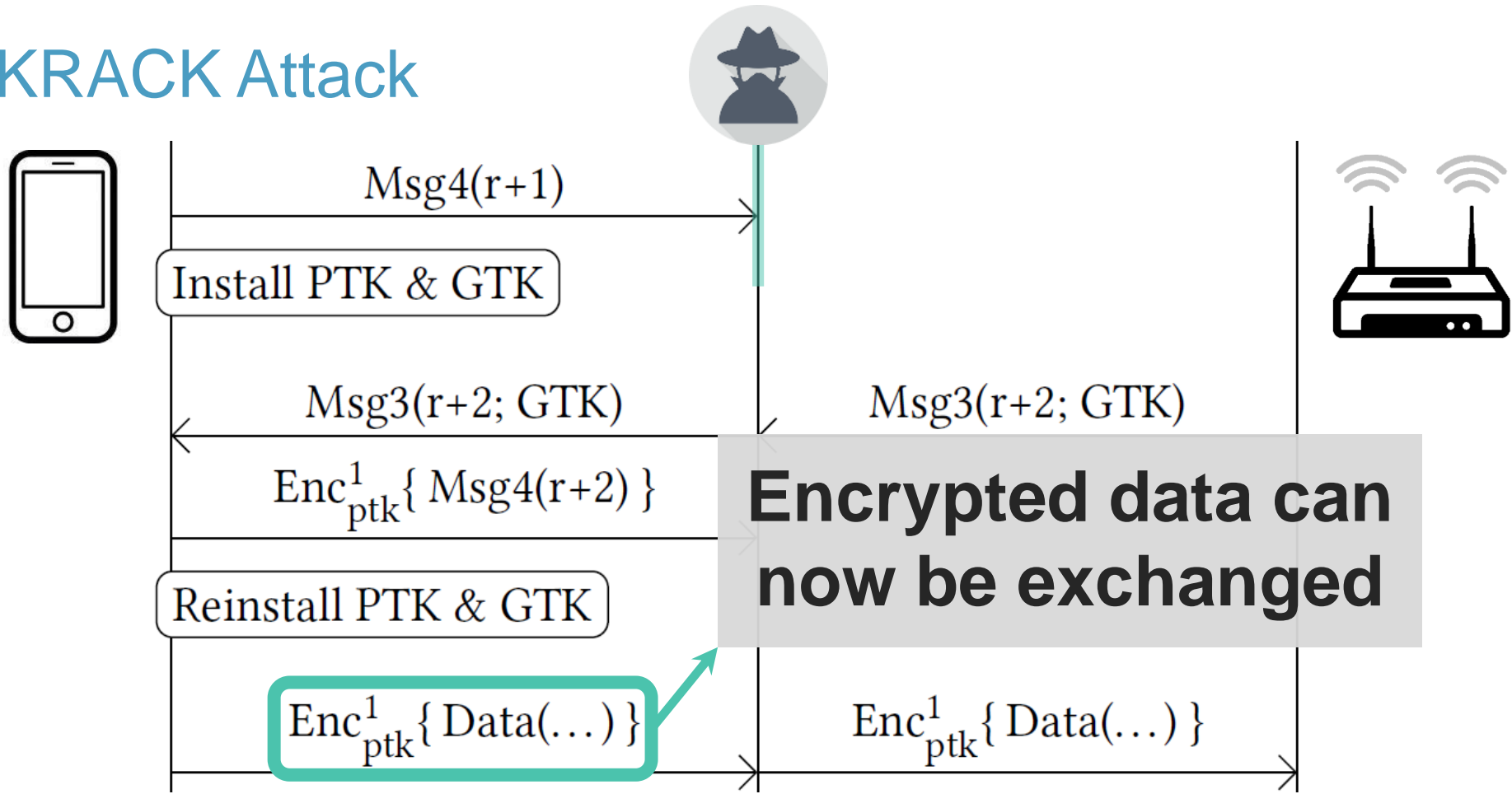
# KRACK Attack



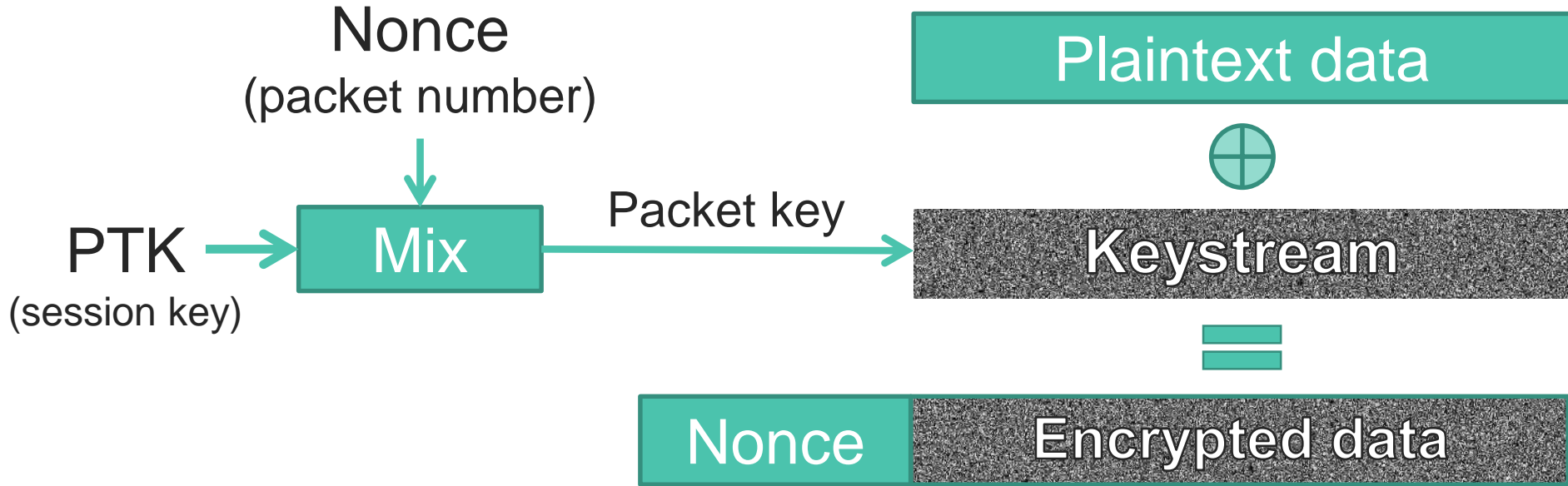
