

WiFi Security

Dr. Kari Kostinen and Prof. Srdjan Capkun

Security of Wireless Networks – Fall 2022

ETH Zurich

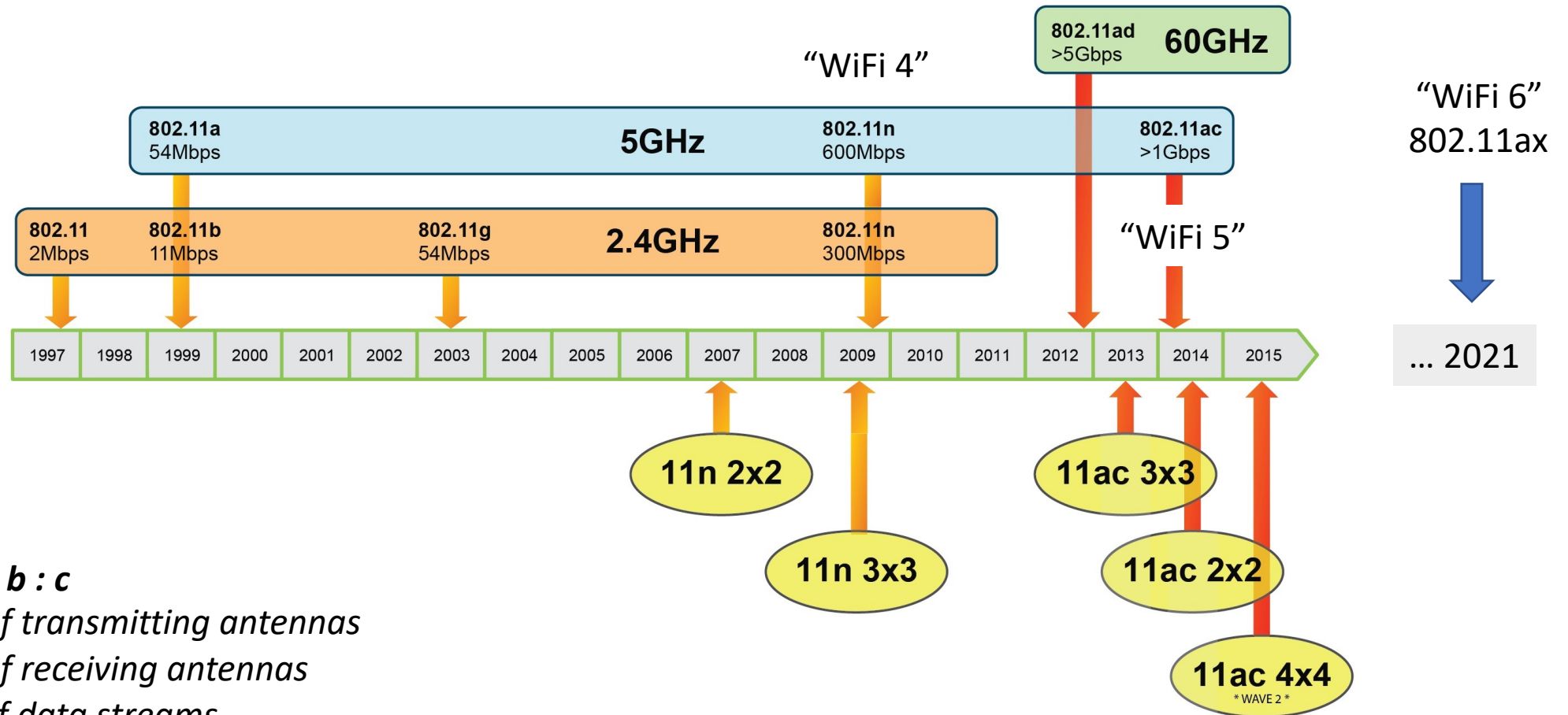
Lecture outline

- **Part 1:** WiFi basics
 - Communication concepts and standards
- **Part 2:** Basic manipulations
 - Building blocks for attacks
- **Part 3:** WiFi security standards and attacks
 - WEP, WPA/TKIP, WPA2, WPA3

Part 1: WiFi basics

Communication concepts and standards

WiFi communication standards



WiFi versions

WiFi 4 WiFi 5

Standard	802.11a	802.11b	802.11g	802.11n	802.11ac	802.11ad
Year introduced	1999	1999	2003	2009	2012	2014
Maximum data transfer speed	54 Mbps	11 Mbps	54 Mbps	65 to 600 Mbps	78 Mbps to 3.2 Gbps	6.76 Gbps
Frequency band	5 GHz	2.4 GHz	2.4 GHz	2.4 or 5 GHz	5 GHz	60 GHz
Channel bandwidth	20 MHz	20 MHz	20 MHz	20, 40 MHz	40, 80, 160 MHz	2160 MHz
Highest order modulation	64 QAM	11 CCK	64 QAM	64 QAM	256 QAM	64 QAM
Spectrum usage	OFDM	DSSS	DSSS, OFDM	OFDM	SC-OFDM	SC, OFDM
Antenna configuration	1×1 SISO	1×1 SISO	1×1 SISO	Up to 4×4 MIMO	Up to 8×8 MIMO, MU-MIMO	1×1 SISO

WiFi 6

Standard: 802.11ax

Year introduced: 2021

Frequency bands: 2.4GHz, 5GHz, 6Ghz

Spectrum use: OFDMA

Modulation: 1024 QAM

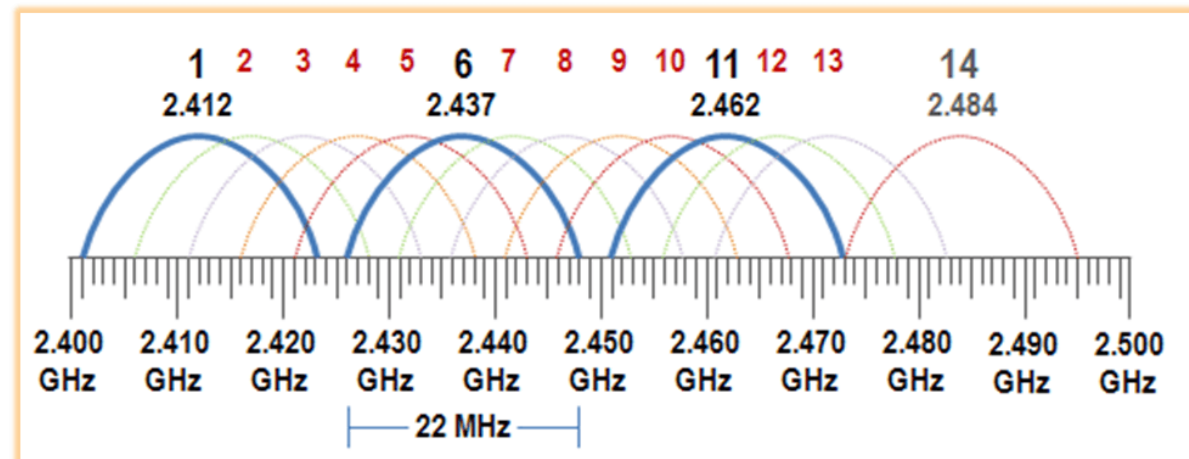
Antennas: MIMO uplink and downlink

Goals:

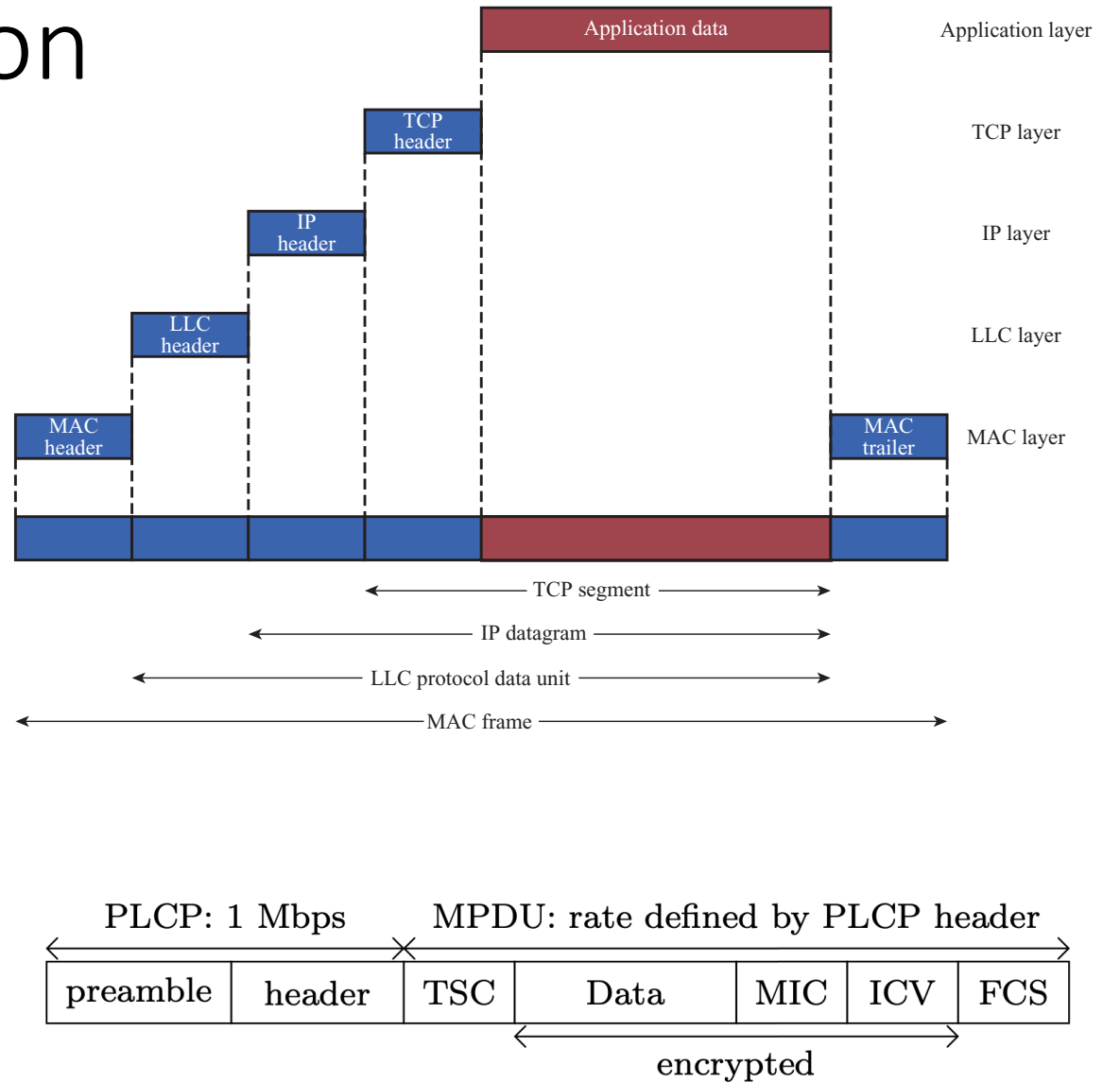
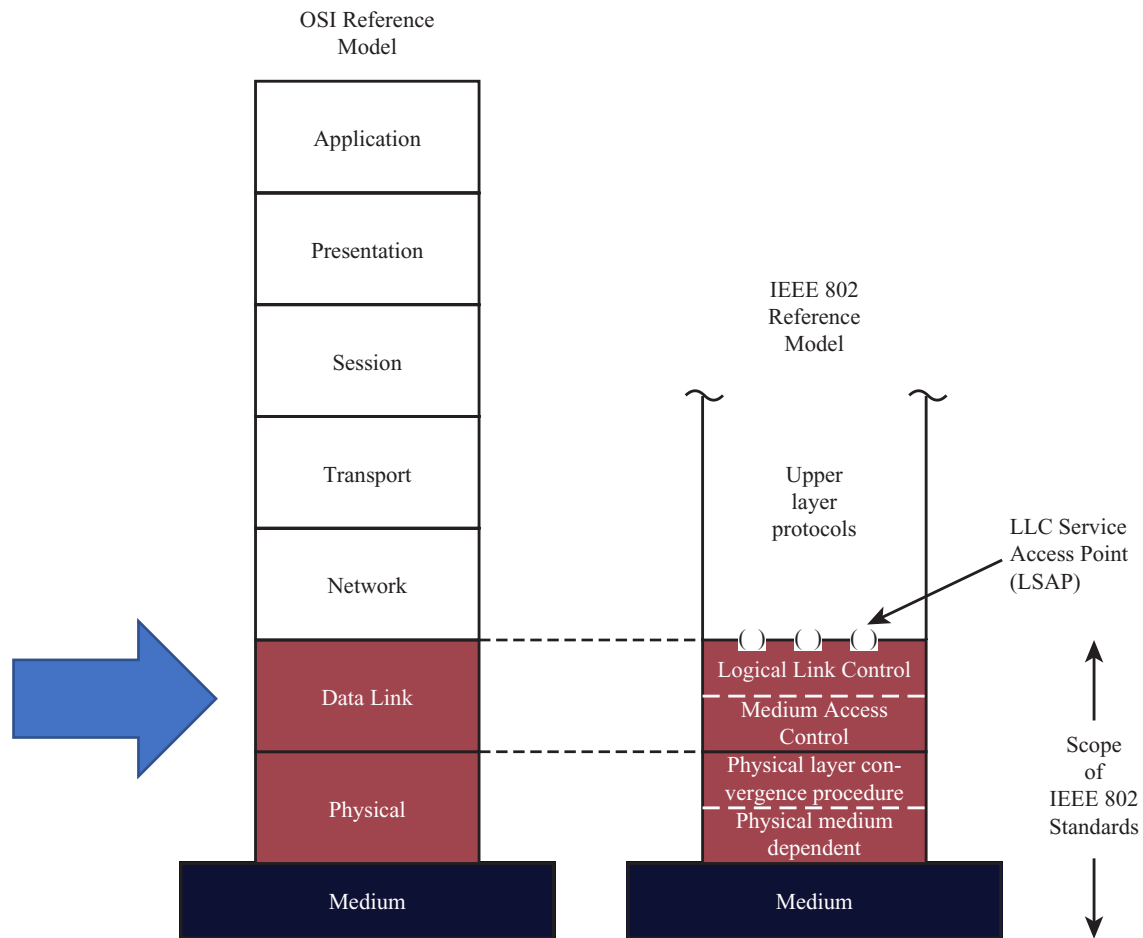
- High client density
- Latency
- Power control

