భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Wi-Fi Security: Threats & Solutions

Bheemarjuna Reddy Tamma

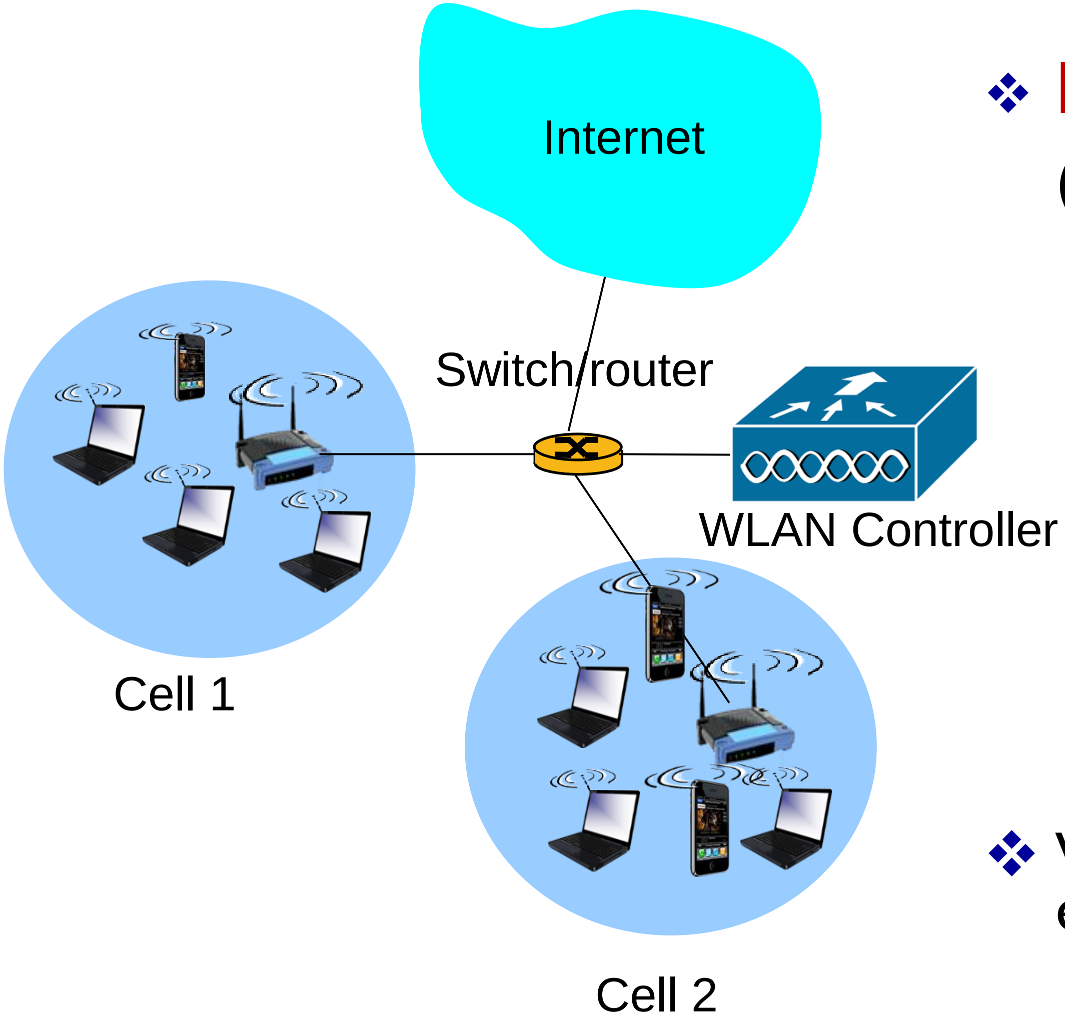
IIT Hyderabad

Credits: *Some slides and pics in this presentation are adapted from William Stallings textbook on Wireless Security, Kurose and Ross textbook on Computer Networking, slides of Mathy Vanhoef, a host of others and Internet sources*

Outline

- □ Wi-Fi Architecture
- □ Why Wi-Fi Security is important?
- □ Wi-Fi Security Threats
- □ Wi-Fi Security Standards
- □ Vulnerabilities in Wi-Fi Security Stds
- □ What WPA3 offers?
- □ Wi-Fi Security: Best Practices to mitigate

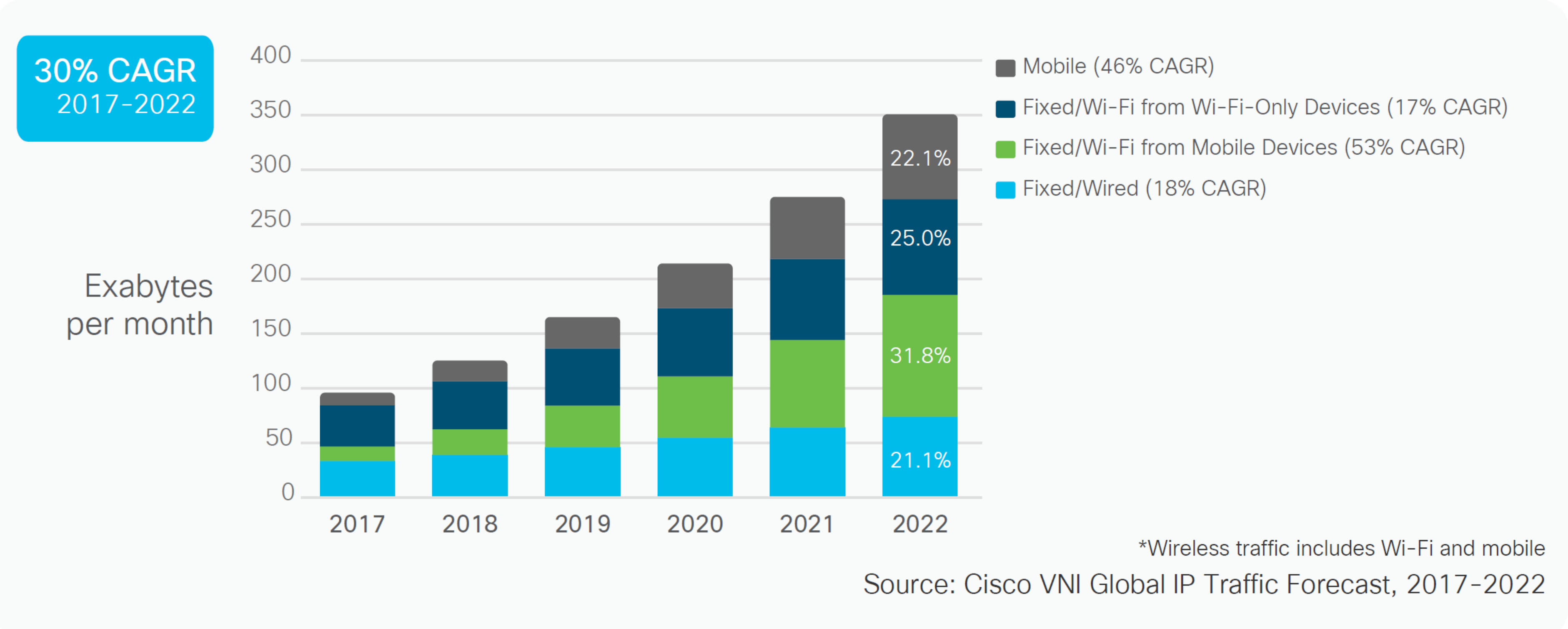
802.11 WLAN (Wi-Fi) Architecture



- Basic Service Set (BSS) (aka "cell")
  - Building block of IEEE 802.11 WLAN
  - In infrastructure mode, a cell contains:
    - Wireless clients/stations
    - Access Point (AP)
- WLAN controller in enterprise deployments

Why Wi-Fi Security is IMP?

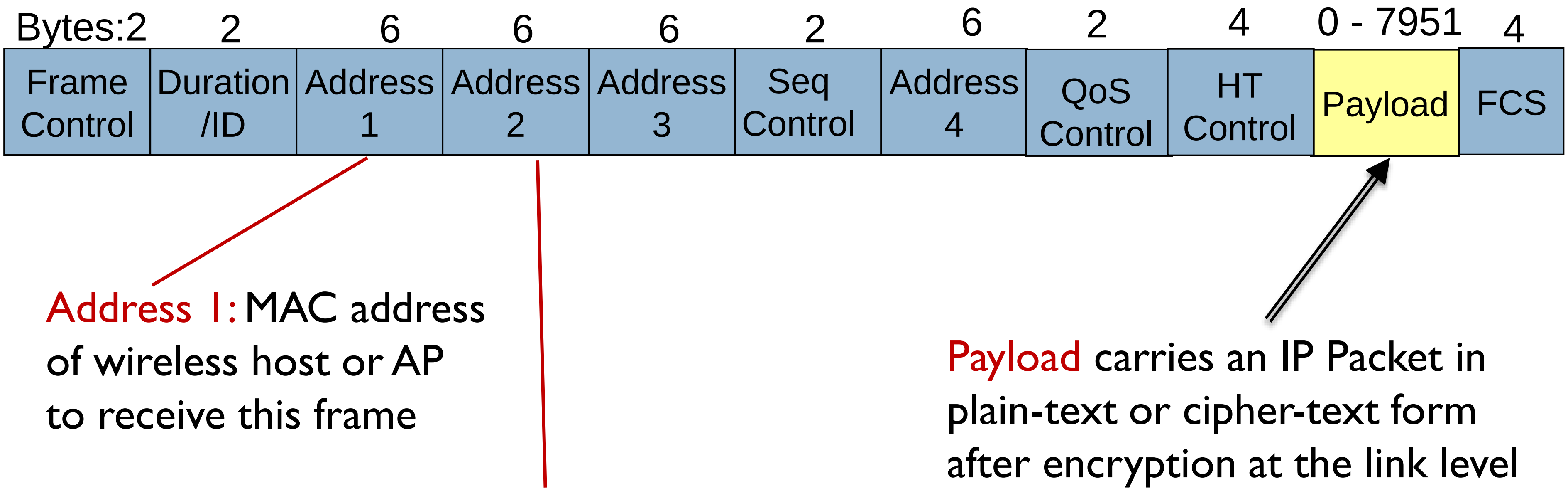
- More than half of world's data is carried by Wi-Fi!



*Wireless traffic includes Wi-Fi and mobile

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022

802.11 (Wi-Fi) Packet Format



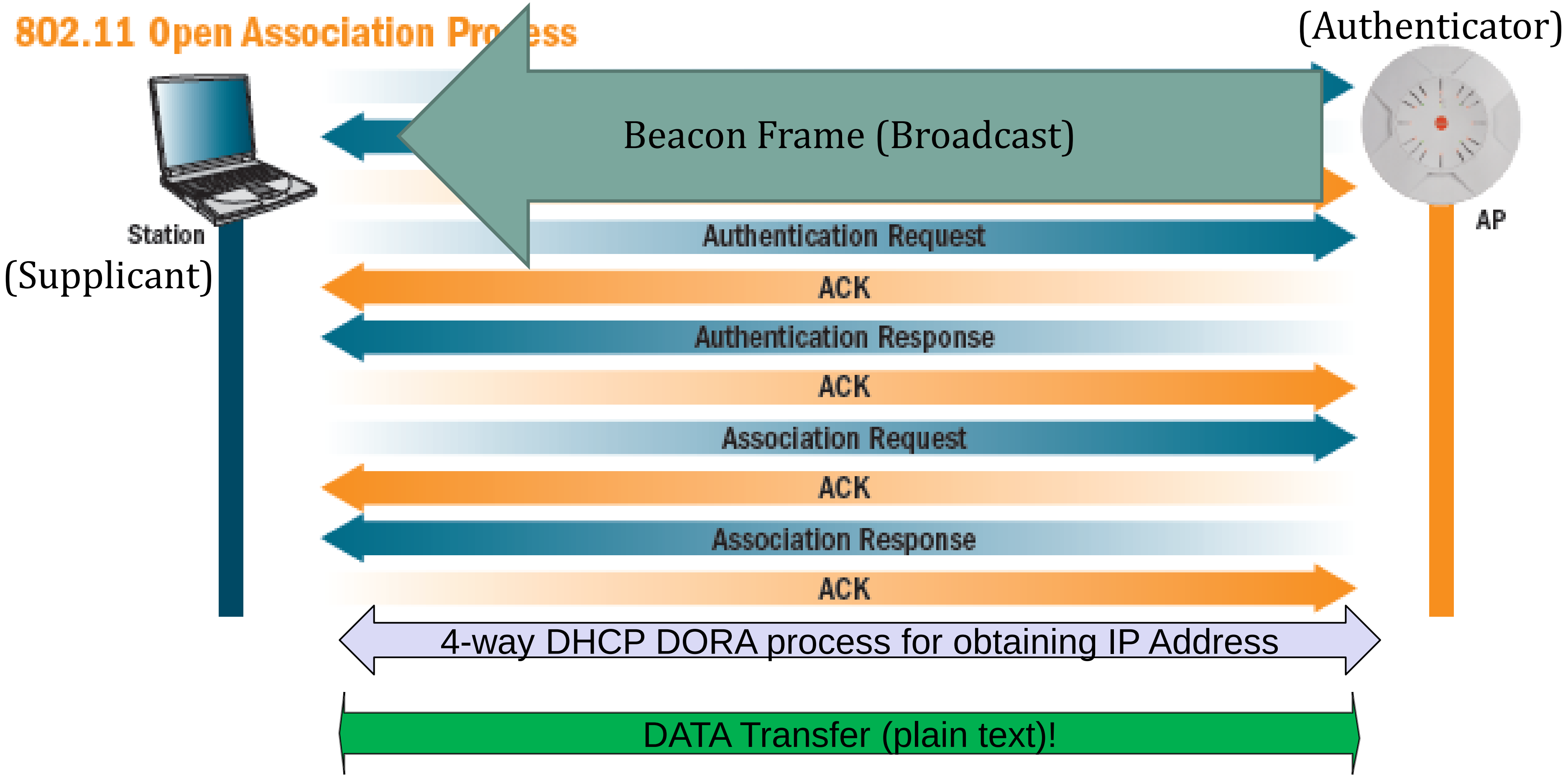
Bytes:	2	2	6	6	6	2	6	2	4	0 - 7951	4
	Frame Control	Duration /ID	Address 1	Address 2	Address 3	Seq Control	Address 4	QoS Control	HT Control	Payload	FCS

Address 1: MAC address of wireless host or AP to receive this frame

Address 2: MAC address of wireless host or AP transmitting this frame

Payload carries an IP Packet in plain-text or cipher-text form after encryption at the link level

How does a STA join Wi-Fi network ?



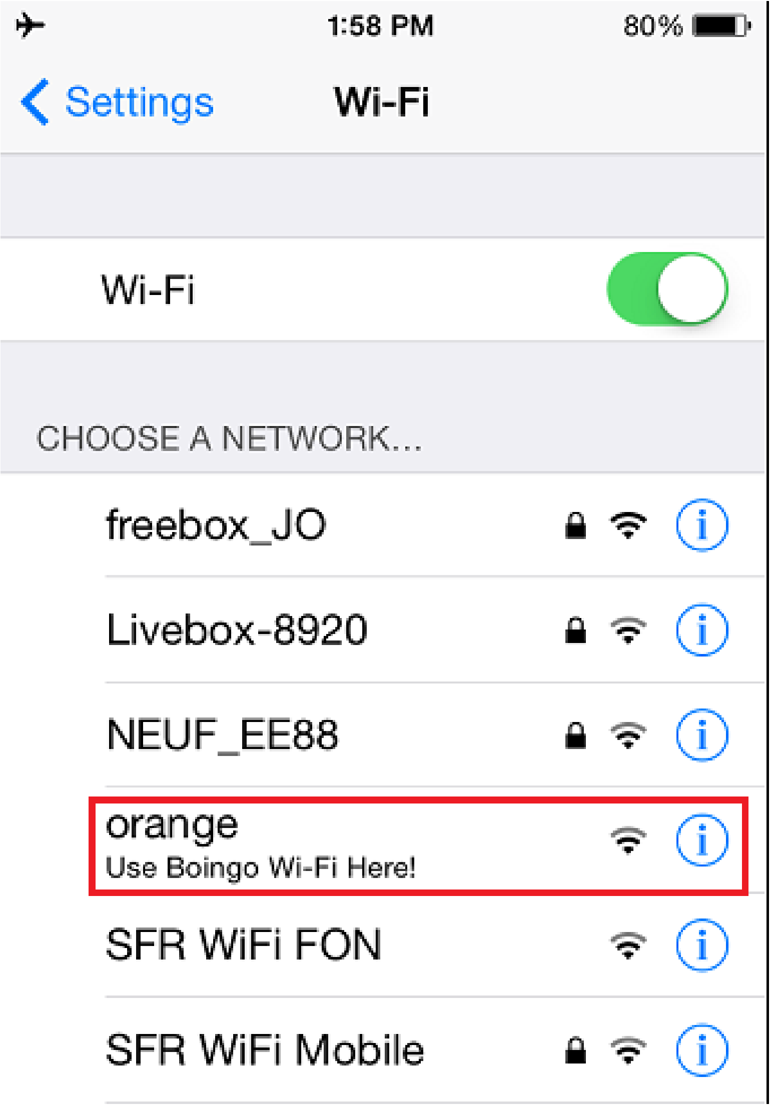
Wi-Fi Security Threats

1) Eavesdropping
2) Denial of Service (DoS) attacks
3) Man-in-the-middle (MITM) attacks
4) Malicious association to rogue (AP) networks
5) AP configuration over HTTP

Hacking Wi-Fi Networks

- Tools of the trade
  - Wireshark/Tcpdump
  - AirCrack-NG
  - Kismet
  - WEPCrack/AirSnort
  - CoWPAtty
  - NetStumbler
  - WiFuzz
  - Pyrit, Fern
  - Cain & Able
  - AirXploit
  - so on…

Free/Paid, Public Wi-Fi is Open!



1:58 PM 80%

< Settings Wi-Fi

Wi-Fi

CHOOSE A NETWORK...

freebox_JO

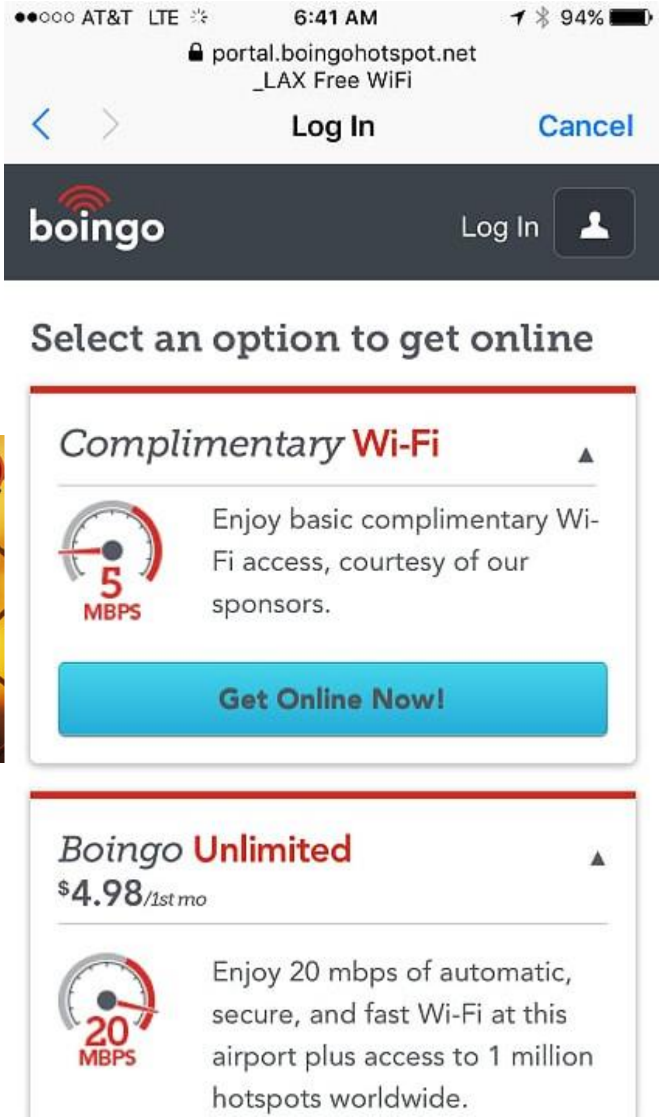
Livebox-8920

NEUF_EE88

orange
Use Boingo Wi-Fi Here!

