भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Wi-Fi Security: Threats ↔ Solutions

Bheemarjuna Reddy Tamma

Dept. of Computer Science & Engineering

IIT Hyderabad

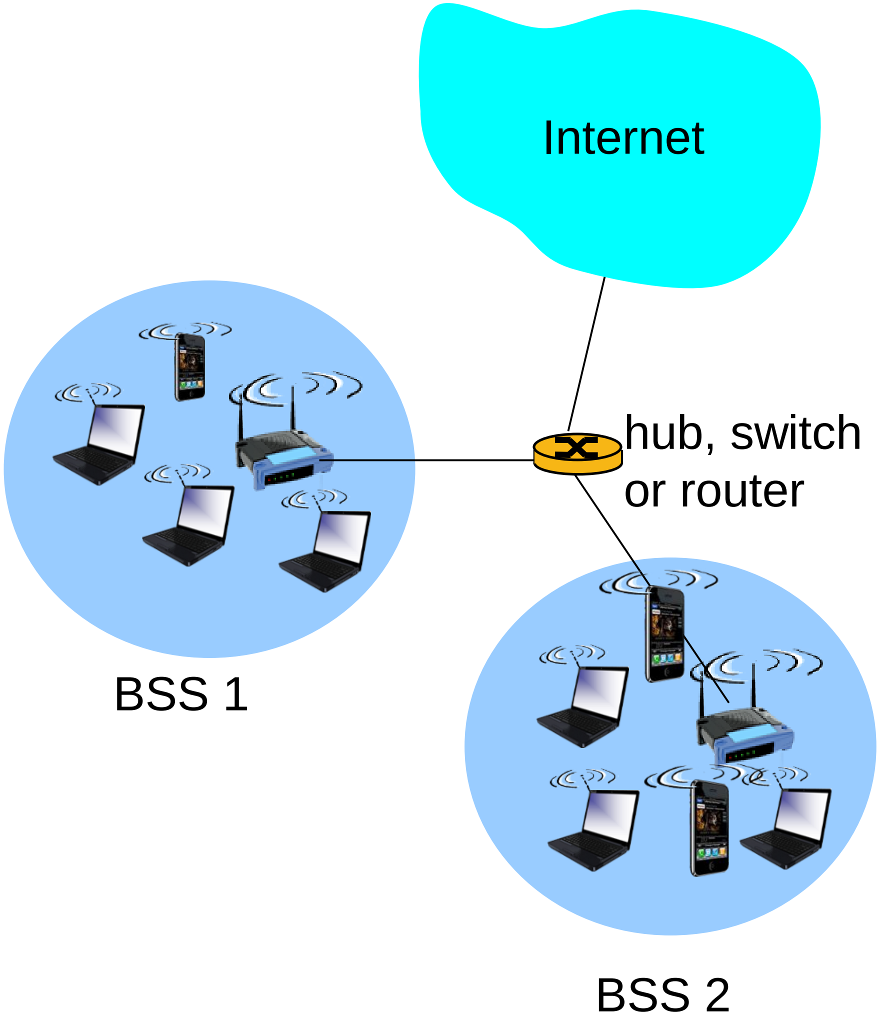
November 3, 2020

One Week Webinar Series on Cloud & Network Security Mechanisms, CSE Dept, IIT (ISM) Dhanbad

Outline

- □ Wi-Fi Architecture
- □ Why Wi-Fi Security is important?
- □ Wi-Fi Security Threats
- □ Wi-Fi Security Standards
- □ Security Vulnerabilities of WPA2
- □ What WPA3 offers?
- □ Wi-Fi Security: Recommendations

802.11 LAN (Wi-Fi) Architecture



- Basic Service Set (BSS) (aka "cell")
  - Building block of IEEE 802.11 WLAN
  - In infrastructure mode, BSS contains:
    - Wireless clients
    - Access Point (AP)

Legacy 802.11 (Wi-Fi) Data Packet

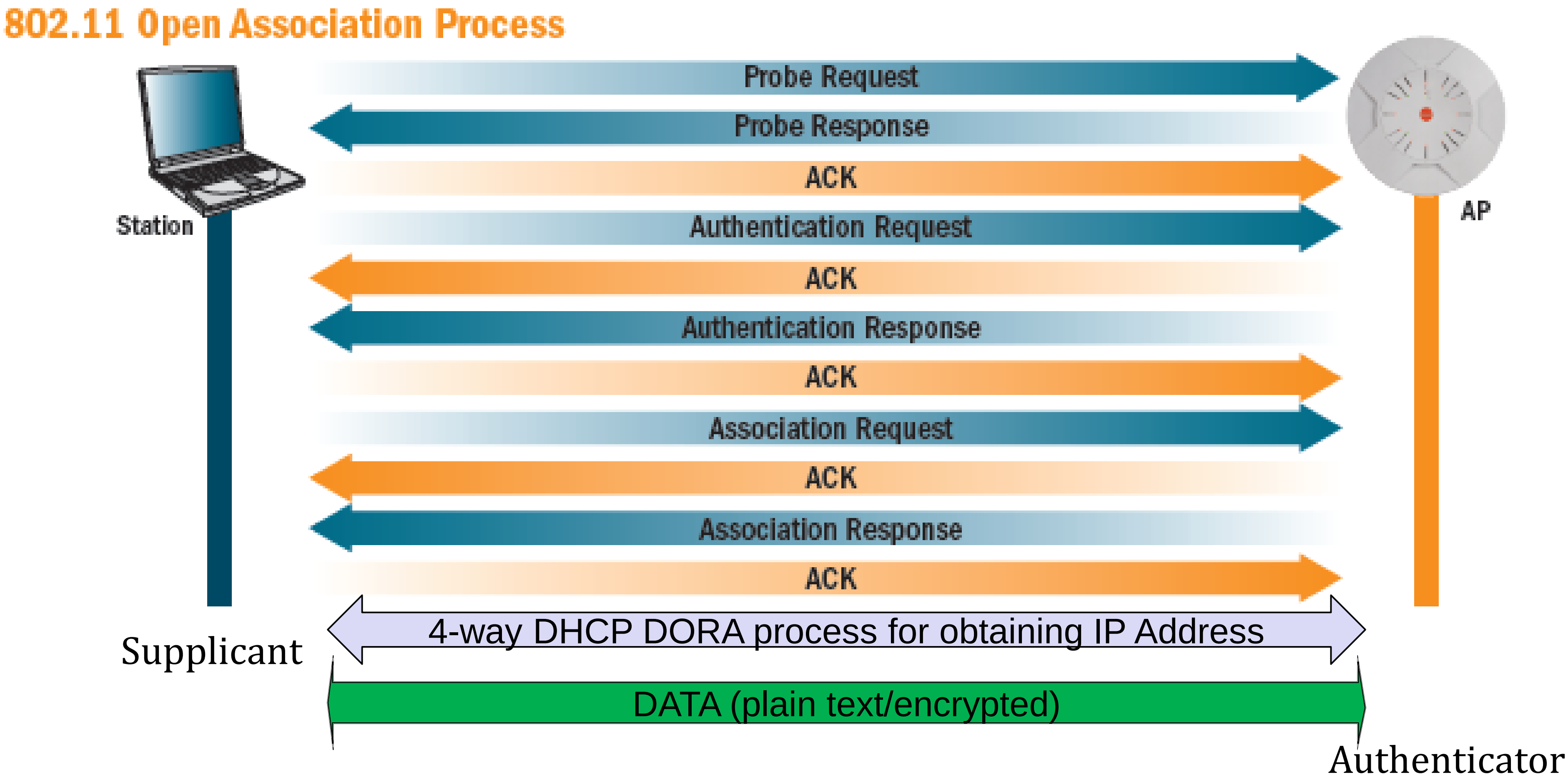
2	2	6	6	6	2	6	0 - 2312	4
frame control	duration	address 1	address 2	address 3	seq control	address 4	payload	CRC

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 IP Packet in plain-text or encryption form

How does a STA join Wi-Fi network ?



Supplicant

4-way DHCP DORA process for obtaining IP Address

DATA (plain text/encrypted)

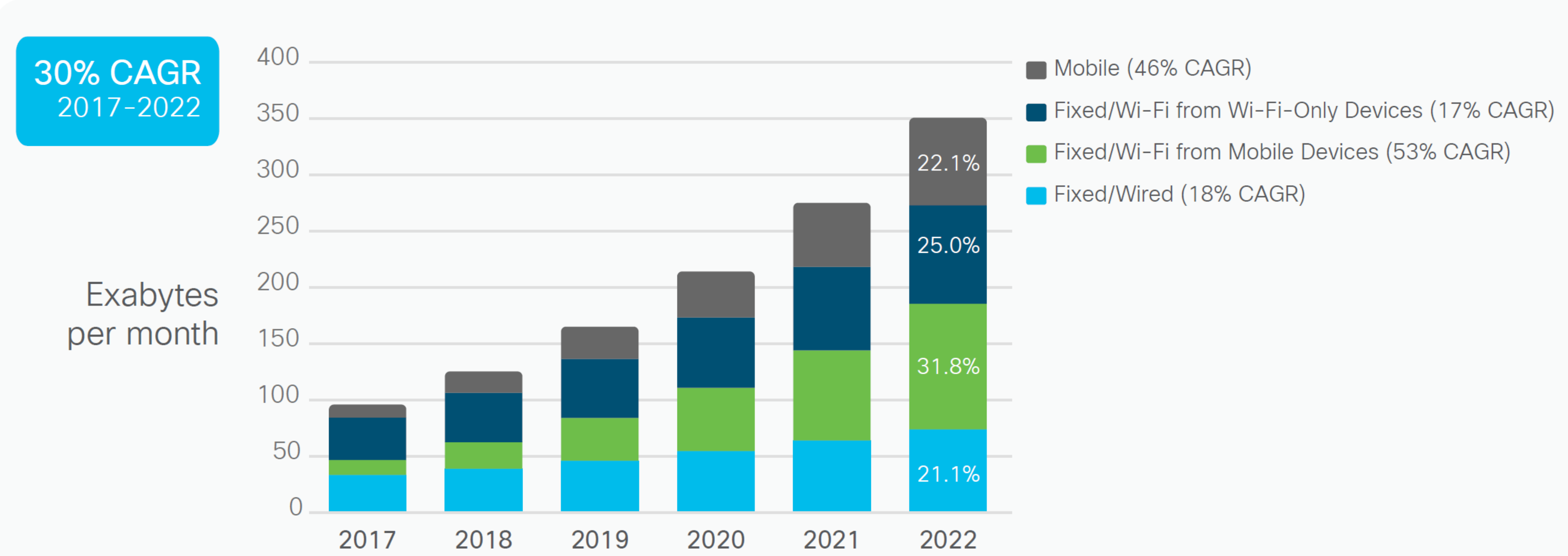
Authenticator

Secure communication requirements

- *confidentiality*: only sender, intended receiver should "understand" message contents
  - sender encrypts message
  - receiver decrypts message
- *authentication:* sender, receiver want to confirm identity of each other
- *message integrity:* sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
- *access and availability*: services must be accessible and available to users

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

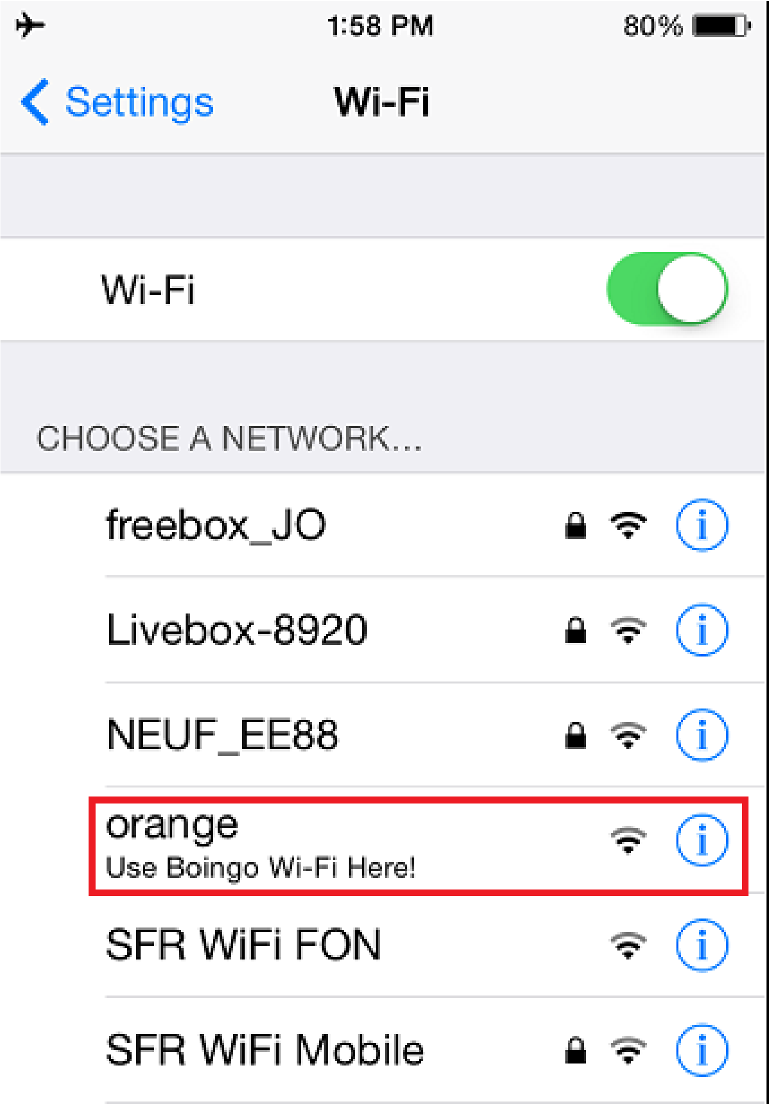
Wi-Fi Security Threats

- □ Eavesdropping
- □ Man-in-the-middle (MITM) attacks
- □ Malicious association to rogue (AP) networks
- □ Denial of Service (DoS) attacks
- □ AP configuration over HTTP

