RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

NOW MATTERS

SESSION ID: TECH-R12

WI-FI SECURITY: THE DETAILS MATTER

Tom Carpenter

CTO
CWNP
@carpentertom | @CWNP

Why Does Wi-Fi Security Matter?

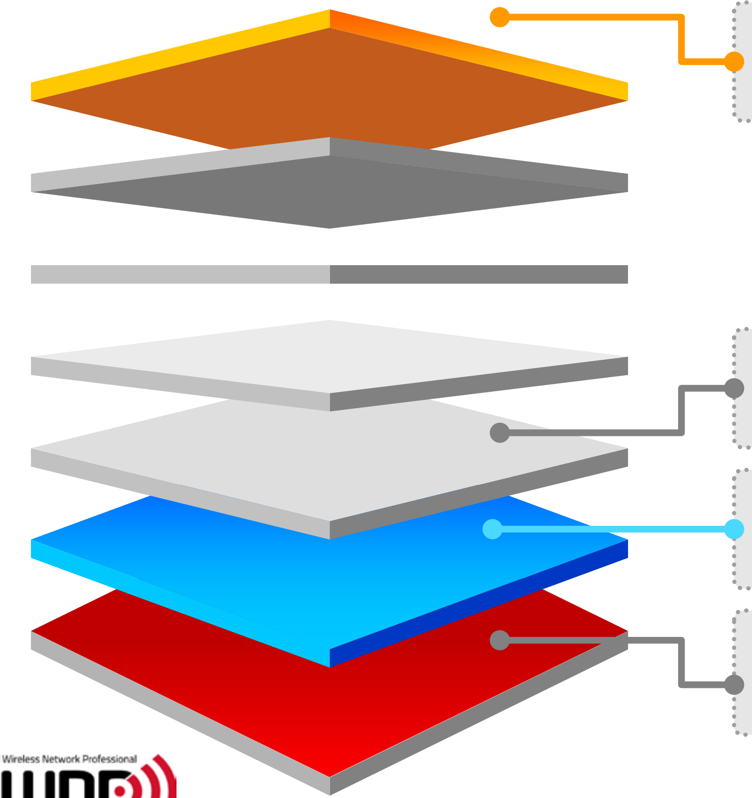
541.6 million hotspots by 2021

500 million new mobile workforce professionals

Wi-Fi is entering new areas every month

Cloud-managed WLAN market to grow to $3.3 billion by 2020

Wi-Fi Security Landscape



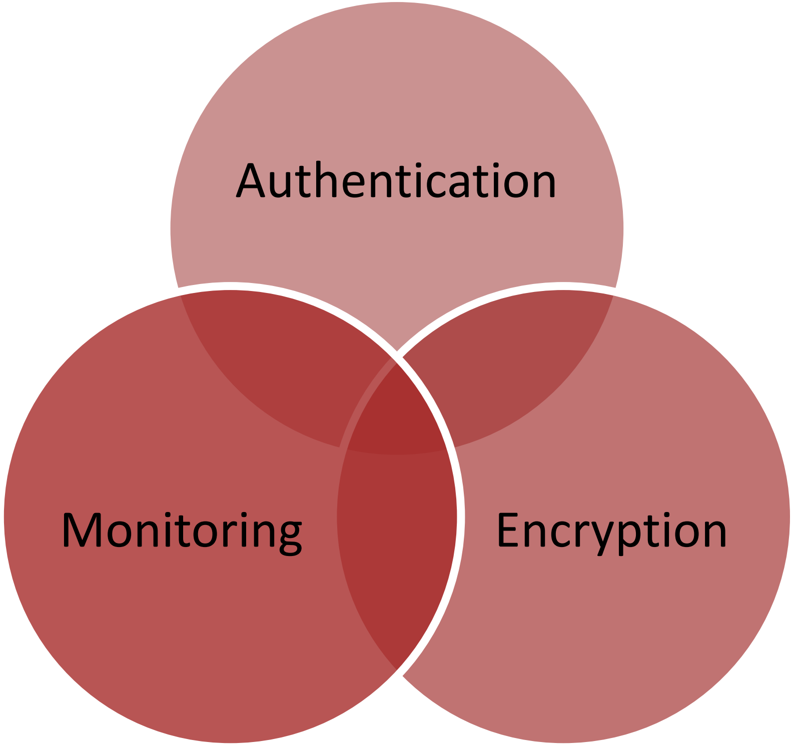
Application Layer
Use of secure applications assists in network security

Network (IP) Layer
Secure infrastructure and protocols should be used

Data Link (MAC) Layer
Data encryption and authentication should be used

Physical Layer
Monitoring and alert systems should be used

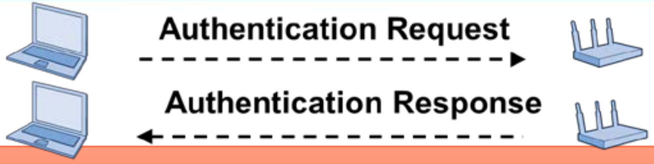
Focus Points



802.11 Authentication Methods

- Open System
- Pre-Shared Key
- 802.1X/EAP
- Shared Key is deprecated as of 802.11i-2004