Channels

- WiFi standards define radio frequency **ranges**
 - Typically: 2.4 GHz, 5 GHz
- Each range is divided into several **channels**
 - Usually **numbered**
 - With 5 **MHz** spacing



Layers and encapsulation



Medium access control

- Multiple devices share the same communication medium
- **Goals**
 - Reliable data delivery
 - Security
- Why handle these services at MAC layer?
 - Can be more efficient than higher layer (e.g., TCP)
 - Can be safer than rely on applications or higher layer protocols

CSMA/CA

- **Carrier-Sense Multiple Access (CSMA)** with collision avoidance

1. **Carrier sense:**

- Prior to sending, monitor shared medium

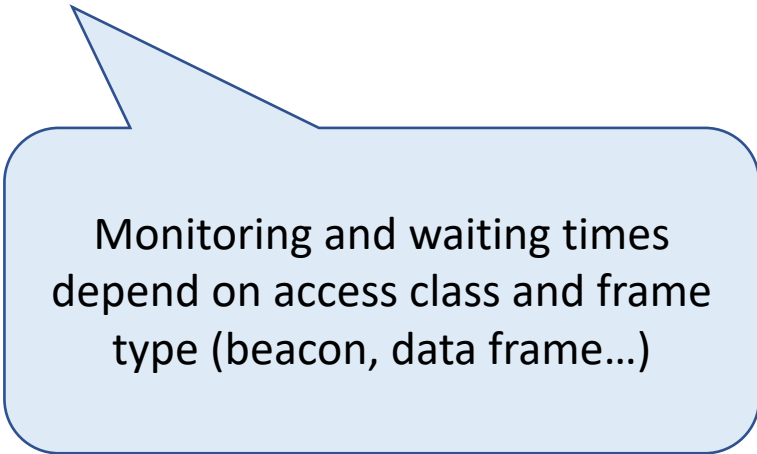
2. **Collision avoidance:**

- If another node detected, wait for randomized “**backoff period**”
- Then listen again

3. **Transmit** entire frame

- Wait for ACK
- If no ACK, wait for backoff period

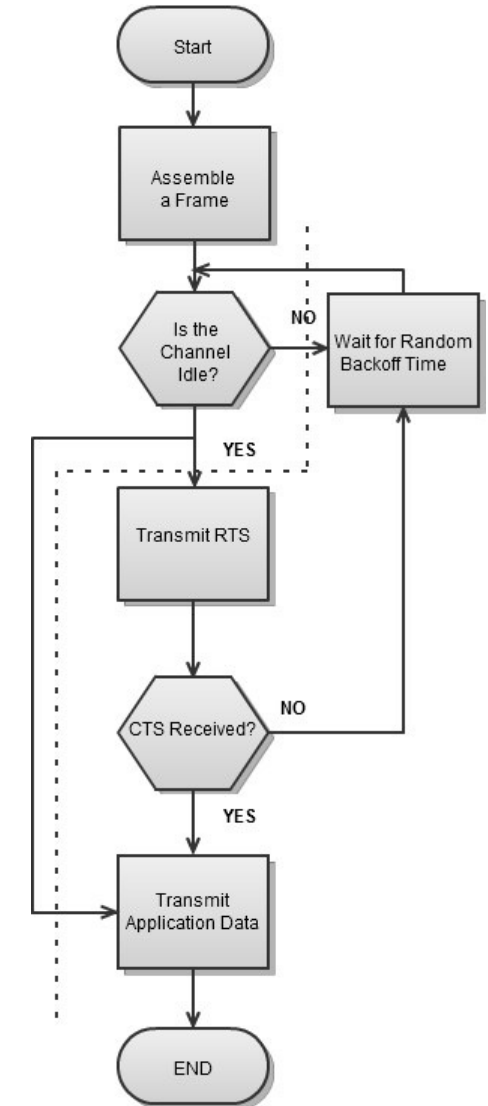
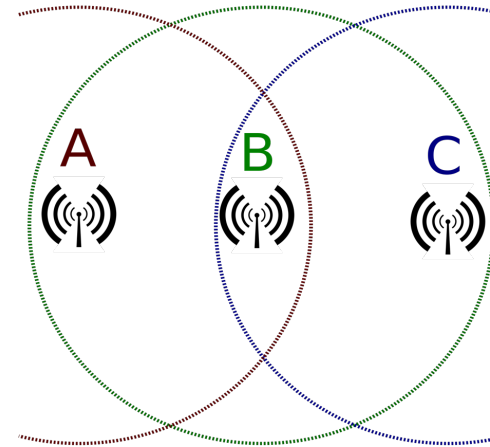
- Distributed Coordination Function (DCF)



Monitoring and waiting times depend on access class and frame type (beacon, data frame...)

Hidden node problem

- Example
 - B can communicate with A and C
 - A cannot communicate with C
- Potential solution
 - After backoff:
 - Transmit **Request to Send (RTS)** to AP
 - Wait for **Clear to Send (CTS)** from AP
- Optional features in 802.11

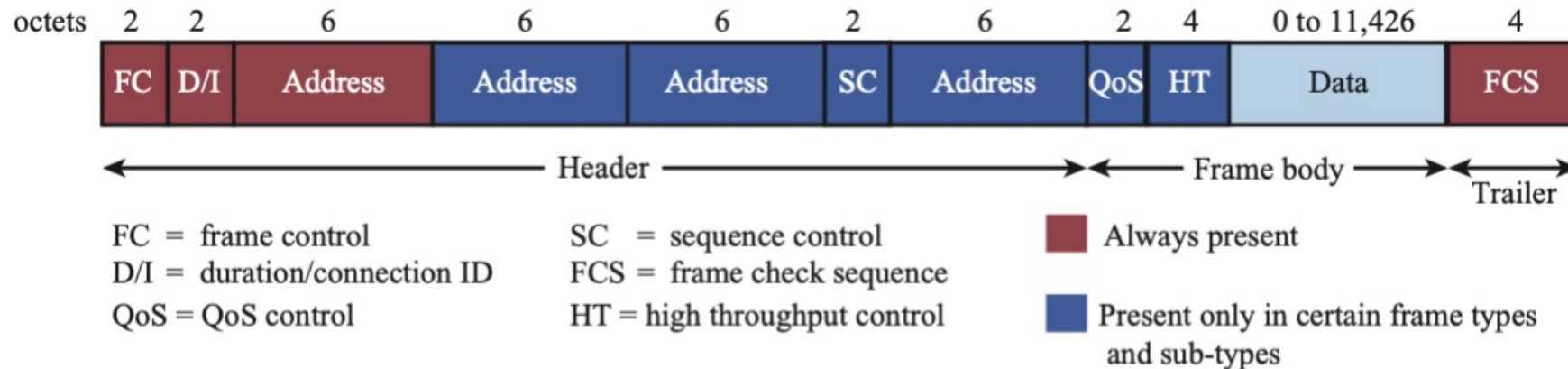


MAC Frames

- 802.11 frame allows 3-4 addresses
 - Common: sender, AP, destination

Frame types

- Data (user traffic)
- Control (e.g., beacons, RTS)
- Management (e.g., power)



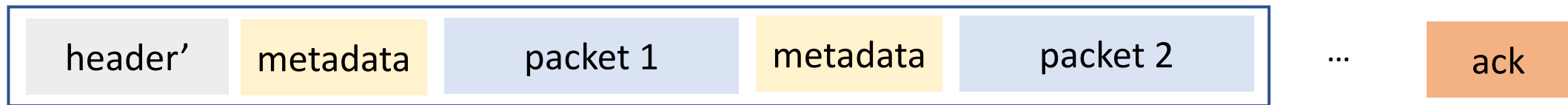
(a) MAC frame

Aggregation

Multiple frames



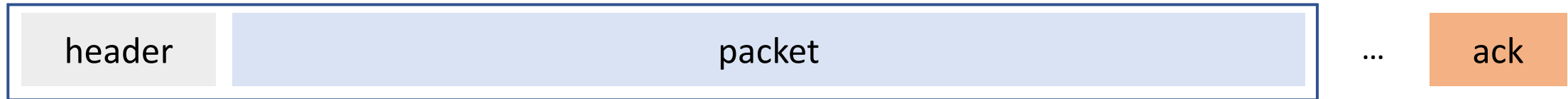
Aggregated frame



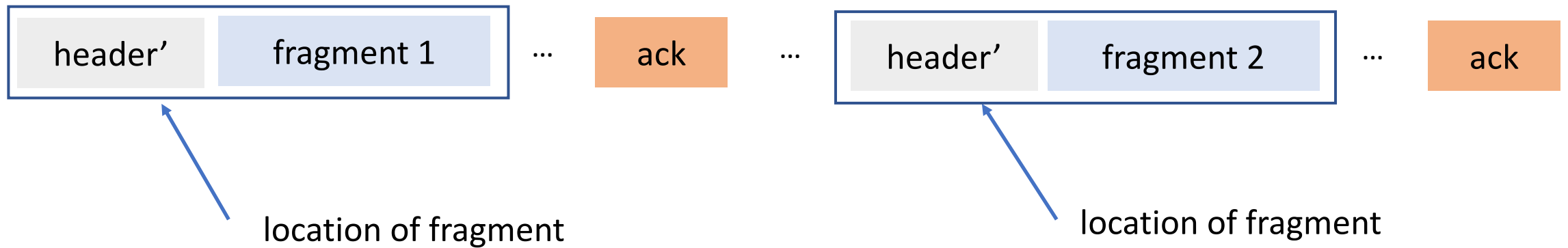
↑
aggregated flag

Fragmentation

Large frames



Fragmented frame



Other features

- **Network discovery**

- AP sends periodic **beacons** (management frames)
- Contains: SSID, supported options...