# KRACK Attack



# KRACK Attack



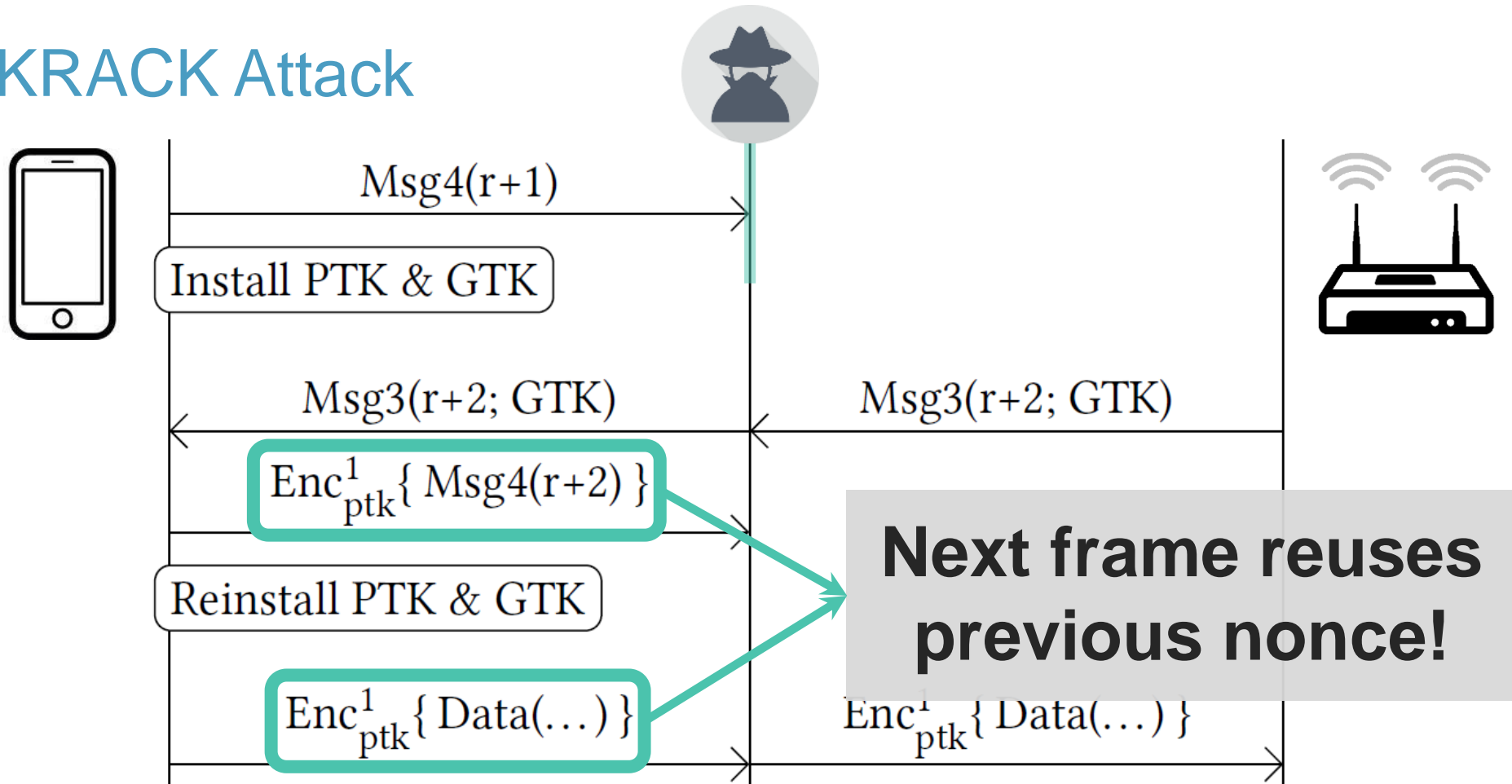
# Quick background: encryption