Hacking Wi-Fi Networks

- Tools of the trade
  - Wireshark/Tcpdump
  - Kismet
  - WEPCrack/AirSnort
  - AirCrack-NG
  - 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

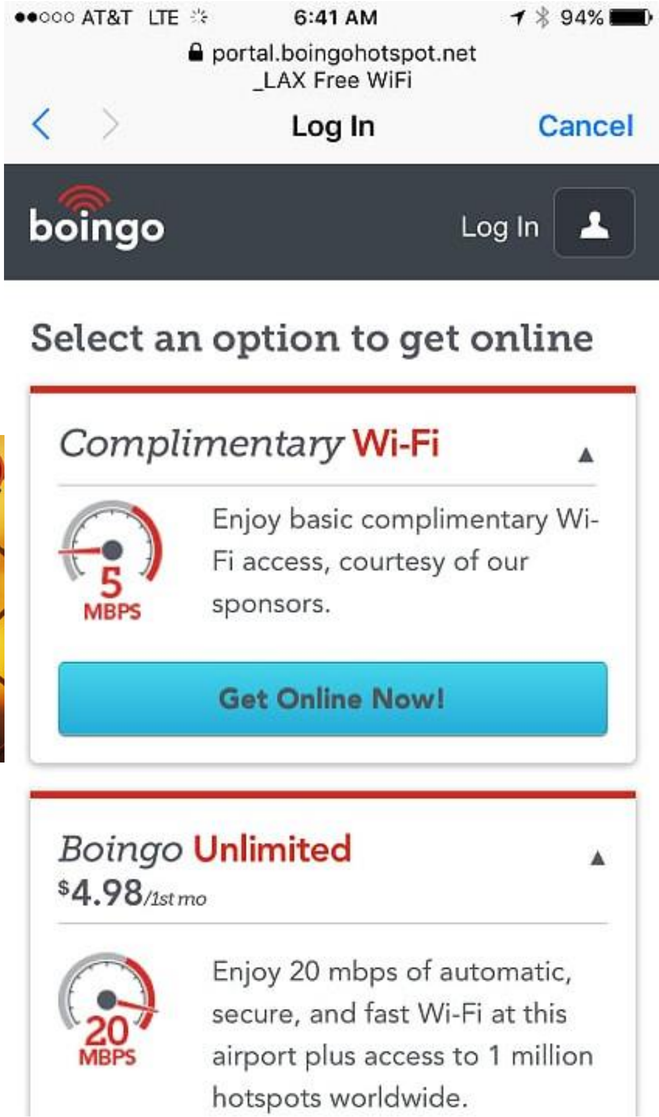
Livebox-8920

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!

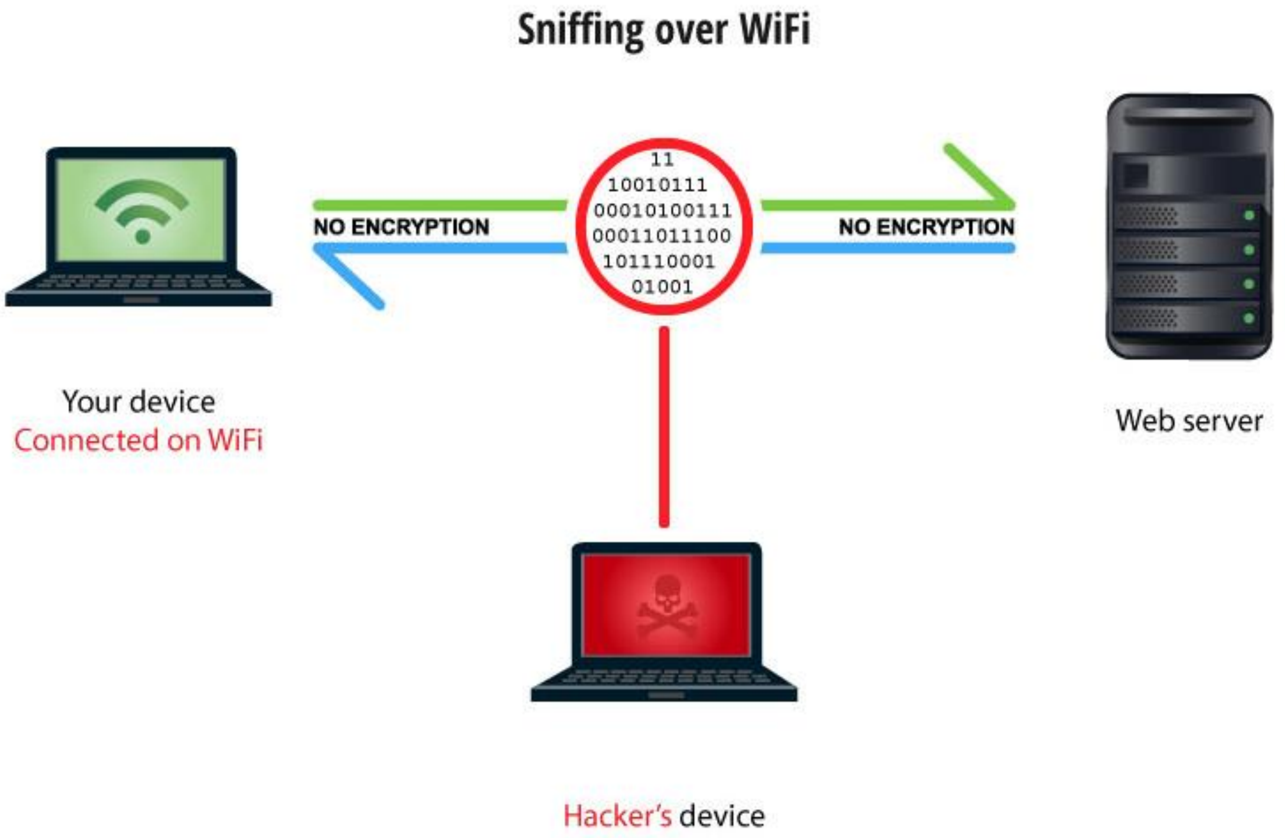
Boingo Unlimited

$4.98/1st mo

20 MBPS

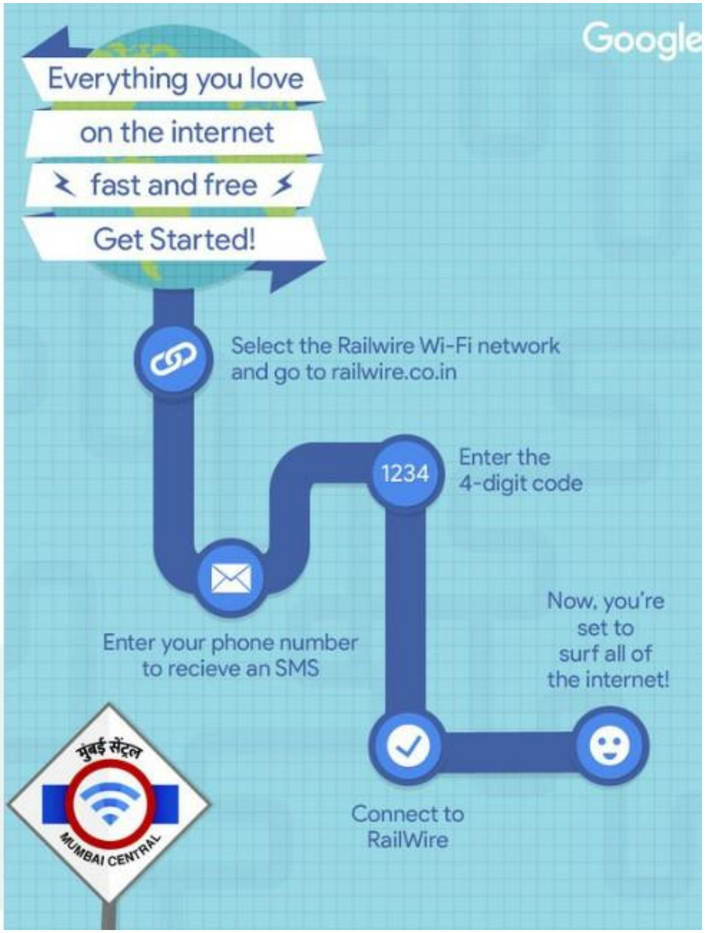
Enjoy 20 mbps of automatic, secure, and fast Wi-Fi at this airport plus access to 1 million hotspots worldwide.

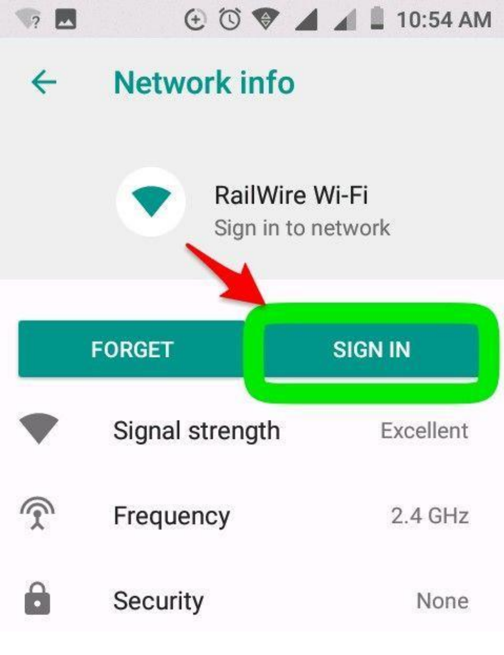
Threat: Eavesdropping on Open Wi-Fi Networks



- □ 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
Enter your phone number to recieve an SMS
1234
Enter the 4-digit code
Now, you're set to surf all of the internet!
Connect to RailWire
मुंबई सेंट्रल
MUMBAI CENTRAL


10:54 AM

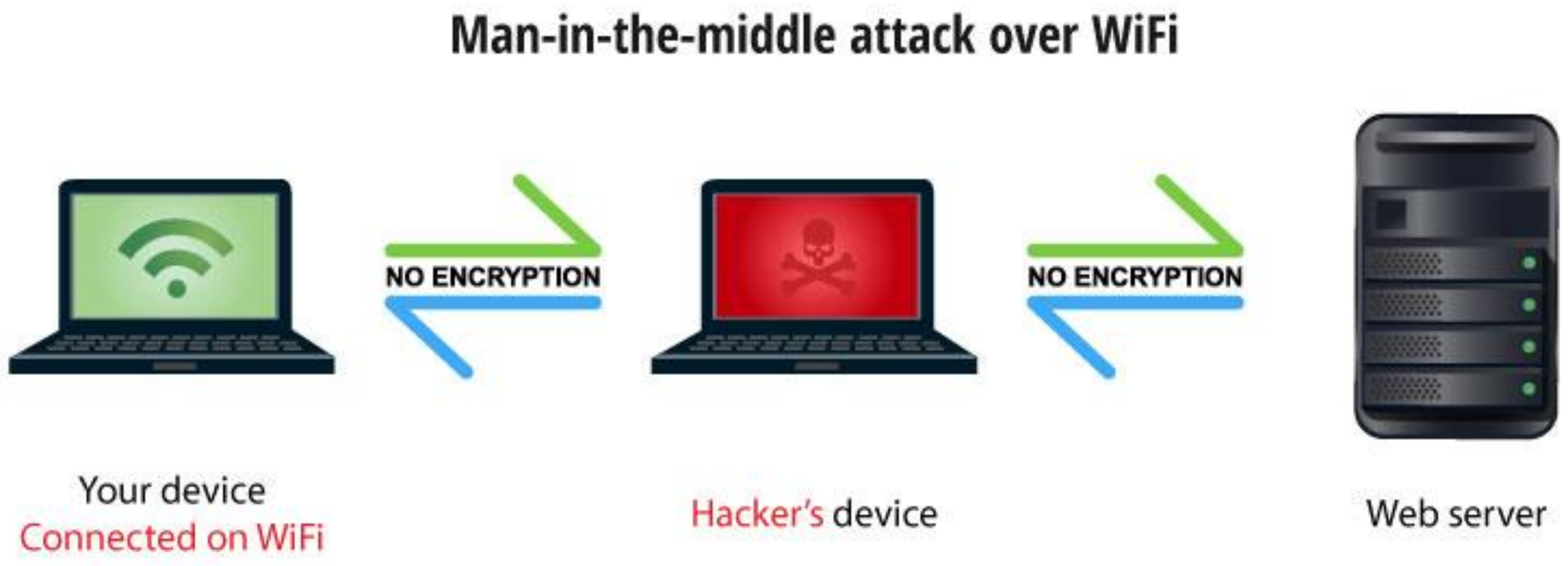
Network info

RailWire Wi-Fi
Sign in to network

FORGET SIGN IN

Signal strength	Excellent
Frequency	2.4 GHz
Security	None

Threat: MITM attacks in Wi-Fi



- Rogue APs with SSID of legitimate Wi-Fi networks
- Malicious Hotspots: Free, open networks that snoop into data sent/received
- Affects confidentiality and integrity of data exchanged