SSID = WiFi network name

- **Quality of Service**

- Separate QoS classes and prioritization

- **Power management**

- Stations/clients can query missed frames using polling frames

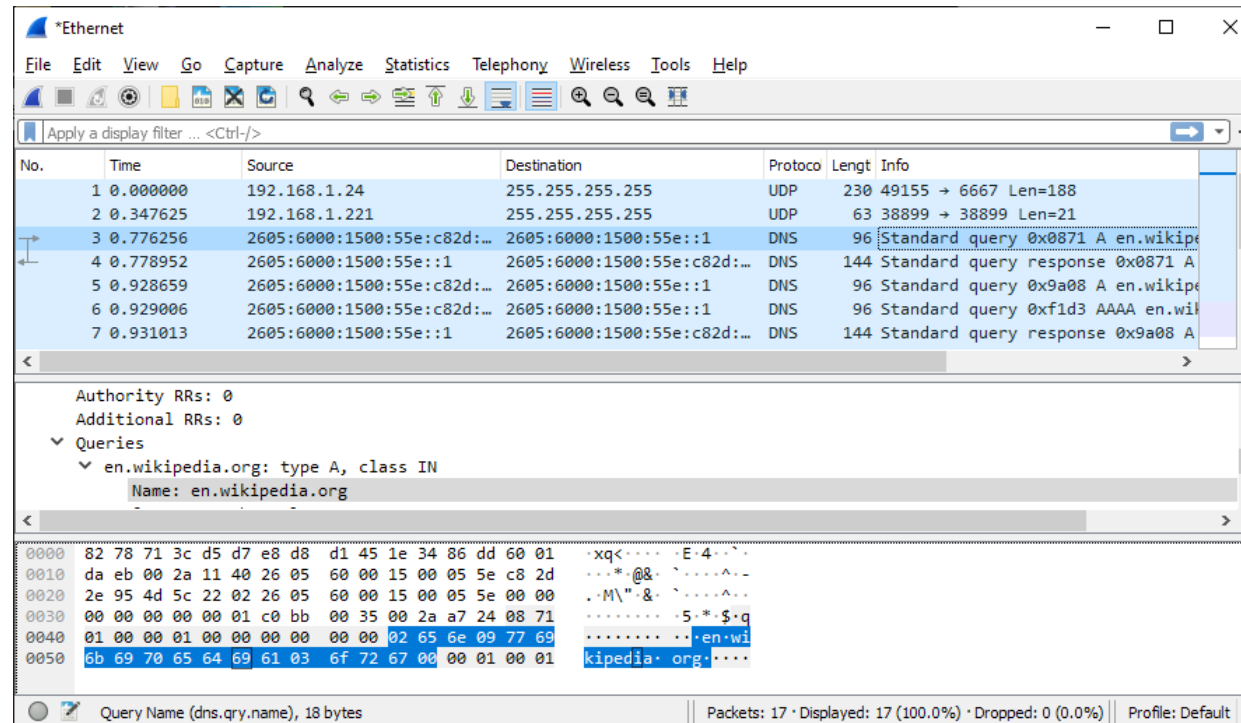
- And many others...

Part 2: Basic manipulations

Building blocks for attacks

Capturing all frames

- **Typically easy**
 - Install packet capture SW that sets WiFi interface to “promiscuous mode”



Wireshark.org

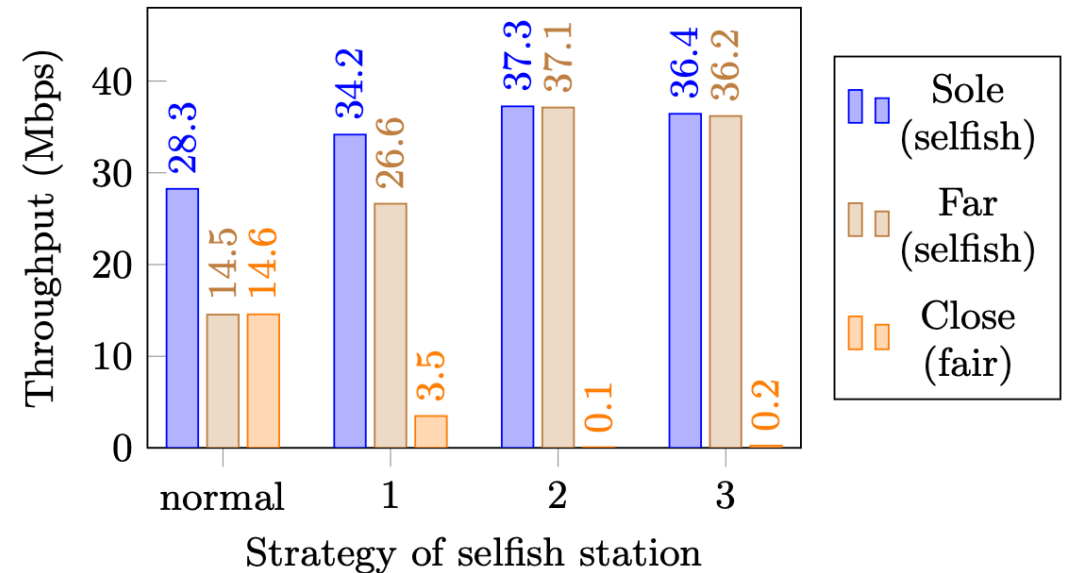
Communication fairness

- Previously mentioned “Distributed Coordination Function” is **fair**
 - Theoretical proof (2004)
- **Assuming all stations follow the rules!**
 - Listen before transmitting
 - If medium busy, wait
 - If no ACK received (likely collision), wait some more

Berger-Sabbatel et al. “Fairness and its impact on delay in 802.11 networks” GLOBECOM, 2004

Unfair channel usage

- How to get **more throughput**?
 1. Buy commodity WiFi device
 2. Modify the driver
 3. Manipulate registers (backoff params)
- Example experiment (2014)
 - Different strategies (1,2,3)
 - Sole selfish or two stations



Vanhoef and Piessens. "Advanced Wi-Fi Attacks Using Commodity Hardware". ACSAC'14.

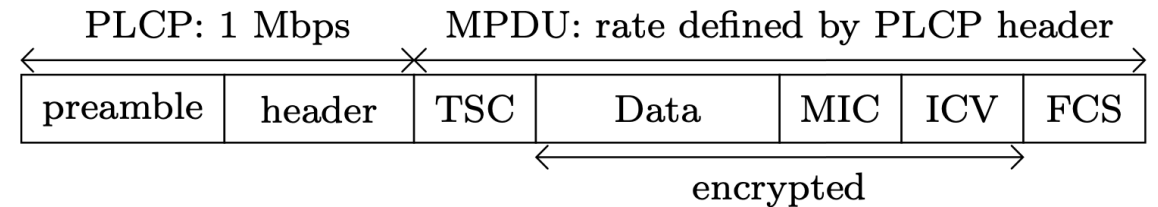
Simple jamming

- How to turn your WiFi device into **continuous jammer**?
 1. Disable carrier sense
 2. Reset interframe space and disable backoff
 3. Don't wait for ACK
 4. Queue large number of frames for transmission
- Different ways to block WiFi frames
 - **Trigger carrier sense** of transmitter (prevent **sending**) – **Less power**
 - **Mangle the frame** at receiver (prevent **receiving**) – **More power**
- Example experiment (2014)
 - Effective from 80 meters

Vanhoeft and Piessens. “Advanced Wi-Fi Attacks Using Commodity Hardware”. ACSAC’14.

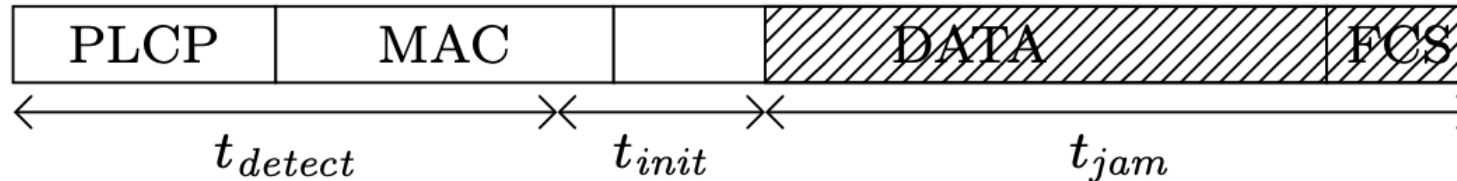
Selective jamming

- Main idea
 - Listen
 - Decode prefix of incoming frame
 - Decide whether to jam
- Challenge
 - **One needs to be fast!**



Recall:

PLCP = Physical Layer Control Protocol



Selective jamming experiment

- Setup

- Victim at 70 cm from AP
- Jammer at 1 meter

Limitation:

Commodity wireless chip writes to RAM with a delay
Example: after 48 bytes decoded

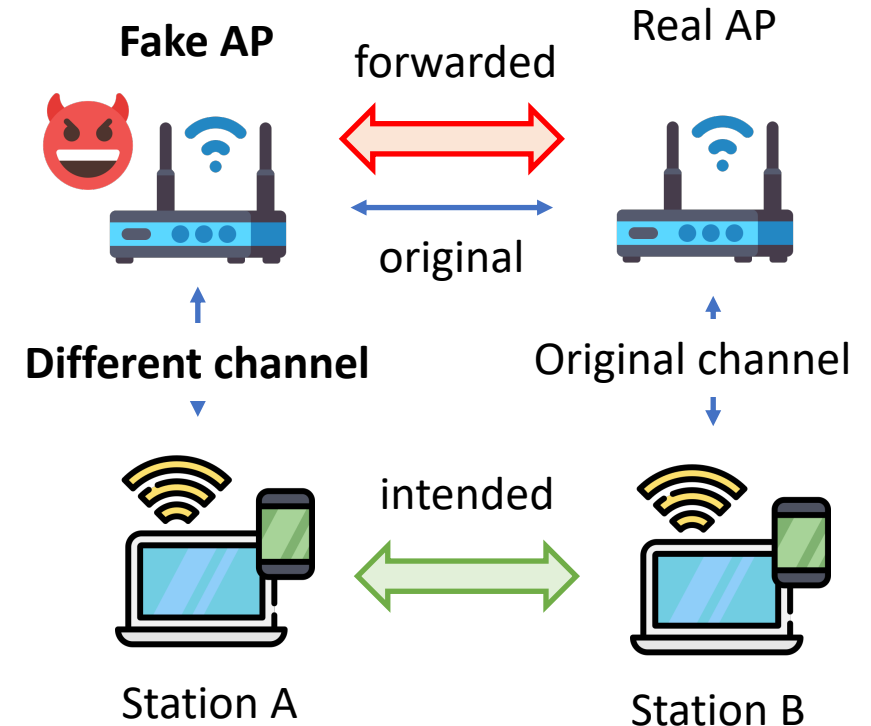
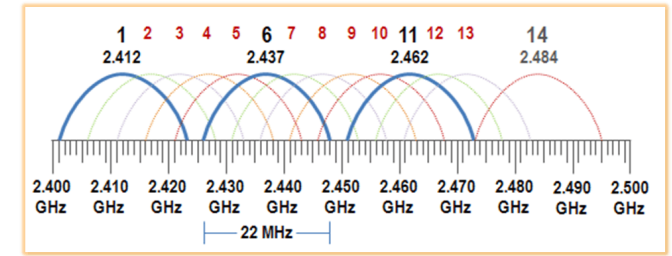
- Results

- 52% to 99% of beacons malformed
- Depending on victim/attacker type of antenna, radio chip, noise filters...