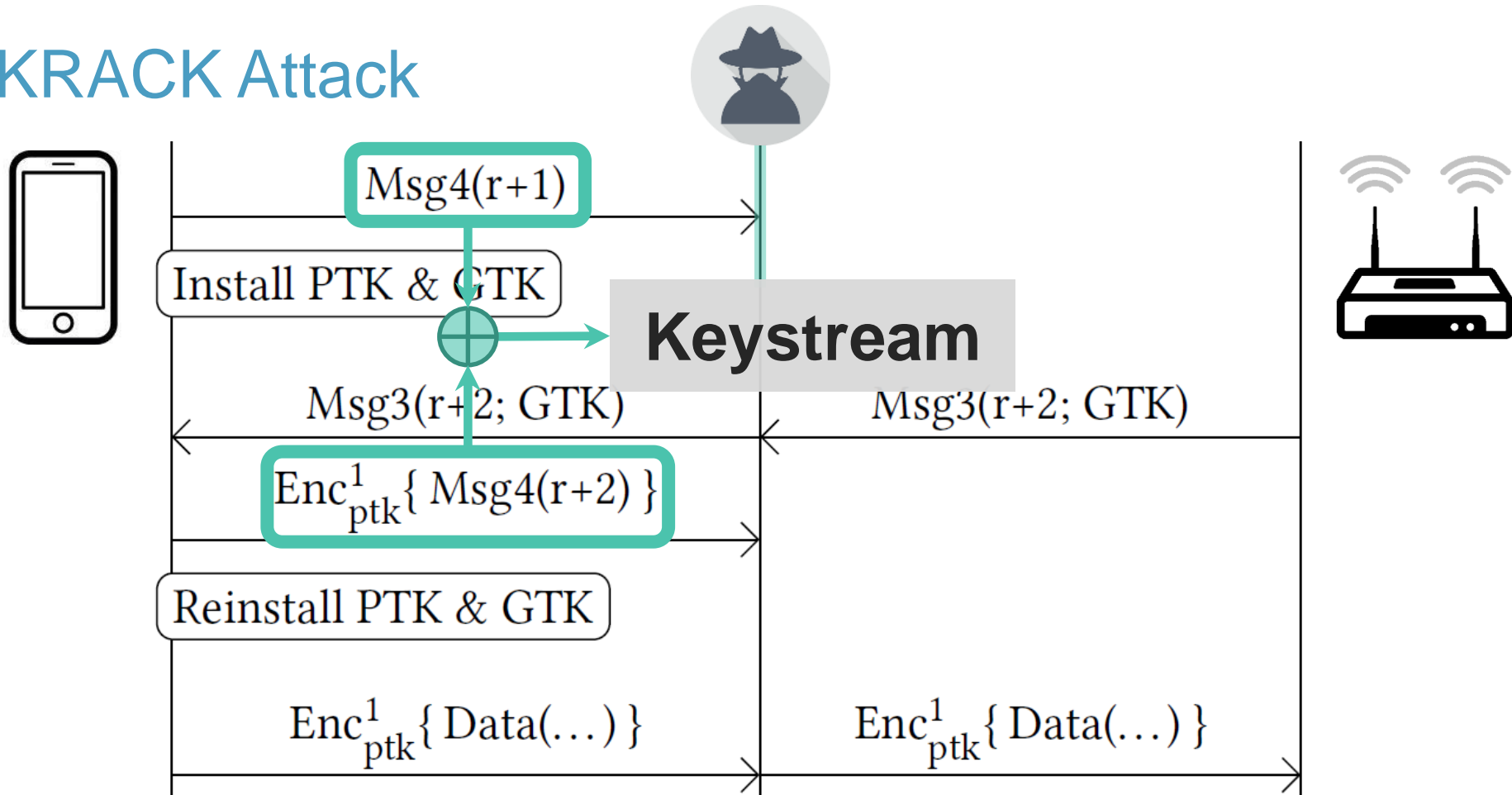
→ Nonce reuse implies keystream reuse (in all WPA2 ciphers)