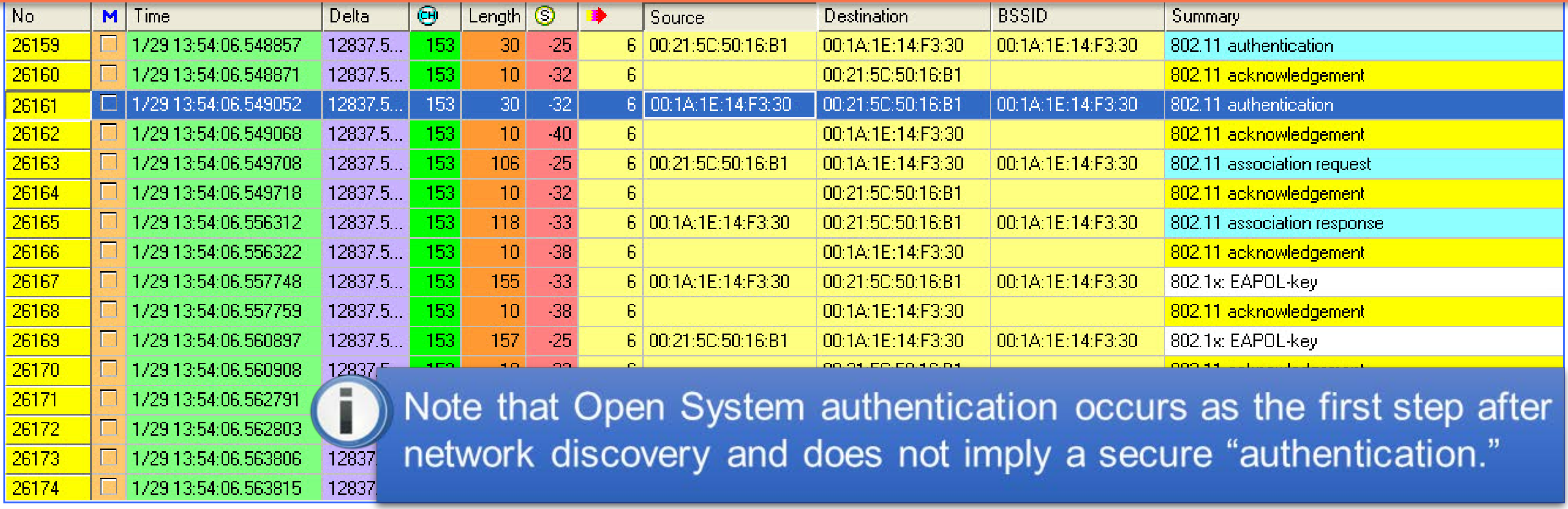
Open System Authentication



Authentication Request

Authentication Response

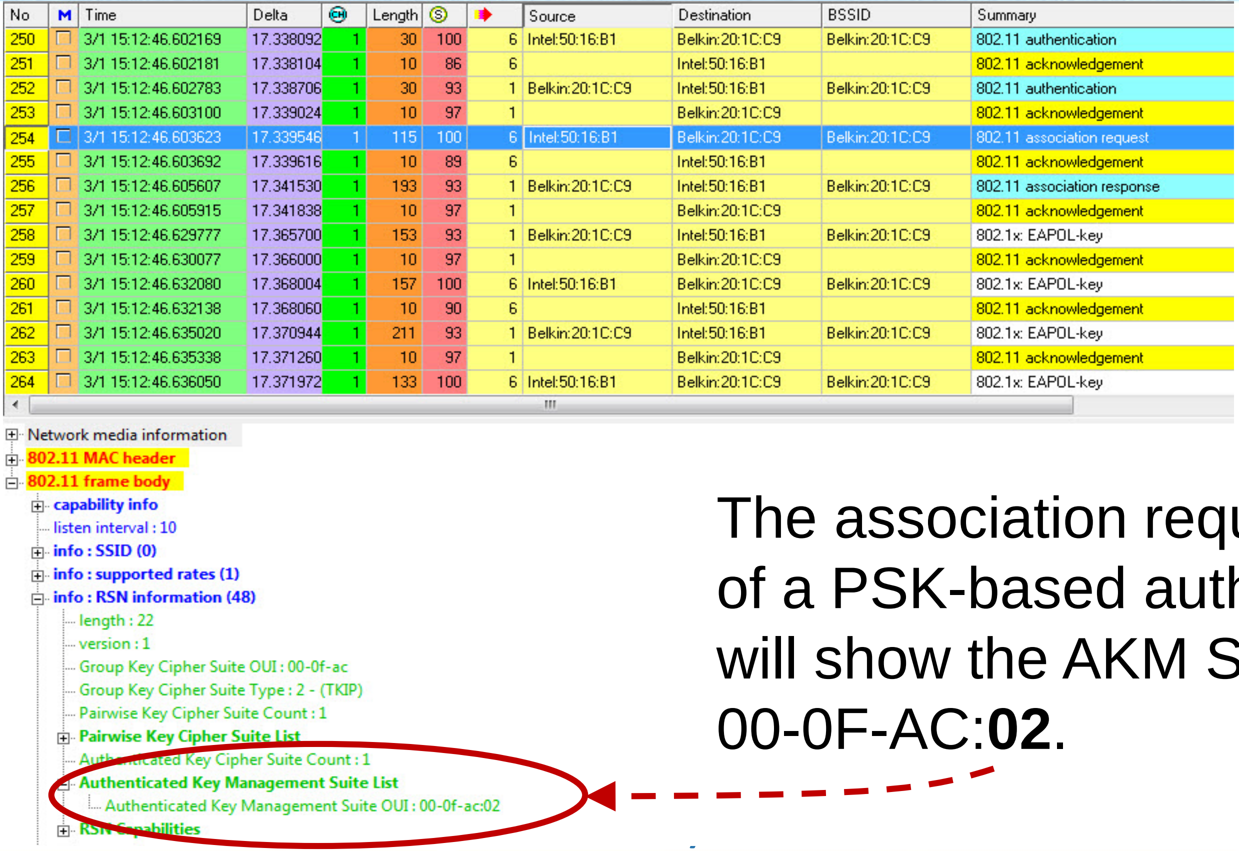
A null authentication method



No	M	Time	Delta	CH	Length	S		Source	Destination	BSSID	Summary
26159	□	1/29 13:54:06.548857	12837.5...	153	30	-25	6	00:21:5C:50:16:B1	00:1A:1E:14:F3:30	00:1A:1E:14:F3:30	802.11 authentication
26160	□	1/29 13:54:06.548871	12837.5...	153	10	-32	6		00:21:5C:50:16:B1		802.11 acknowledgement
26161	□	1/29 13:54:06.549052	12837.5...	153	30	-32	6	00:1A:1E:14:F3:30	00:21:5C:50:16:B1	00:1A:1E:14:F3:30	802.11 authentication
26162	□	1/29 13:54:06.549068	12837.5...	153	10	-40	6		00:1A:1E:14:F3:30		802.11 acknowledgement
26163	□	1/29 13:54:06.549708	12837.5...	153	106	-25	6	00:21:5C:50:16:B1	00:1A:1E:14:F3:30	00:1A:1E:14:F3:30	802.11 association request
26164	□	1/29 13:54:06.549718	12837.5...	153	10	-32	6		00:21:5C:50:16:B1		802.11 acknowledgement
26165	□	1/29 13:54:06.556312	12837.5...	153	118	-33	6	00:1A:1E:14:F3:30	00:21:5C:50:16:B1	00:1A:1E:14:F3:30	802.11 association response
26166	□	1/29 13:54:06.556322	12837.5...	153	10	-38	6		00:1A:1E:14:F3:30		802.11 acknowledgement
26167	□	1/29 13:54:06.557748	12837.5...	153	155	-33	6	00:1A:1E:14:F3:30	00:21:5C:50:16:B1	00:1A:1E:14:F3:30	802.1x: EAPOL-key
26168	□	1/29 13:54:06.557759	12837.5...	153	10	-38	6		00:1A:1E:14:F3:30		802.11 acknowledgement
26169	□	1/29 13:54:06.560897	12837.5...	153	157	-25	6	00:21:5C:50:16:B1	00:1A:1E:14:F3:30	00:1A:1E:14:F3:30	802.1x: EAPOL-key
26170	□	1/29 13:54:06.560908	12837.5								
26171	□	1/29 13:54:06.562791									
26172	□	1/29 13:54:06.562803									
26173	□	1/29 13:54:06.563806	12837.								
26174	□	1/29 13:54:06.563815	12837.								

Note that Open System authentication occurs as the first step after network discovery and does not imply a secure "authentication."

Pre-Shared Key (PSK)