SFR WiFi FON

SFR WiFi Mobile



●●○○○ AT&T LTE 6:41 AM 94%

portal.boingohotspot.net
_LAX Free WiFi

Log In Cancel

boingo Log In

Select an option to get online

Complimentary Wi-Fi

5 MBPS

Enjoy basic complimentary Wi-Fi access, courtesy of our sponsors.

Get Online Now!

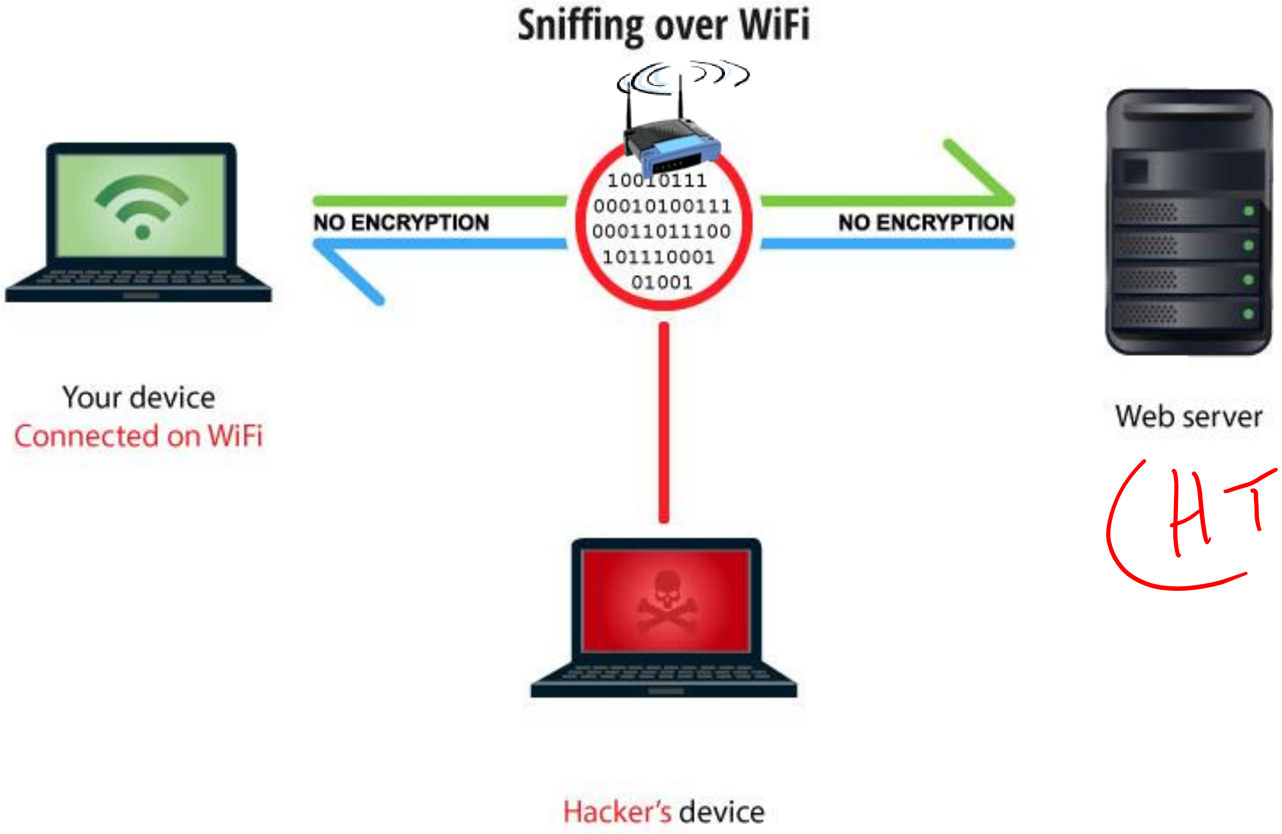
Boingo Unlimited

$4.98/1st mo

20 MBPS

Enjoy 20 mbps of automatic, secure, and fast Wi-Fi at this airport plus access to 1 million hotspots worldwide.

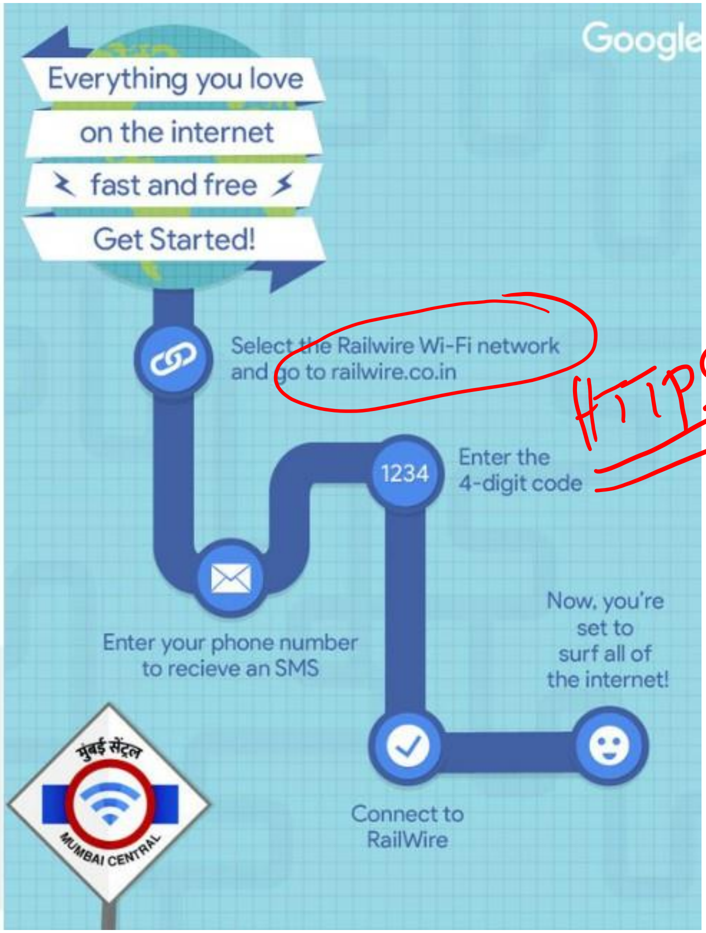
Threat-1: Eavesdropping on Open Wi-Fi Networks



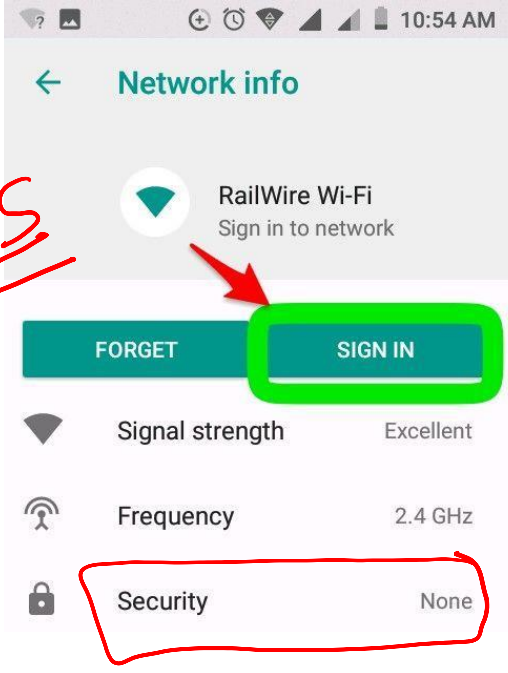
(HTTP)

- □ Here AP is not malicious, just open (no encryption of link b/w AP and STA)
- □ Easy to intercept traffic, but almost impossible to detect ☹
- □ Many tools available: Wireshark/Tcpdump/airdump-ng/...
- □ Affects Confidentiality of data exchanged

Free Wi-Fi led to spike in Cyber attacks!


Google
Everything you love
on the internet
fast and free
Get Started!
Select the Railwire Wi-Fi network and go to railwire.co.in
1234
Enter the 4-digit code
Enter your phone number to recieve an SMS
Now, you're set to surf all of the internet!
Connect to RailWire
मुंबई सेंट्रल
MUMBAI CENTRAL

https



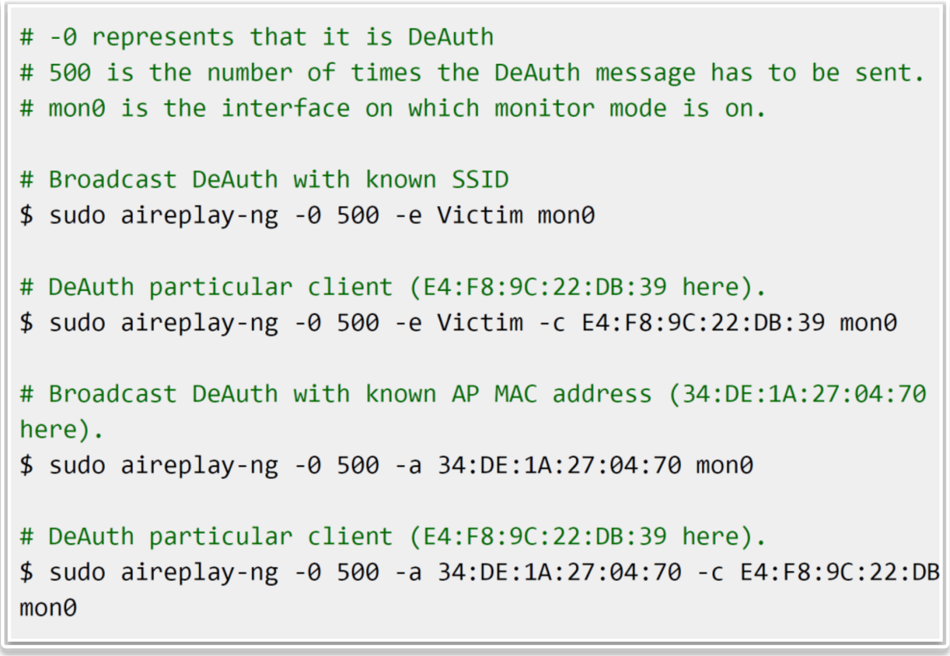
RailWire Wi-Fi
Sign in to network

Signal strength	Excellent
Frequency	2.4 GHz
Security	None

Threat-2: Denial of Service (DoS) attacks

- Frequency jamming
  - Not very technical, but works very well
- Spoofed Deauthentication / Disassociation messages
  - Wi-Fi Control/Mgmt frames are not protected in 802.11i std
  - Can target one specific user or all connectd to AP or Wi-Fi network
- Evil Twin: Rogue APs on legitimate WLAN system
  - Only client-side authentication
- Black hole evil twin
- Battery exhaustion

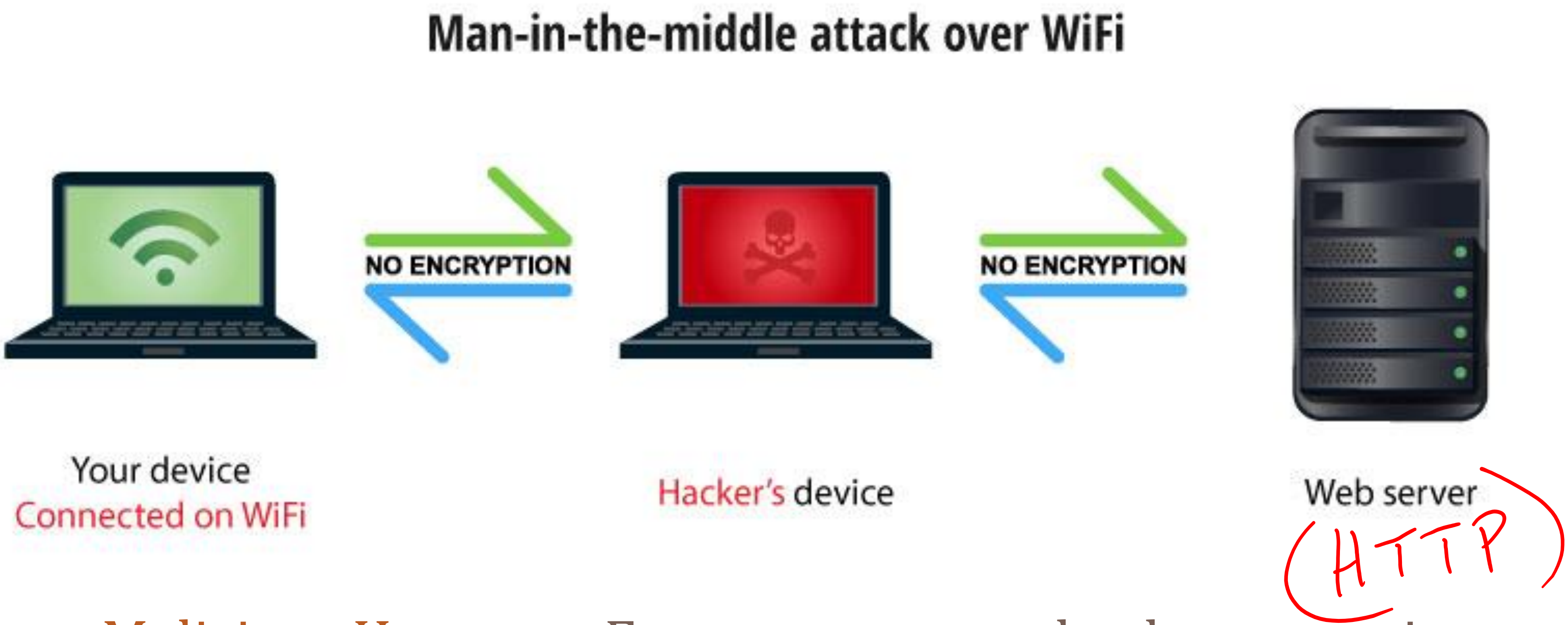
https://aircrack-ng.org/



```
# -0 represents that it is DeAuth
# 500 is the number of times the DeAuth message has to be sent.
# mon0 is the interface on which monitor mode is on.

# Broadcast DeAuth with known SSID
$ sudo aireplay-ng -0 500 -e Victim mon0

# DeAuth particular client (E4:F8:9C:22:DB:39 here).
$ sudo aireplay-ng -0 500 -e Victim -c E4:F8:9C:22:DB:39 mon0

# Broadcast DeAuth with known AP MAC address (34:DE:1A:27:04:70
here).
$ sudo aireplay-ng -0 500 -a 34:DE:1A:27:04:70 mon0

# DeAuth particular client (E4:F8:9C:22:DB:39 here).
$ sudo aireplay-ng -0 500 -a 34:DE:1A:27:04:70 -c E4:F8:9C:22:DB
mon0
```

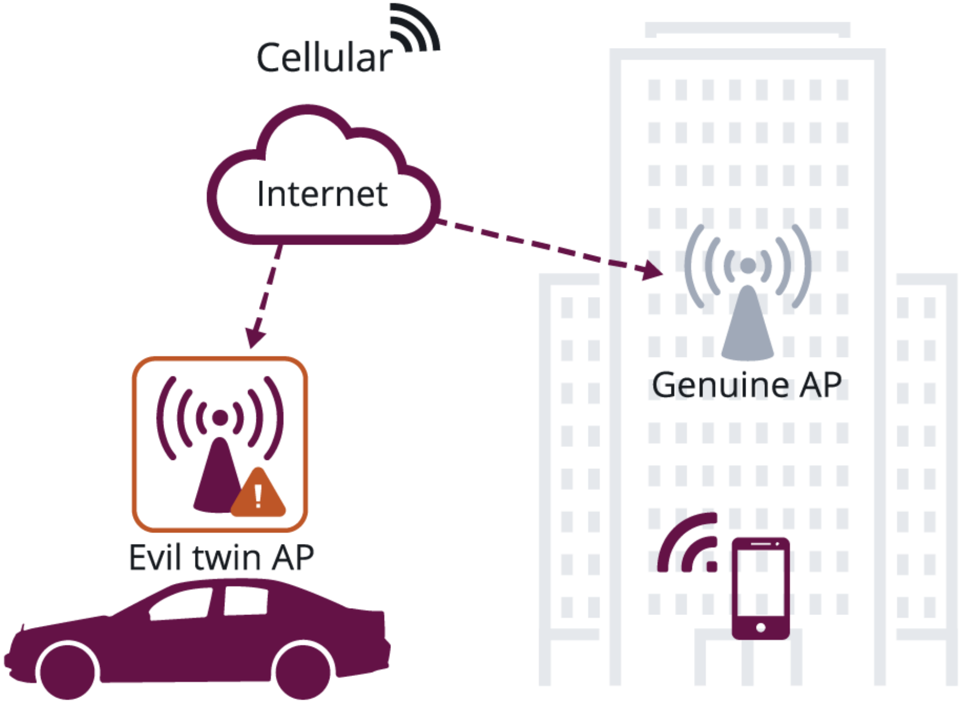
aireplay-ng [Aircrack-ng]

Threat-3a: MITM attacks in Open Wi-Fi



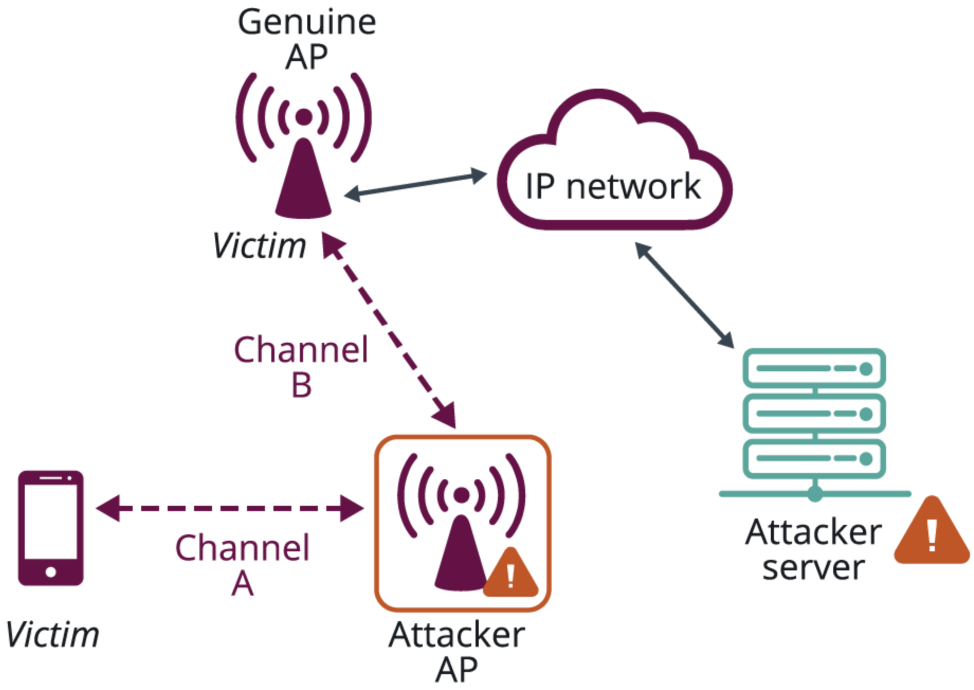
- □ Malicious Hotspots: Free, open networks that snoop into data sent/received
- □ Affects confidentiality and integrity of data exchanged