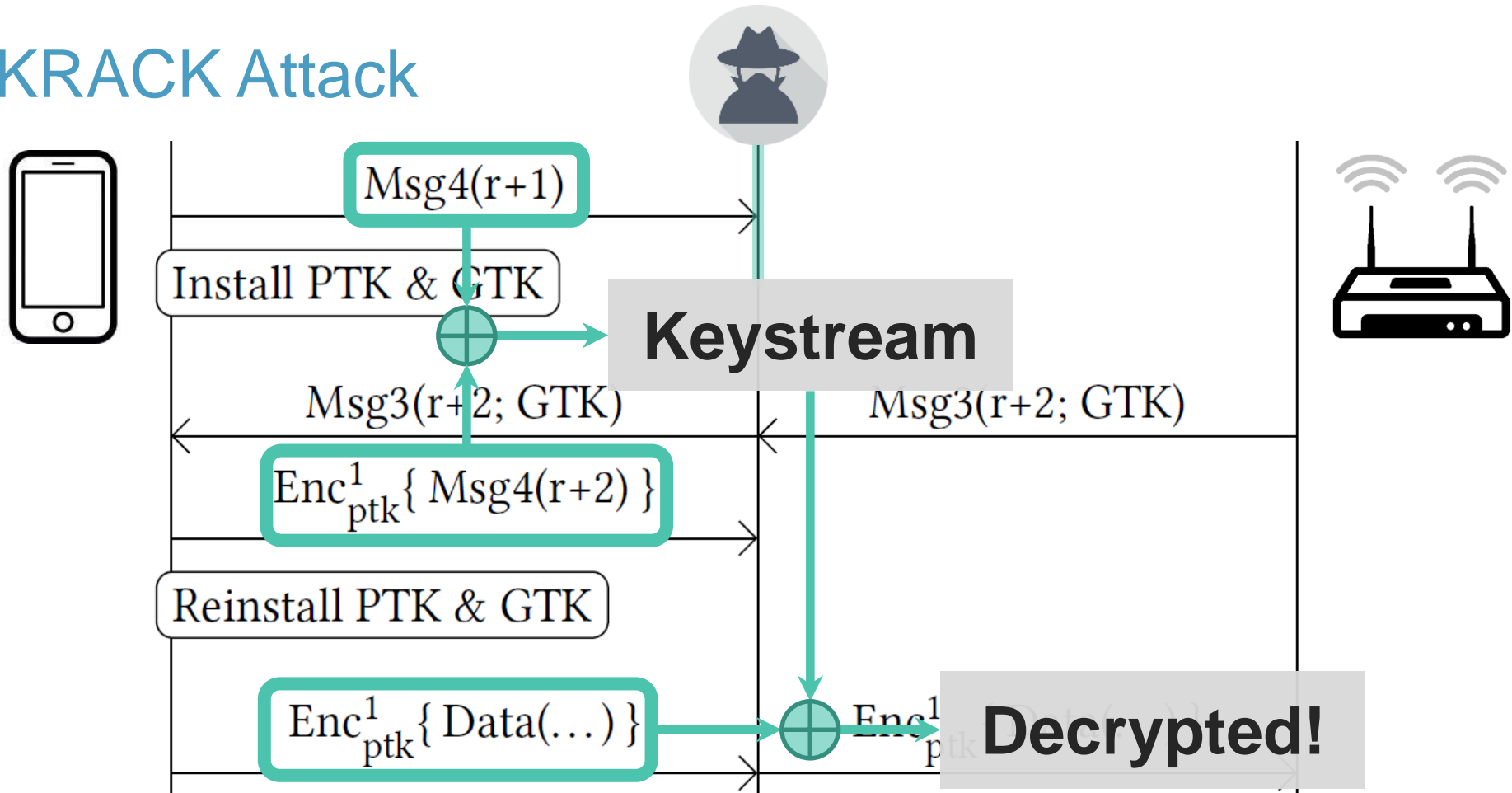
# KRACK Attack



# KRACK Attack



# KRACK Attack



# Conclusion



- › Jamming is cheap
- › Selective jamming also possible
- › Can even use mobile phone!
- › Facilitates KRACK attacks

# Thank you!

## Questions?

[github.com/vanhoefm/modwifi](https://github.com/vanhoefm/modwifi)

# References

1. M. Raya, J.-P. Hubaux, and I. Aad. DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots. In MobiSys, 2004.
2. M. Vanhoef and F. Piessens. Practical verification of WPA-TKIP vulnerabilities. In ASIACCS, 2013.
3. M. Vanhoef and F. Piessens. Advanced Wi-Fi attacks using commodity hardware. In ACSAC, 2014.
4. C. Cox. Hi-tech car thieves hit the streets with £30 jamming devices bought over the internet. In Manchester Evening News, 2014.
5. C. Arthur. Car thieves using GPS 'jammers'. In The Guardian, 2010.
6. J. Weiner. High-tech thieves used phone-jammer in \$74k sunglass heist, cops say. In Orlando Sentinel, 2011.
7. P. Dandumont. Don't trust geolocation! Retrieved 5 October, 2015, from [journaldulapin.com/2013/08/26/dont-trust-geolocation/](http://journaldulapin.com/2013/08/26/dont-trust-geolocation/)
8. Gollakota et al. They can hear your heartbeats: non-invasive security for implantable medical devices. In SIGCOMM, 2011.
9. Schulz et al. Massive Reactive Smartphone-Based Jamming using Arbitrary Waveforms and Adaptive Power Control. In WiSec, 2017.
10. M. Vanhoef and F. Piessens. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. In ACM CCS, 2017.