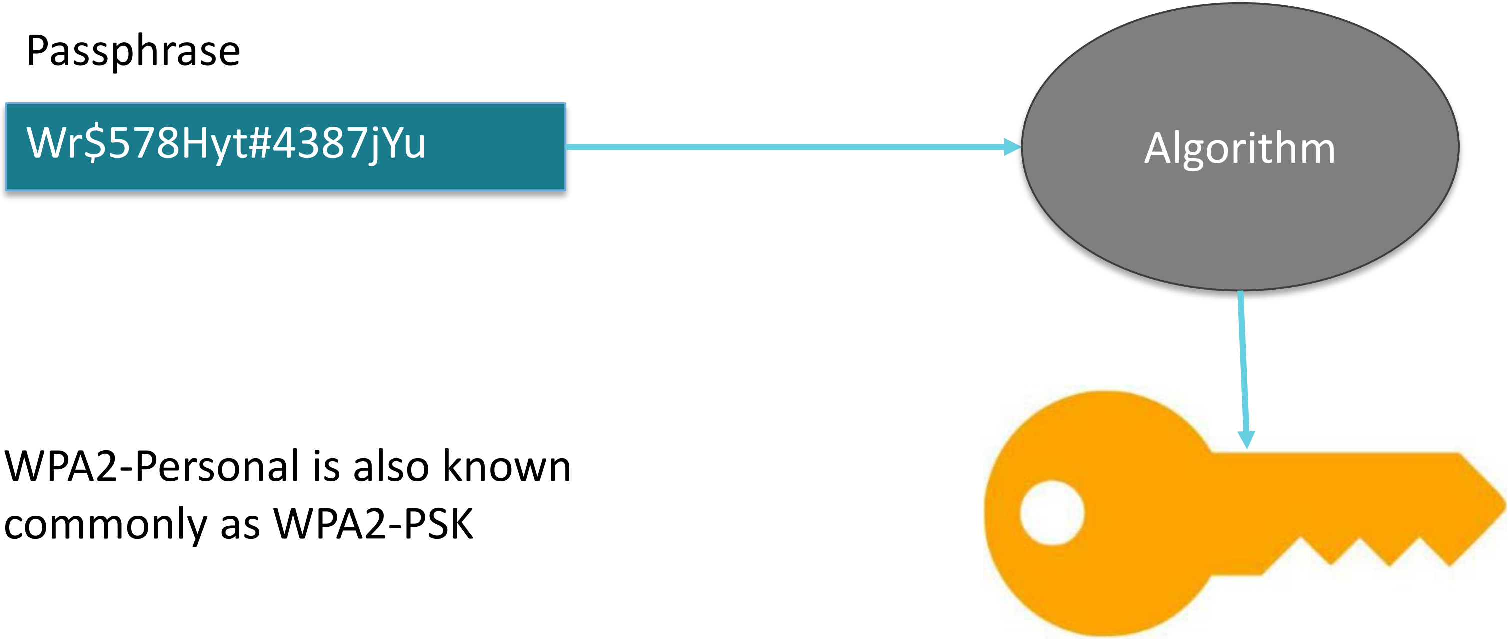


No	M	Time	Delta	CH	Length	Ⓢ	➔	Source	Destination	BSSID	Summary
250	□	3/1 15:12:46.602169	17.338092	1	30	100	6	Intel:50:16:B1	Belkin:20:1C:C9	Belkin:20:1C:C9	802.11 authentication
251	□	3/1 15:12:46.602181	17.338104	1	10	86	6		Intel:50:16:B1		802.11 acknowledgement
252	□	3/1 15:12:46.602783	17.338706	1	30	93	1	Belkin:20:1C:C9	Intel:50:16:B1	Belkin:20:1C:C9	802.11 authentication
253	□	3/1 15:12:46.603100	17.339024	1	10	97	1		Belkin:20:1C:C9		802.11 acknowledgement
254	□	3/1 15:12:46.603623	17.339546	1	115	100	6	Intel:50:16:B1	Belkin:20:1C:C9	Belkin:20:1C:C9	802.11 association request
255	□	3/1 15:12:46.603692	17.339616	1	10	89	6		Intel:50:16:B1		802.11 acknowledgement
256	□	3/1 15:12:46.605607	17.341530	1	193	93	1	Belkin:20:1C:C9	Intel:50:16:B1	Belkin:20:1C:C9	802.11 association response
257	□	3/1 15:12:46.605915	17.341838	1	10	97	1		Belkin:20:1C:C9		802.11 acknowledgement
258	□	3/1 15:12:46.629777	17.365700	1	153	93	1	Belkin:20:1C:C9	Intel:50:16:B1	Belkin:20:1C:C9	802.1x: EAPOL-key
259	□	3/1 15:12:46.630077	17.366000	1	10	97	1		Belkin:20:1C:C9		802.11 acknowledgement
260	□	3/1 15:12:46.632080	17.368004	1	157	100	6	Intel:50:16:B1	Belkin:20:1C:C9	Belkin:20:1C:C9	802.1x: EAPOL-key
261	□	3/1 15:12:46.632138	17.368060	1	10	90	6		Intel:50:16:B1		802.11 acknowledgement
262	□	3/1 15:12:46.635020	17.370944	1	211	93	1	Belkin:20:1C:C9	Intel:50:16:B1	Belkin:20:1C:C9	802.1x: EAPOL-key
263	□	3/1 15:12:46.635338	17.371260	1	10	97	1		Belkin:20:1C:C9		802.11 acknowledgement
264	□	3/1 15:12:46.636050	17.371972	1	133	100	6	Intel:50:16:B1	Belkin:20:1C:C9	Belkin:20:1C:C9	802.1x: EAPOL-key

```
⊞ Network media information
⊞ 802.11 MAC header
⊟ 802.11 frame body
    ⊞ capability info
      listen interval : 10
    ⊞ info : SSID (0)
    ⊞ info : supported rates (1)
    ⊟ info : RSN information (48)
        length : 22
        version : 1
        Group Key Cipher Suite OUI : 00-0f-ac
        Group Key Cipher Suite Type : 2 - (TKIP)
        Pairwise Key Cipher Suite Count : 1
      ⊞ Pairwise Key Cipher Suite List
        Authenticated Key Cipher Suite Count : 1
      ⊟ Authenticated Key Management Suite List
          Authenticated Key Management Suite OUI : 00-0f-ac:02
      ⊞ RSN Capabilities
```

The association request frame of a PSK-based authentication will show the AKM Suite type as 00-0F-AC:**02**.

WPA2-Personal



Passphrase

Wr$578Hyt#4387jYu

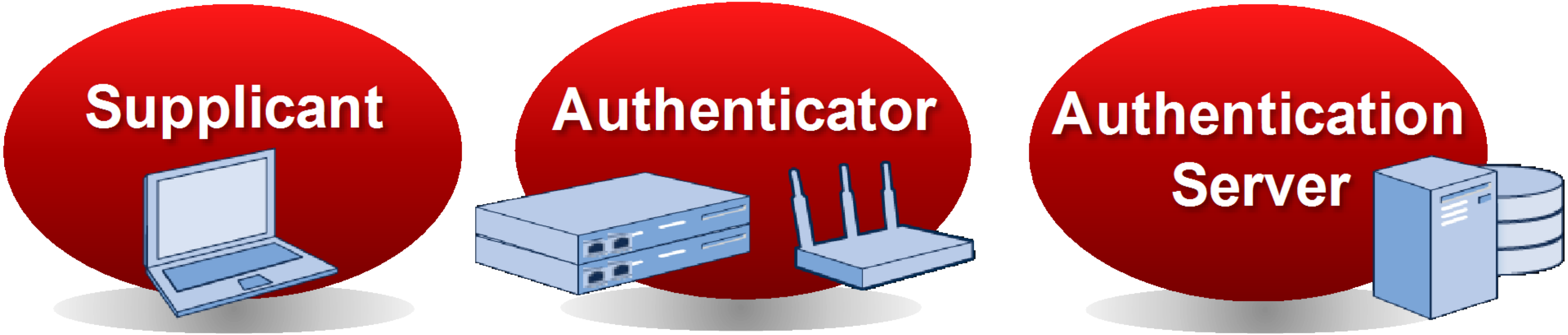
Algorithm

WPA2-Personal is also known commonly as WPA2-PSK

How PSK Authenticates

- Authentication occurs during the 4-way handshake
- Frames 2-4 are MIC-protected
- The MIC calculation includes the KCK, which is part of the PTK, as an input
- Mismatched MIC calculations between the supplicant and authenticator result in termination of the 4-way handshake