Threat-3b: MITM using Evil Twin Hotspot



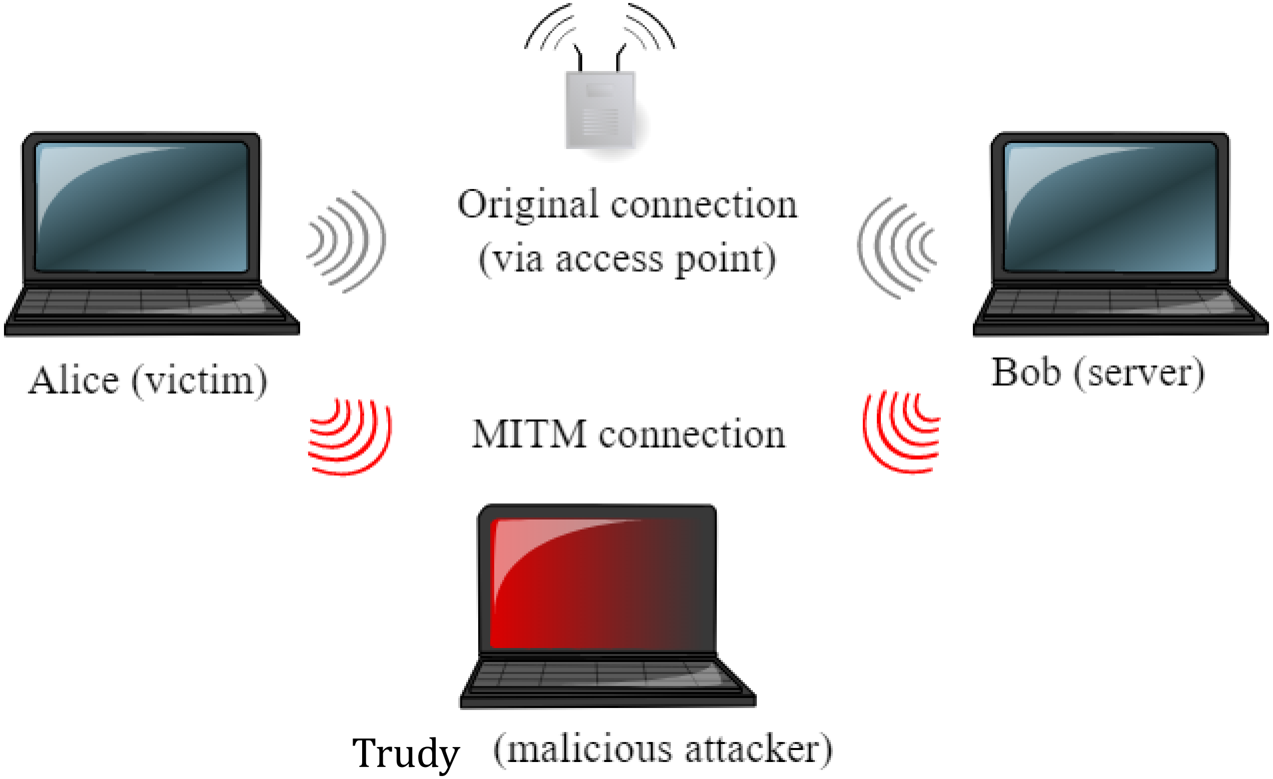
- Rogue APs on legitimate and protected Wi-Fi networks
  - Attacker masquerades as a legitimate (secure) AP to inspect or modify data, or attempt social engineering attacks to obtain personal information

Threat-3c: Multi-Channel MITM Attack



- Attacker /w two Wi-Fi radios (by MAC ID spoofing and using DeAuth/CSA messages) tries to exploit a protocol or implementation weakness by relaying, suppressing, modifying, or injecting messages
  - 2014-acsac-body-raw.pdf (acm.org)
  - Operating Channel Validation: Preventing Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks (mathyvanhoef.com)

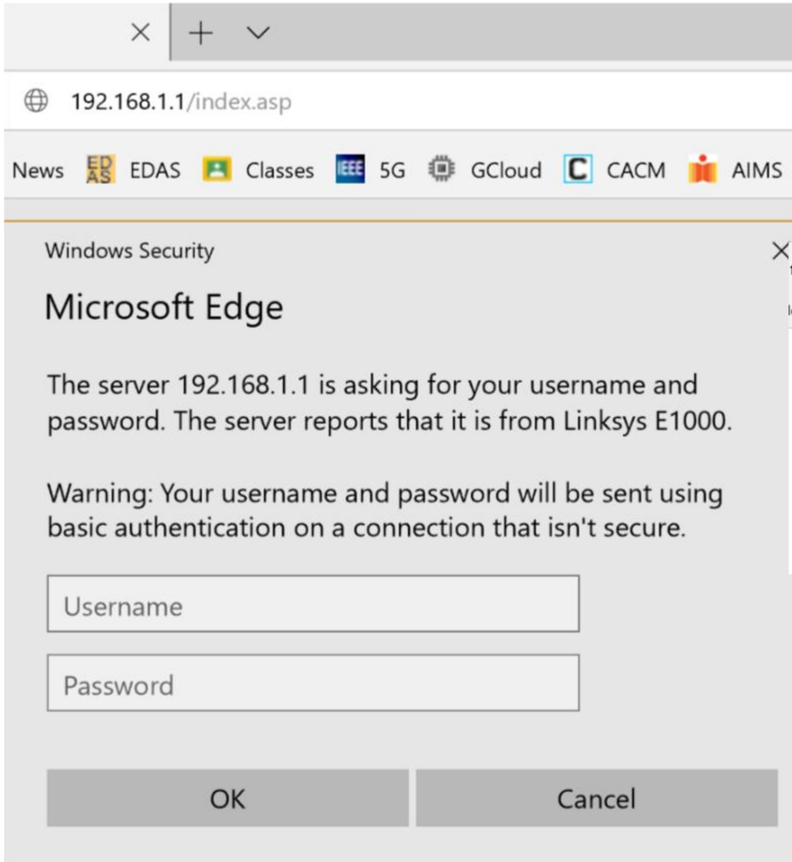
Threat-3d: MITM using ARP Poisoning



- □ Address Resolution Protocol (ARP) requests are used to get MAC address associated with IP address of a device
- □ Trudy send gratuitous ARP messages to Alice giving her MAC address as that of Bob and vice versa☹
  - Run a Man-in-the-Middle attack on a WiFi hotspot (poly.edu) & arp-request_reinjection [Aircrack-ng]

Demo of MITM Attack

Threat-4: Open AP configuration over HTTP



Windows Security

Microsoft Edge

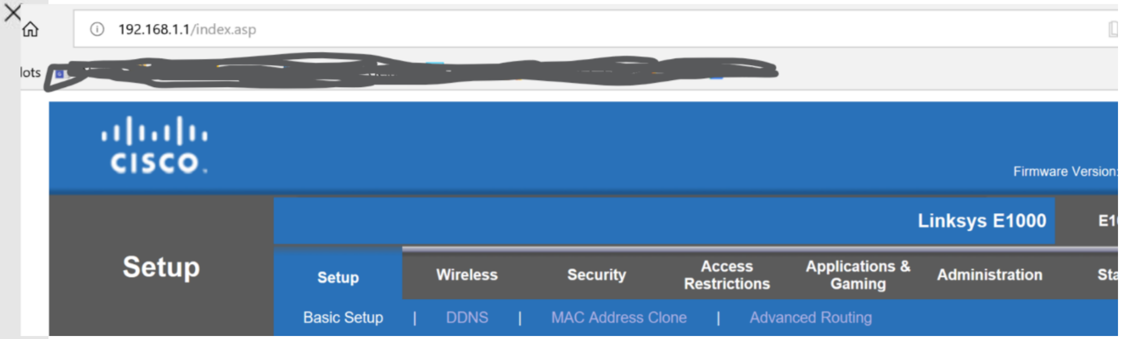
The server 192.168.1.1 is asking for your username and password. The server reports that it is from Linksys E1000.

Warning: Your username and password will be sent using basic authentication on a connection that isn't secure.

Username

Password

OK | Cancel



[redacted]

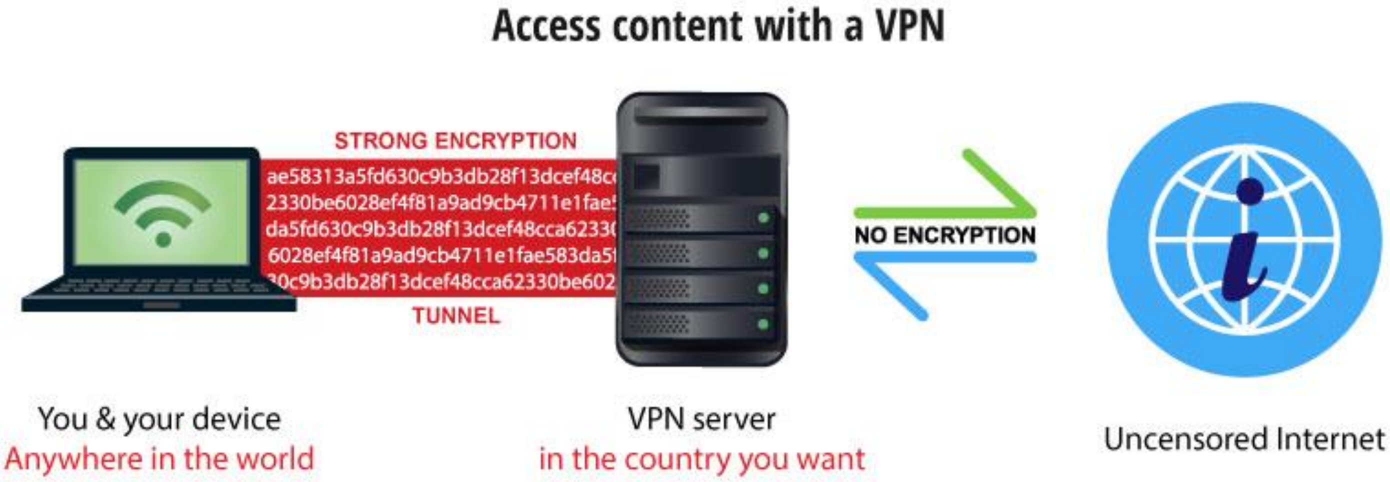
How to stay safe on public Wi-Fi?

√ DO:

- Try VPN (Virtual Private Network) to make your public Wi-Fi connection private
- Only visit sites using

🔒 https://

- Turn OFF file sharing


Access content with a VPN
STRONG ENCRYPTION
ae58313a5fd630c9b3db28f13dcef48c
2330be6028ef4f81a9ad9cb4711e1fae
da5fd630c9b3db28f13dcef48cca62330
6028ef4f81a9ad9cb4711e1fae583da5
0c9b3db28f13dcef48cca62330be602
TUNNEL
NO ENCRYPTION
You & your device
Anywhere in the world
VPN server
in the country you want
Uncensored Internet

How to stay safe on public Wi-Fi?

- Don’t:
  - Allow your Wi-Fi to auto-connect to open networks
  - Log into an App that contains sensitive info. Go to the website instead to verify it uses HTTPS before logging in
  - Leave your Wi-Fi radio on if you are not using it
  - Click unexpected links, attachments, or pop-ups
  - Access websites that hold your sensitive information, such as bank or healthcare accounts and e-commerce sites

References

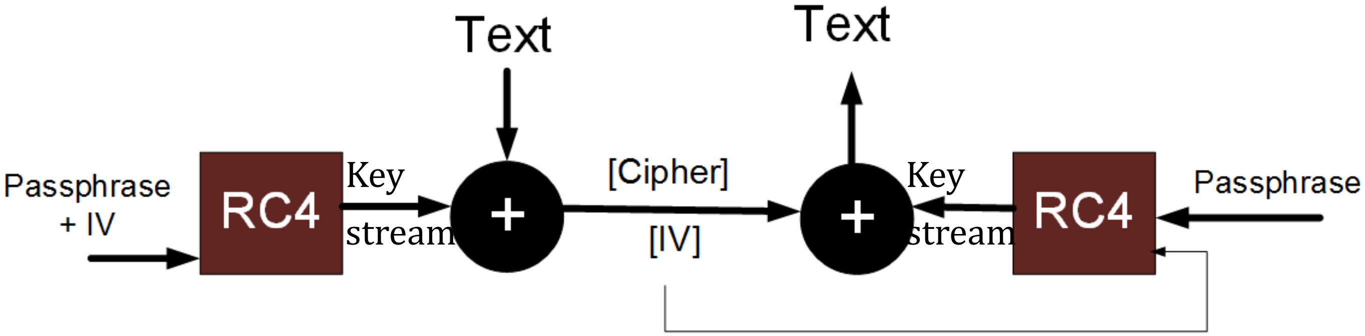
- IEEE 802.11 Std: https://doi.org/10.1109/IEEESTD.2022.9930960
- https://code.google.com/archive/p/wifuzz/wikis/WiFuzz.wiki
- http://www.secdev.org/projects/scapy/
- https://www.eetimes.com/document.asp?doc_id=1206324
- https://thebestvpn.uk/unsecured-wifi-network/
- https://witestlab.poly.edu/blog/conduct-a-simple-man-in-the-middle-attack-on-a-wifi-hotspot/
- https://wirelesslywired.com/2017/07/05/following-the-802-1x-aaa-process-with-packet-captures/
- https://whisperlab.org/introduction-to-hacking/lectures/wifi-exploitation

Wi-Fi Security Standards

- 1997→Wired Equivalent Privacy (WEP)
- 2003→Wireless Protected Access (WPA)
- 2004→WPA2 (IEEE 802.11i)
- 2019→WPA3 (Wi-Fi 6/6E devices support it)

Wired Equivalent Privacy (WEP)

- Original solution offered by IEEE 802.11 std
- Uses RC4 encryption algo (stream cipher) with pre-shared keys (40-bit or 104-bit) and 24-bit Initialization Vectors (IV)



- Flawed design, easily broken
  - There’s no key management
  - All users always share the same WEP key
    - Used for both authentication and encryption ☹
  - IV is too small, sent in clear text and its reuse caused problems
  - Tools to break WEP are widely available (e.g., AirCrack-ng)

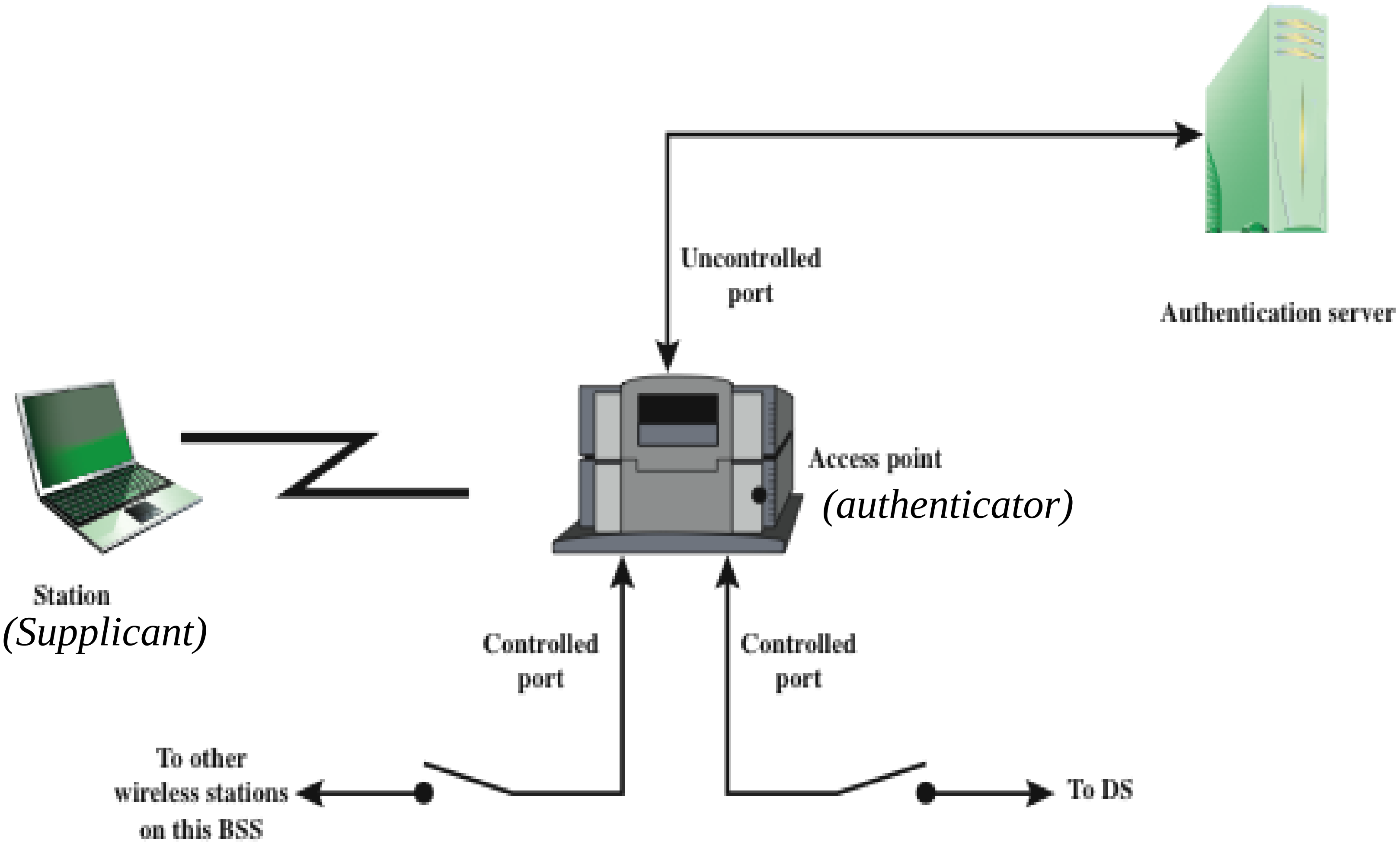
https://asecuritysite.com/encryption/rc4_wep

Using the Fluhrer, Mantin, and Shamir Attack to Break WEP – NDSS Symposium (ndss-symposium.org)

WPA2 and WAP3

- Wireless Protected Access 2 (WPA2)
  - WPA2 is Wi-Fi alliance name for 802.11i amendment
  - Two variants: WPA2-Enterprise and WPA2-Personal
  - WPA2-Enterprise uses 802.1X for access control
    - Uses Extensible Authentication Protocol (EAP) for authentication and key exchange, e.g., EAP-TLS, EAP-PEAP
  - Confidentiality and integrity protocol: AES-CCMP
- WPA3
  - WPA3-Personal, WPA3-Enterprise and Enhanced Open
  - Support for protected management frames and an optional enhanced crypto mode

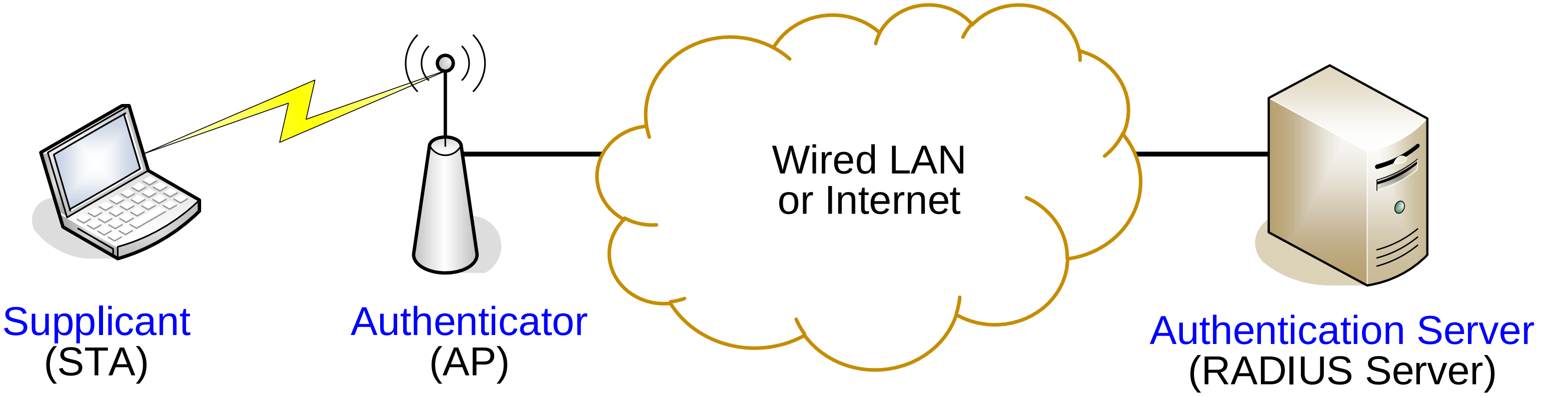
802.1X Access Control in WPA2-Enterprise