Vanhoeft and Piessens. "Advanced Wi-Fi Attacks Using Commodity Hardware". ACSAC'14.

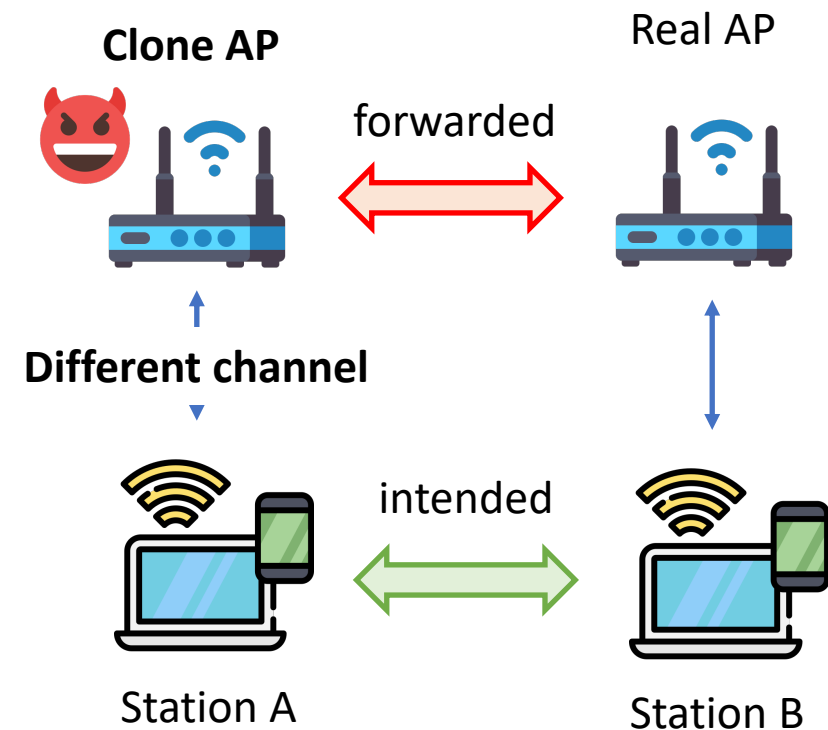
Man-in-the-middle position

- Useful building block for many attacks
 - Route through adversary
 - Intercept and manipulate traffic
- Why not selective jamming?
 - Success rate not 100%
 - Timing requirements → hard to be selective
- Main idea
 - Clone AP on a **different channel** but same MAC address
 - **Forward frames** from fake AP to real AP



MitM continued

- How to make victims connect to fake AP?
- May not work
 - Selectively jam beacons/probes (too short)
- Probably works
 - Continuously jam real AP → clients switch



Part 3: WiFi security standards

WEP, WPA/TKIP, WPA2, WPA3

WiFi security standards



- Adoption of latest standards still low...
- Example study (Schepers, 2021)
 - 30% of examined networks still support WPA/TKIP
 - None of the examined networks supports only WPA3

Schepers et al. "Let Numbers Tell the Tale: Measuring Security Trends in Wi-Fi Networks and Best Practices." WiSec'21

WEP

- The first WiFi security standard (**1997**)
 - Design developed by the industry

WEP = “Wired Equivalence Privacy”

- **Goal:** provide “some level” of communication protection
 1. Confidentiality (encryption)
 2. Integrity (checksum)
 3. Access control (authentication protocol)
- Likely threat model: **passive adversary**
- Few years later WEP fully broken

WEP basic operation

1. Checksum

- Compute plaintext $P = (M, c(M))$
- CRC algorithm as c

2. Encryption

- Encrypt using RC4 stream cipher
- Choose initialization vector IV
- Compute ciphertext $C = P \oplus RC4(IV, k)$



keystream

3. Transmit

- Send (IV, C)

Implementation:

- link-level (MAC frames)

Assumption:

- shared key k

Problem 1: Confidentiality

- Keystream reuse problem
 - If $C_1 = P_1 \oplus RC4(IV, k)$ and $C_2 = P_2 \oplus RC4(IV, k)$
 - Then $C_1 \oplus C_2 = P_1 \oplus P_2$
- Keystream reuse implications
 - **One known plaintext** reveals another
 - **No known plaintext:** often redundancy to recover both from $P_1 \oplus P_2$
- Keystream reuse in WEP
 - **IV space is 24 bits** \rightarrow repeats after half a day (5 Mbps)
 - Standard does not mandate **IV** change
 - Some device set **IV = 0 on reboot**

Ways to predict plaintext

- Known structure like IP headers
- Packet injection from Internet

Borisov et al. "Intercepting Mobile Communications: The Insecurity of 802.11." MobiCom'01

Problem 2: Integrity

- The used checksum c (CRC-32) is not cryptographic MAC

- **Message modification**

- Checksum property: $c(x \oplus y) = c(x) \oplus c(y)$
 - Enables **controlled modifications**

$$\begin{aligned} C' &= C \oplus \langle \Delta, c(\Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M, c(M) \rangle \oplus \langle \Delta, c(\Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M \oplus \Delta, c(M) \oplus c(\Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M', c(M \oplus \Delta) \rangle \\ &= \text{RC4}(v, k) \oplus \langle M', c(M') \rangle. \end{aligned}$$

- **Example**

- Assume ciphertext C with unknown data M
 - One can find C' that decrypts to $M' = M \oplus \Delta$
 - Such that Δ is arbitrary

Problem 3: Access control

- Authentication in WEP
 1. AP sends **challenge in plaintext**
 2. Station replies with **WEP encryption of challenge** (proof of key possession)
 3. AP completes network association
- Simple attack
 - Monitor legitimate authentication → learn plaintext/ciphertext pair
 - Derive keystream (xor)
 - Compute valid response for new challenge

Borisov et al. "Intercepting Mobile Communications: The Insecurity of 802.11." MobiCom'01

WEP cracking tools (ca. 2007)

- Recover key in few minutes
 - http://www.aircrack-ng.org/doku.php?id=simple_wep_crack

Aircrack-ng 0.9

[00:03:06] Tested 674449 keys (got 96610 IVs)

KB	depth	byte(vote)
0	0/ 9	12(15) F9(15) 47(12) F7(12) FE(12) 1B(5) 77(5) A5(3) F6(3) 03(0)
1	0/ 8	34(61) E8(27) E0(24) 06(18) 3B(16) 4E(15) E1(15) 2D(13) 89(12) E4(12)
2	0/ 2	56(87) A6(63) 15(17) 02(15) 6B(15) E0(15) AB(13) 0E(10) 17(10) 27(10)
3	1/ 5	78(43) 1A(20) 9B(20) 4B(17) 4A(16) 2B(15) 4D(15) 58(15) 6A(15) 7C(15)

KEY FOUND! [12:34:56:78:90]

Probability: 100%

WEP summary

- **Recall:** WEP was intended to provide “some protection”
 - Probable thinking: precise “in-flight” message manipulation difficult
 - All above attack possible with **simple recording** (passive adversary)
- Open design process would have probably caught many flaws
- Challenges
 - Fast in cheap/simple hardware
 - Secure key and IV management → devices need to maintain state

WPA/TKIP

- The second WiFi security standard (**2003**)
 - Designed as **transitional mechanism**

Terminology:

WPA = WiFi Protected Access

TKIP = Temporal Key Integrity Protocol

- Main challenge
 - Confidentiality and integrity protection implemented in WiFi chip
 - **Design that works with legacy (WEP) devices**
 - Firmware or driver update
- Long-term view
 - A study from **2013** found that 70% of networks still allow TKIP
 - A study from **2019** found 50%

Schepers et al. "Practical Side-Channel Attacks against WPA-TKIP." ACSAC'19

WPA/TKIP operation

- Goals
 - No keystream reuse
 - Cryptographic MAC
- Constraints
 - Compatible **RC4/WEP** hardware
- Enhancements
 1. Augment encryption with **per-packet key mixing**
 2. New **integrity protection** mechanism (MICHAEL)
 3. Replay protection using **counters** (TSC)

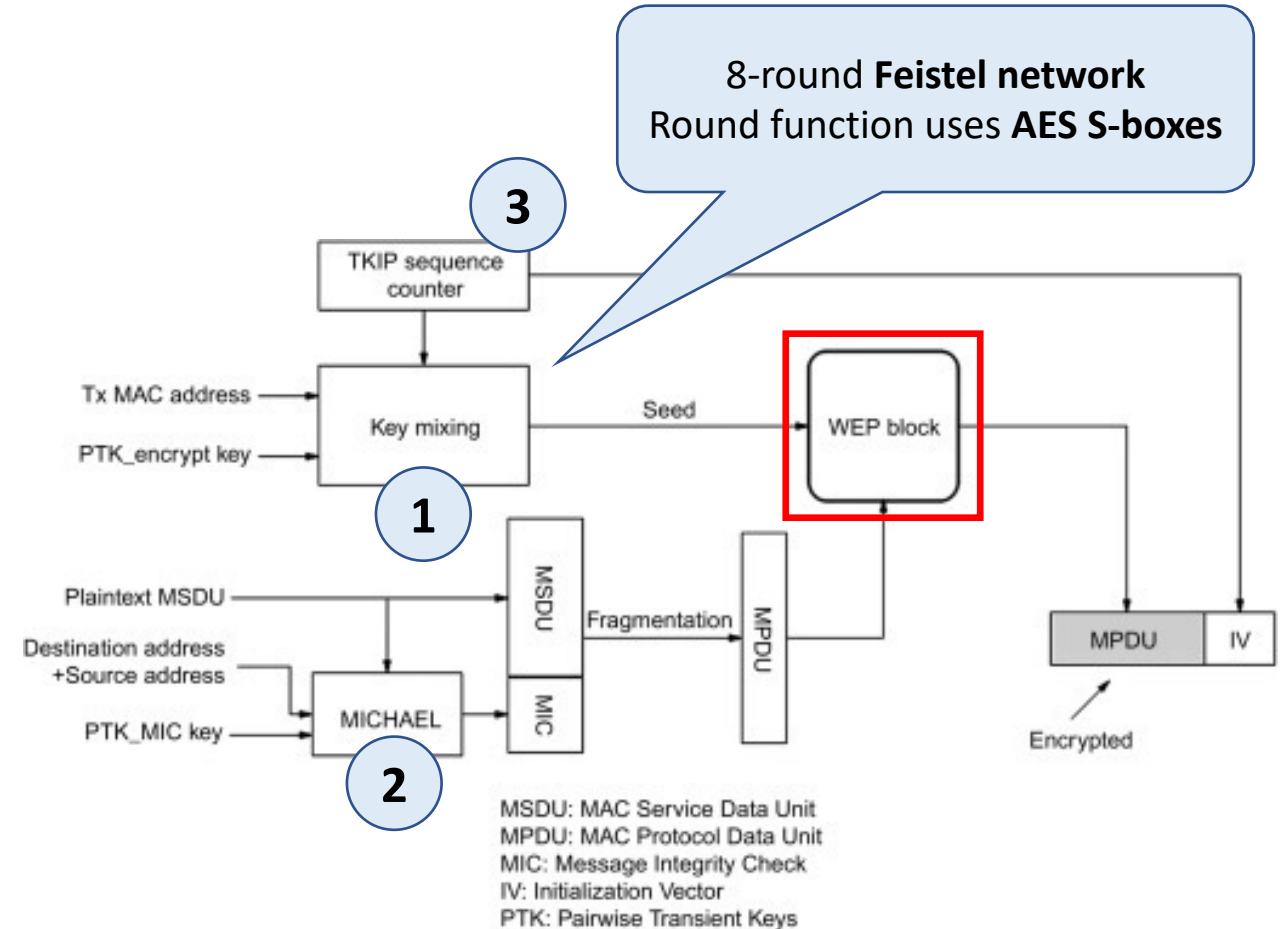


Figure from Bensky, "Short-range Wireless Communication" (2019)