Port-Based 802.1X Access Control



Supplicant

Authenticator

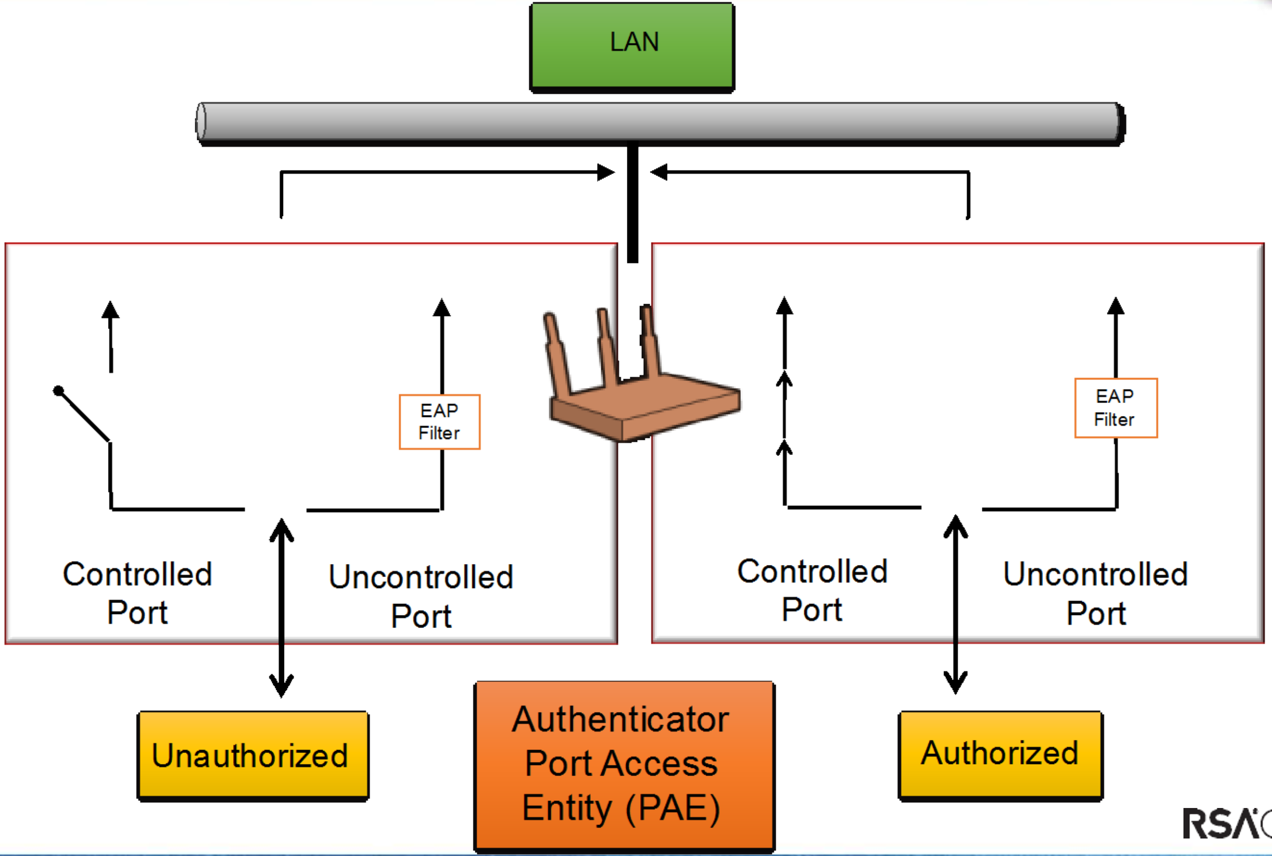
Authentication Server

An entity at one end of a point-to-point LAN segment that is being authenticated by an Authenticator attached to the other end of that link.

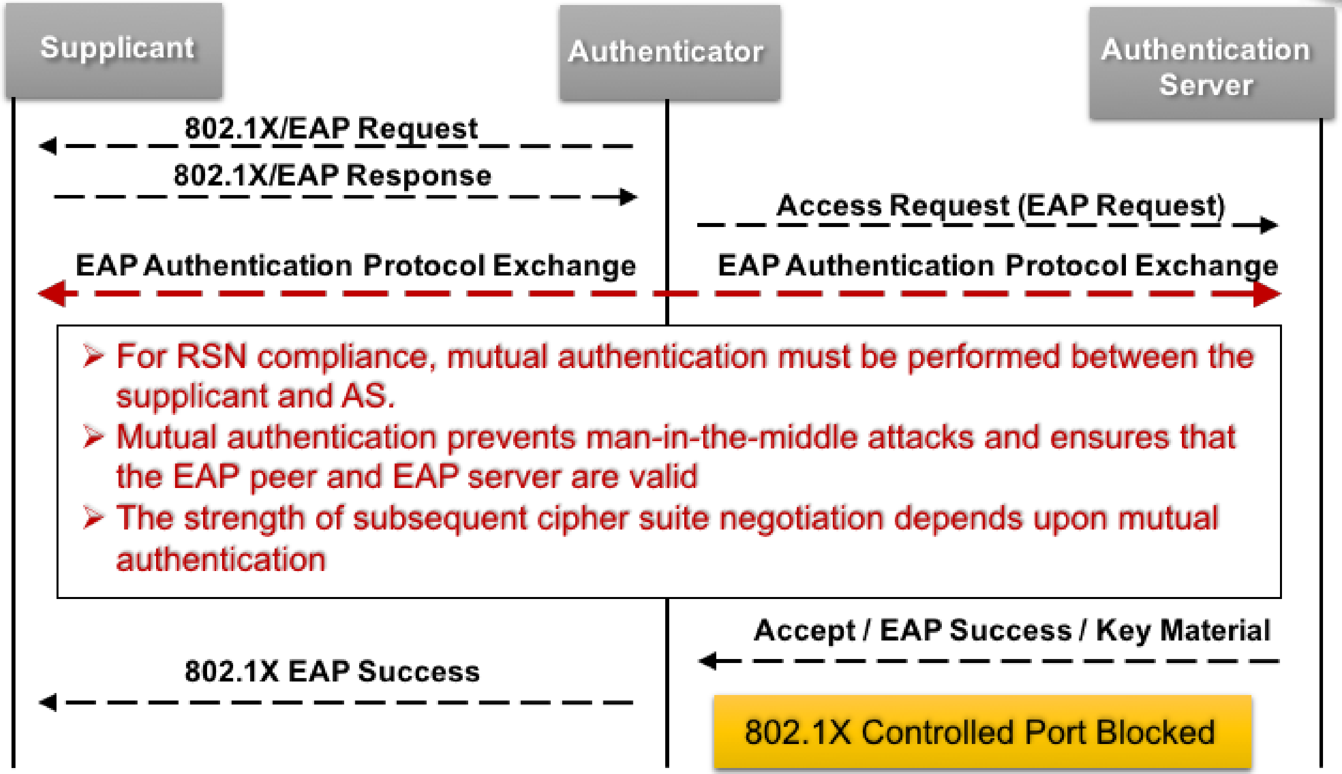
An entity at one end of a point-to-point LAN segment that facilitates authentication of the entity attached to the other end of that link.

An entity that provides an authentication service to an Authenticator. This service determines, from the credentials provided by the Supplicant, whether the Supplicant is authorized to access the services provided by the Authenticator.

802.1X Port Functions



802.1X/EAP



Supplicant

Authenticator

Authentication Server

802.1X/EAP Request

802.1X/EAP Response

Access Request (EAP Request)

EAP Authentication Protocol Exchange

EAP Authentication Protocol Exchange

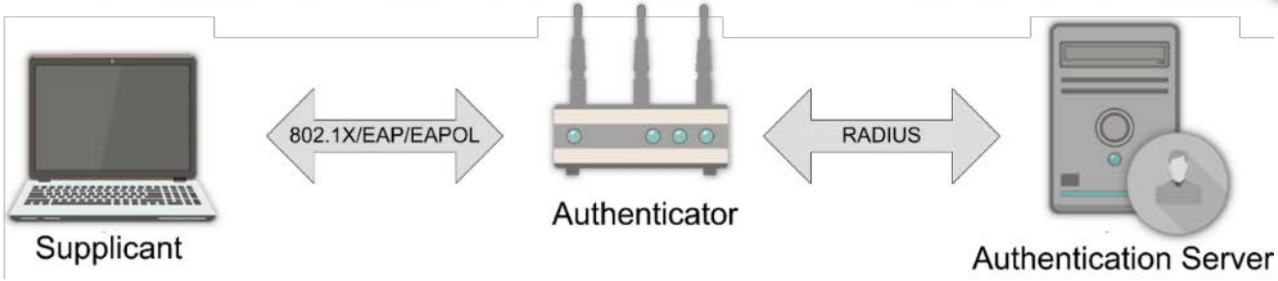
- For RSN compliance, mutual authentication must be performed between the supplicant and AS.
- Mutual authentication prevents man-in-the-middle attacks and ensures that the EAP peer and EAP server are valid
- The strength of subsequent cipher suite negotiation depends upon mutual authentication