Demo of MITM Attack

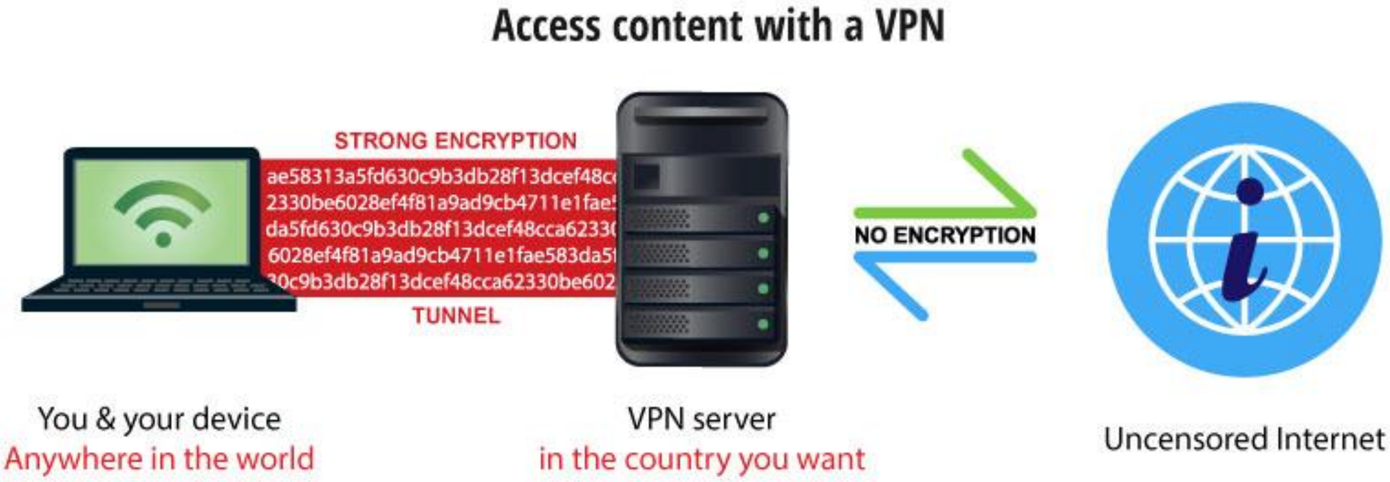
How to stay safe on public Wi-Fi?

√ DO:

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

https://

- Turn OFF file sharing



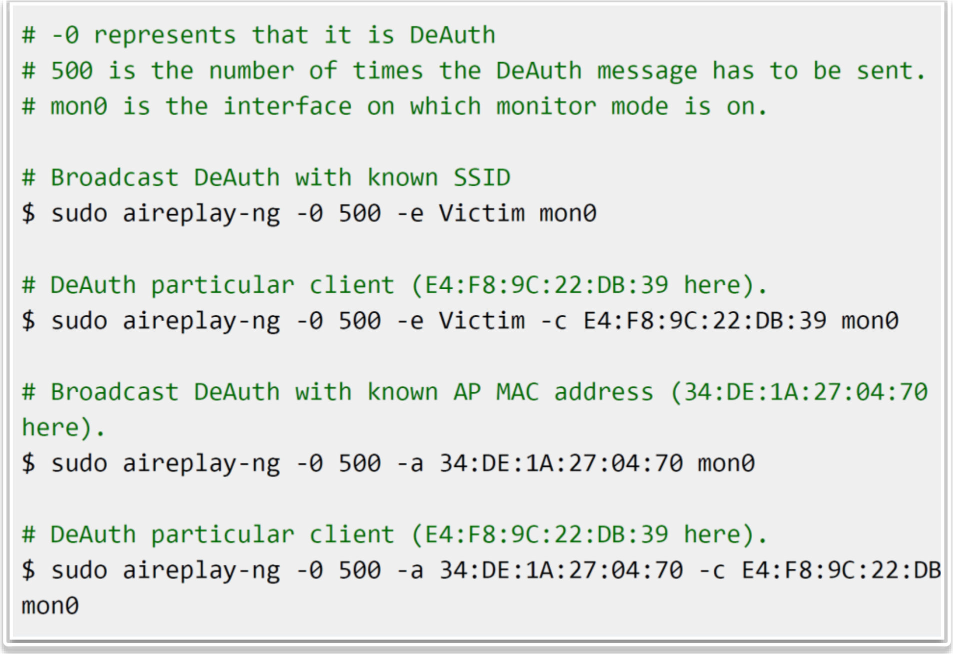
Access content with a VPN
STRONG ENCRYPTION
ae58313a5fd630c9b3db28f13dcef48c
2330be6028ef4f81a9ad9cb4711e1fae
da5fd630c9b3db28f13dcef48cca6233
6028ef4f81a9ad9cb4711e1fae583da5
0c9b3db28f13dcef48cca62330be602
TUNNEL
NO ENCRYPTION
You & your device
Anywhere in the world
VPN server
in the country you want
Uncensored Internet

How to stay safe on public Wi-Fi?

- Don’t:
  - Allow your Wi-Fi to auto-connect to open networks
  - Log into any account via 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
  - Access websites that hold your sensitive information, such as bank or healthcare accounts

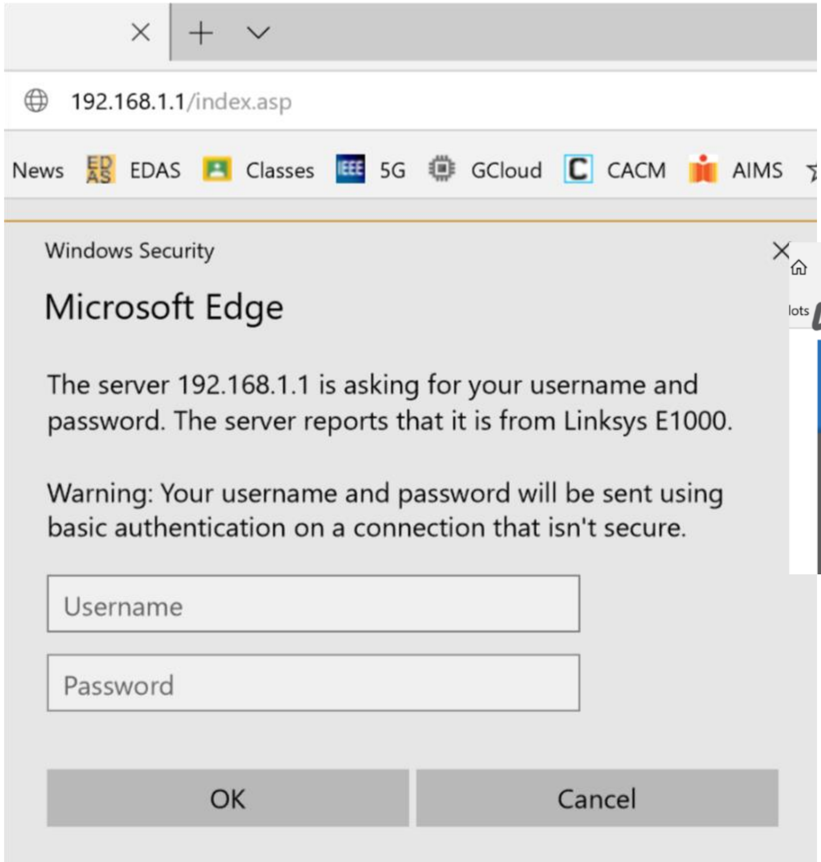
Threat: Denial of Service (DoS) attacks

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



```
# -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
```

Open AP configuration over HTTP



Windows Security

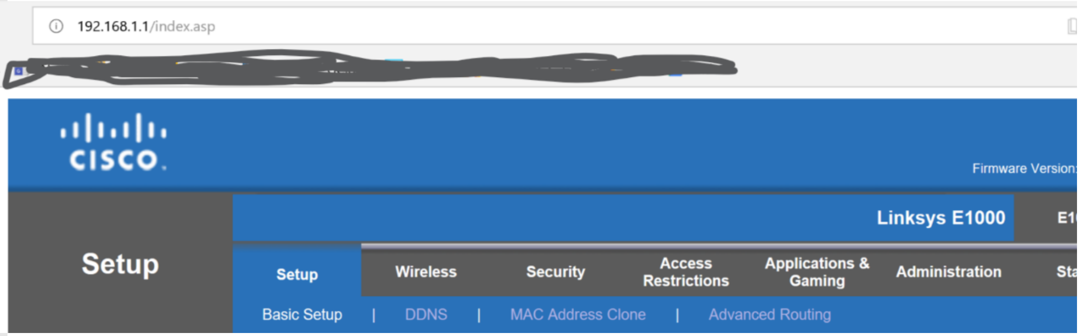
Microsoft Edge

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



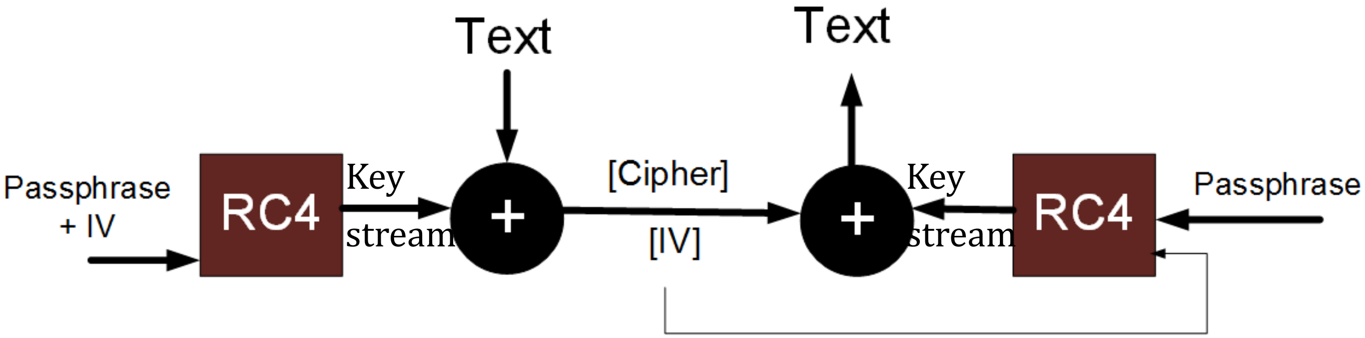
[redacted]

Wi-Fi Security Standards

- 1997→Wired Equivalent Privacy (WEP)
- 2003→Wireless Protected Access (WPA)
- 2004→WPA2 (IEEE 802.11i)
- 2019→WPA3 (Some products having Wi-Fi 6 radios 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)

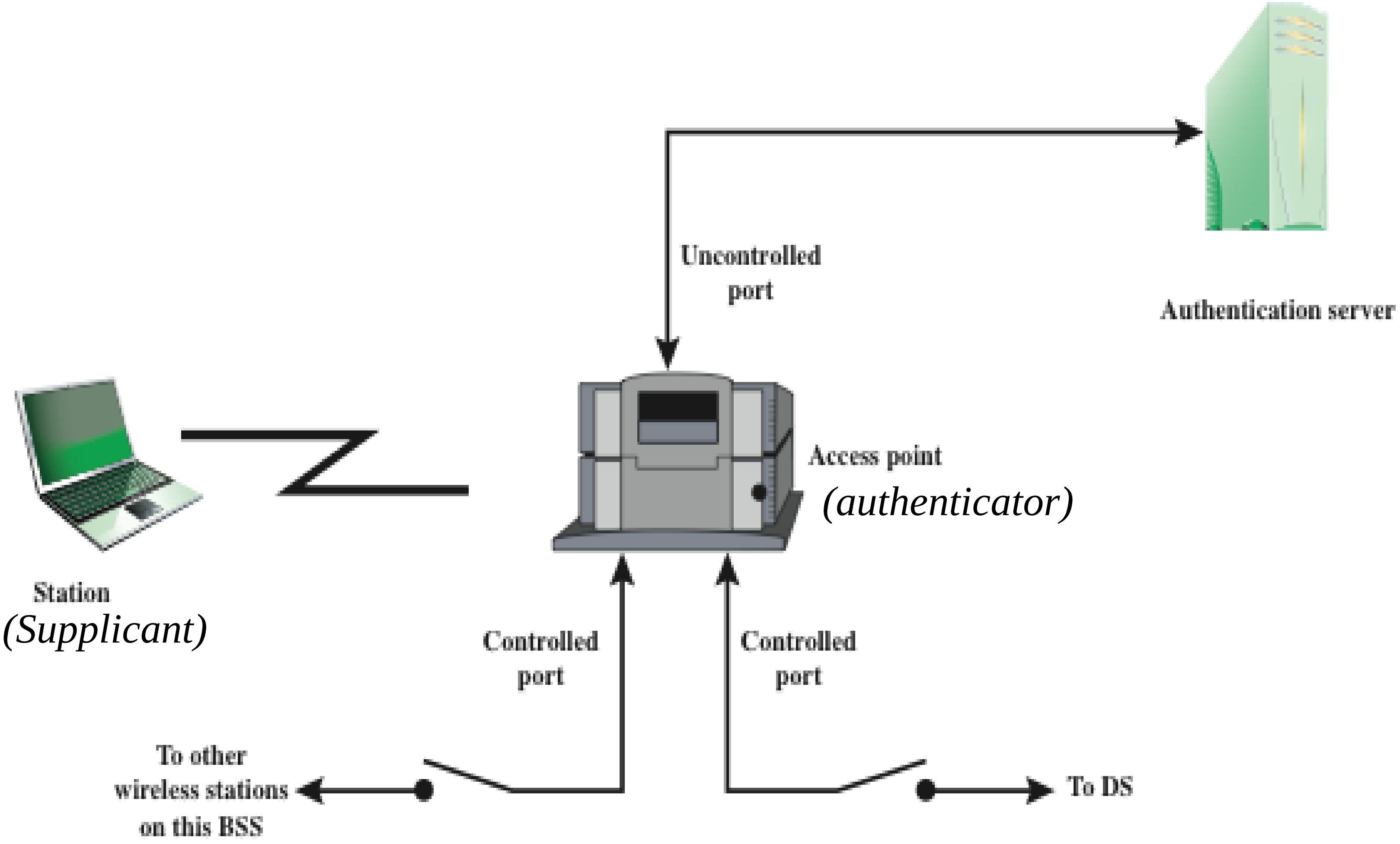
http://www.dartmouth.edu/~madory/RC4/wepexp.txt
https://asecuritysite.com/encryption/rc4_wep