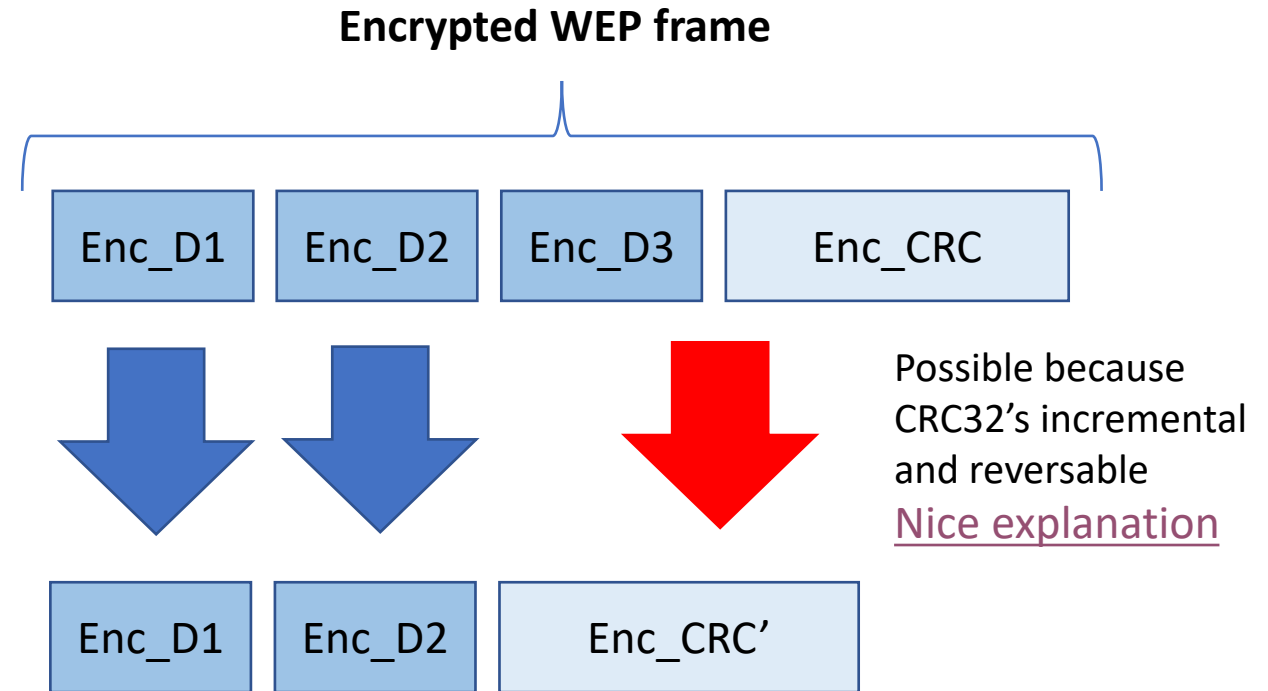
ChopChop technique

- **Starting point**

- Unknown key
- Unknown plaintext

- **Steps**

1. Record encrypted frame
2. Remove last data byte
3. **Guess chopped plaintext byte (D3)**
4. **Construct encrypted checksum CRC'**
5. **Check guess correctness using AP as oracle**
6. Correct guess → learn plaintext byte
7. **Repeat/chop** next byte



End result:

- Known plaintext packet →
- Known keystream for one packet

WPA/TKIP attack

- **Observations**

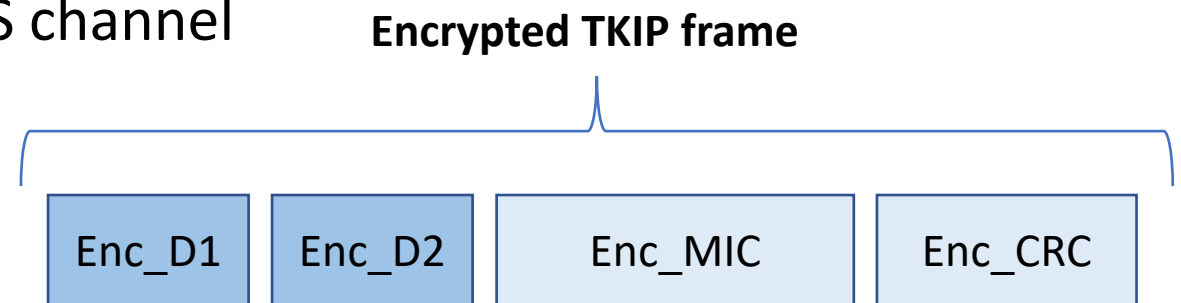
- TKIP frame has MIC (MICHAEL) and CRC
- False MIC → failure report (**oracle!**)
- MIC algorithm is reversible
- Counter (STC) is separate for each QoS channel

- **Attack**

1. ChopChop approach (+ waiting)
2. Learn plaintext MIC value
3. Reverse to learn MIC key

End result:

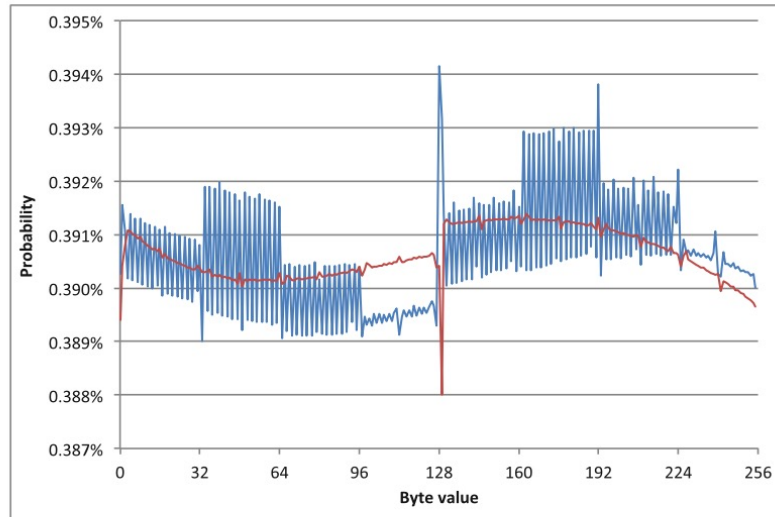
- Decrypt **one packet** (AP → station)
- Ability to inject 7 packets (per QoS)



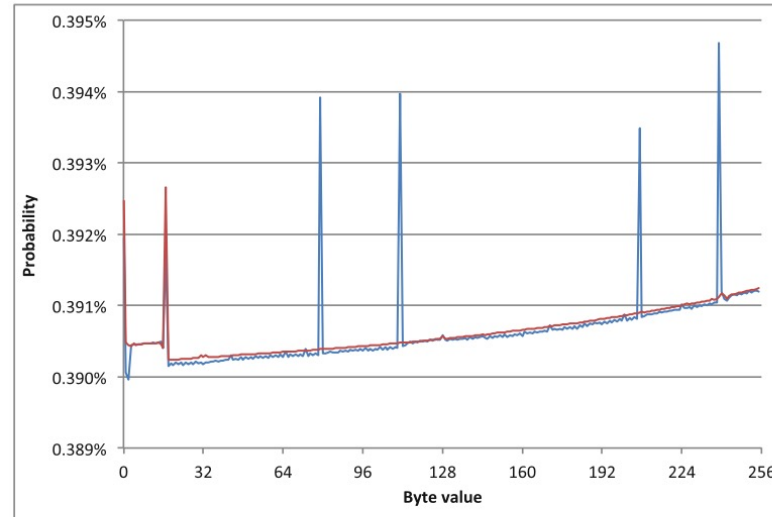
Beck and Tews. "Practical attacks against WEP and WPA." WiSec'09

WPA/TKIP key stream analysis

- **Observation:** statistical tests reveal **biases** in WPA/TKIP key stream



(a) Biases at position Z_1



(b) Biases at position Z_{17}

Prior **RC4** analysis in TLS context afound biases

Biases found for WPA/TKIP:
More **complex behaviour**

→ Attacking **easier/harder** depending on position

WPA/TKIP attack using biases

- **Setup**

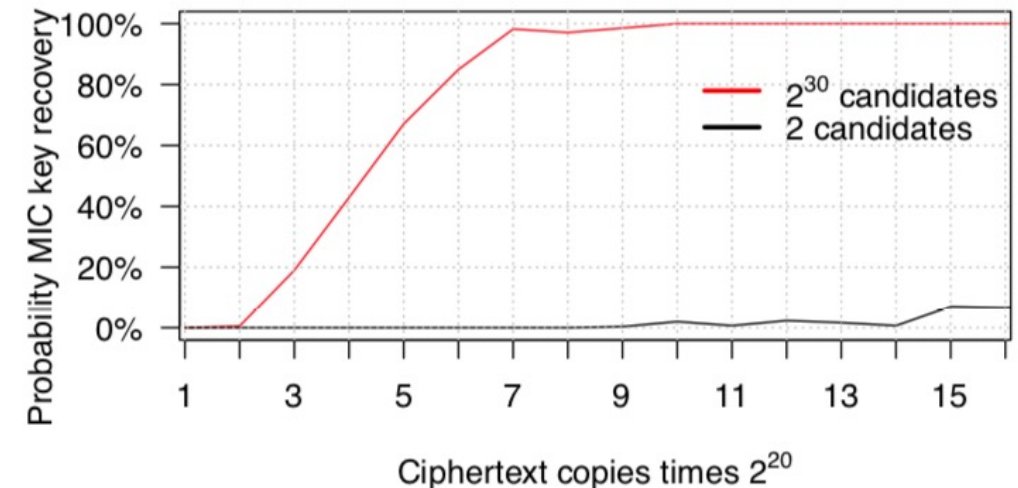
- 10 CPU years of statistics generation

- **Attack example**

- Capture 10M packets (e.g., 1h traffic)
- **Leverage biases to recover plaintext**
- Derive MIC key

- **End result**

- Decrypted packet
- Ability to inject



Vanhoef and Piessens. "All Your Biases Belong To Us: Breaking RC4 in WPA-TKIP and TLS." In *USENIX Security*, 2015.