# Multi-channel MitM also enables other attacks



## Traffic Analysis

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

## Attacking broadcast TKIP

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

Cho~~p~~Chop

# Multi-channel MitM also enables other attacks

Exploit implementation bugs

- › **Block** certain handshake messages
- › E.g. bugs in 4-way handshake



Specialized attack scenarios

- › E.g. **modify** advertised capabilities
- › See [X] for details



# 1. Attack Wi-Fi Geolocation

Location determined by nearby SSIDs



Geolocation attack [7]

- › Inject SSIDs of another location
- › Problem: can only spoof locations with more APs
- › Solution: selectively jam nearby Aps

→ Never blindly trust Wi-Fi geolocation!

## 2. Use as a defense system

Use jamming to **protect** a network

- › Selectively jam rouge APs
- › Wearable shield to protect medical implants that constantly sends jamming signal [8]
- › ... (it's an active research topic)

## 2. Use as a defense system

Legal aspects are unclear

Blocking personal hotspots:

- › Done by Marriott and Smart City Holdings
- › Complaint was filled to the FCC
- › Settled for fine of \$600,000 and \$750,000



Is blocking **malicious or rogue** hotspots legal?

# DOMINO defense system

Also capable of detecting selective jammers

- › Assumes MAC header is still valid
- › Attacker has low #(corrupted frames)
- › Thrown of the network

Unfortunately it's flawed

- › Jammer (corrupted) frames are not authenticated
- › **We can pretend that a client is jamming others**