WPA2

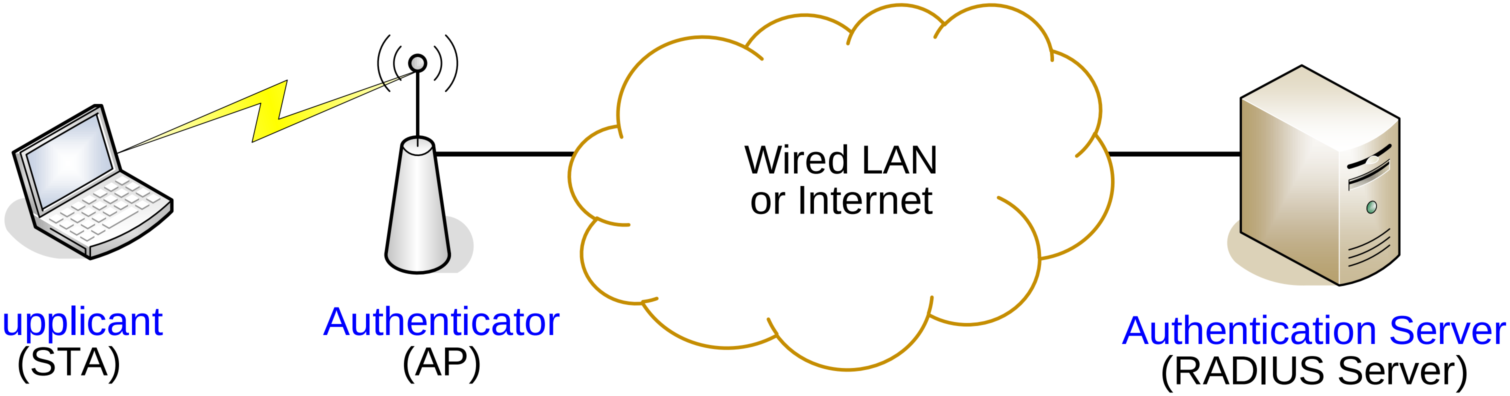
- Wireless Protected Access 2 (WPA2)
  - WPA2 is Wi-Fi alliance name for 802.11i amendment
  - Uses 802.1X for access control
  - Uses Extensible Authentication Protocol (EAP) for authentication and key exchange, e.g., EAP-TLS, EAP-PEAP
  - Confidentiality and integrity protocol: AES-CCMP

- Historical: WPA
  - Used in the transition period before the 11i standard was finalized and before AES support in NIC hardware
  - TKIP encryption = RC4 with frequently changing keys and other enhancements
  - Security of TKIP and WPA is now considered broken; always disable them in your (old) AP!

802.1X Access Control in WPA2

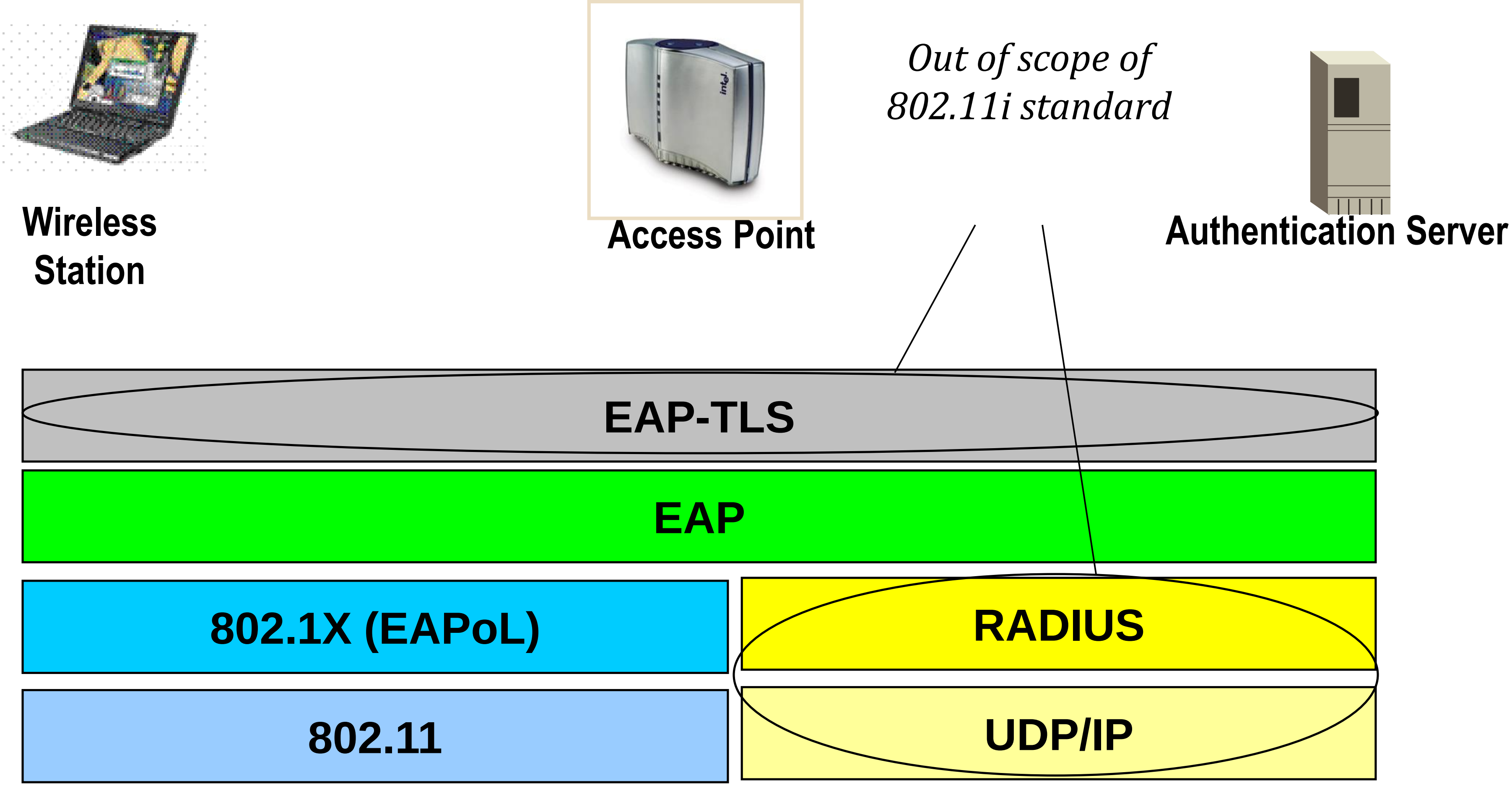


WPA2/802.1X architecture



- 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



Out of scope of 802.11i standard

Wireless Station

Access Point

Authentication Server

EAP-TLS

EAP

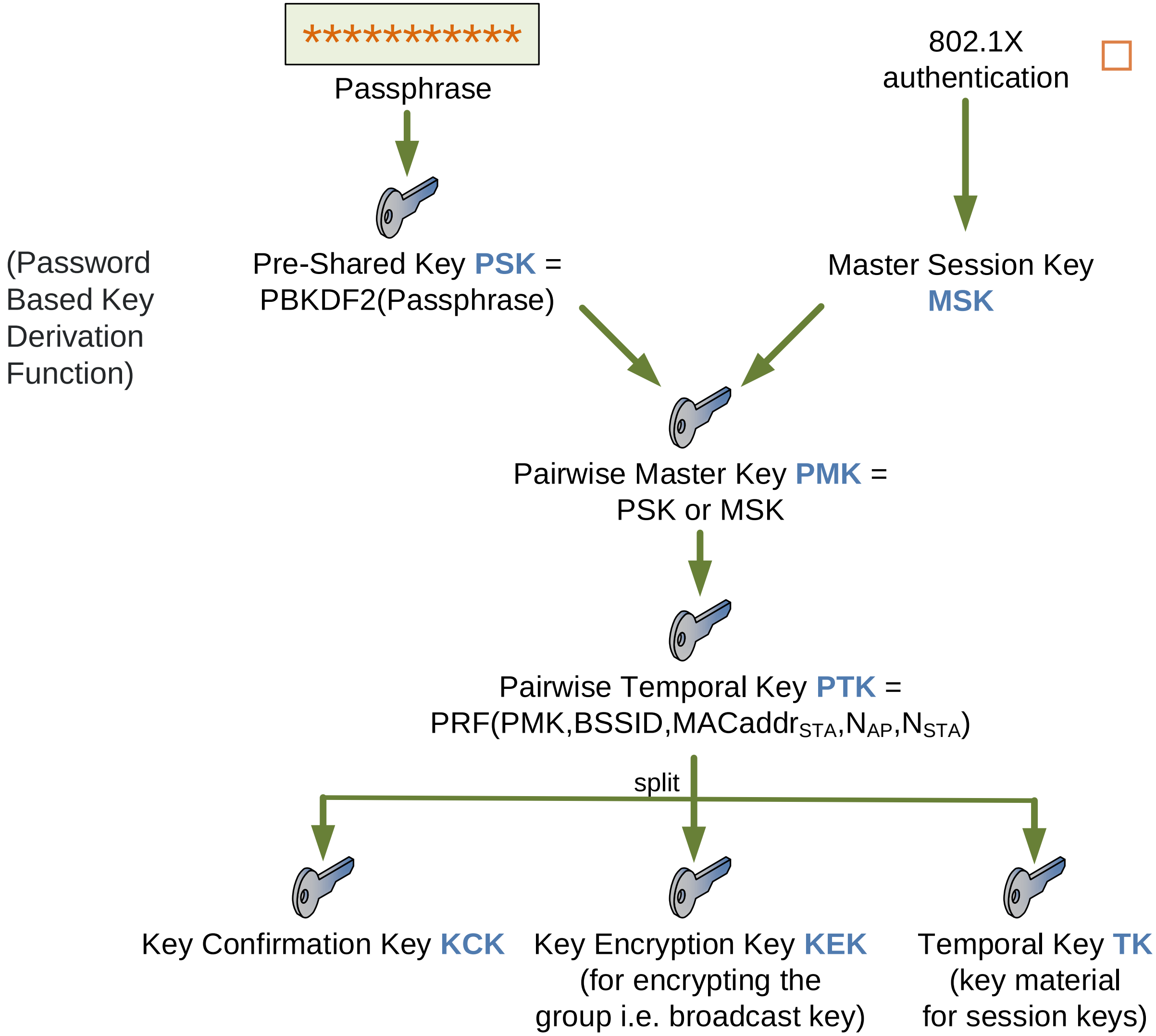
802.1X (EAPoL)