WPA/TKIP summary

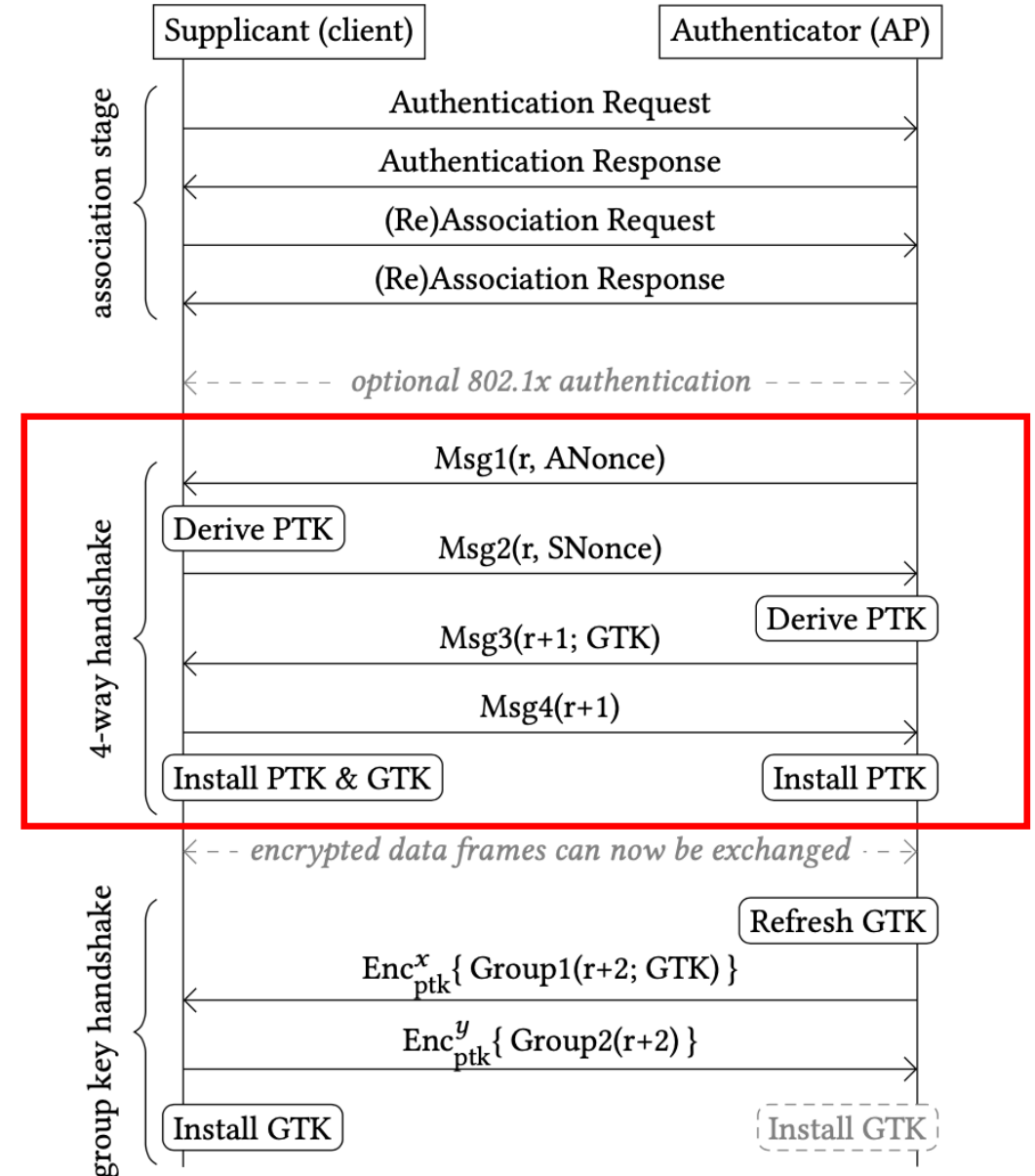
- Difficult starting point
 - Can't change already deployed chips: **must use WEP block**
- How well did the WPA/TKIP “fix” work?
 - Raise the bar for attacks significantly
 - First attacks (Beck 2009)
 - More attacks (Paterson 2014, Vanhoef 2015)
- Main issue: continued use!

WPA2

- Introduced in **2004**
- Better communication protection
 - New MAC-level crypto mechanisms called **CCMP**
 - **Confidentiality** using **AES-128** in counter mode
 - **Integrity** using **CBC-MAC**
 - “Authenticate-then-encrypt”
- Better access control
 - **Authentication** based on **new 4-way handshake**

WPA2 Handshake

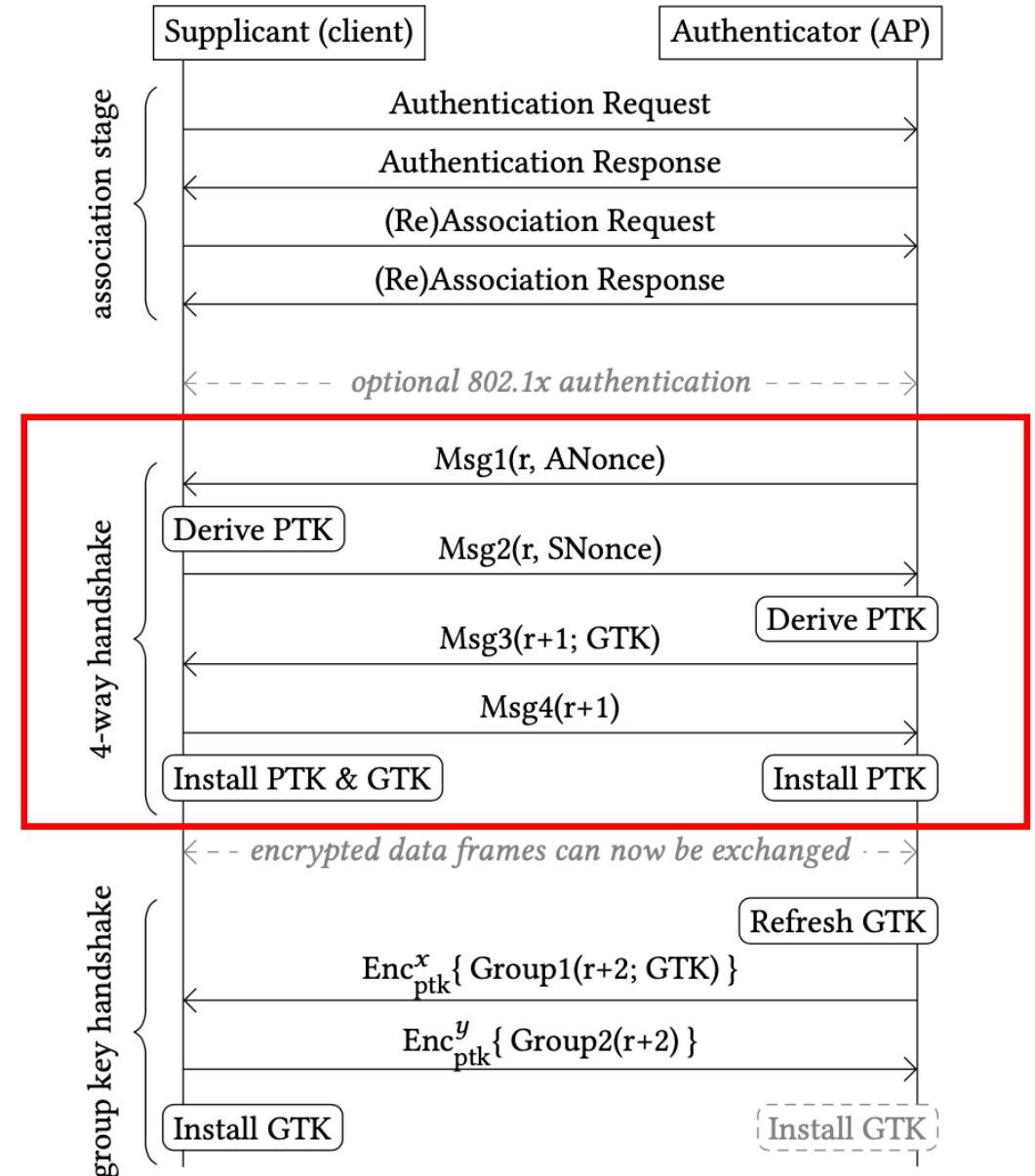
- Starting point
 - Pre-shared key (PMK)
- Goals
 - **Mutual authentication**
 - Session key agreement (PTK)
- Main steps
 - Msg1/2: Exchange nonces
 - Derive PTK
 - Use PMK, nonces, MAC addresses
 - Msg3/4: Authenticate exchange
 - **Install PTK**
 - Note: Msg2/3/4 integrity protected



Security proof

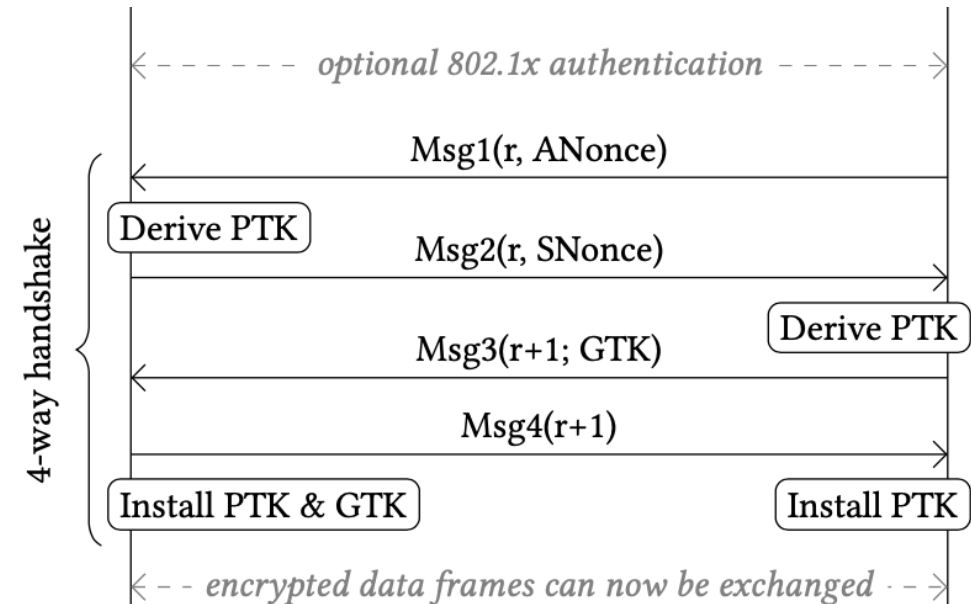
- Protocol “proven secure” in 2005
- Proved properties
 1. Existence of PMK on peer
 2. Established key (PTK) is fresh
 3. Synchronize PTK into MAC
 4. ...

He et al. “A Modular Correctness Proof of IEEE 802.11i and TLS.” In CCS’05.