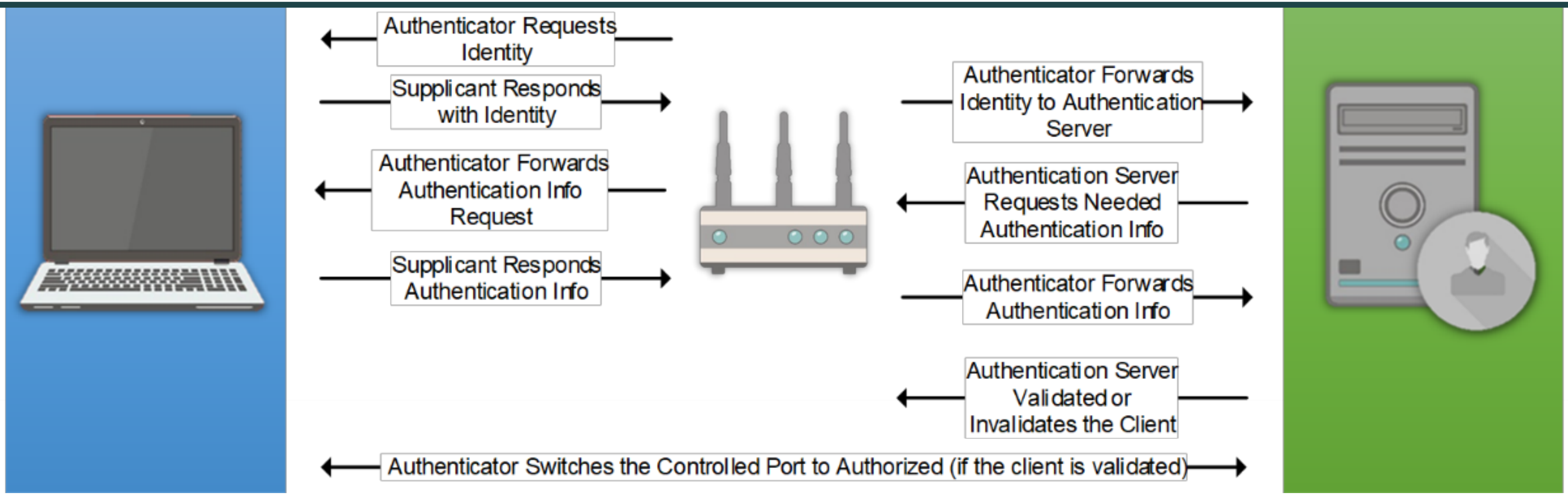
Accept / EAP Success / Key Material

802.1X EAP Success

802.1X Controlled Port Blocked

802.1X/EAP Architecture





Authenticator Requests Identity

Authenticator Forwards Identity to Authentication Server

Supplicant Responds with Identity

Authenticator Forwards Authentication Info Request

Authentication Server Requests Needed Authentication Info

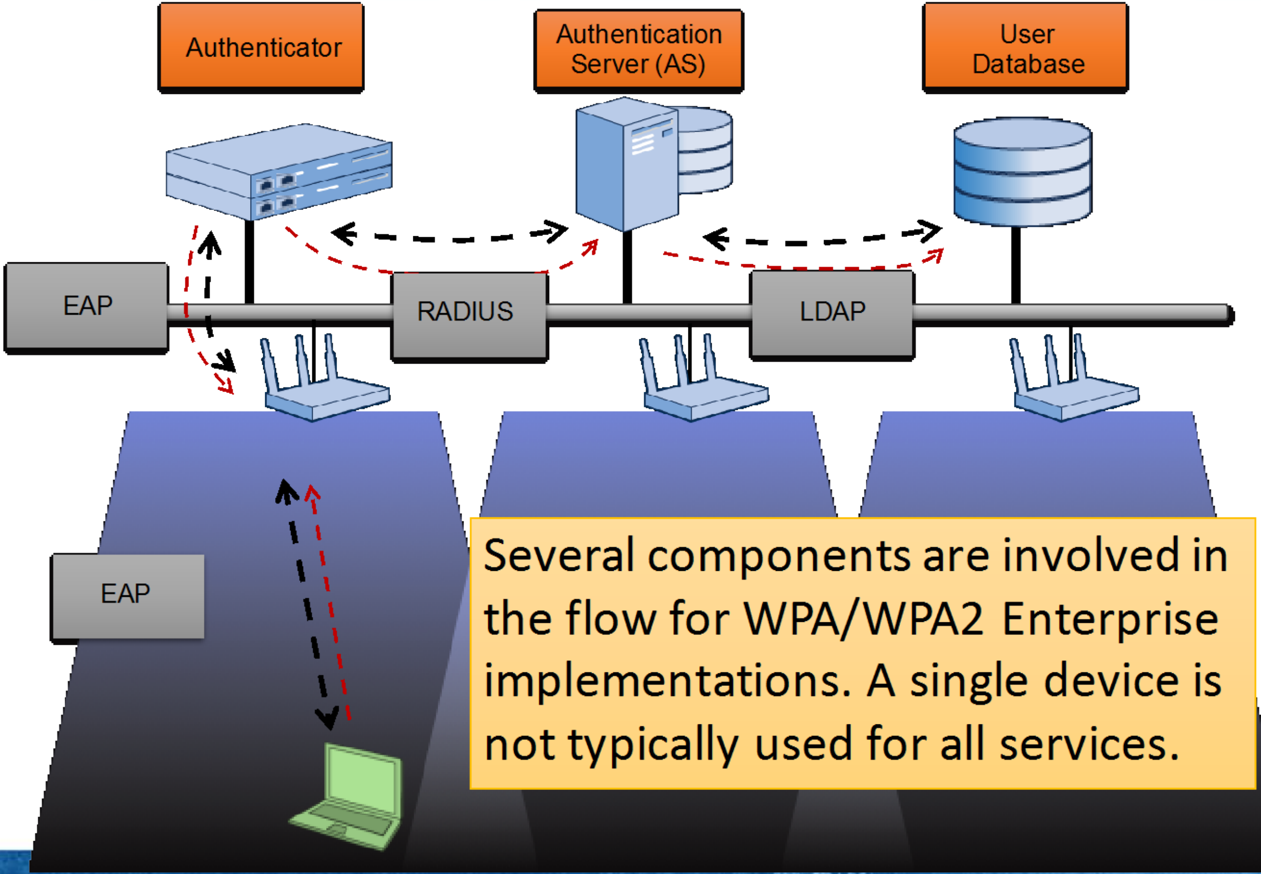
Supplicant Responds Authentication Info

Authenticator Forwards Authentication Info

Authentication Server Validated or Invalidates the Client

Authenticator Switches the Controlled Port to Authorized (if the client is validated)