RADIUS

802.11

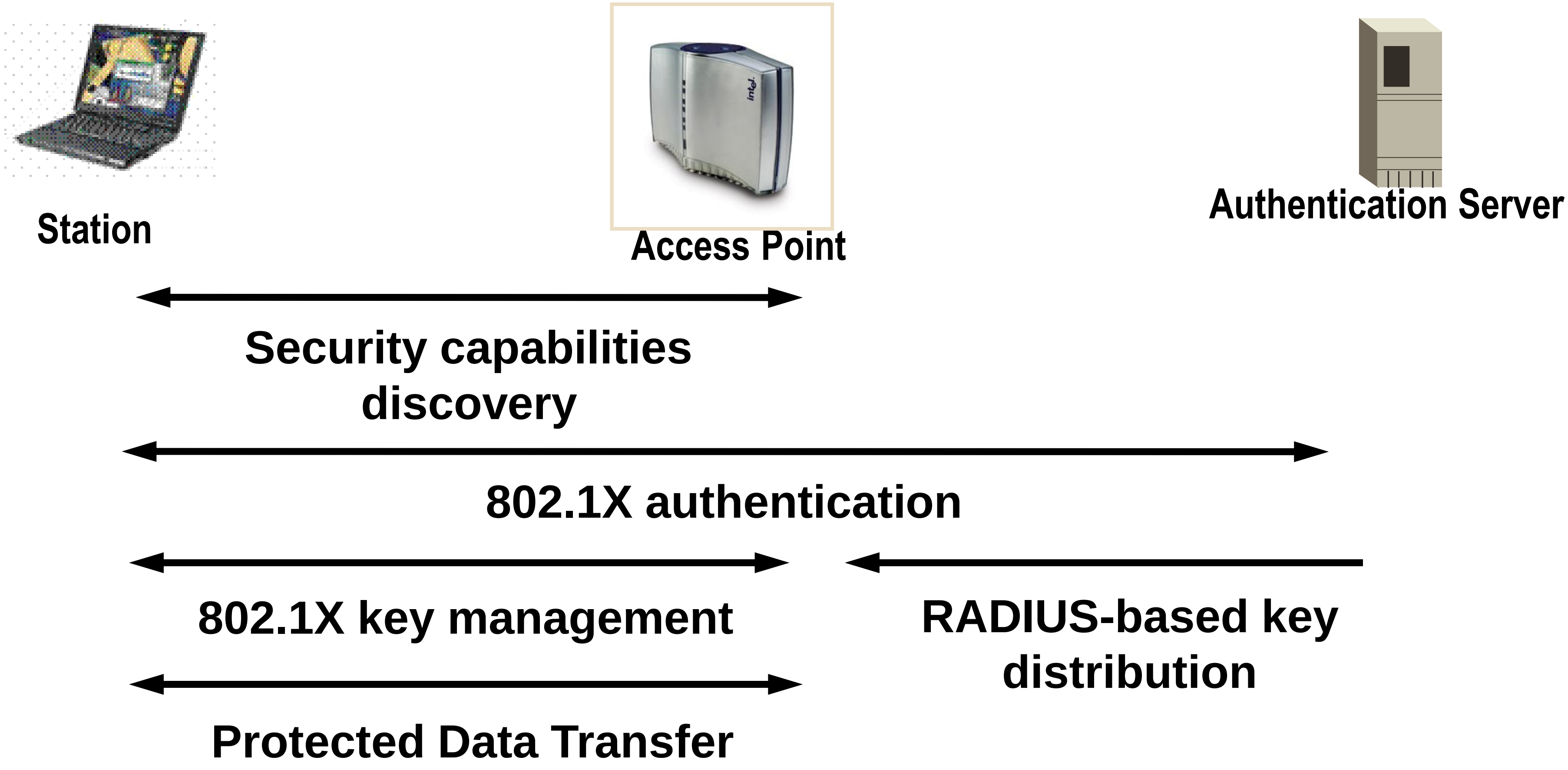
UDP/IP

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: Operational Phases



Station

Access Point

Authentication Server

Security capabilities discovery

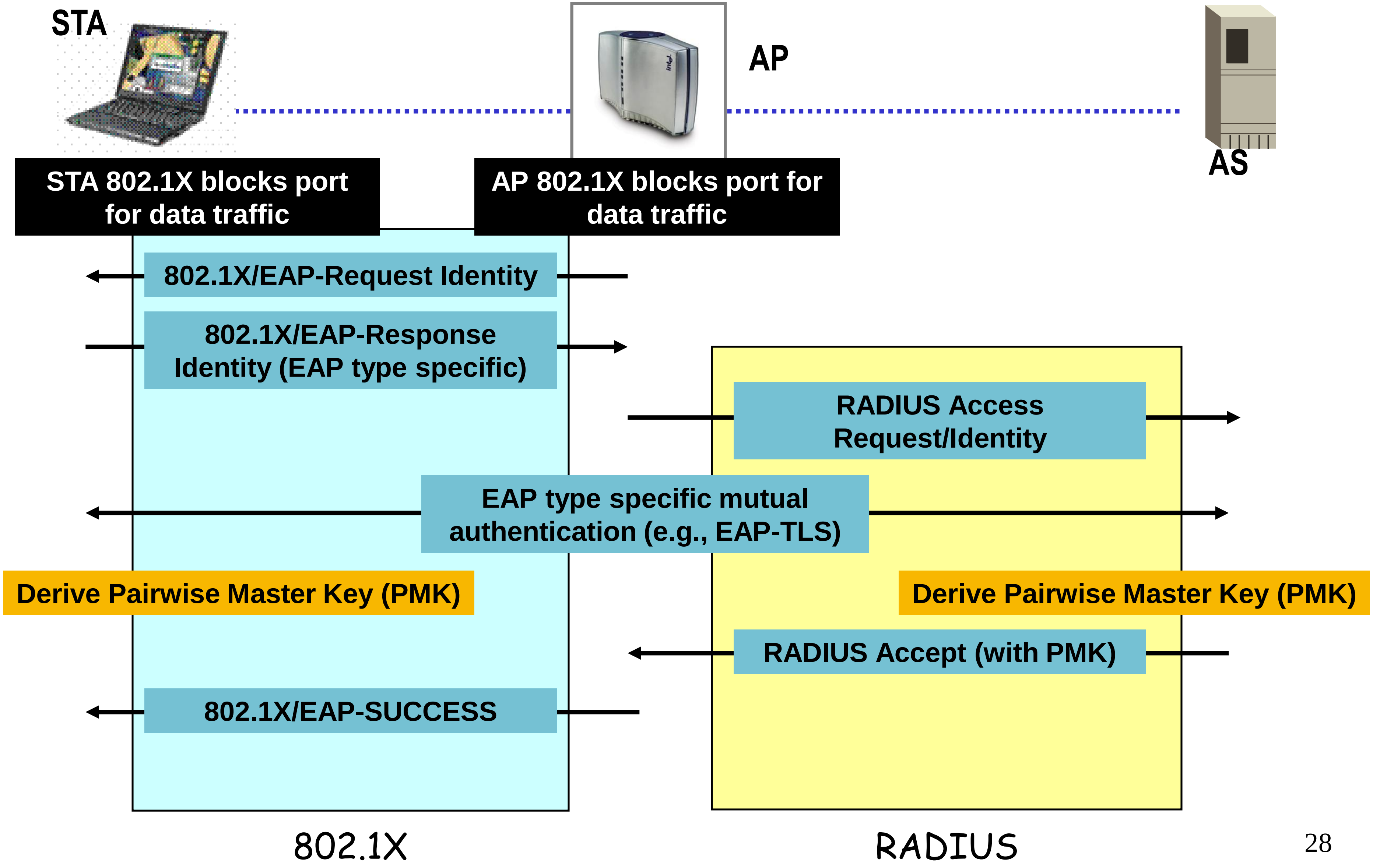
802.1X authentication

802.1X key management

RADIUS-based key distribution

Protected Data Transfer

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)

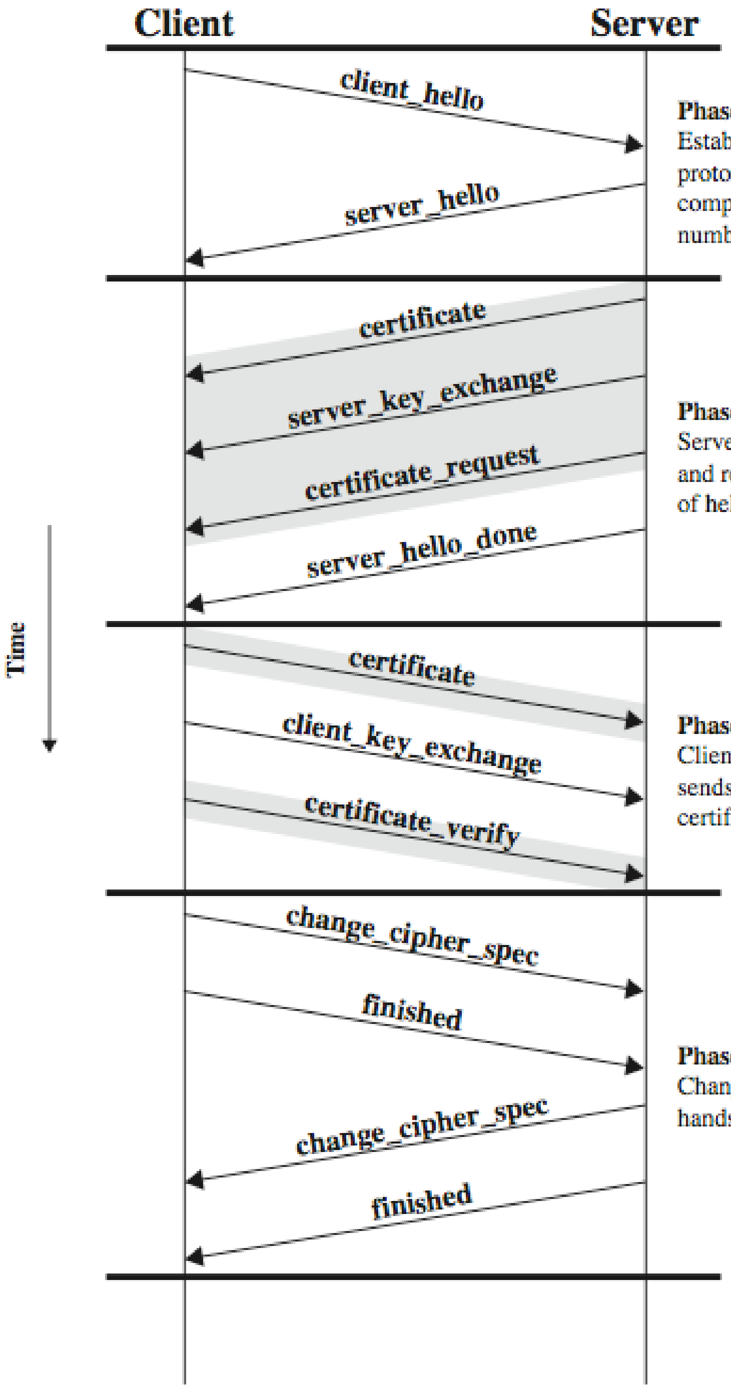
RADIUS Accept (with PMK)

802.1X/EAP-SUCCESS

802.1X

RADIUS

Two-Way TLS/SSL Handshaking



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

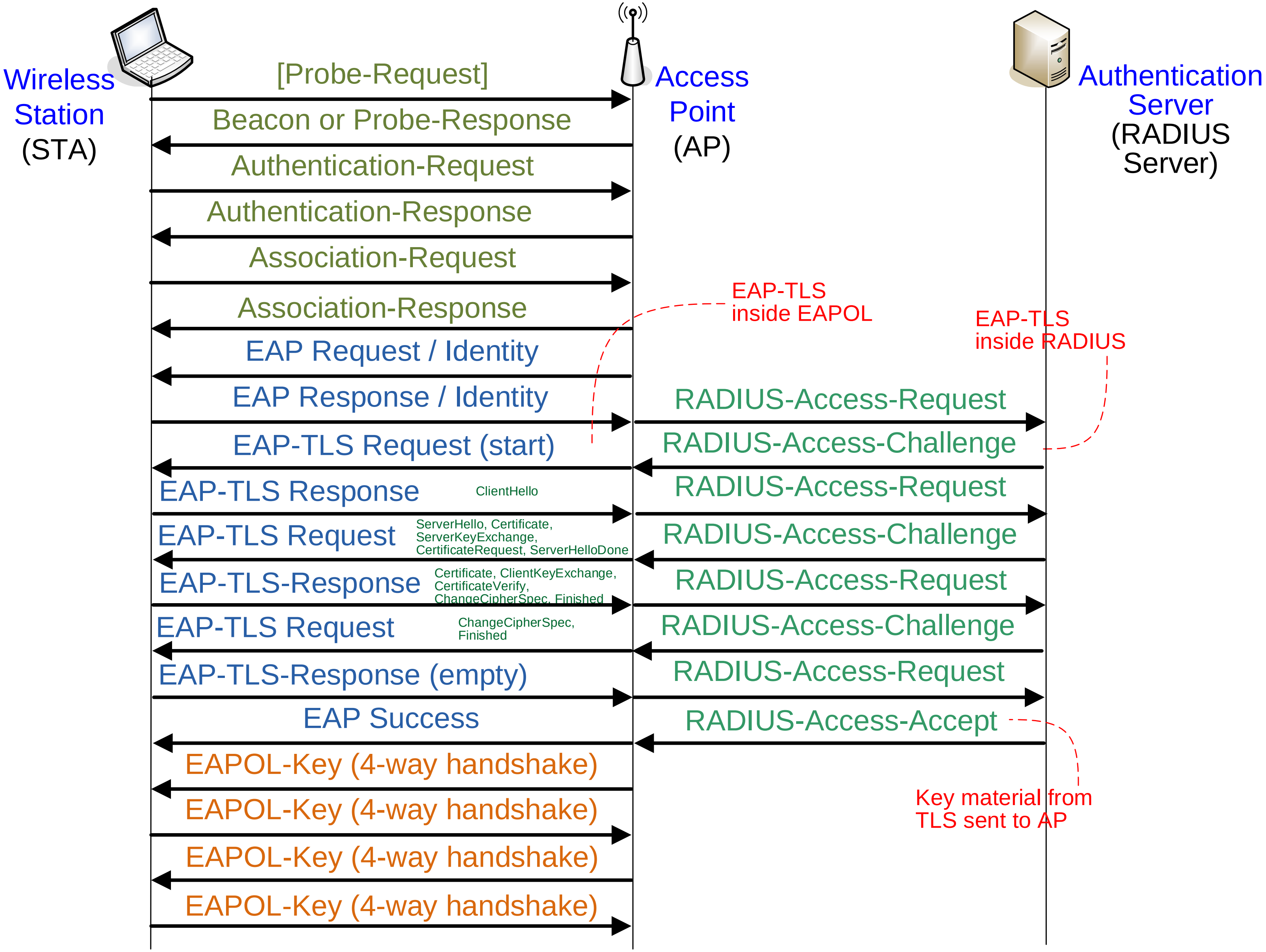
Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

