Wi-Fi Technology Fundamentals

WI-FI TECHNOLOGY
FUNDAMENTALS COURSE

Module-4

Security in Wi-Fi

Session-4b

Authentication and Encryption Mechanisms

Last Session Recap……

Module-4

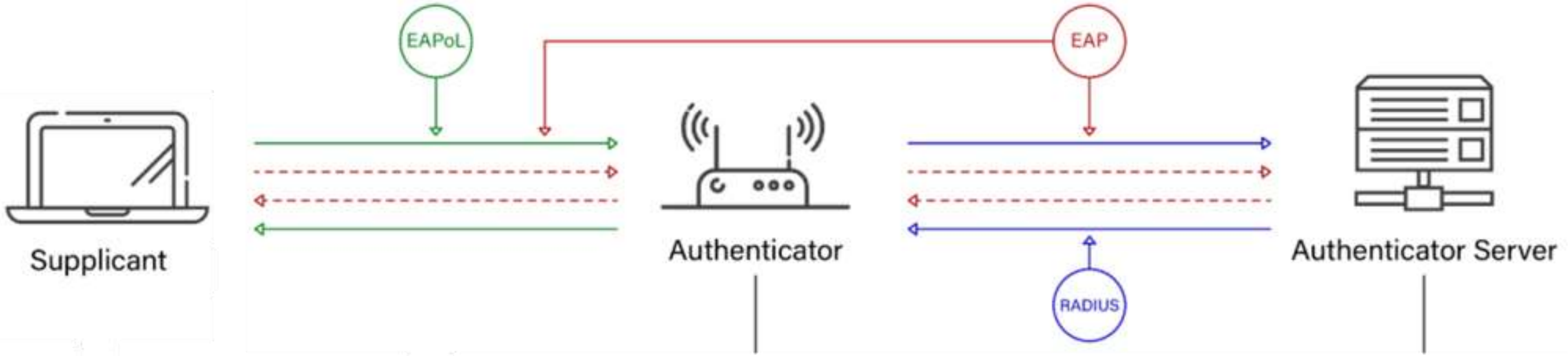
Security in Wi-Fi

Session-4a

Security Basics, Various Security Protocols

- ✓ Authentication, Confidentiality and Integrity
- ✓ Supplicant, Authenticator and Authentication Server
- ✓ Personal and Enterprise Security
- ✓ WEP, WPA, WPA2, WPA3
- ✓ Demo

The Three Enforcers of WiFi Security

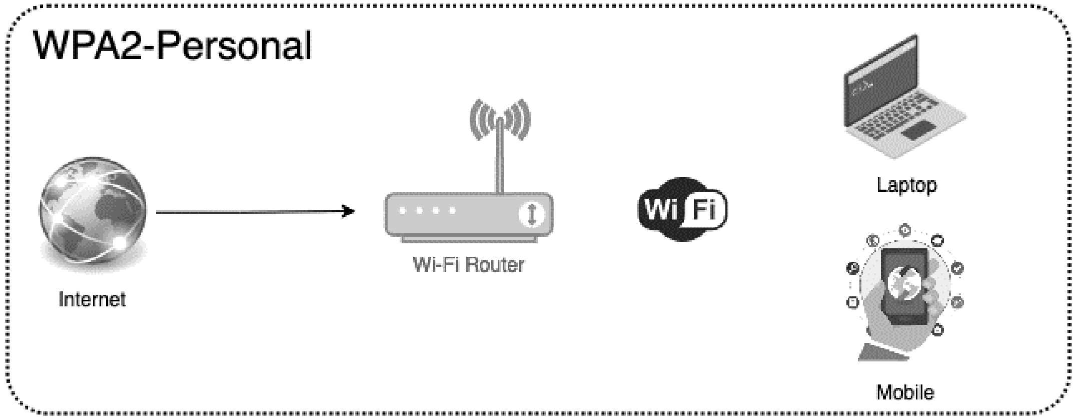


Supplicant: This is the application running on the endpoint or the client's device. It exchanges messages with the authenticator for authentication and encryption

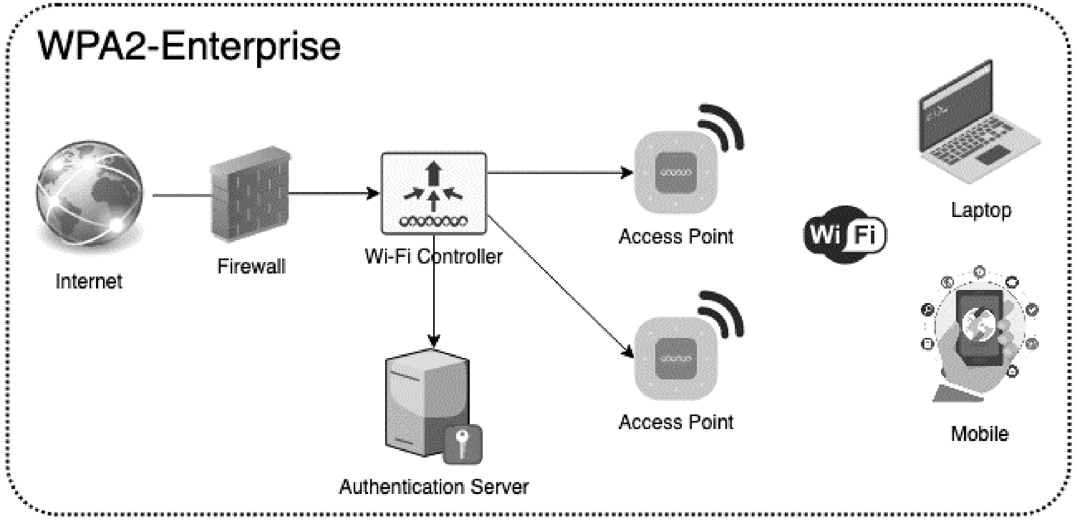
Authenticator: Wireless access point, or wireless LAN controller acts as authenticator who is the middle man between the supplicant and the authentication server.