(authenticator)

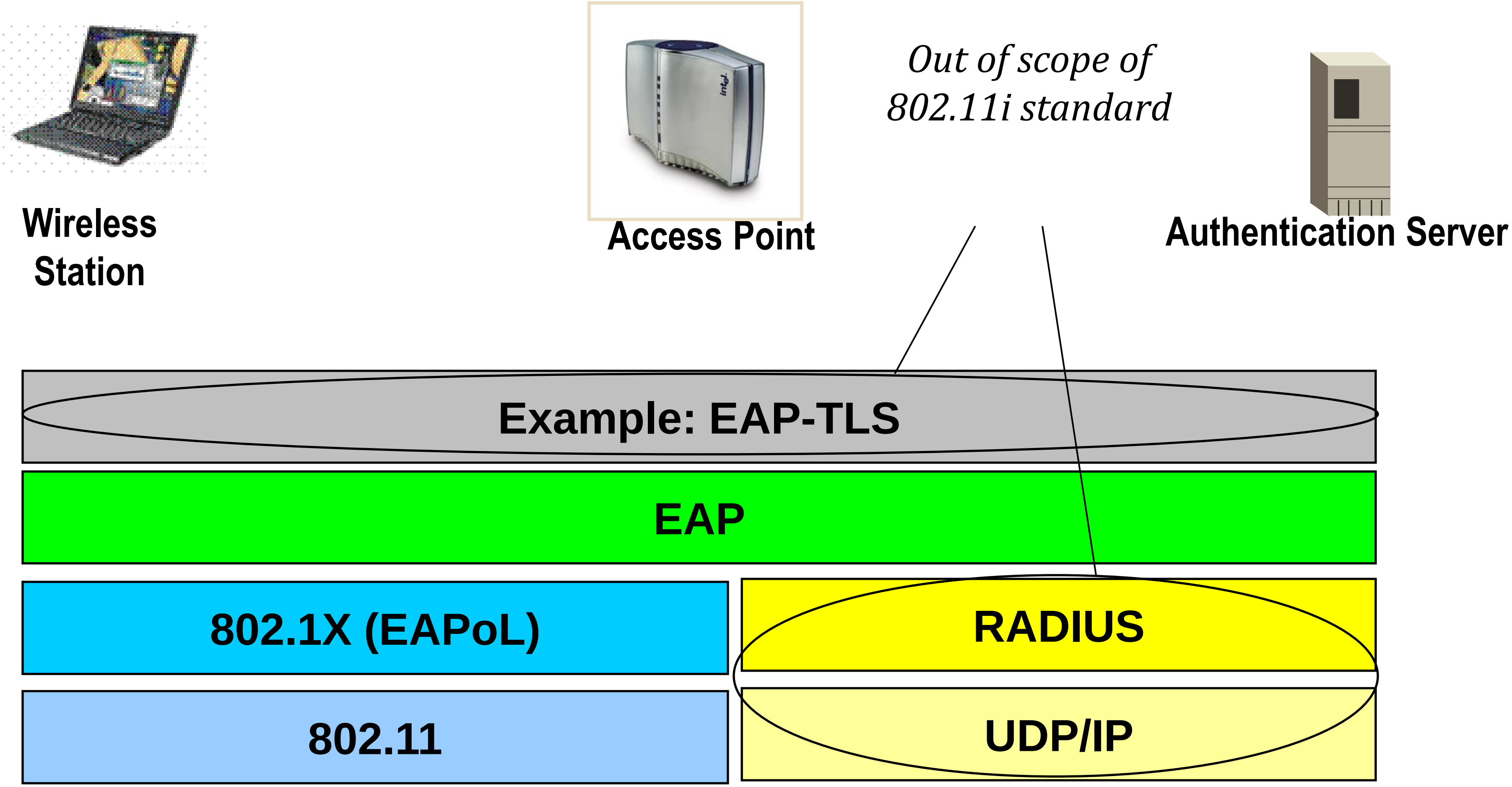
(Supplicant)

WPA2/802.1X architecture



- Supplicant wants to access the wired network via the AP, so it sends Authentication credentials to Authentication Server (AS) with 802.1X (EAP)
- AS authenticates the supplicant and ”tells” the AP whether access to controlled ports should be allowed or not
  - So, AP is simply a pass-through device during authentication process
- Authenticator (AP) then enables network access for the supplicant after successful authentication
- E.g., Enterprise Wi-Fi and Eduroam services

WPA2: Authentication and Key Management Architecture



Wireless Station

Access Point

Out of scope of 802.11i standard

Authentication Server

Example: EAP-TLS

EAP

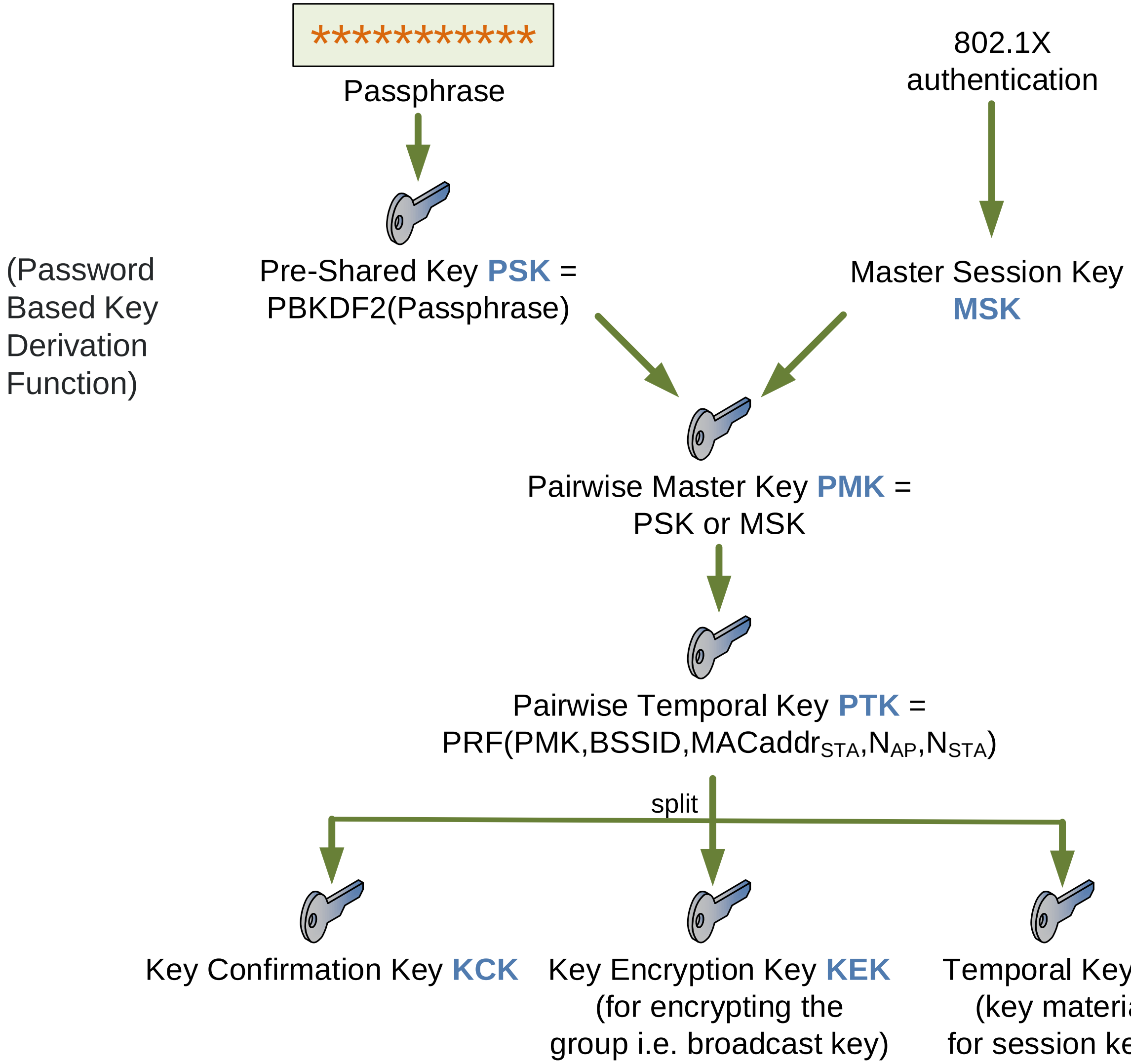
802.1X (EAPoL)

RADIUS

802.11

UDP/IP

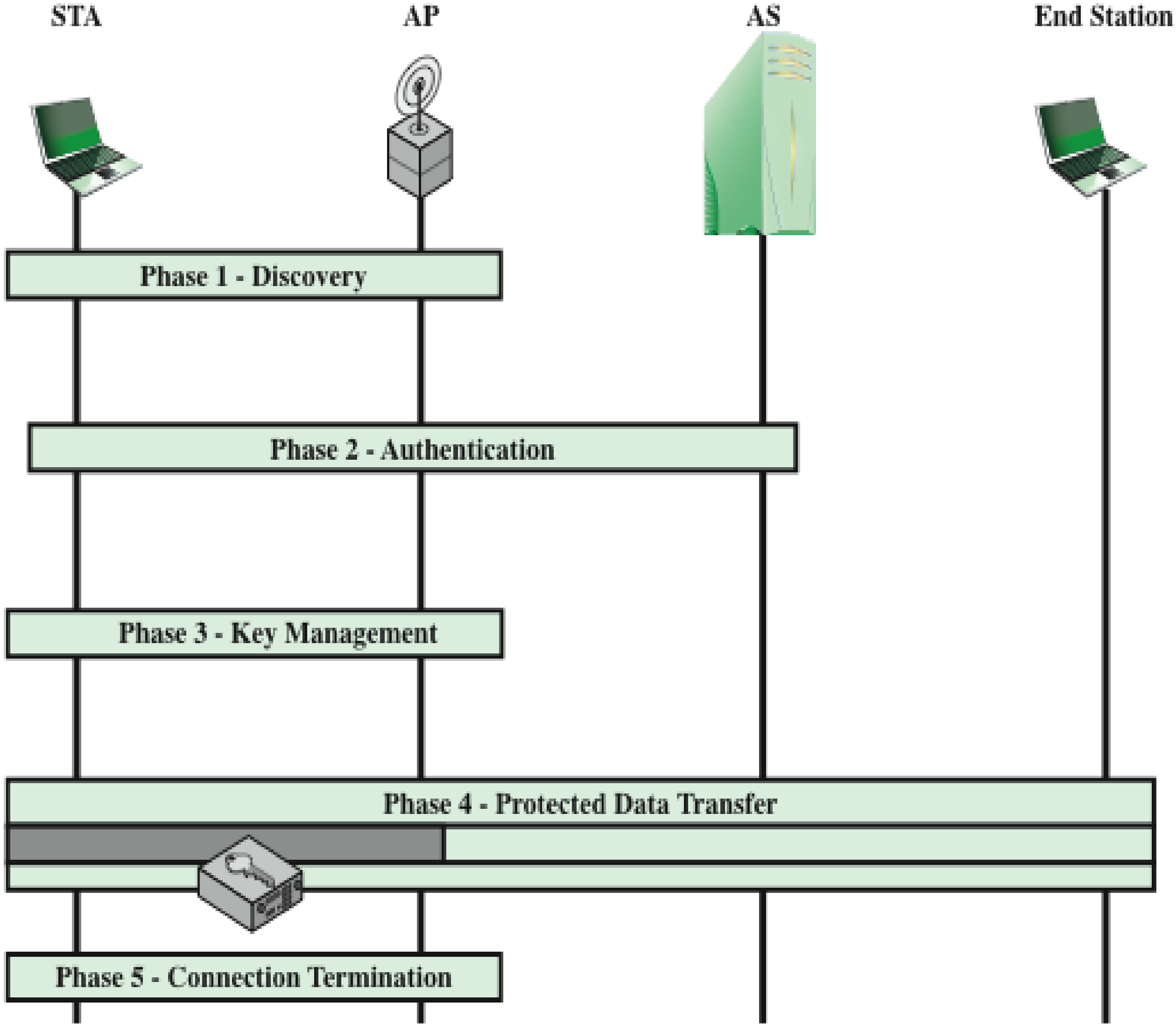
WPA2: Key Hierarchy



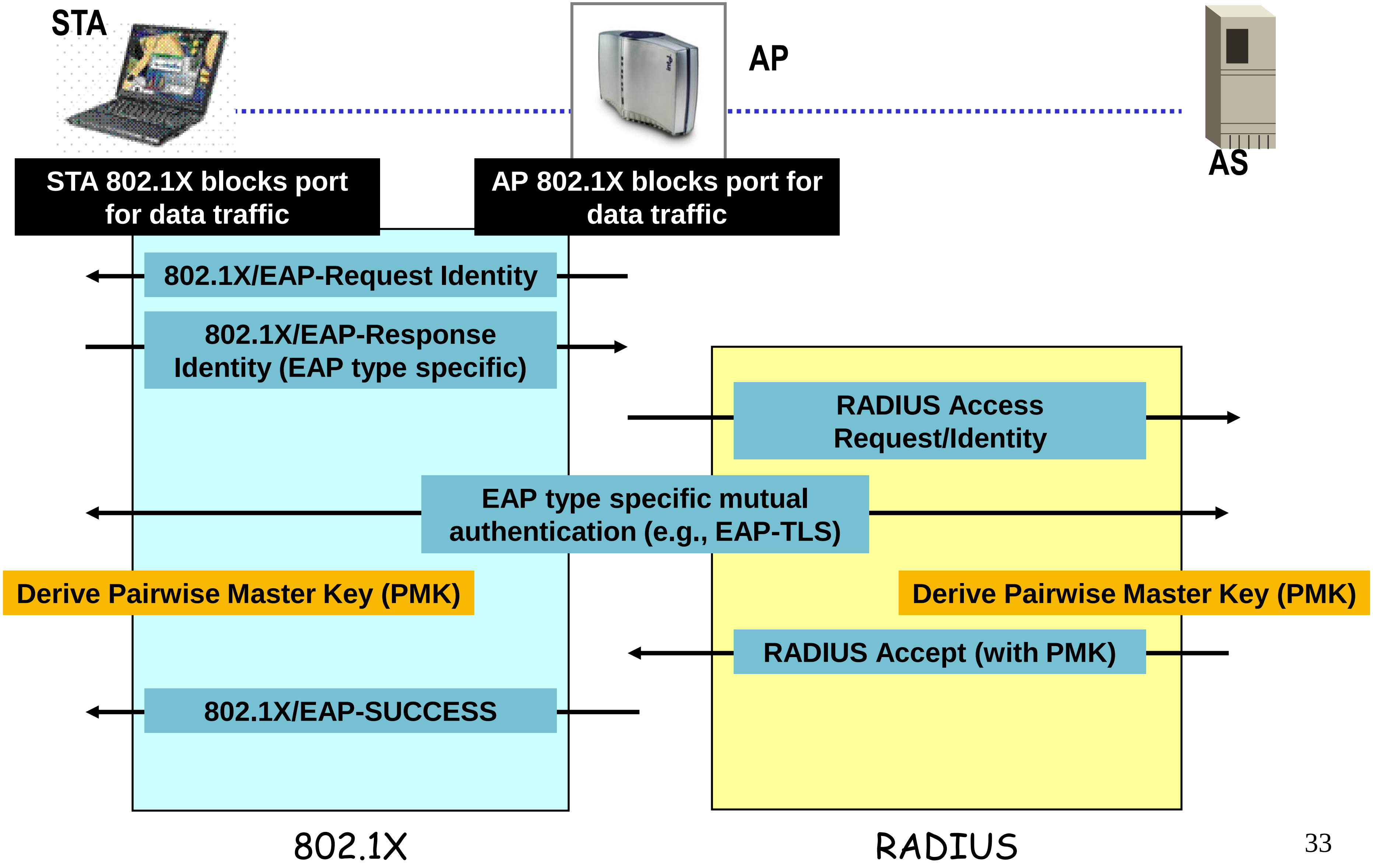
****
Passphrase
802.1X authentication
(Password Based Key Derivation Function)
Pre-Shared Key PSK = PBKDF2(Passphrase)
Master Session Key MSK
Pairwise Master Key PMK = PSK or MSK
Pairwise Temporal Key PTK = PRF(PMK,BSSID,MACaddr$_{STA}$,N$_{AP}$,N$_{STA}$)
split
Key Confirmation Key KCK
Key Encryption Key KEK (for encrypting the group i.e. broadcast key)
Temporal Key TK (key material for session keys)

- Two alternative ways to obtain keys:
  I. 802.1X authentication= WPA2-EAP = WPA2-Enterprise
     - Mutual auth of STA/AP
  II. Preshared key (PSK) authentication = WPA2-PSK = WPA2-Personal
     - Home/small business
     - No AS in network
     - Only STA auth by AP

WPA2-Enterprise: Operational Phases



Authentication Overview



STA

AP

AS

STA 802.1X blocks port for data traffic

AP 802.1X blocks port for data traffic

802.1X/EAP-Request Identity

802.1X/EAP-Response Identity (EAP type specific)

RADIUS Access Request/Identity

EAP type specific mutual authentication (e.g., EAP-TLS)

Derive Pairwise Master Key (PMK)

Derive Pairwise Master Key (PMK)