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)

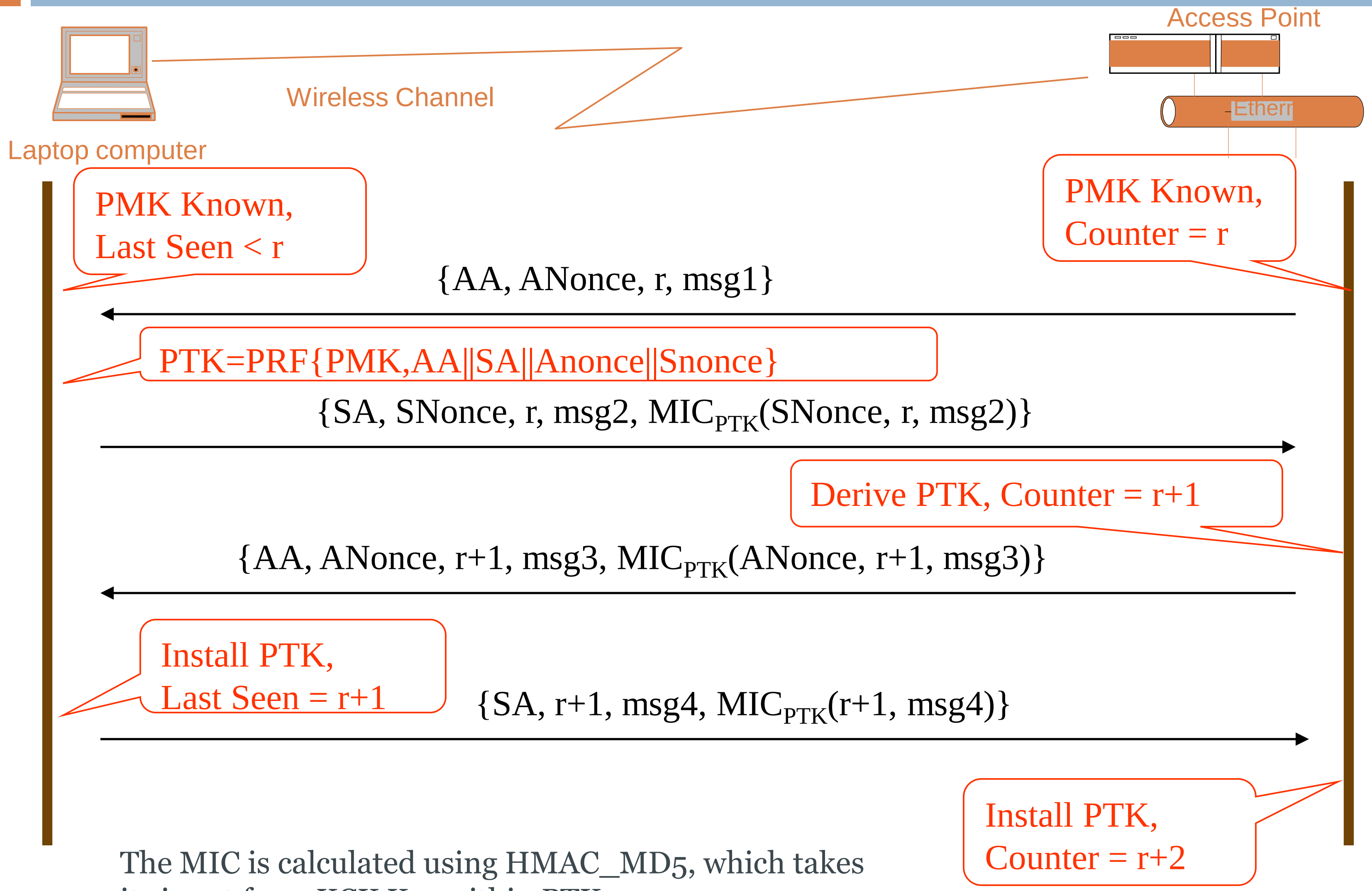
EAPOL-Key (4-way handshake)

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)}

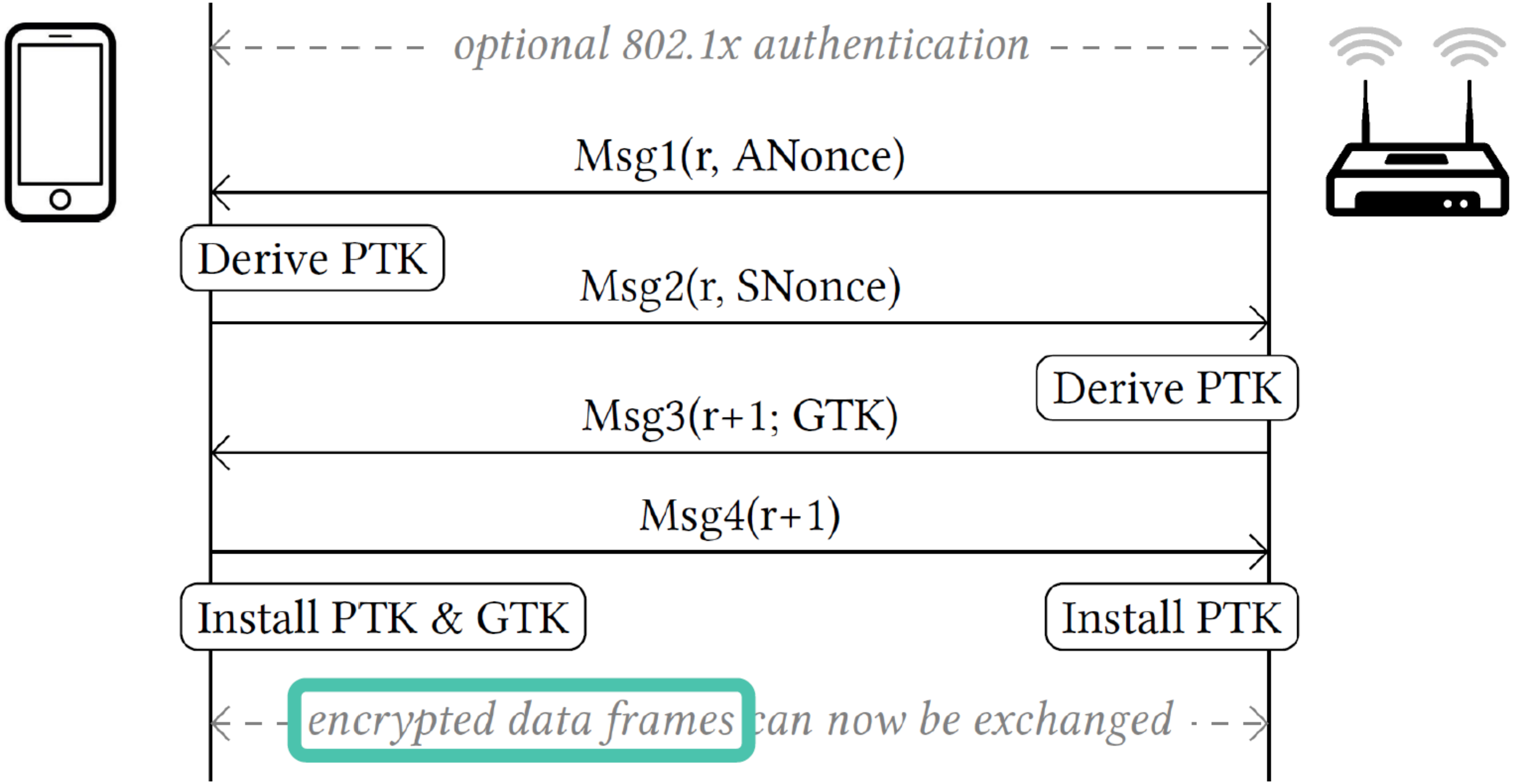
Install PTK,
Last Seen = r+1

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

Install PTK,
Counter = r+2

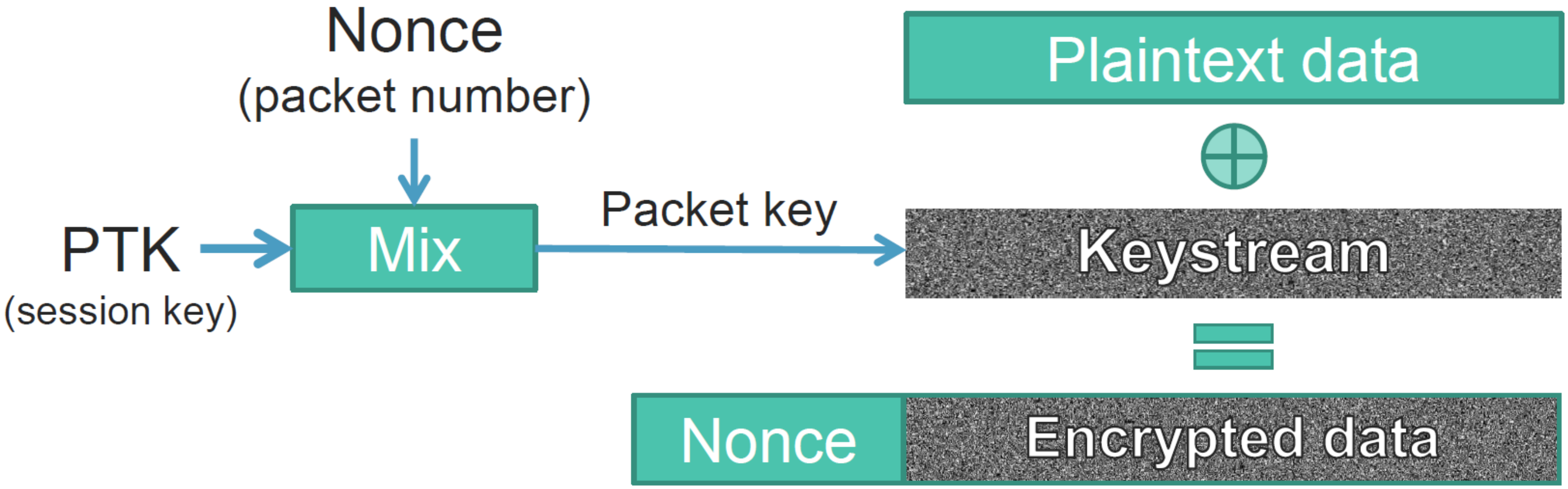
The MIC is calculated using HMAC_MD5, which takes its input from KCK Key within PTK.

WPA2-PSK/EAP: 4-Way Handshake



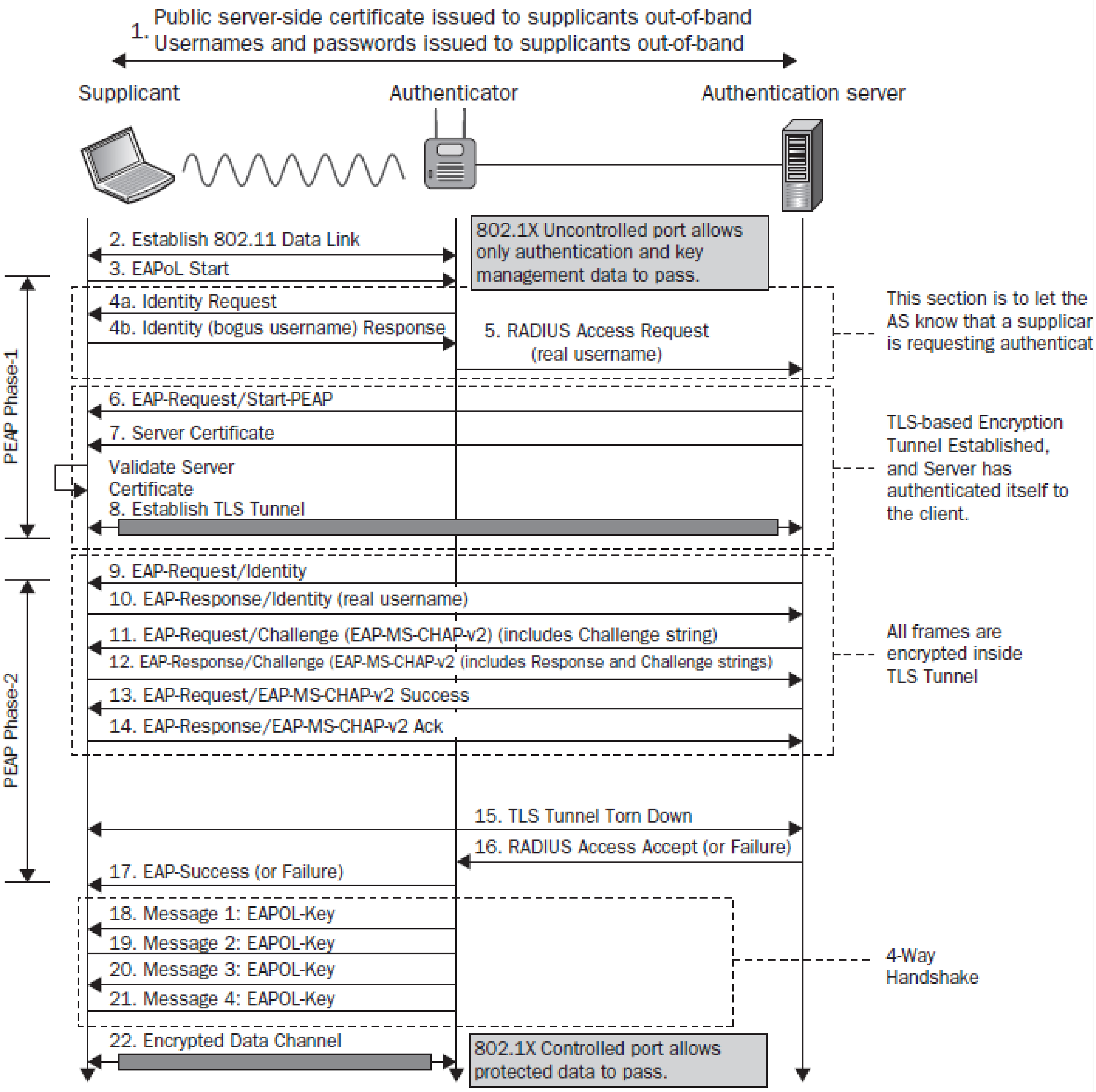
Both WPA2-PSK & EAP make use of AES-CCMP to encrypt data

Encryption of 802.11 MAC Payloads



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

Full WPA2 Authentication (PEAP) & Key Exchange



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
PEAP Phase-1
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-2
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)
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.