Key Reinstallation Attack (KRACK) – 2017

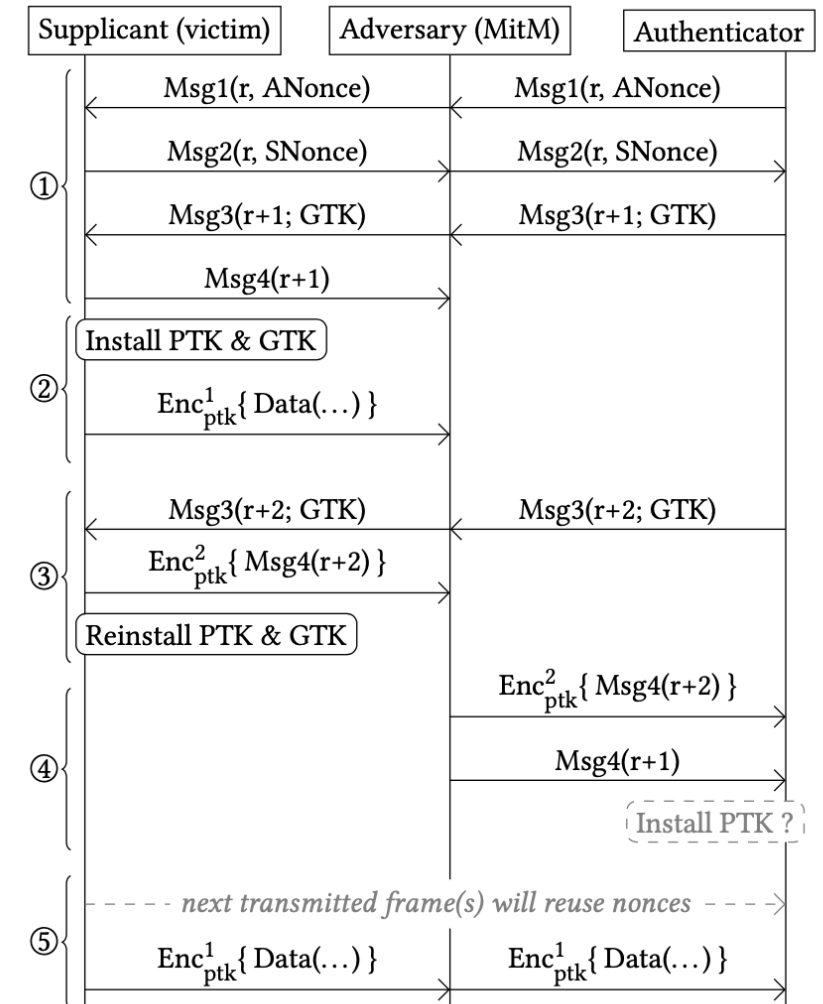
- **Goal:** keystream reuse
- **Observations**
 - AP retransmits Msg3 if no response
 - Each time client **reinstalls** same PTK
 - Reset counter for CCMP protocol
- **Approach:** replay Msg3
- Interesting point
 - Proven properties still hold
 - Models do not capture **when** key installed



Vanhoef and Piessens. “Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2 .” In CCS 2017.

Some KRACK attack details

- **Step 1:** Establish **man-in-the-middle** position
 - Recall that selective jamming difficult
 - Use forwarding based on cloned AP
 - Cannot use different MAC address...
 - but **different channel OK!**
- **Step 2:** Trigger retransmission of Msg3
 - **Blocking** Msg4
- **Impact**
 - Decryption and replay of packets
 - Many devices affected (e.g., Windows and iOS not)
 - Patch soon after



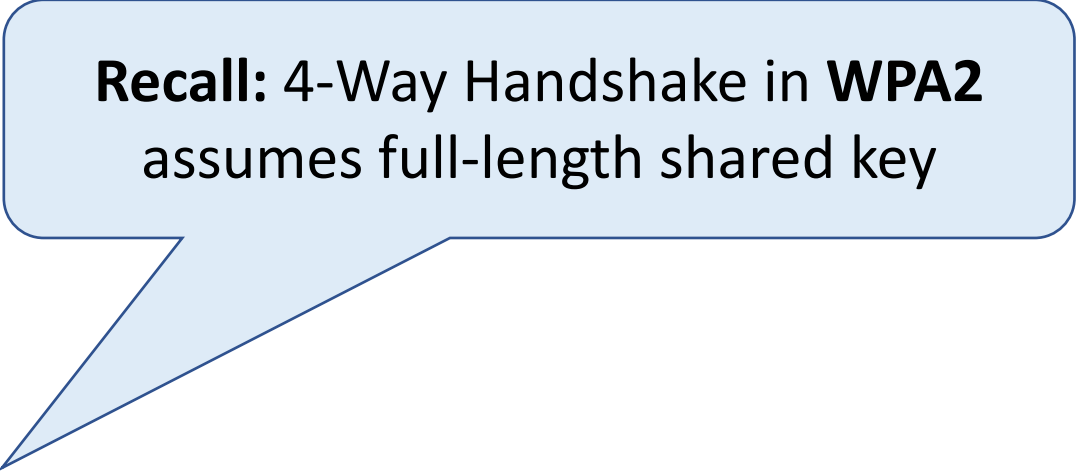
Improved KRACK attacks

- Follow-up work in **2018**
 - Improvements to original KRACK attack
- New findings
 - More practical attack variants
 - Patching is harder than believed
 - Break official 802.11 countermeasure
- Current status
 - <https://www.krackattacks.com/>

Vanhoef and Piessens. “Release the Kraken: New KRACKs in the 802.11 Standard.” In CCS 2018.

WPA3

- The latest WiFi security standard (**2018**)
- Updated cryptography
 - **Confidentiality** using **AES-128/256**
 - **Integrity** using **SHA-384 HMAC**
- New handshake
 - **Password-based** authentication and key agreement
 - Variant of Dragonfly Key Exchange ([RFC 7664](#))



Recall: 4-Way Handshake in **WPA2**
assumes full-length shared key

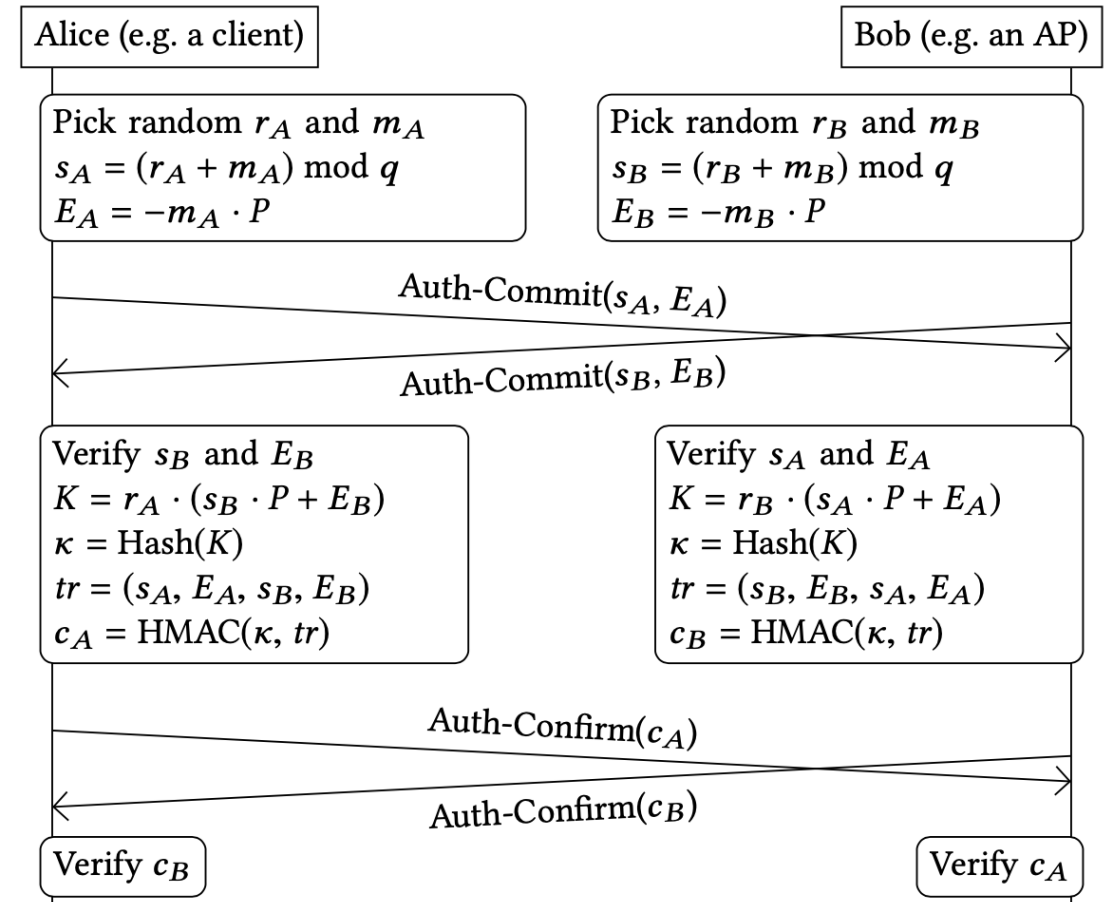
WPA3 handshake

- **Goal**

- Start with **low-entropy password**
- Turn it into **high-entropy key**
- Forward secrecy
- Dictionary attack protection

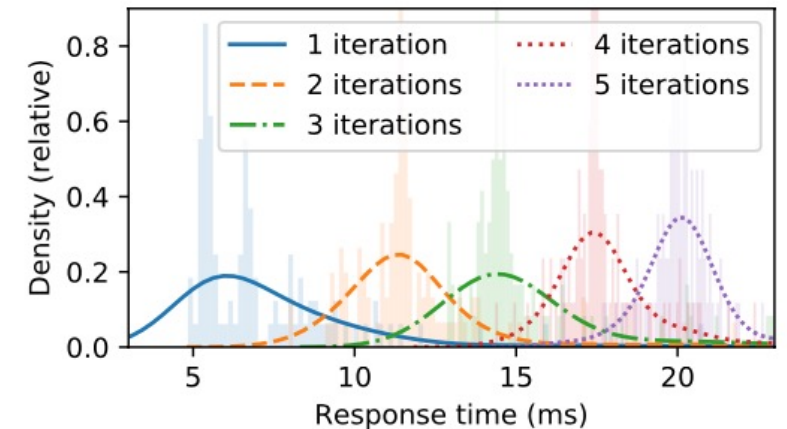
- **Main steps**

1. Commit to *pwd* and nonces
2. Verify commitments
3. Derive shared key *k*
4. Confirm phase



WPA3 handshake analysis (2020)

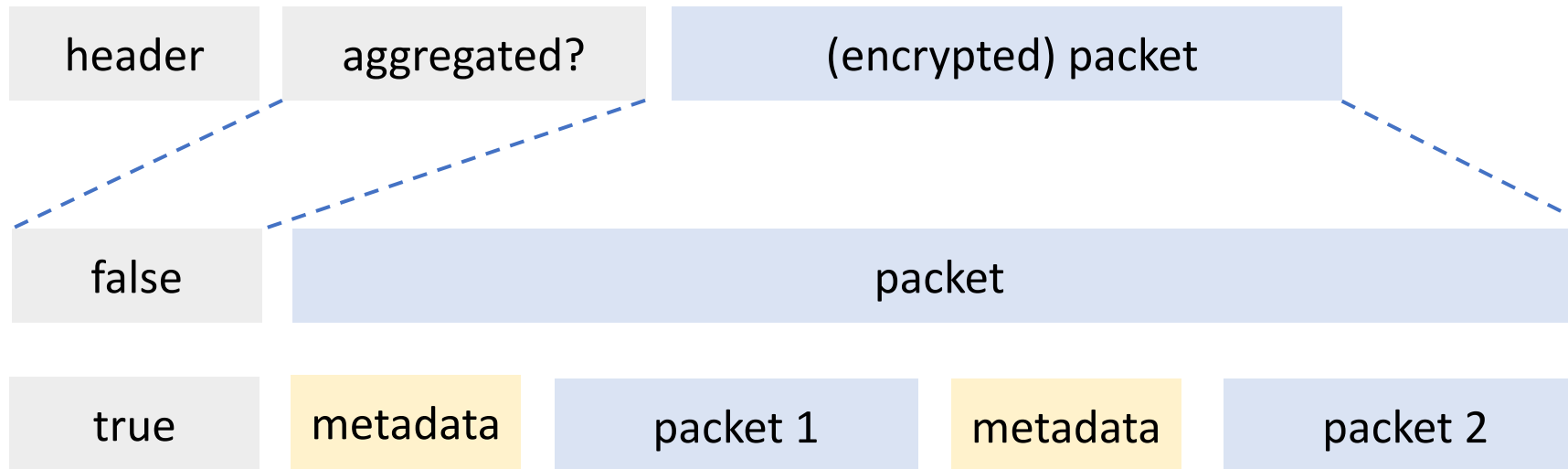
- Downgrade attacks
 - Transition mode: WPA3 and WPA2 handshake
 - Downgrade to WPA2 → detected → record handshake → dictionary attack
- Timing-based side-channel
 - Recall: pwd is converted to group element
 - Execution time (iterations) may depends on pwd