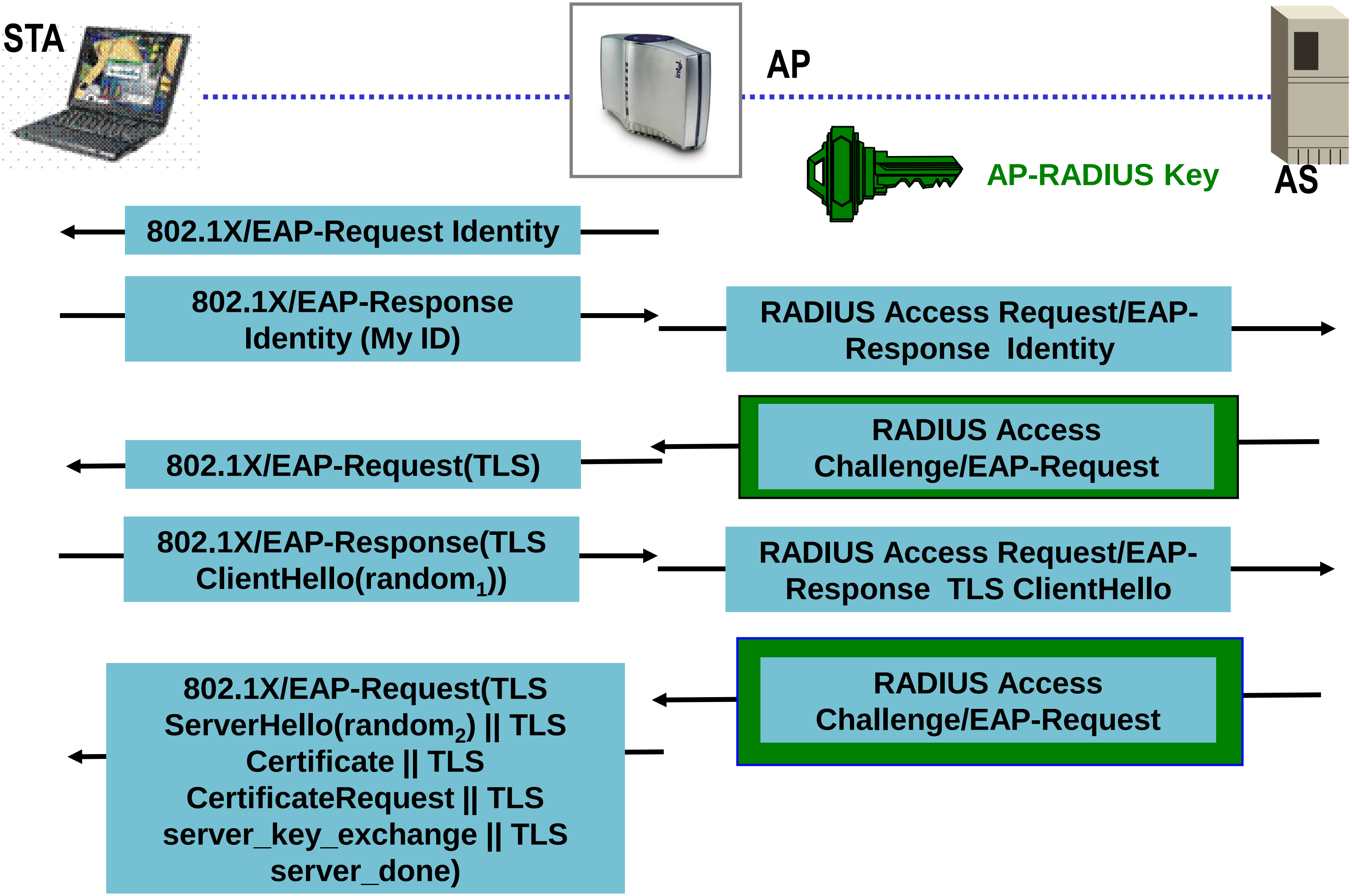
RADIUS Accept (with PMK)

802.1X/EAP-SUCCESS

802.1X

RADIUS

Example: EAP-TLS (1/2)



STA

AP

AS

AP-RADIUS Key

802.1X/EAP-Request Identity

802.1X/EAP-Response Identity (My ID)

RADIUS Access Request/EAP-Response Identity

RADIUS Access Challenge/EAP-Request

802.1X/EAP-Request(TLS)

802.1X/EAP-Response(TLS ClientHello(random$_1$))

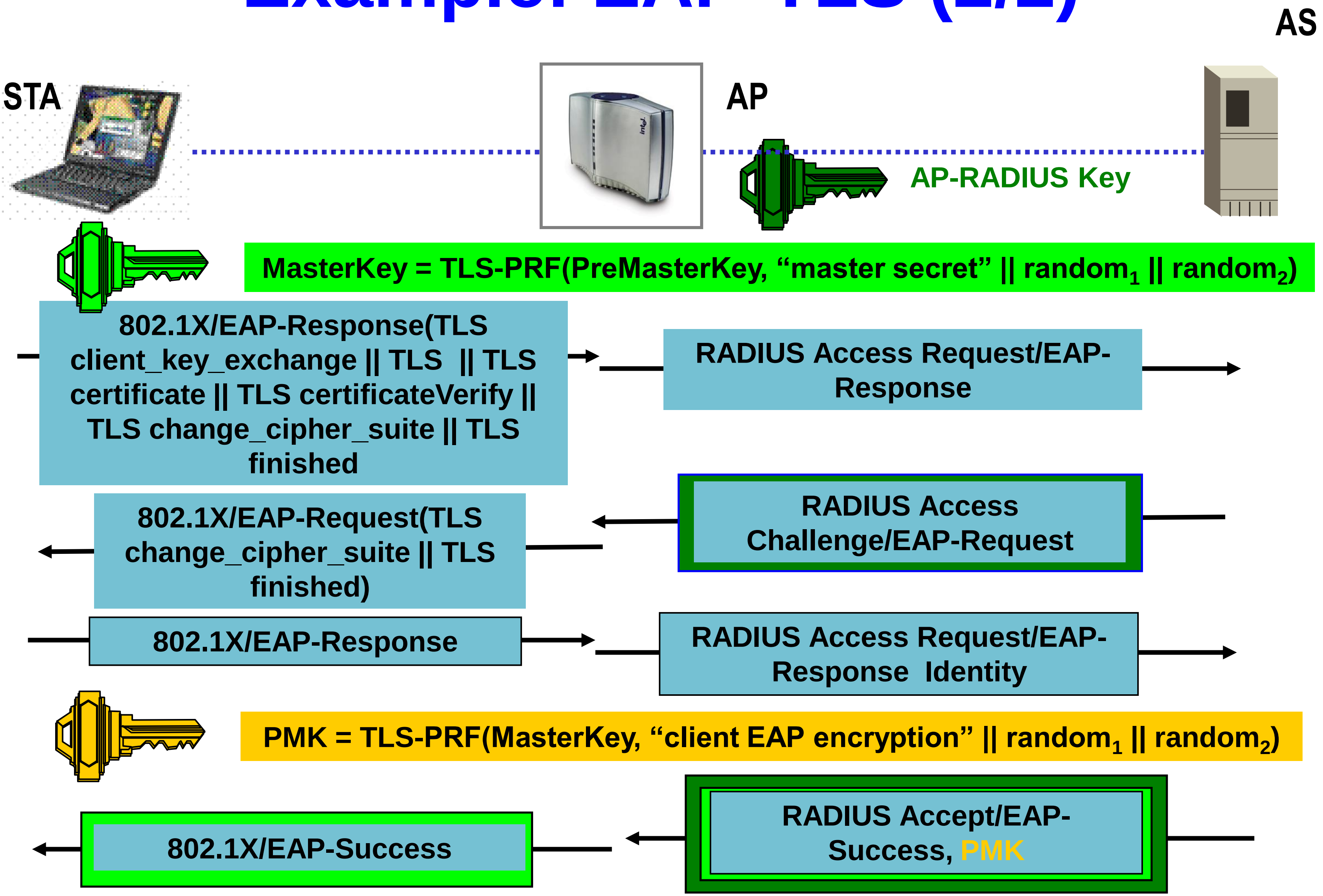
RADIUS Access Request/EAP-Response TLS ClientHello

RADIUS Access Challenge/EAP-Request

802.1X/EAP-Request(TLS ServerHello(random$_2$) || TLS Certificate || TLS CertificateRequest || TLS server_key_exchange || TLS server_done)

EAP-TLS is a certificate-based authentication protocol

Example: EAP-TLS (2/2)



MasterKey = TLS-PRF(PreMasterKey, “master secret” || random$_1$ || random$_2$)

802.1X/EAP-Response(TLS client_key_exchange || TLS || TLS certificate || TLS certificateVerify || TLS change_cipher_suite || TLS finished

RADIUS Access Request/EAP-Response

RADIUS Access Challenge/EAP-Request

802.1X/EAP-Request(TLS change_cipher_suite || TLS finished)

802.1X/EAP-Response

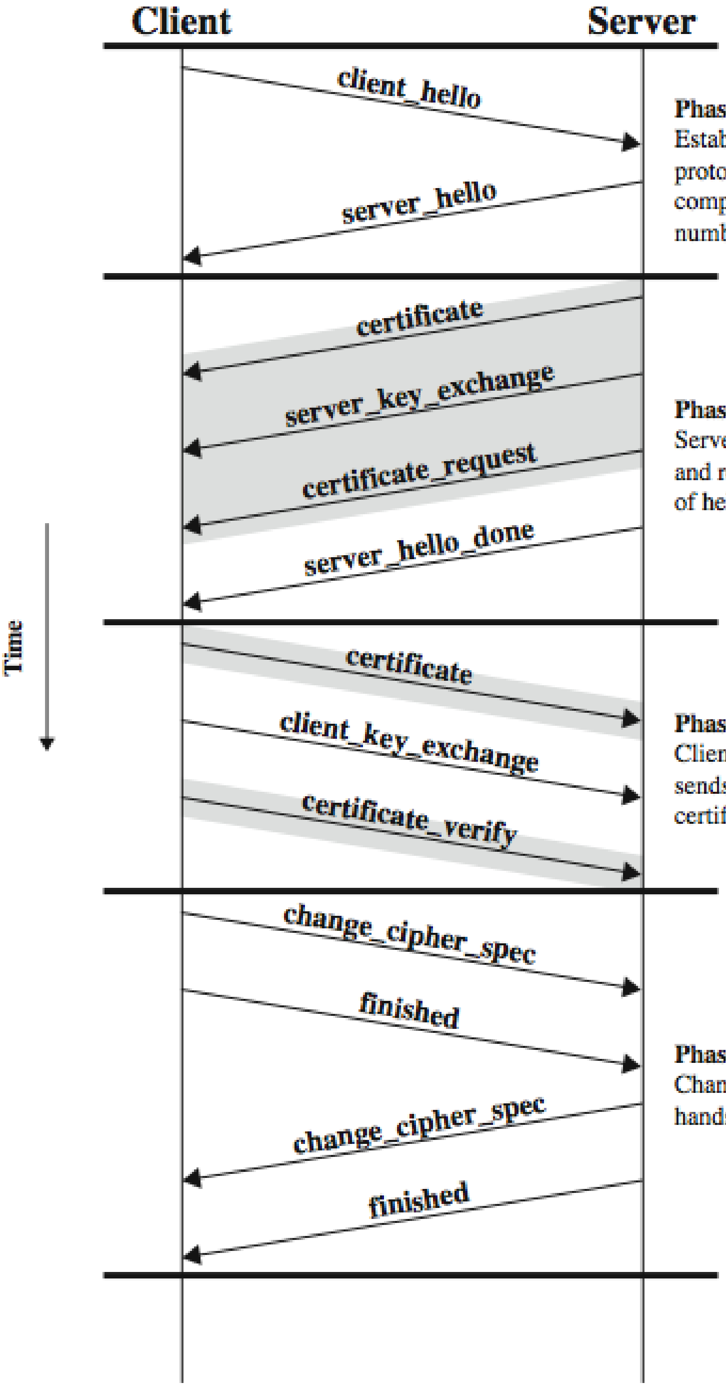
RADIUS Access Request/EAP-Response Identity

PMK = TLS-PRF(MasterKey, “client EAP encryption” || random$_1$ || random$_2$)

RADIUS Accept/EAP-Success, PMK

802.1X/EAP-Success

Two-Way TLS/SSL Handshaking



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

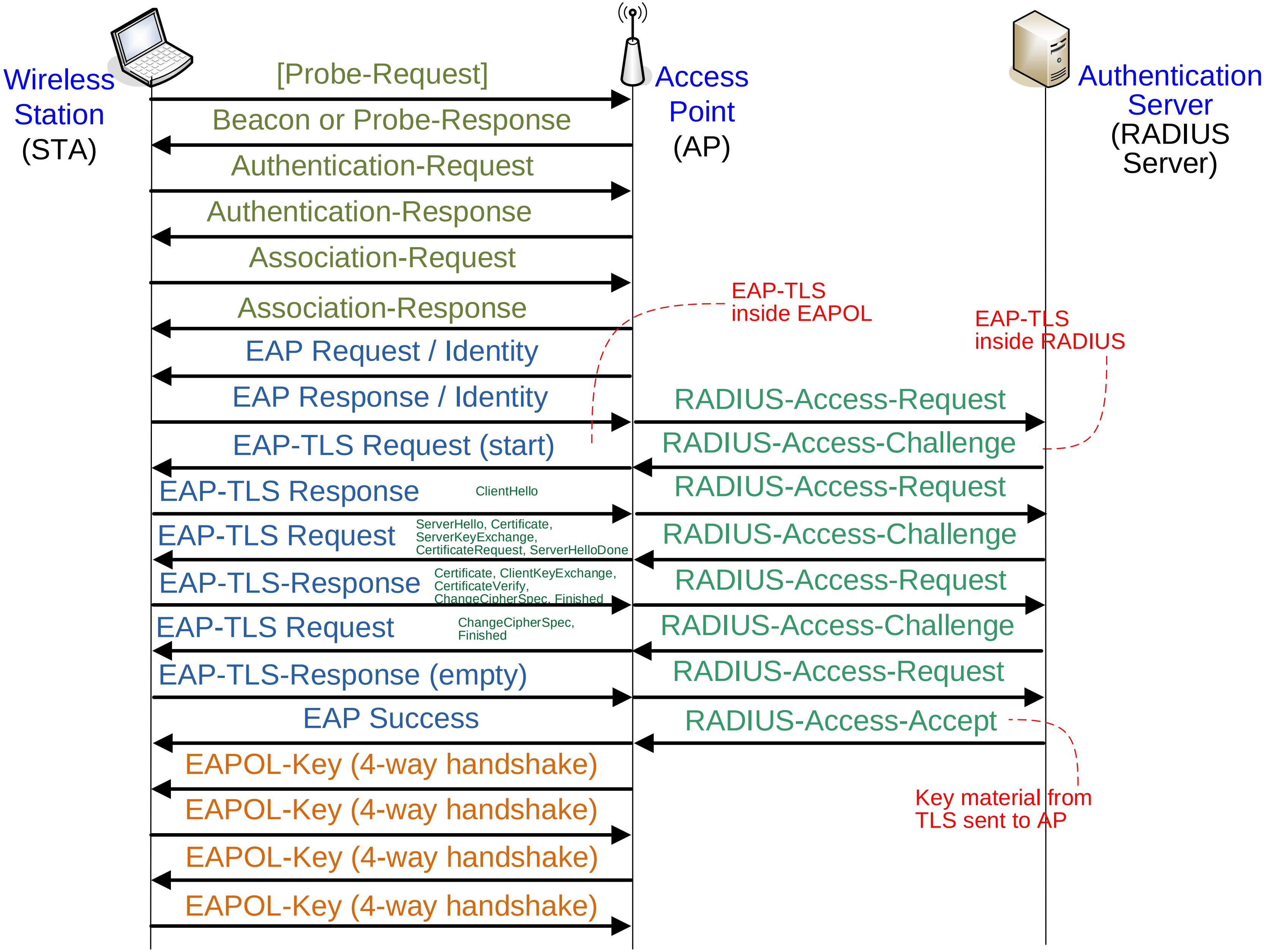
Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Full WPA2 Authentication (EAP-TLS) & Key Exchange



Wireless Station (STA)

Access Point (AP)

Authentication Server (RADIUS Server)

[Probe-Request]

Beacon or Probe-Response

Authentication-Request

Authentication-Response

Association-Request

Association-Response

EAP Request / Identity

EAP Response / Identity

RADIUS-Access-Request

EAP-TLS Request (start)

RADIUS-Access-Challenge

EAP-TLS Response ClientHello

RADIUS-Access-Request

EAP-TLS Request ServerHello, Certificate, ServerKeyExchange, CertificateRequest, ServerHelloDone

RADIUS-Access-Challenge

EAP-TLS-Response Certificate, ClientKeyExchange, CertificateVerify, ChangeCipherSpec, Finished

RADIUS-Access-Request

EAP-TLS Request ChangeCipherSpec, Finished

RADIUS-Access-Challenge

EAP-TLS-Response (empty)

RADIUS-Access-Request

EAP Success

RADIUS-Access-Accept

EAPOL-Key (4-way handshake)

EAPOL-Key (4-way handshake)

EAPOL-Key (4-way handshake)

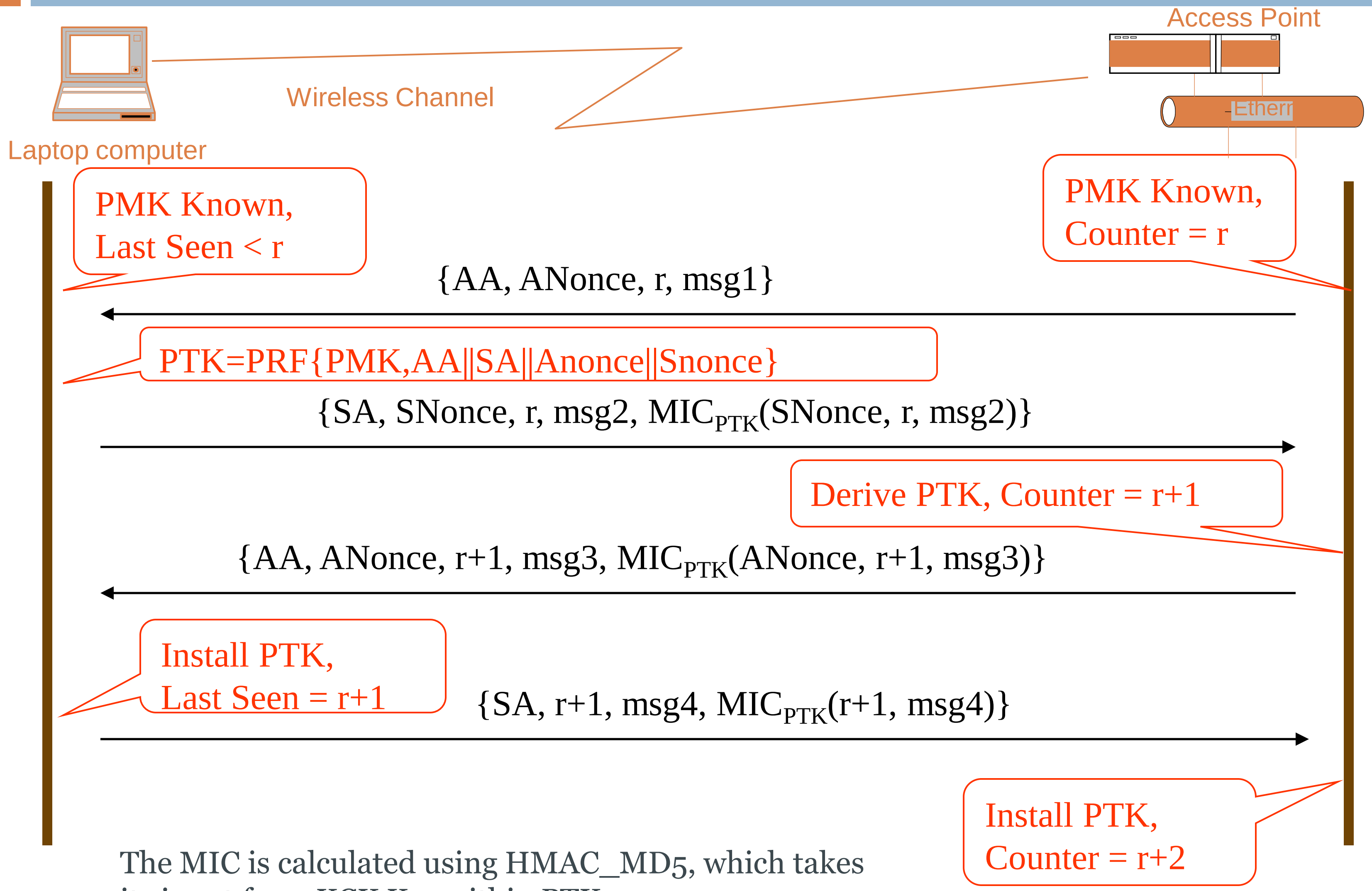
EAPOL-Key (4-way handshake)