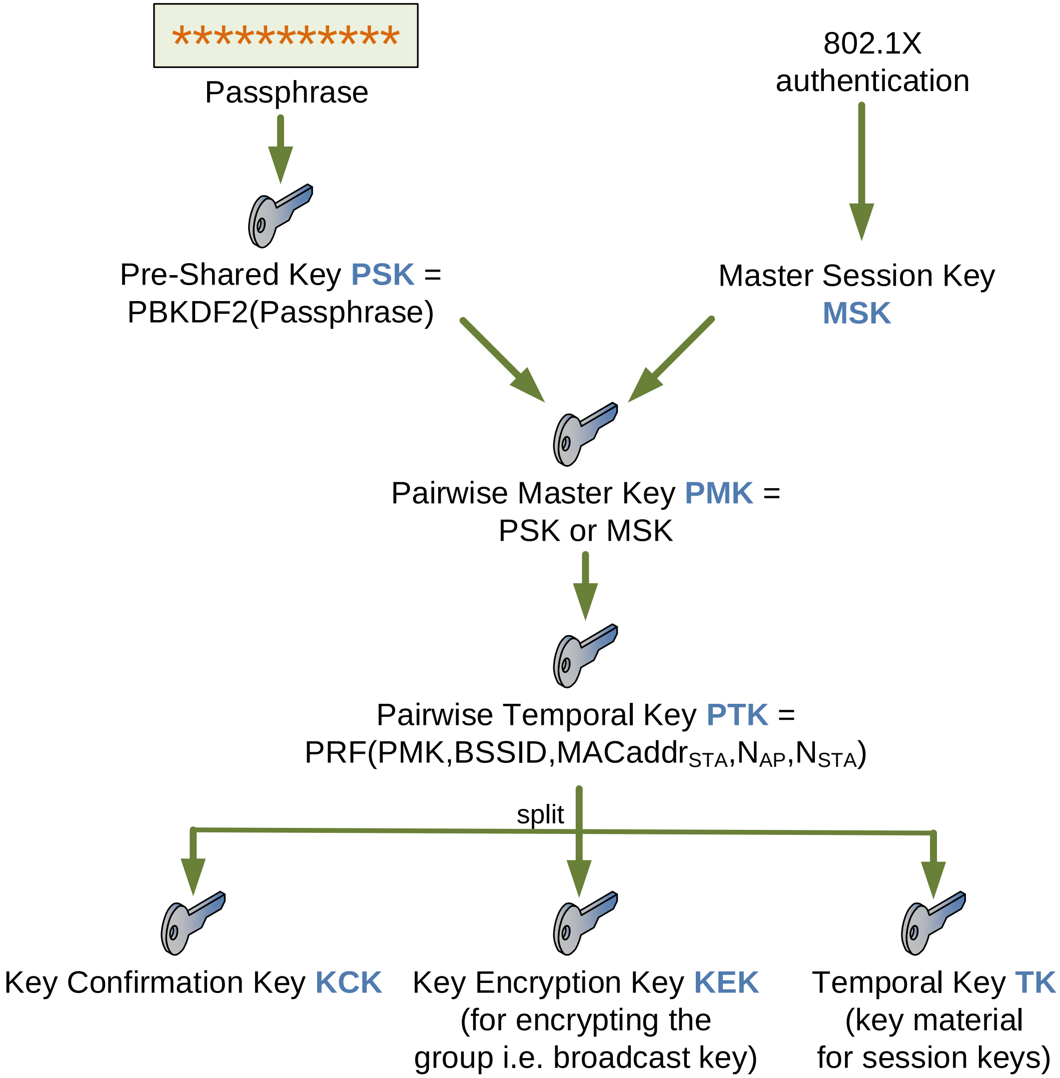
https://mrncciew.com/2014/08/25/cwsp-eap-peap/

Security issues with WPA2

- Sniffing (esp OPEN networks)
- WPA2-PSK:MITM attacks
  - Rogue/malicious AP association
- WAP2-PSK: Offline dictionary attacks
- WPA2-PSK/EAP:- KRACK
- AP configuration over HTTP
- Denial of Service (DoS) attacks

WPA2-PSK OFFLINE DICTIONARY ATTACK

WPA2: Key Hierarchy (recap)


****
Passphrase
802.1X authentication
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)

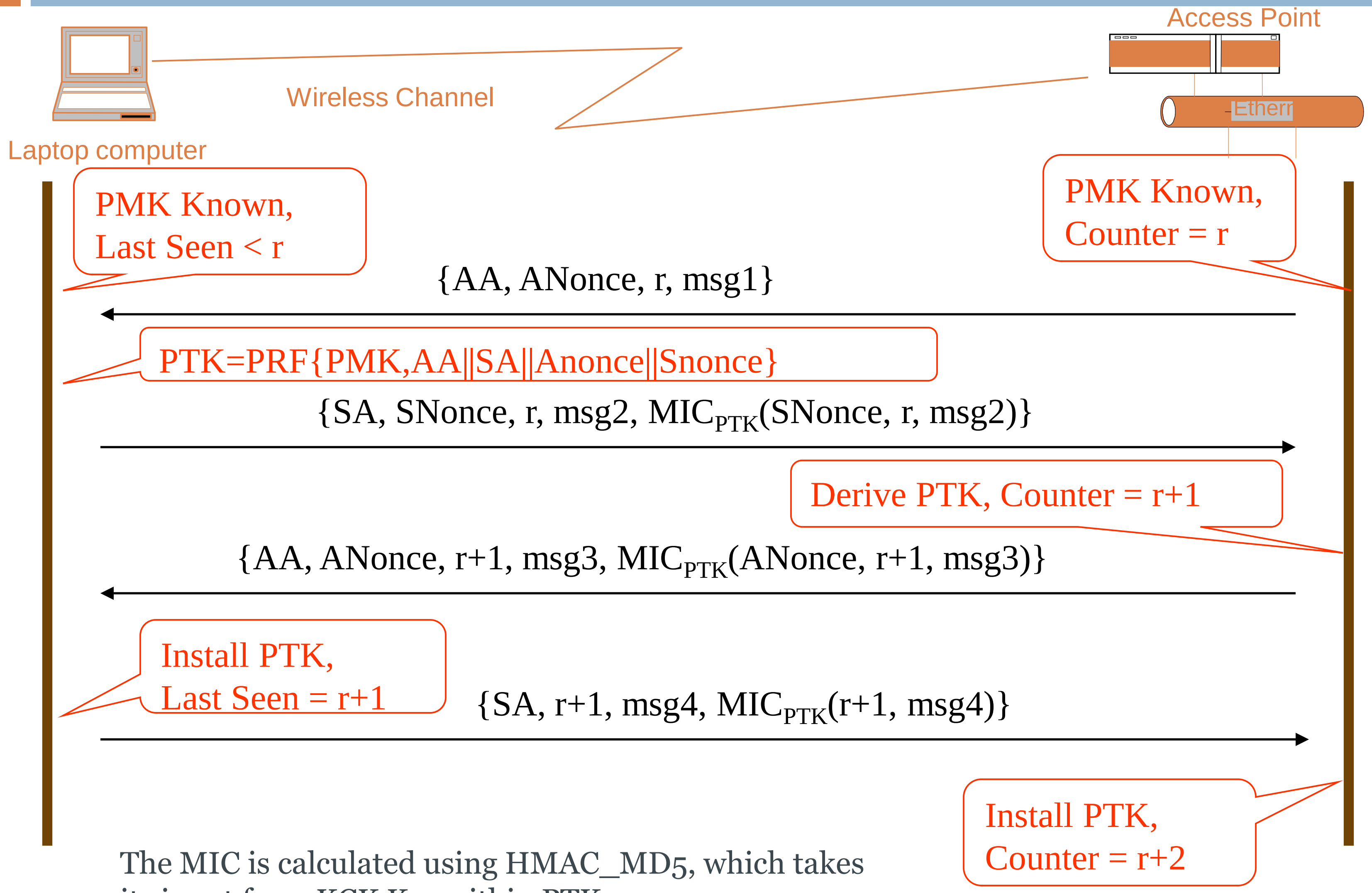
PBKDF2=Password Based Key Derivation Function #2

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)}

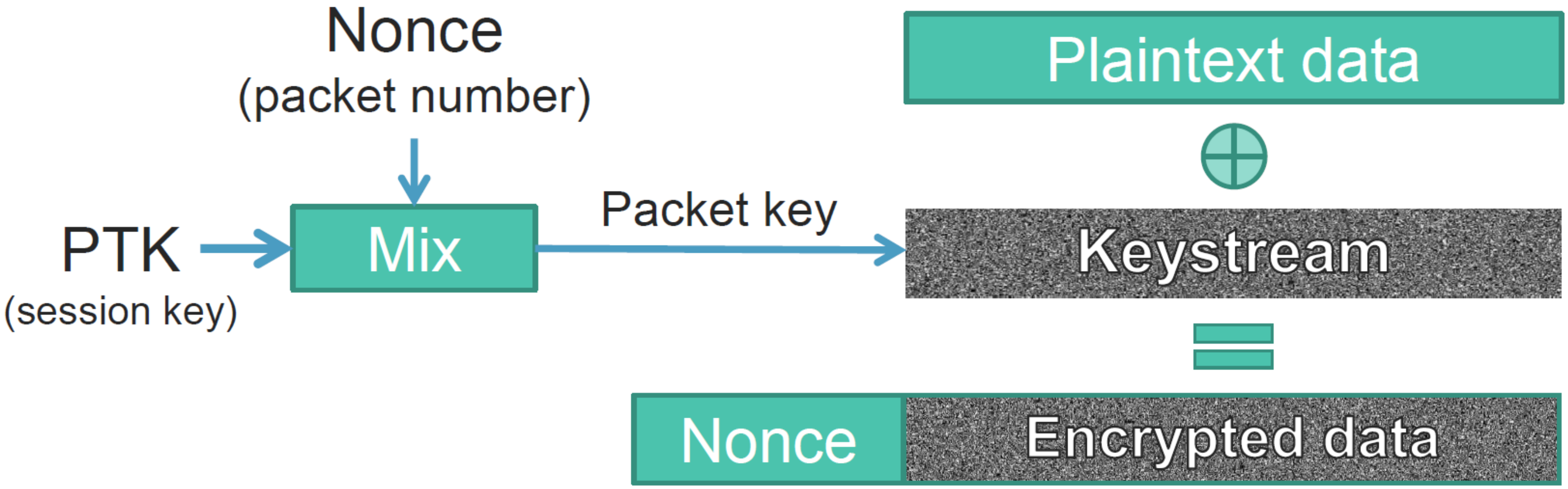
Install PTK,
Counter = r+2

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

KRACK: Key Reinstallation Attacks on WPA2

- Discovered by Mathy Vanhoef, KU Leuven in 2017
- Kind of weakness/ambiguity in .11i std, so effects vary across OS implementations
- So, many devices with Wi-Fi radio were affected
  - Linux and Android 6.0 or higher are highly vulnerable
  - All data from victim can 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.