Enterprise 802.1X/EAP Deployment

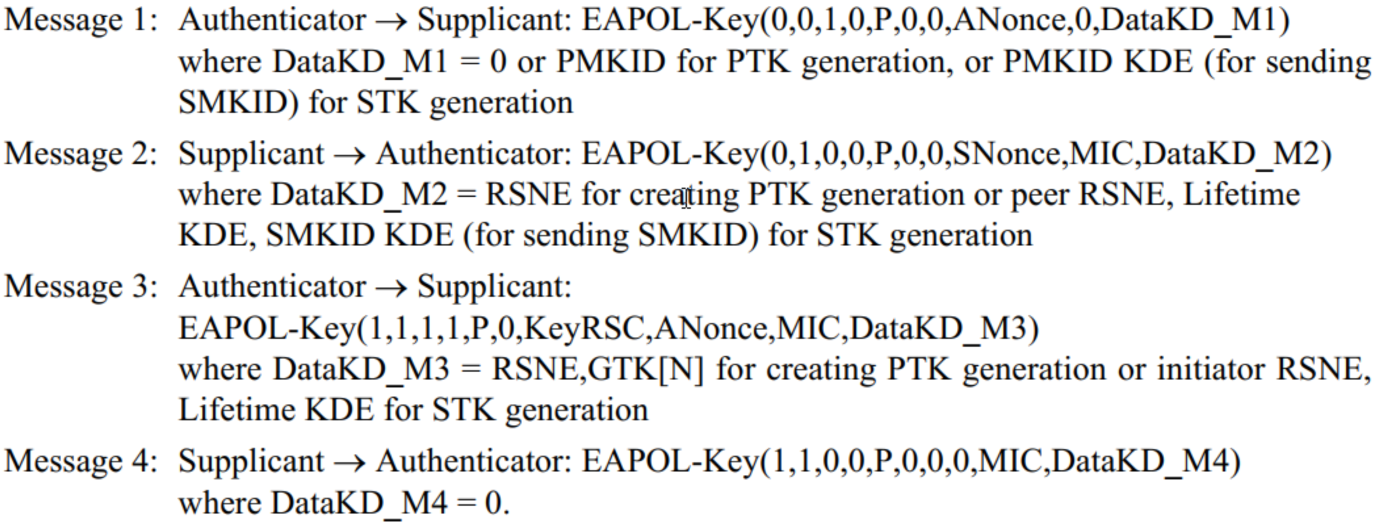


Several components are involved in the flow for WPA/WPA2 Enterprise implementations. A single device is not typically used for all services.

802.11 Encryption Methods

- Authentication and Key Management suites
  - Temporal Key Integrity Protocol (TKIP) – Deprecated
  - Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP)
- Encryption algorithms
  - Rivest Cipher 4 (RC4) - Deprecated
  - Advanced Encryption Standard (AES)
- Modern Wi-Fi generates encryption keys during the 4-way handshake

4-Way Handshake



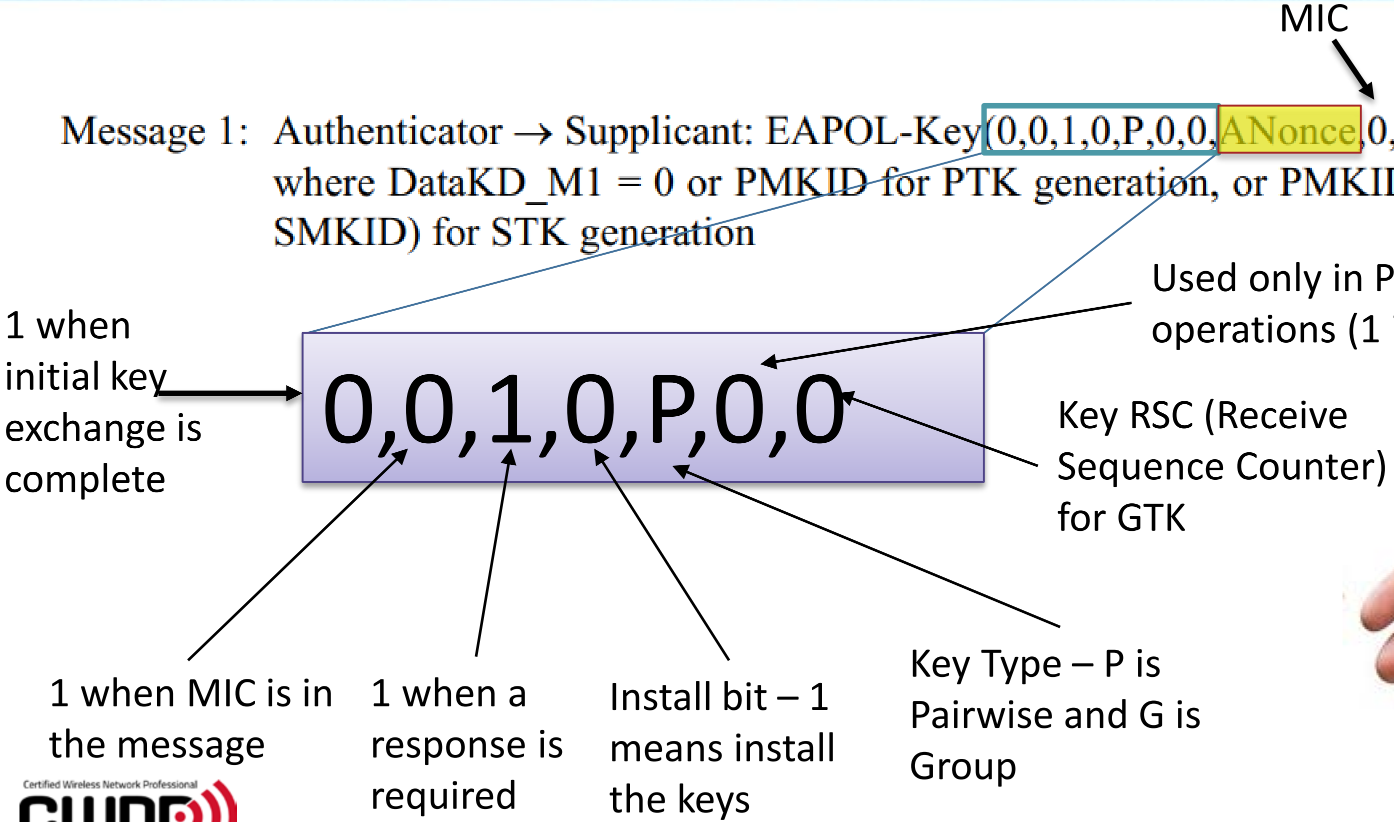
Message 1: Authenticator → Supplicant: EAPOL-Key(0,0,1,0,P,0,0,ANonce,0,DataKD_M1) where DataKD_M1 = 0 or PMKID for PTK generation, or PMKID KDE (for sending SMKID) for STK generation

Message 2: Supplicant → Authenticator: EAPOL-Key(0,1,0,0,P,0,0,SNonce,MIC,DataKD_M2) where DataKD_M2 = RSNE for creating PTK generation or peer RSNE, Lifetime KDE, SMKID KDE (for sending SMKID) for STK generation

Message 3: Authenticator → Supplicant: EAPOL-Key(1,1,1,1,P,0,KeyRSC,ANonce,MIC,DataKD_M3) where DataKD_M3 = RSNE,GTK[N] for creating PTK generation or initiator RSNE, Lifetime KDE for STK generation

Message 4: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,DataKD_M4) where DataKD_M4 = 0.

Message One



MIC

Elements defining the key

Message 1: Authenticator → Supplicant: EAPOL-Key(0,0,1,0,P,0,0,ANonce,0,DataKD_M1)
where DataKD_M1 = 0 or PMKID for PTK generation, or PMKID KDE (for sending SMKID) for STK generation

0,0,1,0,P,0,0

1 when initial key exchange is complete

Used only in PeerKey operations (1 is PeerKey)