EAP-TLS inside EAPOL

EAP-TLS inside RADIUS

Key material from TLS sent to AP

WPA2-PSK/EAP: 4-Way Handshake



PMK Known,
Last Seen < r

PMK Known,
Counter = r

{AA, ANonce, r, msg1}

PTK=PRF{PMK,AA||SA||Anonce||Snonce}

{SA, SNonce, r, msg2, MIC_{PTK}(SNonce, r, msg2)}

Derive PTK, Counter = r+1

{AA, ANonce, r+1, msg3, MIC_{PTK}(ANonce, r+1, msg3)}

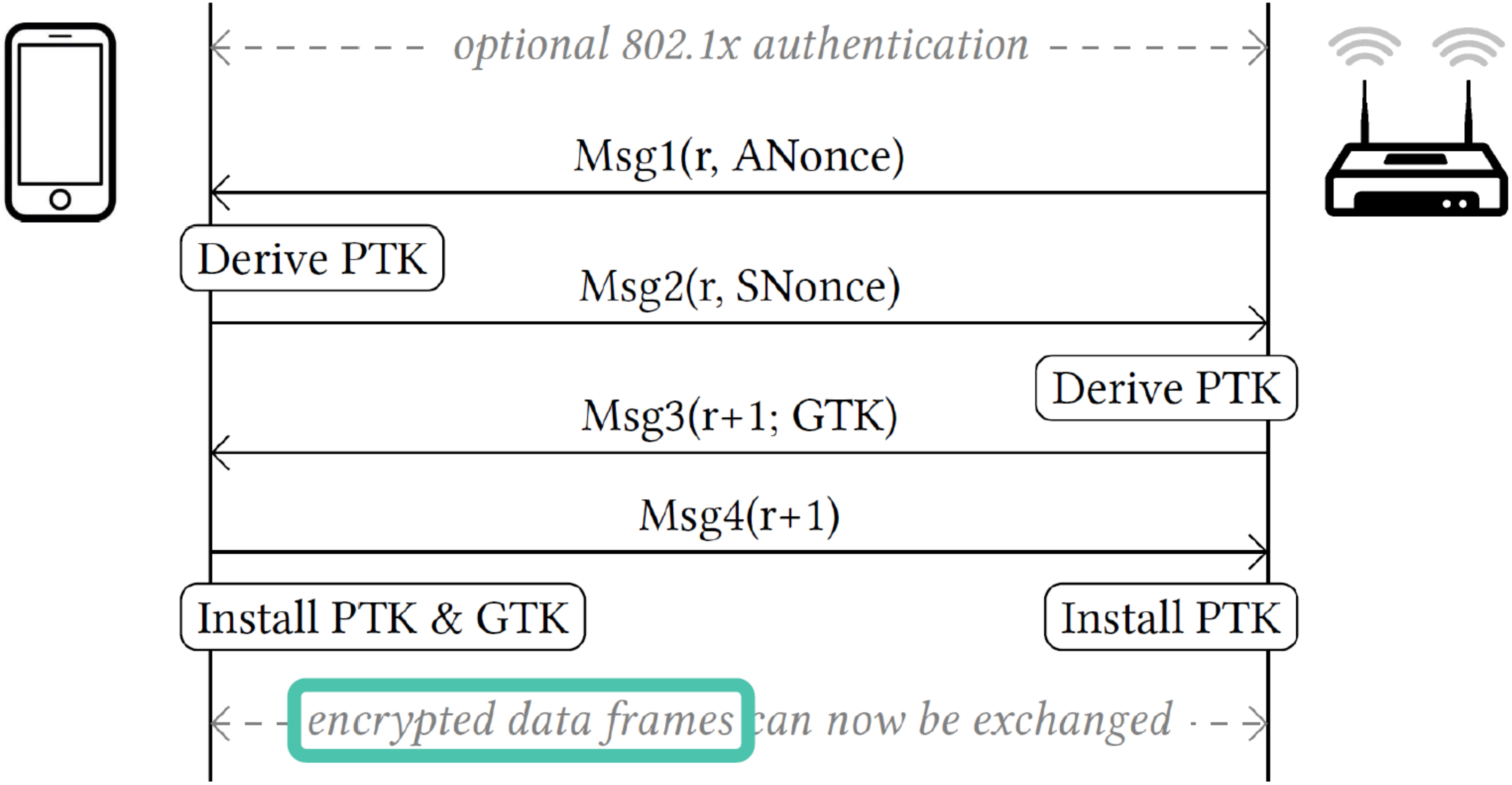
Install PTK,
Last Seen = r+1

{SA, r+1, msg4, MIC_{PTK}(r+1, msg4)}

Install PTK,
Counter = r+2

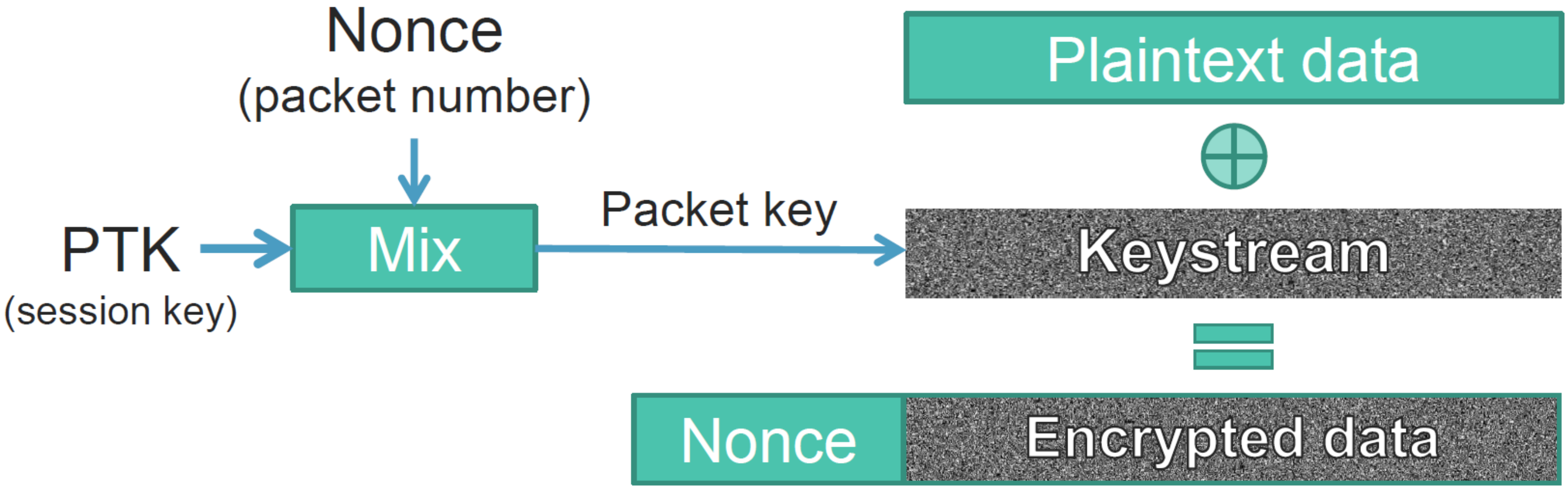
The MIC is calculated using HMAC_MD5, which takes its input from KCK Key within PTK.

WPA2-PSK/EAP: 4-Way Handshake



Both WPA2-PSK & EAP make use of AES-CCMP to encrypt data

Encryption of 802.11 MAC Payloads



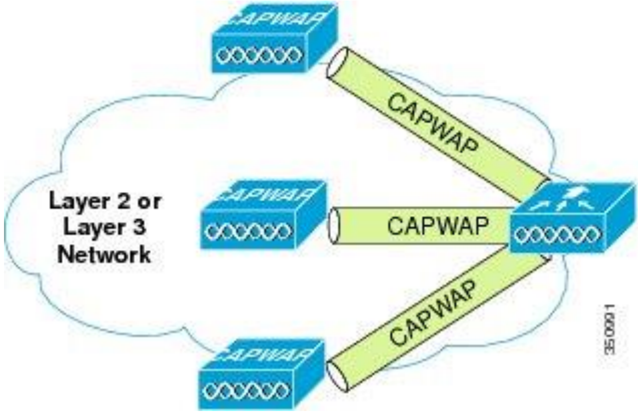
Both WPA2-PSK & EAP make use of AES-CCMP (**C**ounter Mode-**C**ipher Block Chaining **M**essage Authentication Code **P**rotocol) to encrypt data (confidentiality, /w Counter Mode) and to offer integrity protection (/w MAC/MIC)

[CWSP – CCMP Encryption Method | mrn-cciew (mrncciew.com)]

IITH Wi-Fi

❑ Cisco Aironet 3700 Series Access Points

- Dual-band 2.4 and 5 GHz with 802.11ac Wave 1 (draft std) support
- Servers 11a/b/g/n/ac STAs **/w integrated radios**
- Supports 20-, 40- and 80 MHz channels
- Max Tx Power of 23 dBm (200 mW)
- 4*4 MIMO with 3 spatial streams
- A-MSDU and A-MPDU aggregation, WMM (11e)
- 802.11 Dynamic Frequency Selection (DFS)
- PHY data rates up to 1.3 Gbps (80 MHz on 5 GHz)
- **Data Sheet**

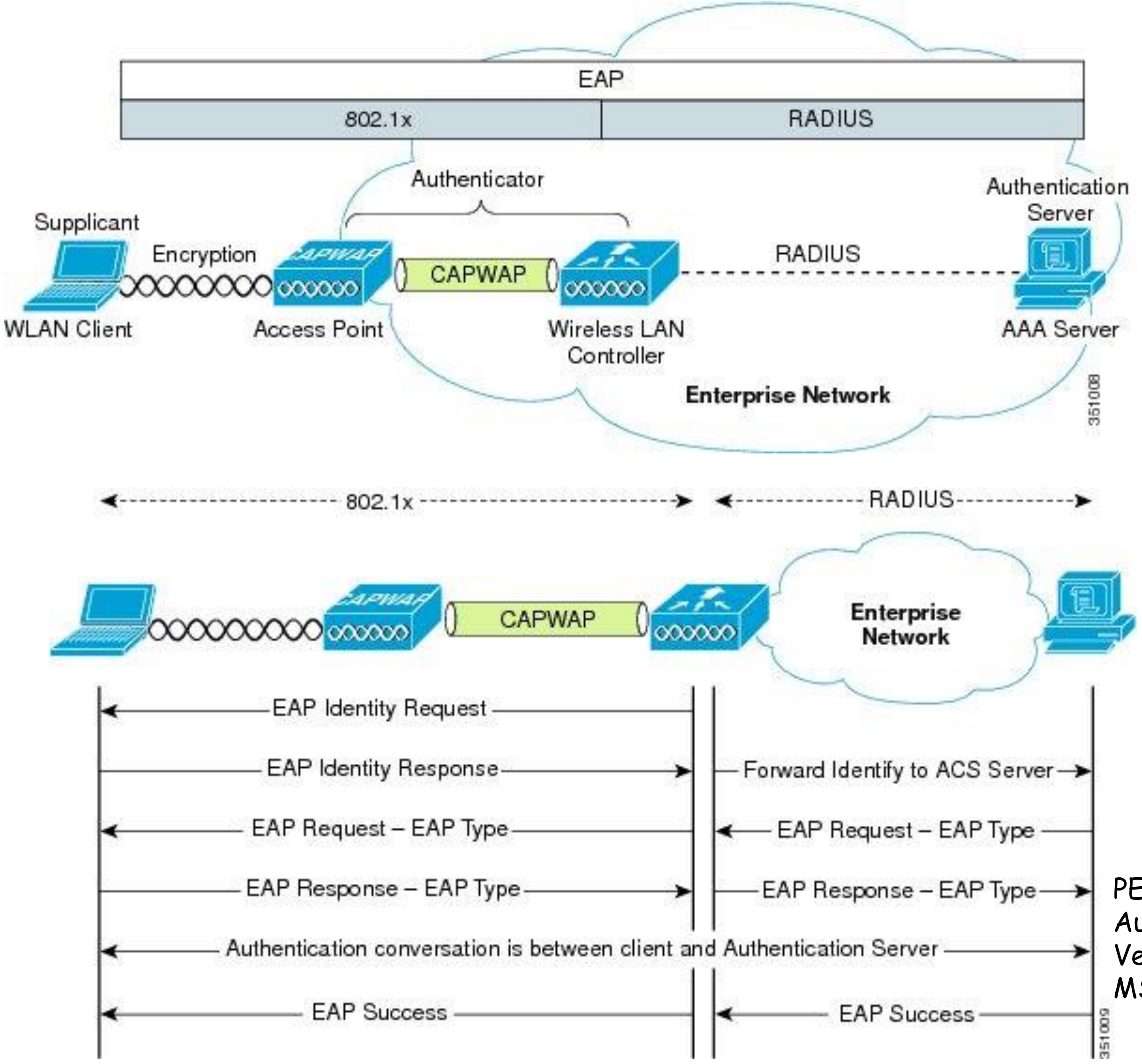


❑ Cisco 5508 WLAN Controller

- CAPWAP Architecture where APs are kept in light-weight (split-MAC) mode
  - CAPWAP: Control and Provisioning of Wireless Access Points, IETF std
  - Timing-dependent operations are generally managed locally on CAPWAP AP, while more complex, less time-dependent operations are managed on the WLC
    - Beacons, control and data frames, encryption by CAPWAP AP, rest by WLC
  - Central configuration, management of APs & two-way (UDP) tunneling of traffic b/w Controller and APs
  - Load-balancing, interference management (DFS), Uninterrupted network access when roaming, QoS, power control, etc
- Supports up to 500 APs and 7000 STAs

❑ **Data Sheet**

IITH Wi-Fi

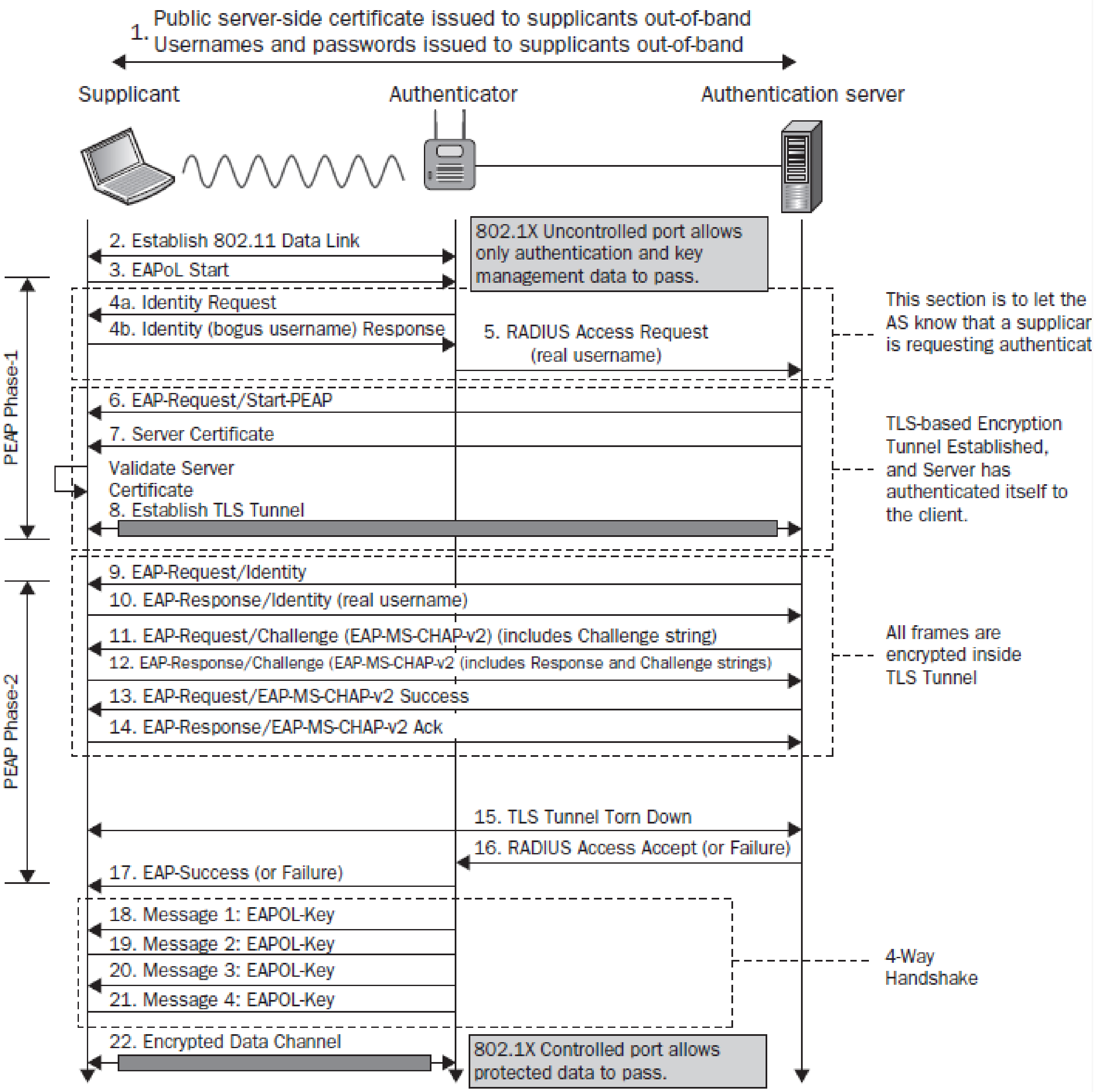


PEAP-Microsoft Challenge Authentication Protocol Version 2 (PEAP-MSCHAPv2): TLS tunnel

Secure Wireless Topology, EAP Message Flow, Credit: Cisco

https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/emob73dg/emob73/ch4_Secu.html

PEAP



1. Public server-side certificate issued to supplicants out-of-band
Usernames and passwords issued to supplicants out-of-band

Supplicant

Authenticator

Authentication server

2. Establish 802.11 Data Link

3. EAPoL Start

802.1X Uncontrolled port allows only authentication and key management data to pass.

4a. Identity Request

4b. Identity (bogus username) Response

5. RADIUS Access Request
(real username)

This section is to let the AS know that a supplicar is requesting authenticat

6. EAP-Request/Start-PEAP

7. Server Certificate

Validate Server Certificate

8. Establish TLS Tunnel

TLS-based Encryption Tunnel Established, and Server has authenticated itself to the client.