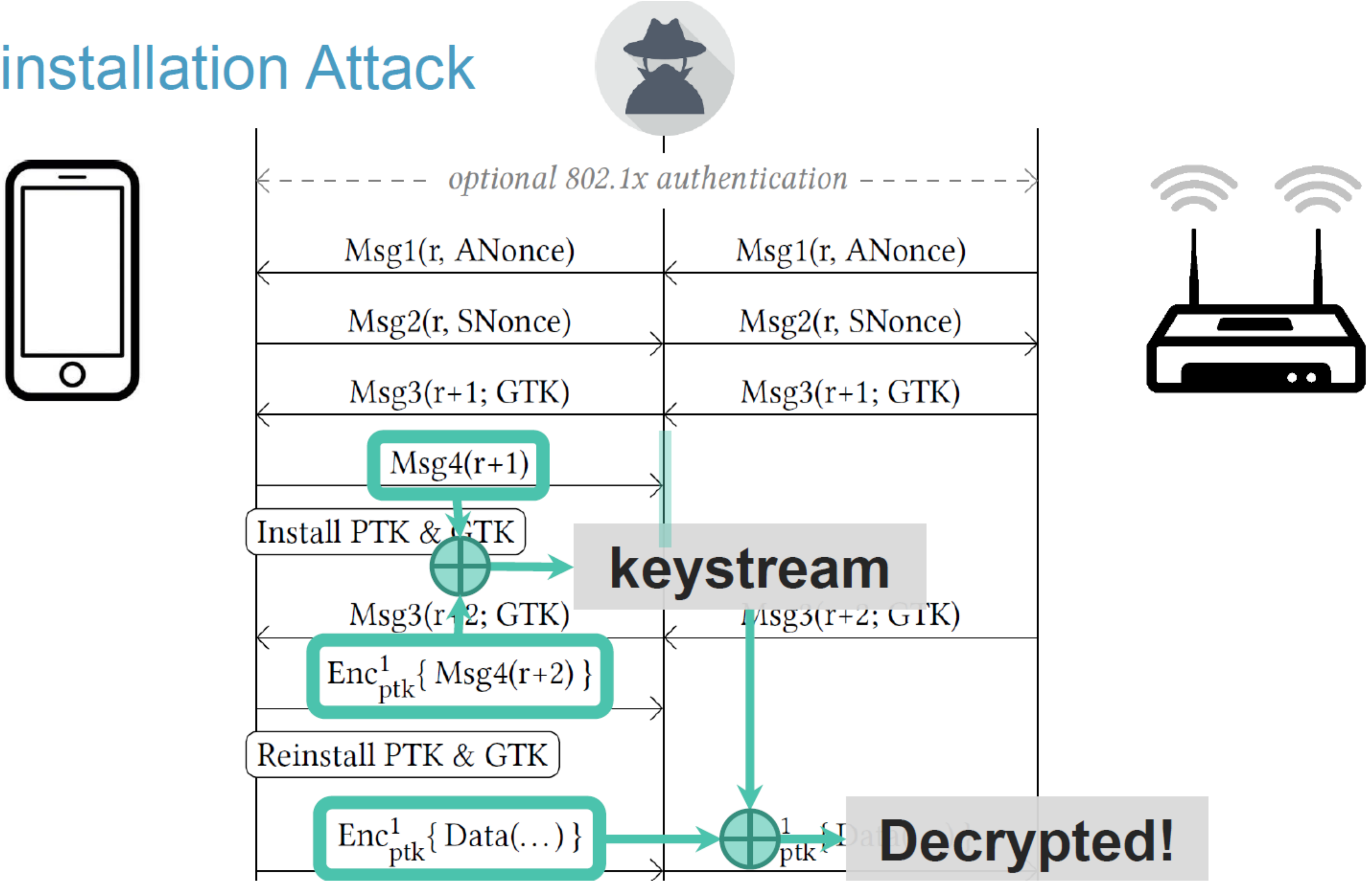
Encryption of 802.11 MAC Payloads



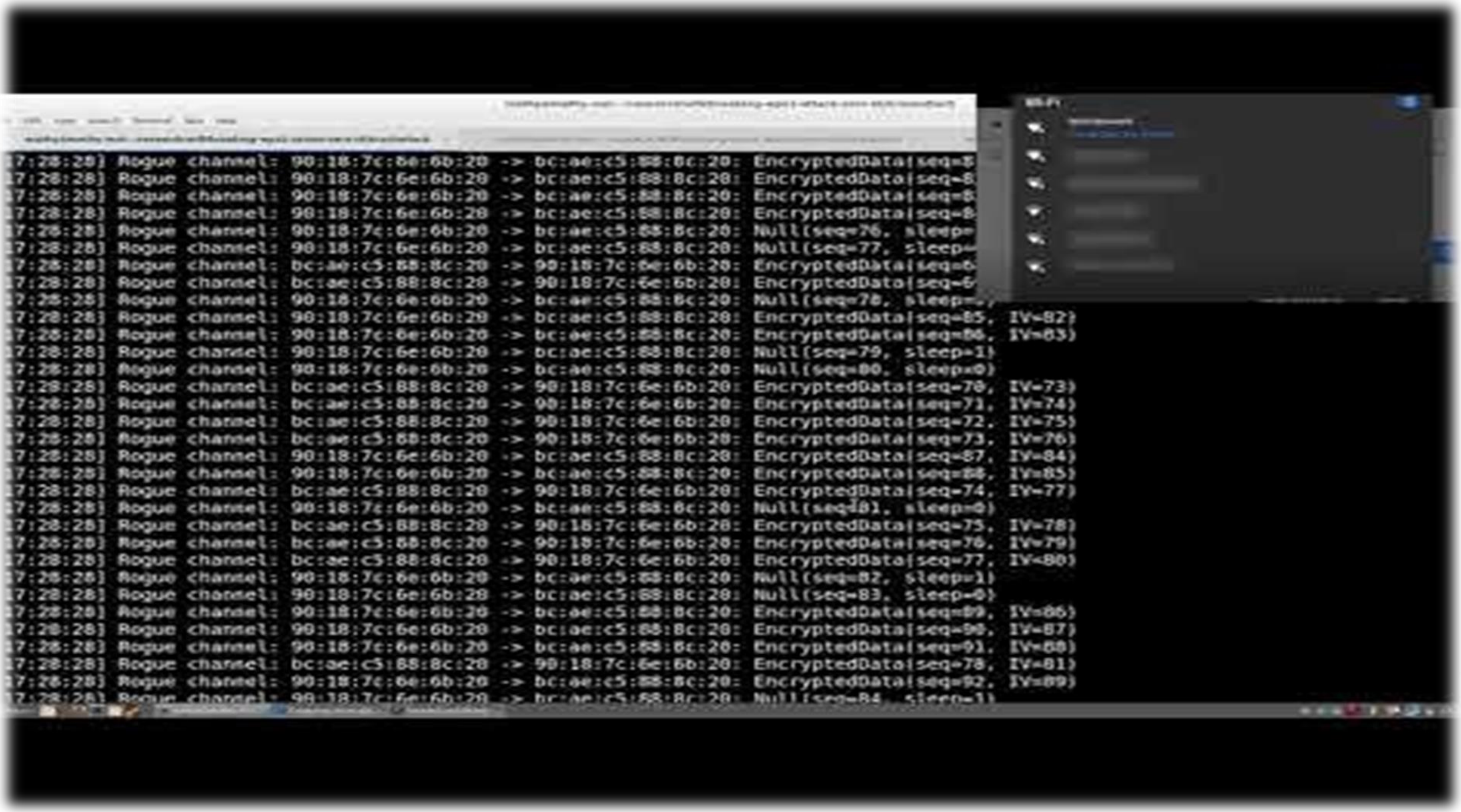
→ Nonce reuse implies keystream reuse (in all WPA2 ciphers)

KRACK: MITM attack on 4-Way H/S

Reinstallation Attack



KRACK: 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)
```

WPA3: OWE

- **OWE: Opportunistic Wireless Encryption for Open SSIDs**
  - IETF RFC 8110
- Encryption w/o authentication like HTTPS browsing
- Meant for open/public APs
- Diffie Hellman key exchange, does n't 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)
  - Sol for client-side AUTH: Captive portal
  - No sol for server-side AUTH
    - Rogue APs (Evil Twins) can still be setup

WPA3: 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)
- It uses Diffie Hellman key exchange to facilitate both the encryption key generation and mutual AUTH
  - SAE handshake 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

Wi-Fi Security Guidelines for Administrators

Use 802.1x based Auth & Protected Mgmt Frames

Allow only specific devices to access your wireless network

Use WIPS, anti-virus and anti-spyware software and a firewall

Change your router's pre-set password for administration and login over https

Turn off SSID broadcasting, apply patches, deploy WPA3 ASAP

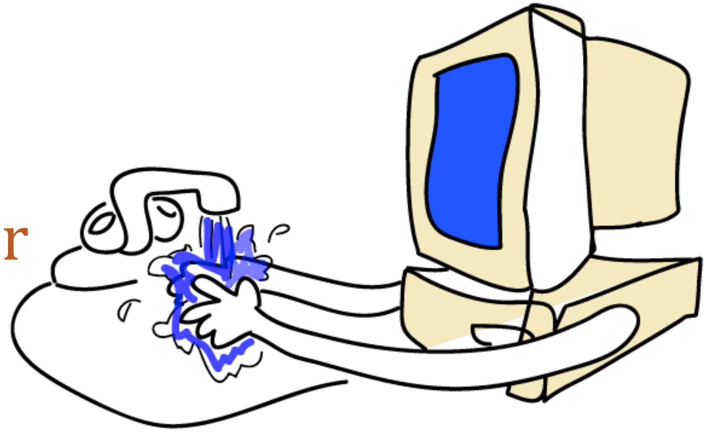
Change the SSID on your router from the default

A Simple way to convert OPEN AP into Protected AP!

- WPA3 takes a while for penetration into the market
- WPA2-PSK supports *session isolation*
  - Other users on the network can't easily see your traffic.
- So enable WPA2-PSK and publicize the passphrase
  - Big signs on the walls, than to offer OPEN Wi-Fi
  - Another option is to include passphrase in SSID, such as "Guest-WiFi-pwd-is-FREEACCESS" and "IITH-Guest-PWD-IITH@2020"!

Digital Hygiene

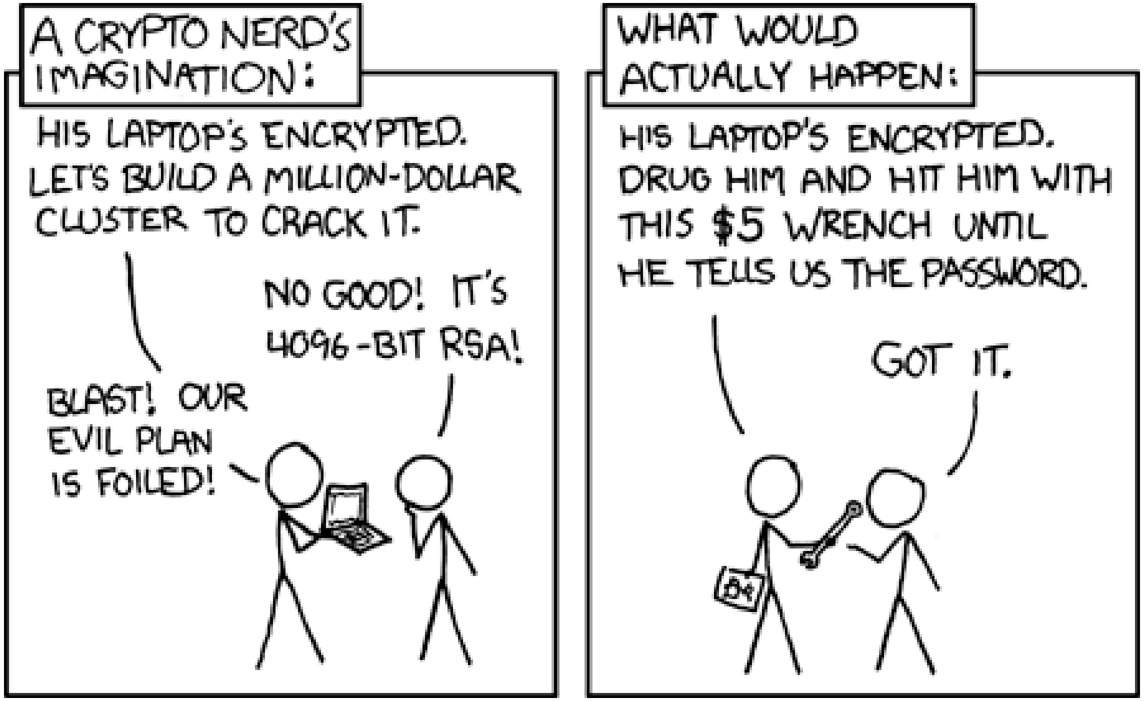
- □ Keep all programs and OS up-to-date
- □ **Backup, backup and backup again**
- □ Change default passwords & create strong, secure passwords (**password managers)**
- □ Avoid sharing using USB flash drives
- □ Don’t fall prey to social engineering or phishing
- □ Only browse sites with HTTPS
- □ Keep Bluetooth & Wi-Fi OFF when not in use esp in public places
- □ Use security software: Firewalls, Antivirus, etc
- □ Buy *cyber insurance* policy!



Limitations of Cryptography!

Cryptography works when used correctly !!

… but is not the solution to all security problems


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LETS BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

XKCD 538

References & Acknowledegments

- 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://thebestvpn.uk/unsecured-wifi-network/
- https://asecuritysite.com/encryption/
- WPA3:
  - https://blog.mojonetworks.com/wpa3-security-enhancements
  - http://www.mathyvanhoef.com/2018/03/wpa3-technical-details.html

KRACK: WPA2 Attacks (Videos)

- KRACK
  - https://www.youtube.com/watch?v=Oh4WURZoR98
- https://blog.mojonetworks.com/wpa2-vulnerability
- YouTube Playlist on WPA2 Attacks
- https://www.youtube.com/watch?v=fOgJswt7nAc

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Thank You

Email: tbr@iith.ac.in
Homepage: http://www.iith.ac.in/~tbr
Google Scholar Profile: http://goo.gl/JdgRB
NeWS Lab: https://newslab.iith.ac.in/