Key RSC (Receive Sequence Counter) for GTK

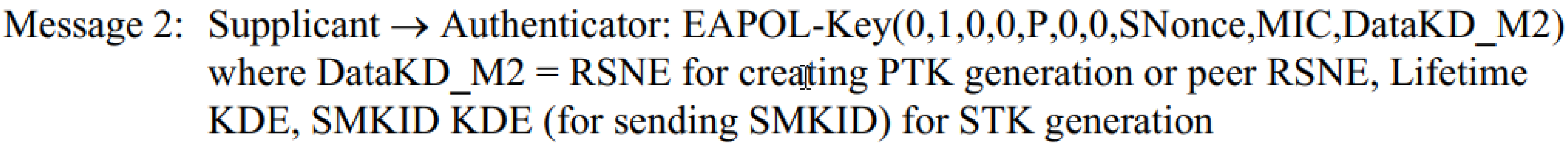
1 when MIC is in the message

1 when a response is required

Install bit – 1 means install the keys

Key Type – P is Pairwise and G is Group

Message Two

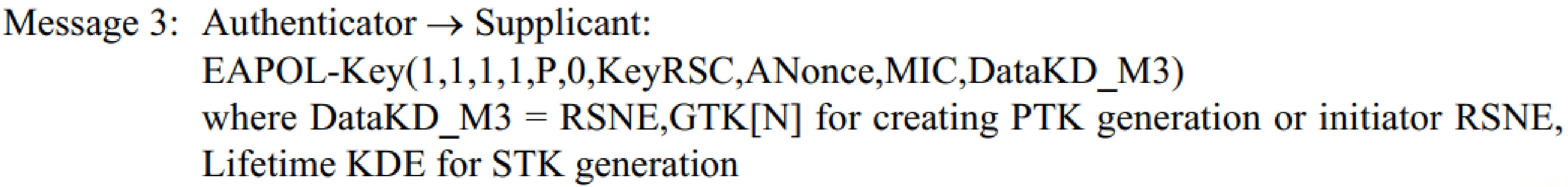


Message 2: Supplicant → Authenticator: EAPOL-Key(0,1,0,0,P,0,0,SNonce,MIC,DataKD_M2)
where DataKD_M2 = RSNE for creating PTK generation or peer RSNE, Lifetime KDE, SMKID KDE (for sending SMKID) for STK generation

The client now sends its NONCE (SNONCE) to the AP/Controller

At this point the client and the AP both have all that's required to generate the Pairwise Transient Key (PTK)

Message 3

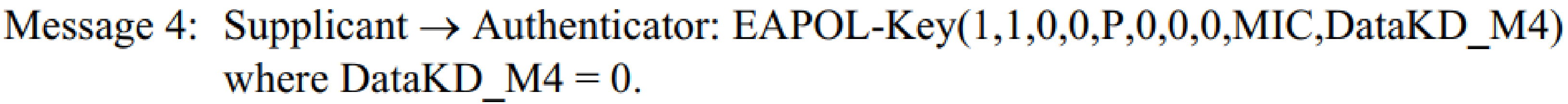


Message 3: Authenticator → Supplicant:
EAPOL-Key(1,1,1,1,P,0,KeyRSC,ANonce,MIC,DataKD_M3)
where DataKD_M3 = RSNE,GTK[N] for creating PTK generation or initiator RSNE, Lifetime KDE for STK generation

The AP/Controller can now send the GTK to the client and the install bit (bit 4) is set to 1

This is the point at which KRACK operates

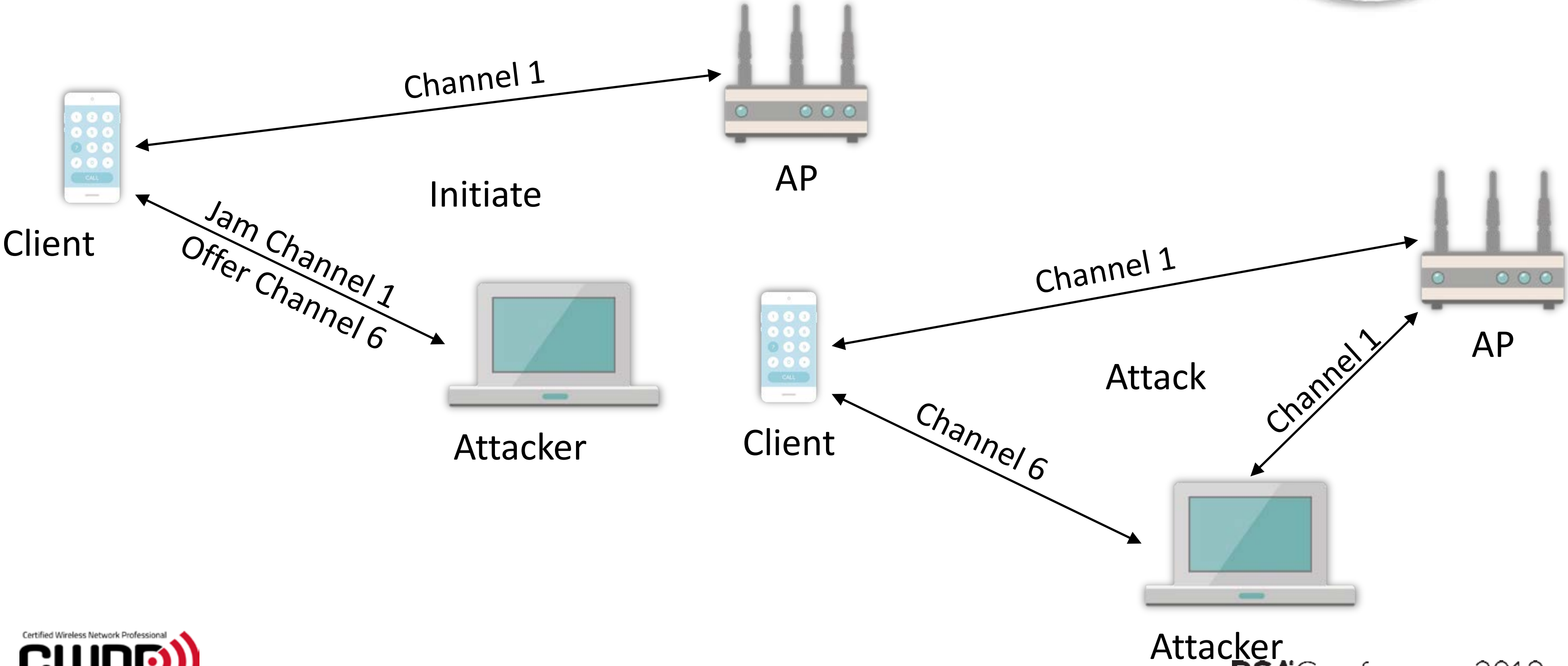
Message 4



Message 4: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,DataKD_M4)
where DataKD_M4 = 0.

This is really just the "all is good" message so the AP/Controller knows the client has the PTK and GTK installed