PEAP Phase-1

9. EAP-Request/Identity

10. EAP-Response/Identity (real username)

11. EAP-Request/Challenge (EAP-MS-CHAP-v2) (includes Challenge string)

12. EAP-Response/Challenge (EAP-MS-CHAP-v2 (includes Response and Challenge strings)

13. EAP-Request/EAP-MS-CHAP-v2 Success

14. EAP-Response/EAP-MS-CHAP-v2 Ack

All frames are encrypted inside TLS Tunnel

15. TLS Tunnel Torn Down

16. RADIUS Access Accept (or Failure)

17. EAP-Success (or Failure)

PEAP Phase-2

18. Message 1: EAPOL-Key

19. Message 2: EAPOL-Key

20. Message 3: EAPOL-Key

21. Message 4: EAPOL-Key

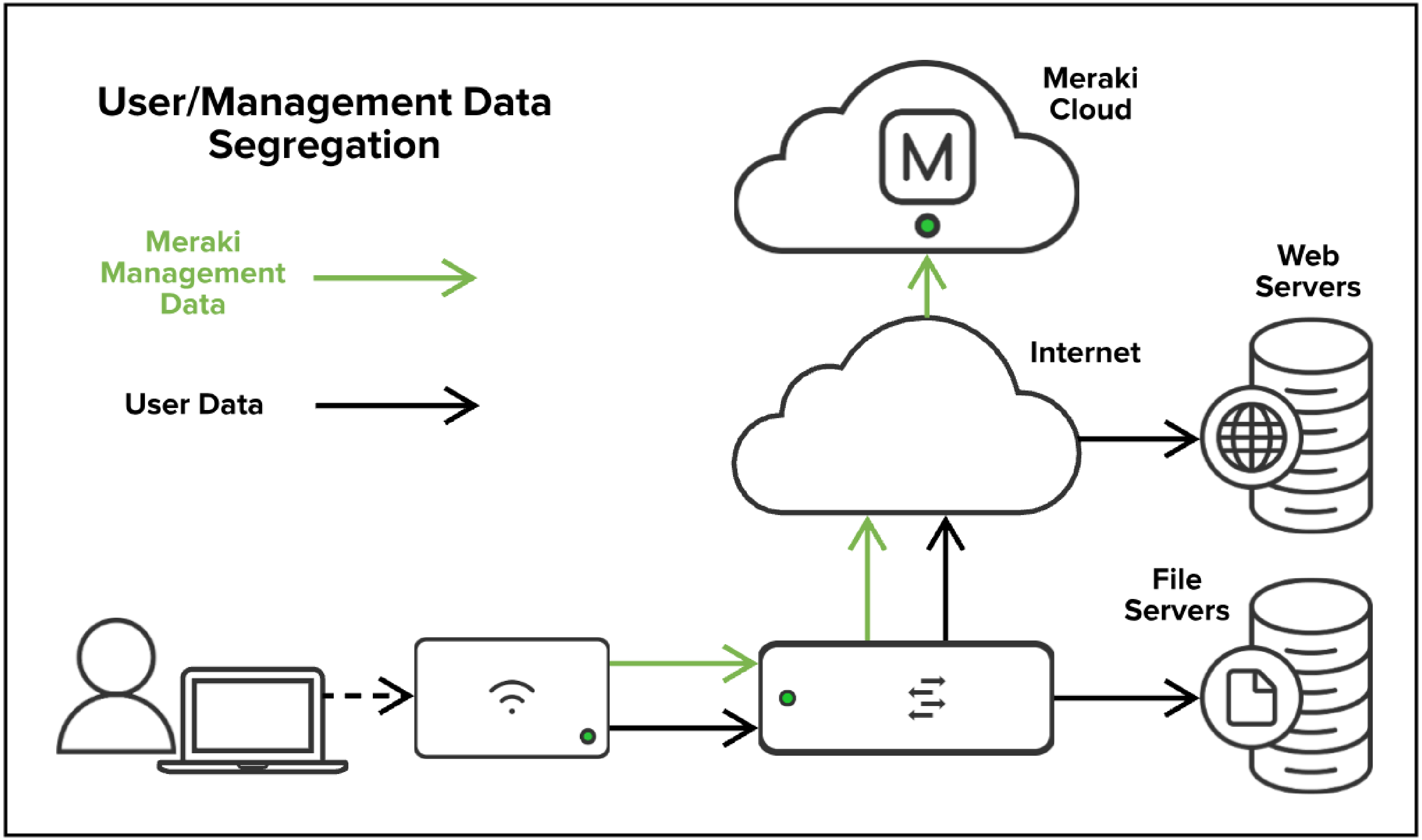
4-Way Handshake

22. Encrypted Data Channel

802.1X Controlled port allows protected data to pass.

https://mrncciew.com/2014/08/25/cwsp-eap-peap/

Cloud based Wi-Fi Mgmt



Cisco Meraki Best Practice Design - Cisco Meraki

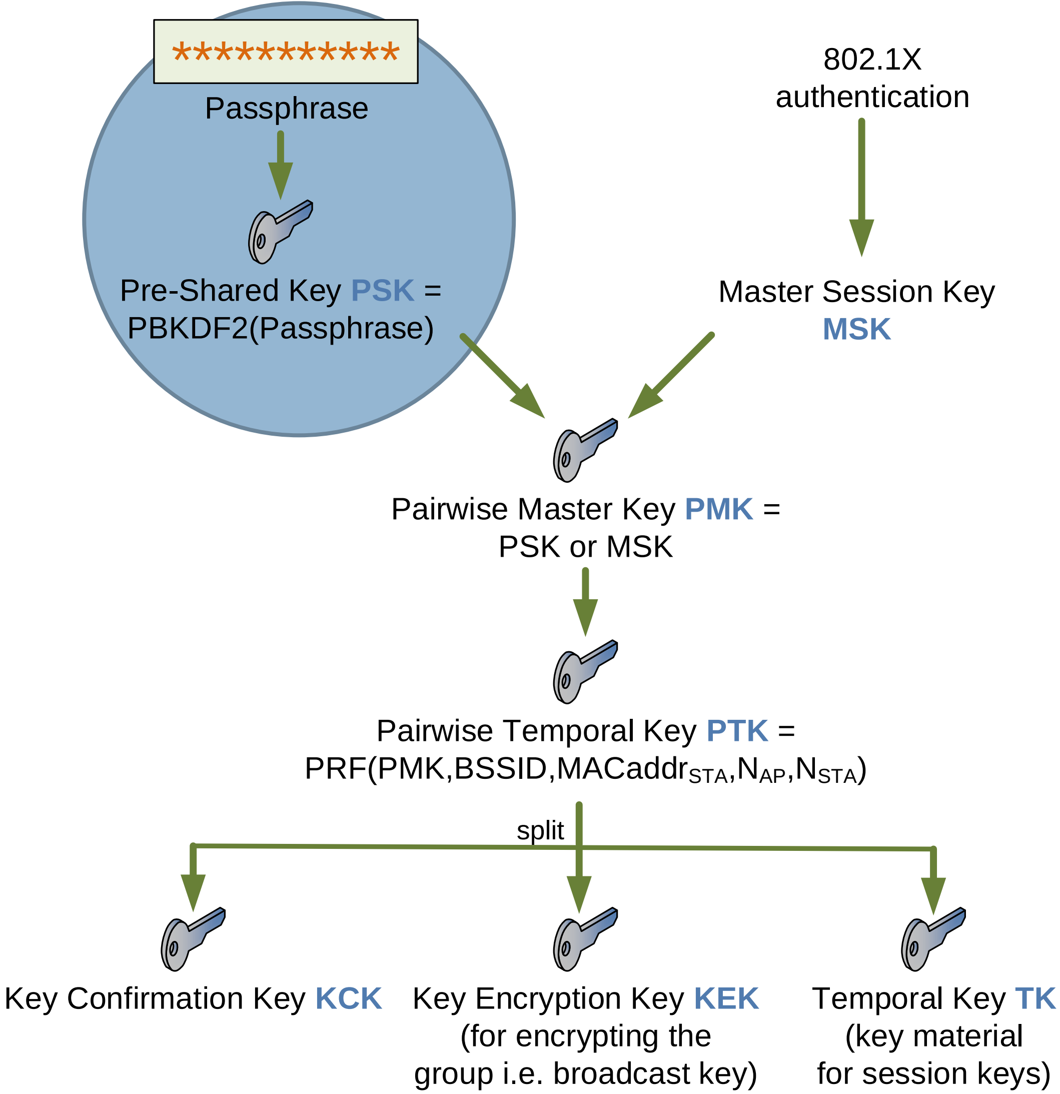
Attacks on WPA2!

- Eavesdropping (esp OPEN networks)
- WPA2-PSK: MITM attacks
  - Association with Evil Twin APs
- WPA2-PSK: Offline dictionary attacks
- WPA2-PSK/EAP: KRACK attacks
- AP configuration over HTTP
- Denial of Service (DoS) attacks

CRACKING WPA2-PSK PASSWORD WITH OFFLINE DICTIONARY ATTACK

Cracking_wpa [Aircrack-ng]

WPA2: Key Hierarchy (recap)



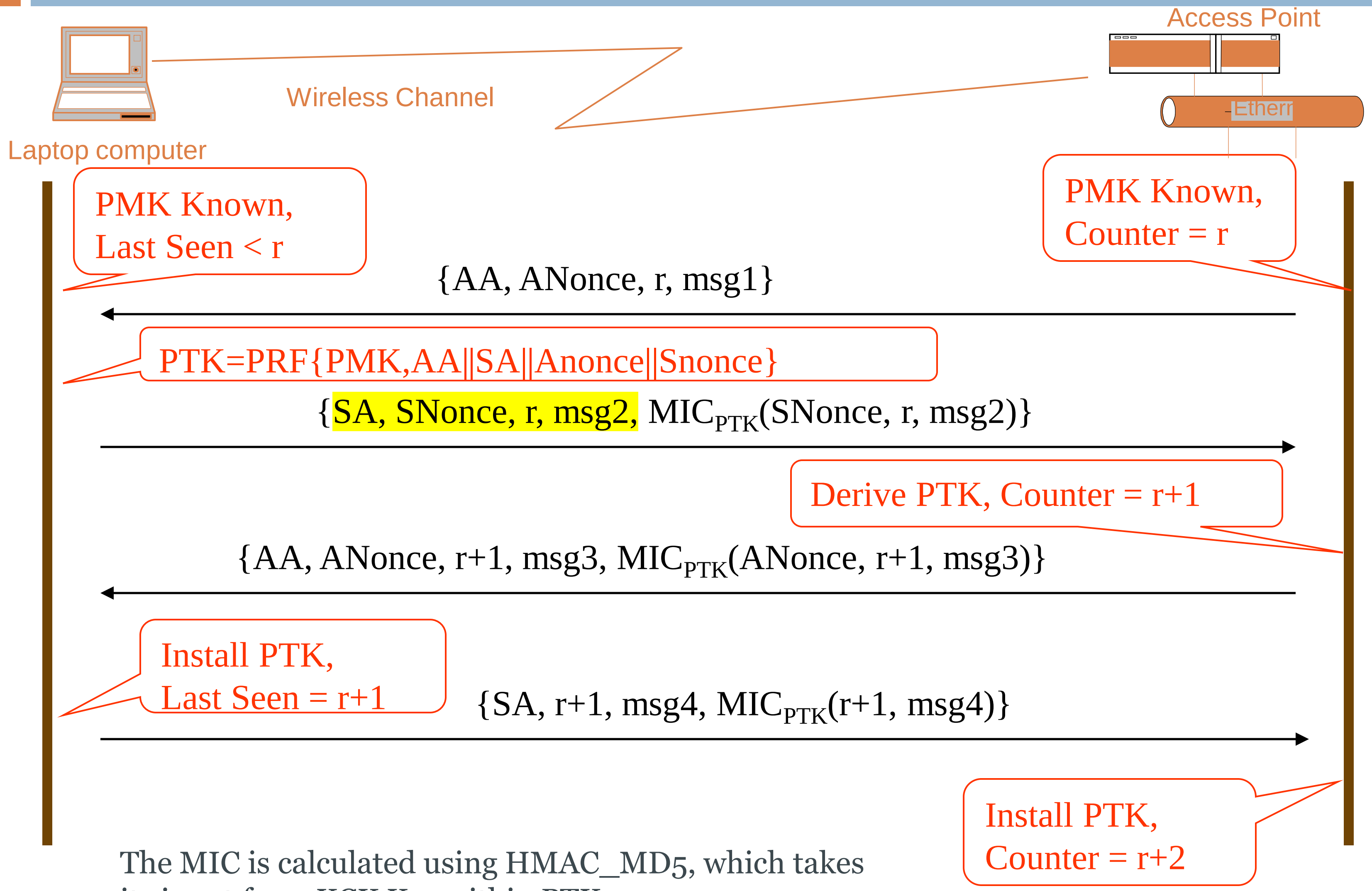
PBKDF2=Password Based Key Derivation Function #2

PSK = PBKDF2(HMAC−SHA1, passphrase, SSID, 4096, 256)

HMAC-SHA1 is a hash based Message Authentication code using SHA1 with passphrase as key and SSID as salt

N_{AP}: Nonce of AP
Nonce: Numbed used once!

WPA2-PSK Offline Dictionary Attack



PMK Known,
Last Seen < r

PMK Known,
Counter = r

{AA, ANonce, r, msg1}

PTK=PRF{PMK,AA||SA||Anonce||Snonce}

{SA, SNonce, r, msg2, MIC$_{PTK}$(SNonce, r, msg2)}

Derive PTK, Counter = r+1

{AA, ANonce, r+1, msg3, MIC$_{PTK}$(ANonce, r+1, msg3)}

Install PTK,
Last Seen = r+1

{SA, r+1, msg4, MIC$_{PTK}$(r+1, msg4)}

Install PTK,
Counter = r+2

The MIC is calculated using HMAC_MD5, which takes its input from KCK Key within PTK.

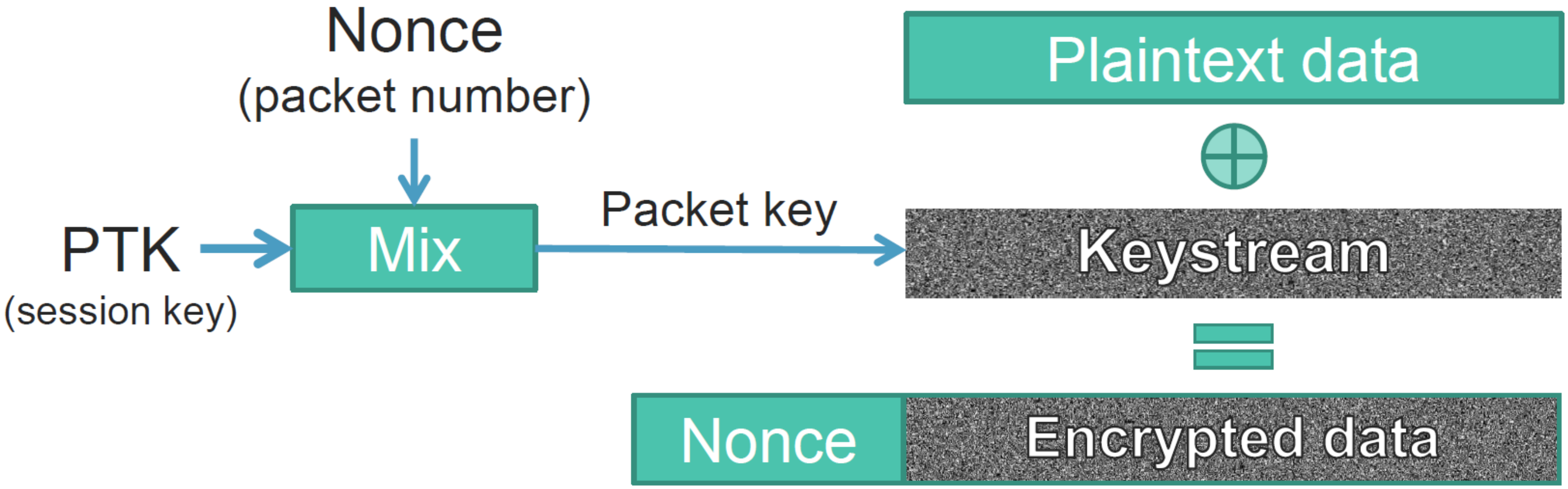
Demo of Cracking WPA2-PSK

https://www.youtube.com/watch?v=WfYxrLaqlN8
https://www.youtube.com/watch?v=Usw0IlGbkC4

KRACK: Key Reinstallation Attacks on WPA2

- Discovered by Mathy Vanhoef, KU Leuven in 2017
- Kind of weakness/ambiguity in .11i std, so effects varied across OS implementations
- So, many devices with Wi-Fi radio were affected
  - Linux and Android 6.0 or higher were highly vulnerable
  - All data from victim could be decrypted
- Main attack is against the 4-way handshake of the WPA2 protocol
  - Both WPA2-Personal and WPA2-Enterprise were vulnerable
- **It does not recover passphrase of Wi-Fi network**
  - Also does not recover (any parts of) the fresh encryption key (PTK) that is negotiated during the 4-way handshake.

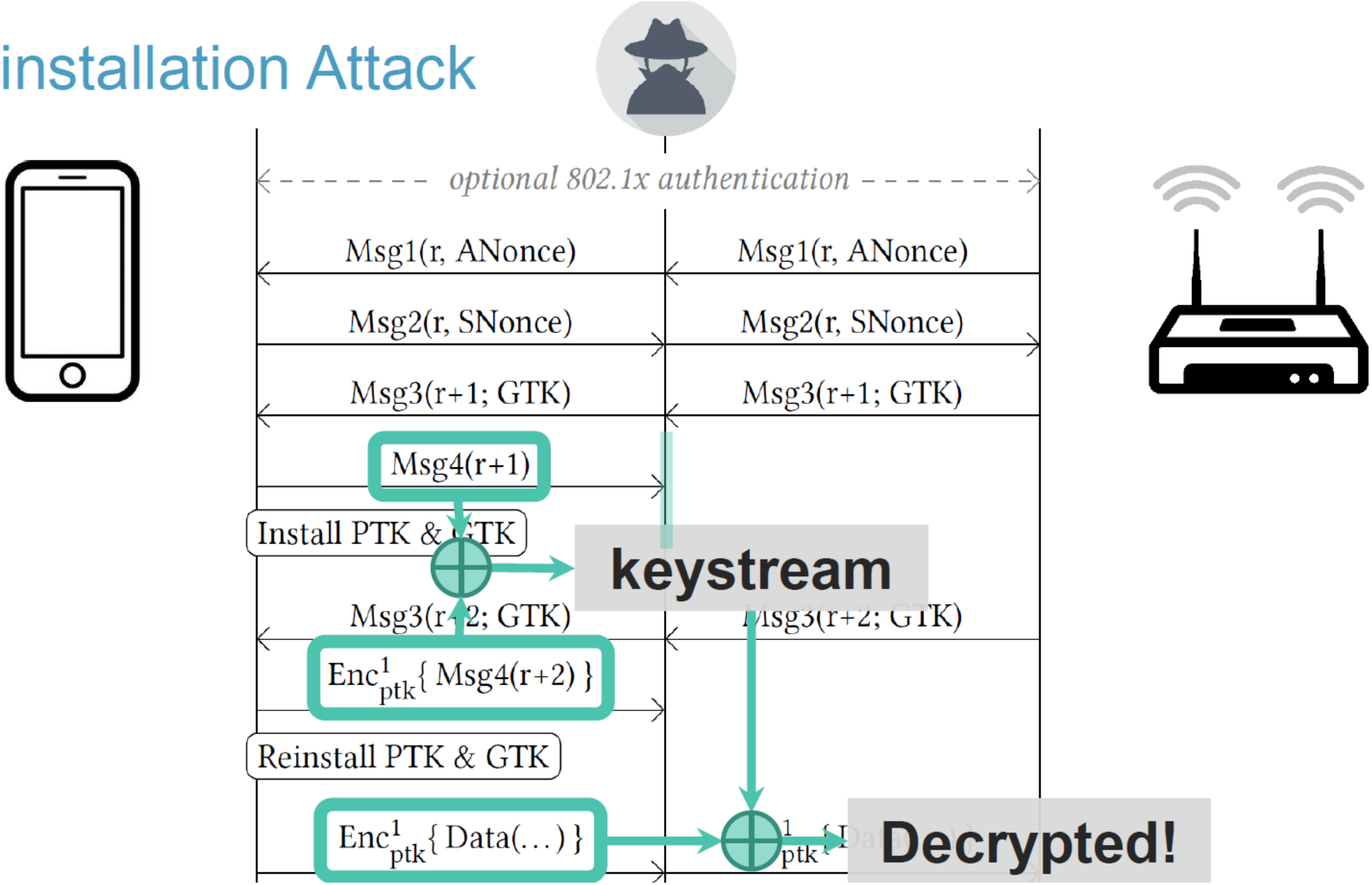
Encryption of 802.11 MAC Payloads



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

KRACK: MITM attack on 4-Way H/S

Reinstallation Attack



optional 802.1x authentication

Msg1(r, ANonce)

Msg1(r, ANonce)

Msg2(r, SNonce)

Msg2(r, SNonce)

Msg3(r+1; GTK)

Msg3(r+1; GTK)

Msg4(r+1)

Install PTK & GTK

keystream

Msg3(r+2; GTK)

Msg3(r+2; GTK)