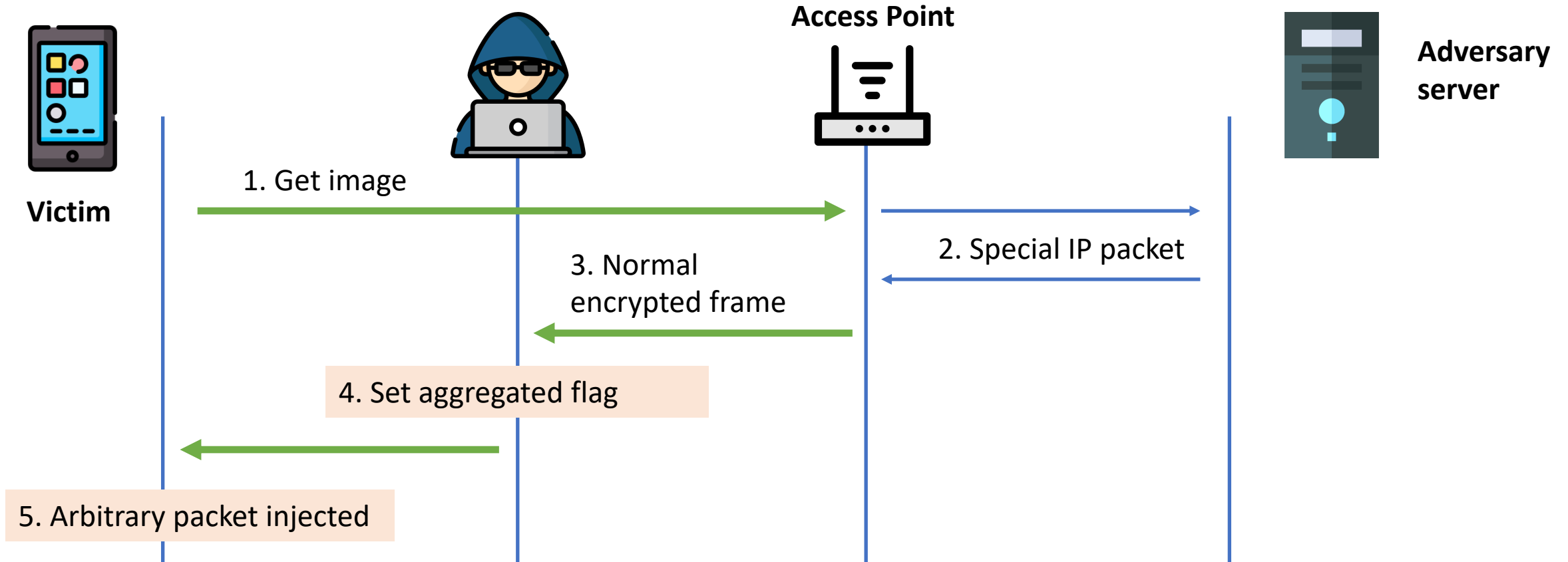
Vanhoef and Ronen. “Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd.” In S&P 2020. <https://wpa3.mathyvanhoef.com/>

Aggregation attack (2021)

Observation:
Aggregated flag not
protected!



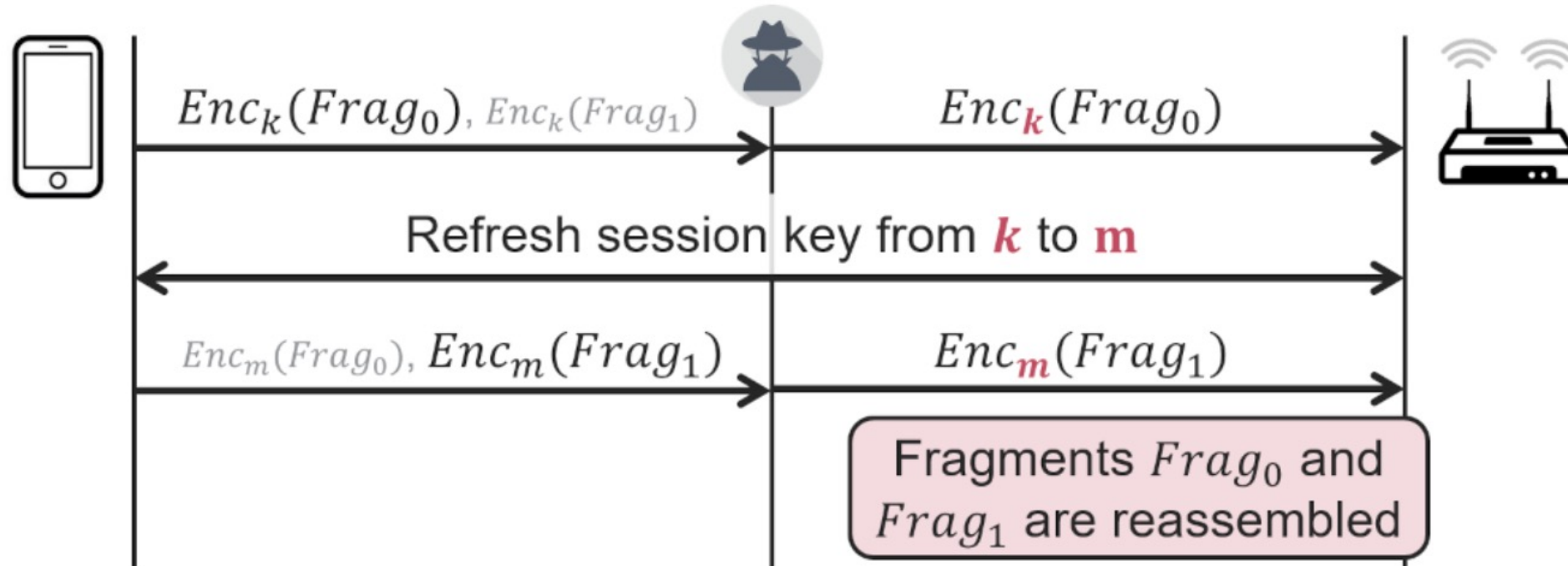
Aggregation attack



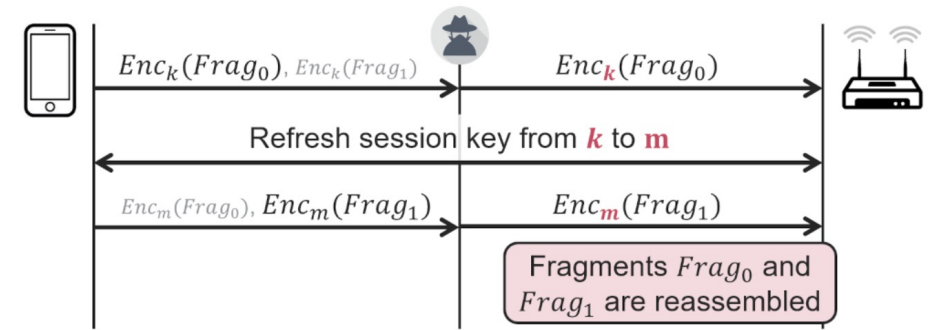
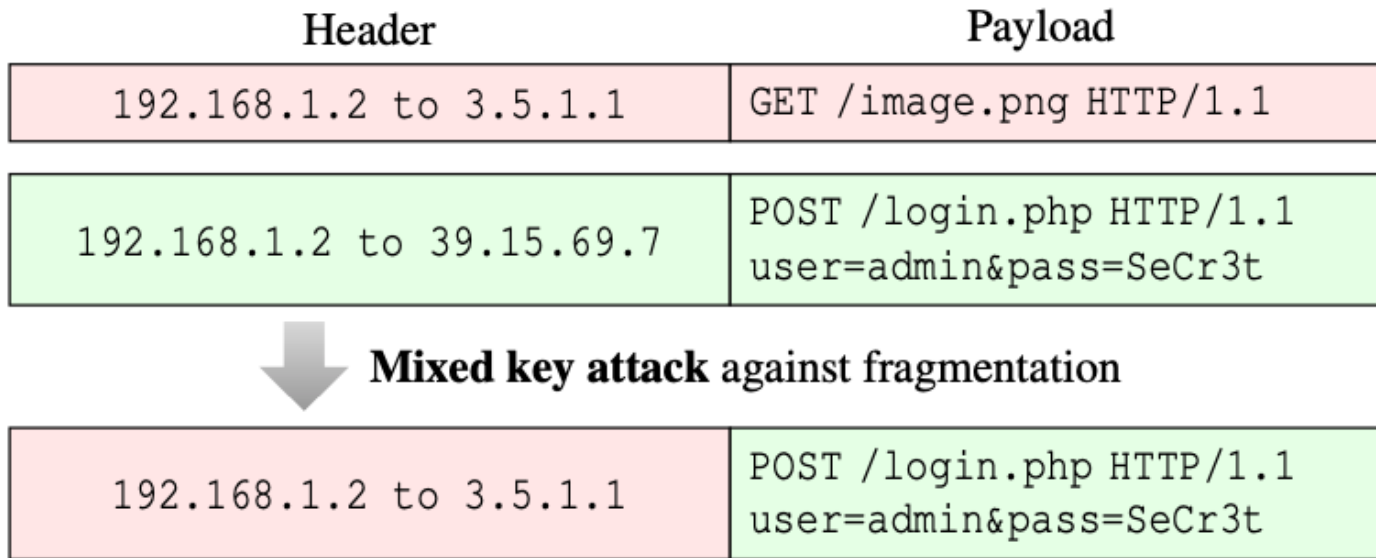
Example: ICMP with malicious DNS server

Vanhoef. "Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation." In USENIX Security 2021. <https://www.fragattacks.com/>

Observation: fragmentation reassembly



Fragmentation attack (2021)



Vanhoef "Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation" In USENIX Security 2021. <https://www.fragattacks.com/>

Aggregation and fragmentation discussion

- Practical attacks require certain assumptions to hold
 - Man-in-the-middle position
 - Social engineering of victim
 - And so on...
- The paper also reports a number of implementation flaws that reduce some of these assumptions
- Attack strategies apply from WEP to WPA3
 - Undiscovered more than two decades!

Lecture summary

- Evolution of WiFi security
 - From **90s** to **today**
- Lessons learnt
 - Constraints, open design, threat model (WEP)
 - Legacy makes things harder (WPA/TKIP)
 - Strong crypto and proven protocols → systems can still break in subtle and unexpected ways (WPA2 and WPA3)

Thank you!