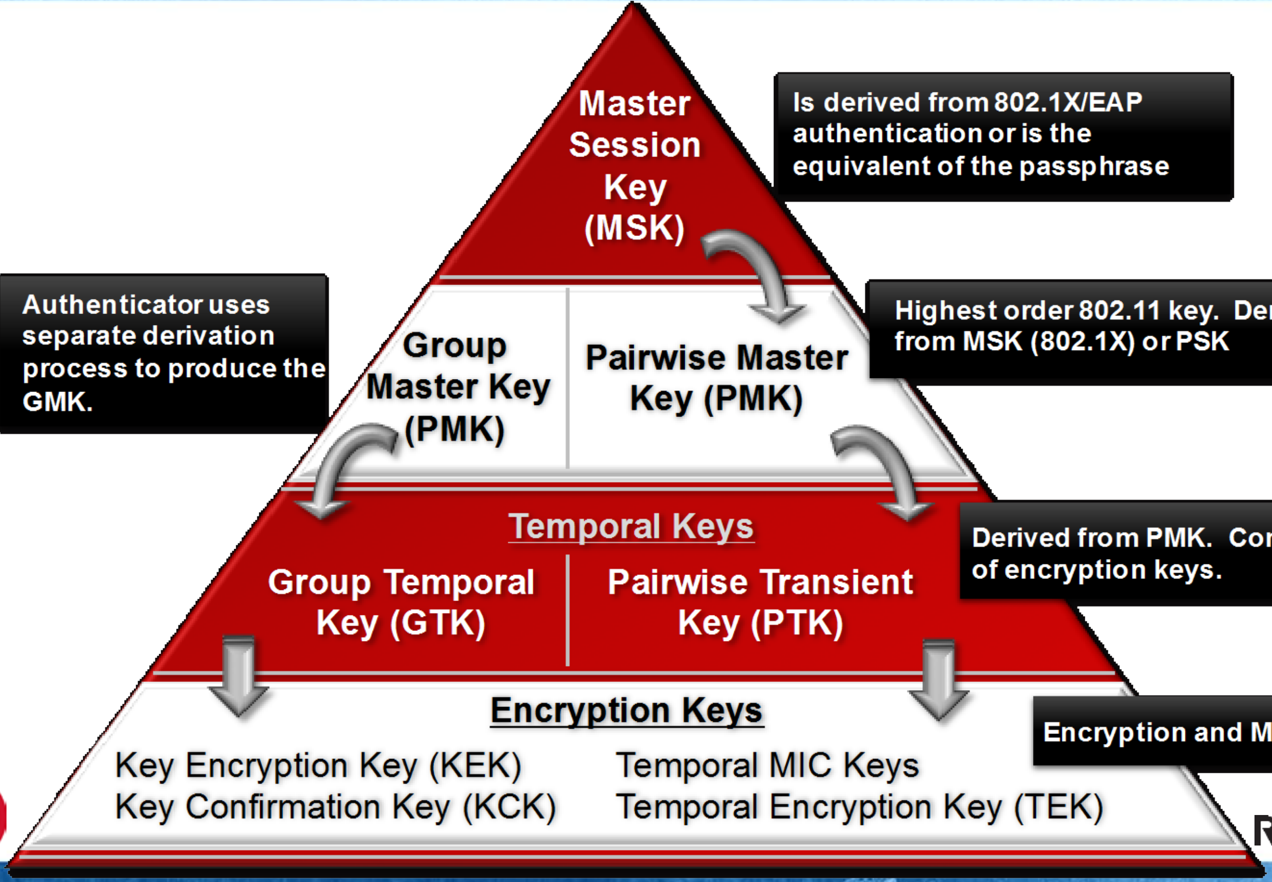
KRACK Operation



Who is to blame for KRACK?

- Great question; Complex answer
- Some say the IEEE because of closed processes and lack of availability of the standard early after release
  - Tom’s take: the 802.11i amendment has been easily available for 13 years with no fee most of that time, if someone noted the problem, the IEEE could have easily included a fix in 11n, 11ac, or any other amendment since then – not sure this is the real problem
- Some say the vendors because they should have implemented the flexible state machine more securely
  - Tom’s take: this is a hard one, the standard leaves a lot of flexibility, so each vendor would do it differently and if they make it too complex they could introduce compatibility problems
- Tom’s opinion: Time
  - Tom’s take: time is to blame; nearly every security solution degrades over time as the most brilliant minds may create it, but other brilliant minds want to thwart it – time is usually on the side of the attackers
- End result: Security is a process not an event

802.11 Key Hierarchy



Master Session Key (MSK)

Is derived from 802.1X/EAP authentication or is the equivalent of the passphrase

Group Master Key (PMK)

Pairwise Master Key (PMK)

Authenticator uses separate derivation process to produce the GMK.

Highest order 802.11 key. Derived from MSK (802.1X) or PSK

Temporal Keys

Group Temporal Key (GTK)

Pairwise Transient Key (PTK)

Derived from PMK. Composed of encryption keys.

Encryption Keys

Key Encryption Key (KEK)
Key Confirmation Key (KCK)

Temporal MIC Keys
Temporal Encryption Key (TEK)

Encryption and MIC keys

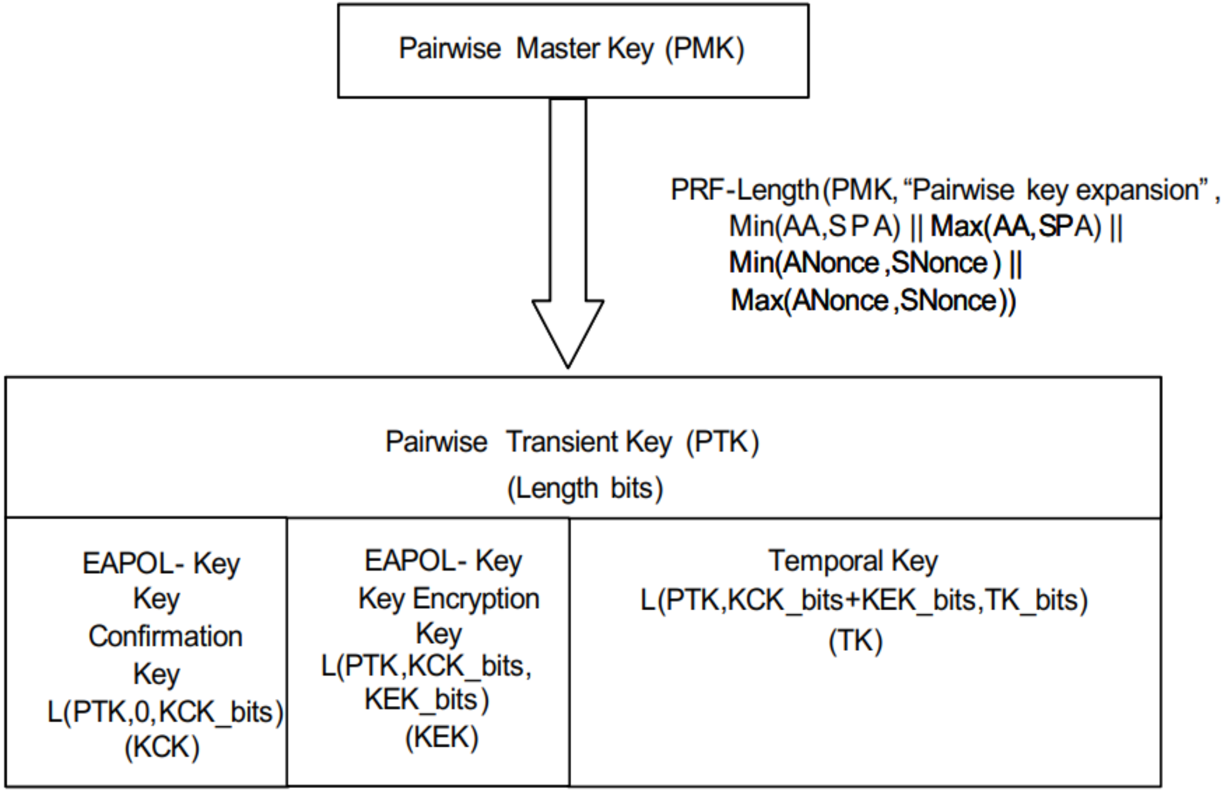
Pairwise Transient Key (PTK)

The PTK is comprised of three keys: KCK, KEK and TK

KCK used for key integrity

KEK used to encrypt and send keys (GTK)

The TK is used to encrypt data payloads



Pairwise Master Key (PMK)

PRF-Length(PMK, "Pairwise key expansion",
Min(AA,SPA) || Max(AA,SPA) ||
Min(ANonce,SNonce) ||
Max(ANonce,SNonce))

Pairwise Transient Key (PTK)
(Length bits)

EAPOL- Key Key Confirmation Key L(PTK,0,KCK_bits) (KCK)	EAPOL- Key Key Encryption Key L(PTK,KCK_bits, KEK_bits) (KEK)	Temporal Key L(PTK,KCK_bits+KEK_bits,TK_bits) (TK)

Wi-Fi Monitoring Methods

- Infrastructure solutions
- Overlay solutions
- Mobile solutions

Where do I go from here?

- Immediately
  - Validate the proper security of your existing Wi-Fi gear
    - Verify patches
    - Verify configuration
- In the next 2-3 months
  - Ensure all newly acquired equipment supports WPA2 (amended) or WPA3
    - Anything certified after November 2017 is tested for KRACK patching
- In the next six months
  - Consider a dedicated Wi-Fi security monitoring solution
    - Monitor configurations, new RF devices, anomalies
  - Many performance tools integrate security metrics, such as 7signal

THANK YOU