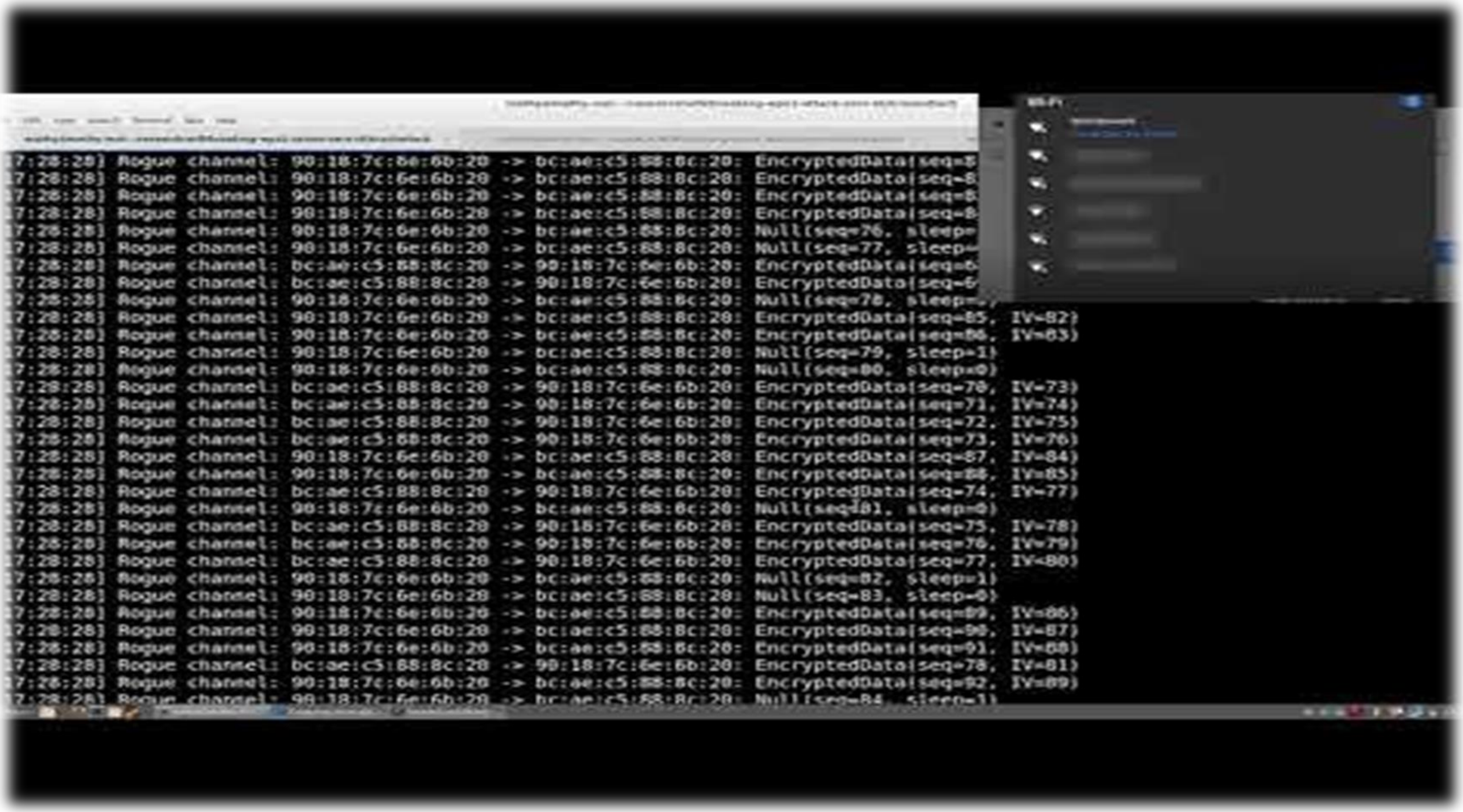
$\mathrm{Enc}^{1}_{\mathrm{ptk}}\{\,\mathrm{Msg4}(r+2)\,\}$

Reinstall PTK & GTK

$\mathrm{Enc}^{1}_{\mathrm{ptk}}\{\,\mathrm{Data}(\ldots)\,\}$

Decrypted!

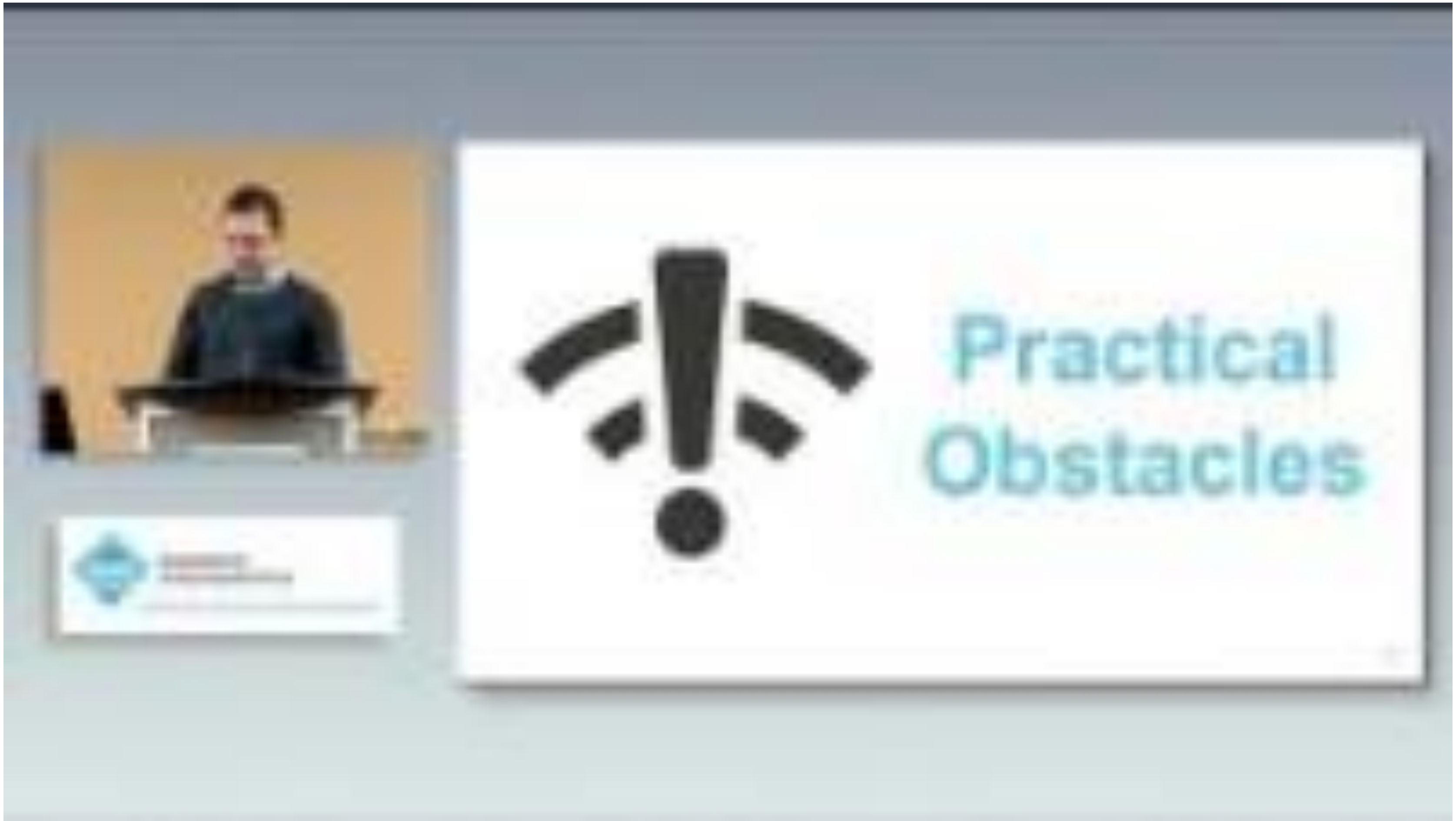
KRACK Attack: Demo



```
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=8
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=8
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=8
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=8
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=76, sleep=
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=77, sleep=
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=6
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=6
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=78, sleep=
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=85, IV=82)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=86, IV=83)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=79, sleep=1)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=80, sleep=0)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=70, IV=73)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=71, IV=74)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=72, IV=75)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=73, IV=76)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=87, IV=84)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=88, IV=85)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=74, IV=77)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=81, sleep=0)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=75, IV=78)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=76, IV=79)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=77, IV=80)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=82, sleep=1)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=83, sleep=0)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=89, IV=86)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=90, IV=87)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=91, IV=88)
7:28:28] Rogue channel: bc:ae:c5:88:8c:20 -> 90:18:7c:6e:6b:20: EncryptedData(seq=78, IV=81)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: EncryptedData(seq=92, IV=89)
7:28:28] Rogue channel: 90:18:7c:6e:6b:20 -> bc:ae:c5:88:8c:20: Null(seq=84, sleep=1)
```

[KRACK - Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2 - YouTube]

KRACK Attack: Presentation



Release the Kraken: New KRACKs in the 802.11 Standard - YouTube

How to defend against KRACK?

- 802.11i std was amended as follows:
  - When an already-in-use key is being reinstalled, the standard now states that the associated transmit nonce and receive replay counter should not be reset!
    - But it does not prevent group key reinstallation attacks ☹

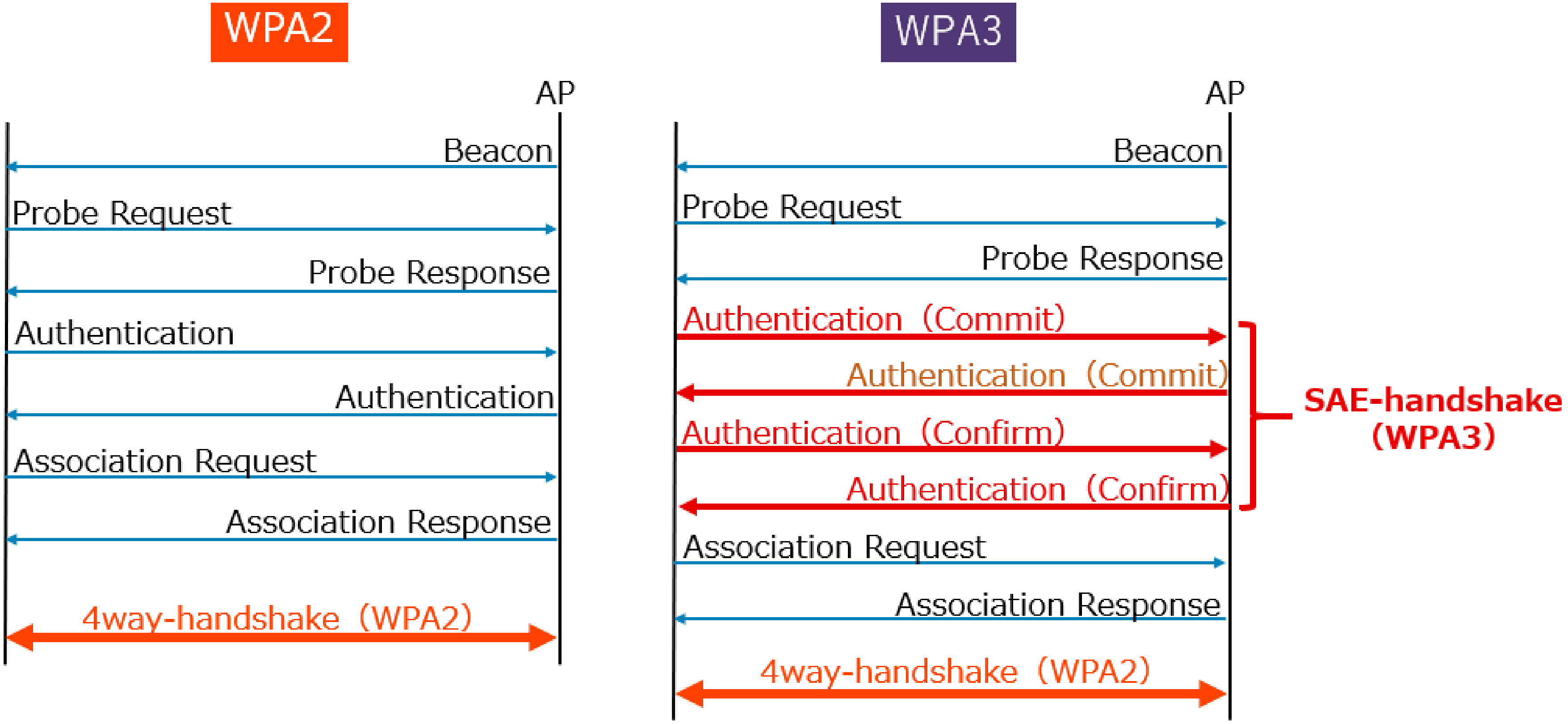
WPA3: Must for Wi-Fi 6 & Beyond

I. WPA3-Enterprise
II. WPA3-Personal leverages Simultaneous Authentication of Equals (SAE) to protect users against offline dictionary attacks
III. Enhanced Open for encryption without authentication in Open networks

WPA3-Enterprise

- Supports Management Frame Protection (MFP)
- Still leverages 802.1X/EAP for authentication like WPA2
- 3 modes of operation
  - WPA3-Enterprise Only
  - WPA3-Enterprise Transition
  - WPA3-Enterprise 192-bit (optional)
    - 256-bit GCMP/AES instead of 128-bit CCMP/AES
    - BIP-GMAC-256 for MFP instead of BIP-CMAC-128
    - EAP-TLS as the authentication protocol

WPA2-Personal vs WPA3-Personal



https://balramdot11b.com/2020/05/17/wpa3-and-dragonfly-sae/

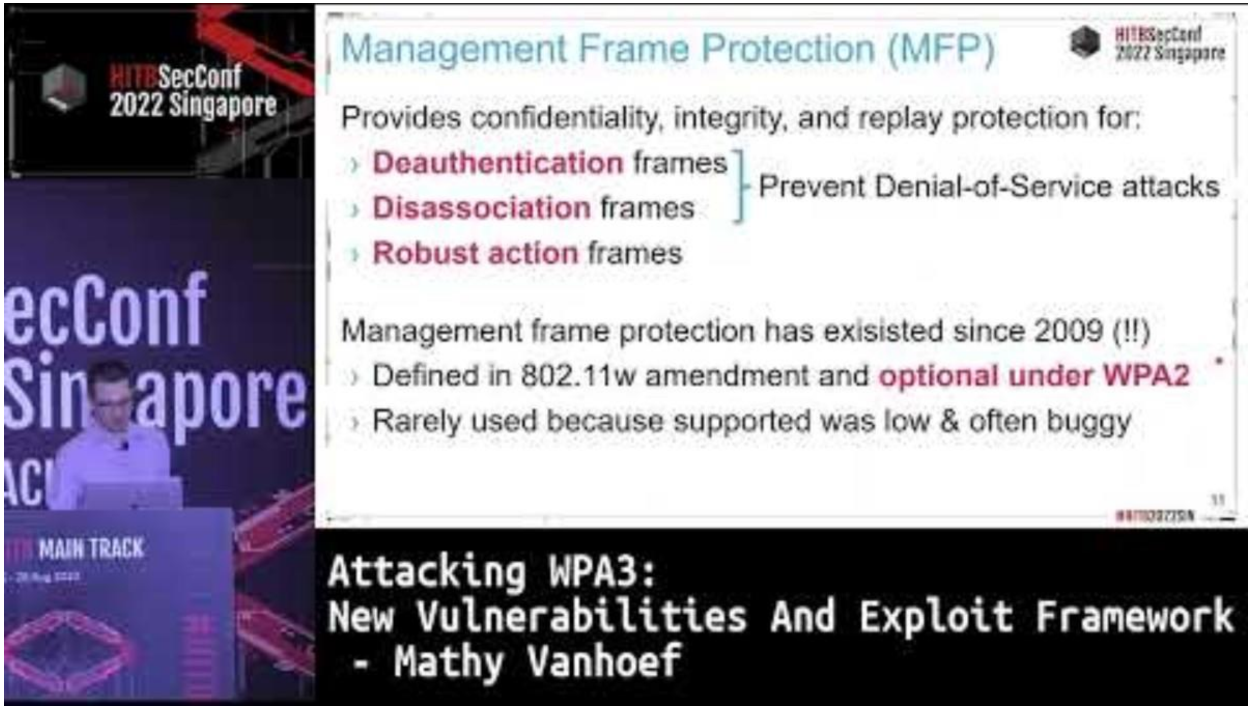
WPA3-Personal: Dragonfly

- **Dragonfly: Offline Dictionary Attack Resistance for PSK Passwords**
  - Even when users choose weak passwords
  - IETF RFC 7664 and Section 12.4 (SAE) of IEEE 802.11 Std
    - Simultaneous Authentication of Equals (SAE)
- SAE is a variant of Diffie-Hellman key exchange to facilitate both encryption key generation & mutual AUTH
  - SAE handshake (commit and confirm msgs with password) to derive a fresh PMK at STA and AP after mutual AUTH
  - PMK is used to get PTK by doing 4-way handshake as usual
- Forward secrecy: Even if passphrase is leaked at a later point in time, it still cannot be used to decrypt the eavesdropped packets from the past unlike WPA2

WPA3: OWE

- **OWE: Opportunistic Wireless Encryption for Open SSIDs**
  - Meant for open/public APs
  - Encryption w/o authentication like securely reading https://www.thehindu.com/ without login
  - Diffie-Hellman key exchange, does not require any certs
    - OWE handshake using Re(association) REQ/RES negotiates a new PMK b/w STA and AP
  - Not a replacement for any of existing auth methods
  - Does not offer AUTH (both client-side and AP-side)
    - Solution for client-side AUTH: Captive portal
    - No solution for server-side AUTH
      - Rogue APs (Evil Twins) can still be setup

Attacks on WPA3!



Management Frame Protection (MFP)

Provides confidentiality, integrity, and replay protection for:

- Deauthentication frames
- Disassociation frames
- Robust action frames

Deauthentication and Disassociation frames: Prevent Denial-of-Service attacks

Management frame protection has exisisted since 2009 (!!)

- Defined in 802.11w amendment and optional under WPA2
- Rarely used because supported was low & often buggy

Attacking WPA3:
New Vulnerabilities And Exploit Framework
- Mathy Vanhoef

https://www.youtube.com/watch?v=MWaIhYaQuM8
https://www.youtube.com/watch?v=tRWMp3jXlRg
https://www.youtube.com/watch?v=44I1wfgGT80

Announcements

- Quiz-2 paper distribution
  - April 1st
- Quiz-3
  - April 30th morning session
  - Topics: HTTPS, IPSEC, DNSSEC, Wi-Fi Security
- Secure-chat assignment evaluations
  - April 2nd week
  - Contact TAs for the slot assignment

References

- IEEE 802.11 Stds: http://standards.ieee.org/about/get/802/802.11.html
  - 802.11i and 802.11w
- https://code.google.com/archive/p/wifuzz/wikis/WiFuzz.wiki
- http://www.secdev.org/projects/scapy/
- https://www.eetimes.com/document.asp?doc_id=1206324
- https://www.krackattacks.com/
- https://www.aircrack-ng.org/
- https://thebestvpn.uk/unsecured-wifi-network/
- https://asecuritysite.com/encryption/
- https://networkwizkid.com/2019/11/16/capturing-eapol-and-radius-using-wireshark/
- https://witestlab.poly.edu/blog/conduct-a-simple-man-in-the-middle-attack-on-a-wifi-hotspot/
- https://wirelesslywired.com/2017/07/05/following-the-802-1x-aaa-process-with-packet-captures/
- https://whisperlab.org/introduction-to-hacking/lectures/wifi-exploitation
- https://mrncciew.com/2014/08/19/cwsp-ccmp-encryption-method/

WPA2 & WPA3 Attacks (Videos)

- KRACK (2017)
  - https://www.youtube.com/watch?v=Oh4WURZoR98
- YouTube Playlist on WPA2 Attacks
  - https://www.youtube.com/watch?v=fOgJswt7nAc
  - WPA2 Encryption Basics | Part 1 | WPA2 Key Installation KRACK Attacks - YouTube
- FragAttacks (2021)
  - https://www.fragattacks.com/
  - https://www.youtube.com/watch?v=88YZ4061tYw&t=11s
- Dragonblood
  - https://wpa3.mathyvanhoef.com/