Authentication Server: Only used for enterprise security. This is responsible for authenticating clients. Authentication servers check the legitimacy of the endpoint and report back to the authenticator with approval or denial.

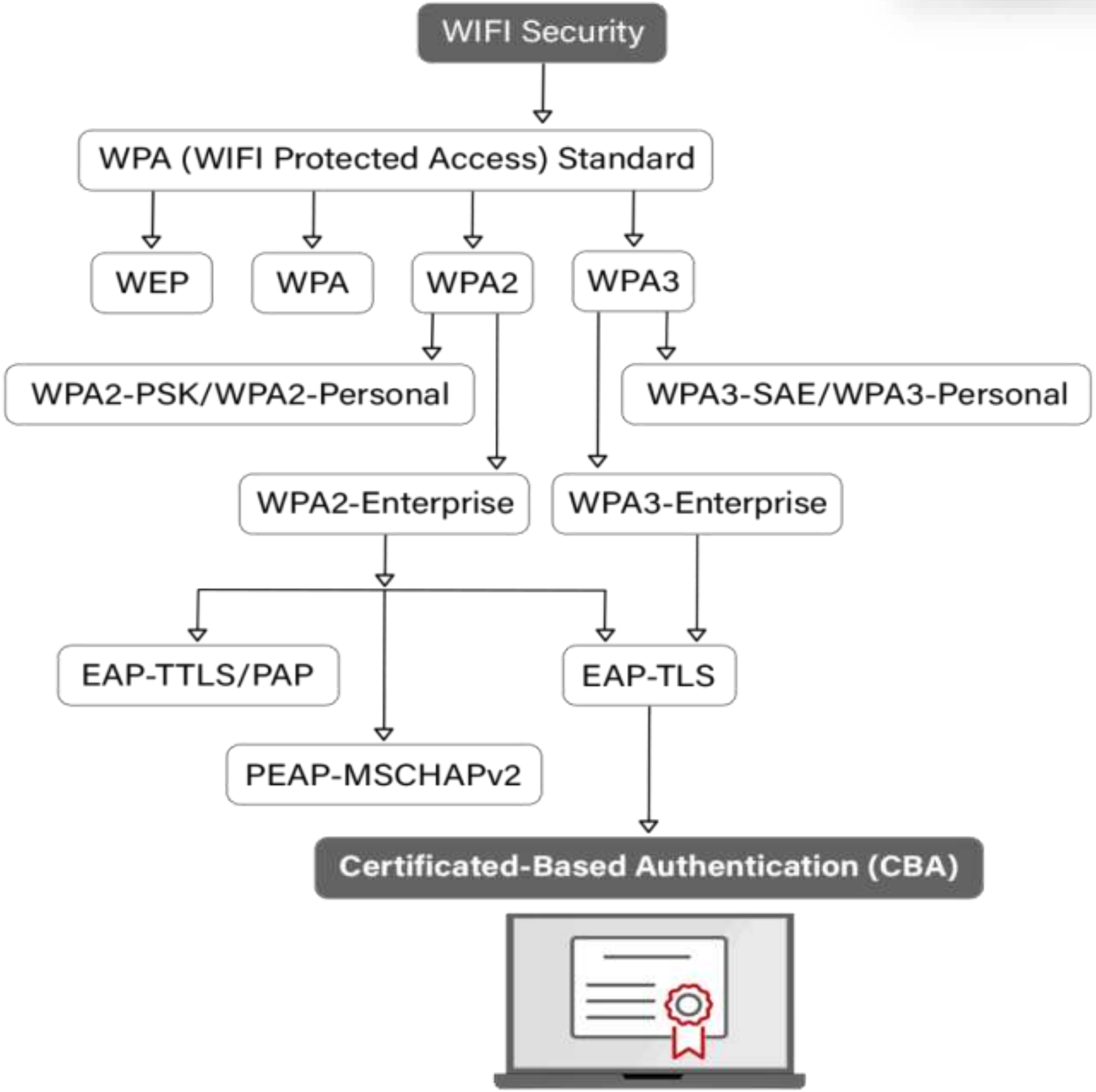
Various Enterprise and Personal Security Methods



WPA2-Personal



WPA2-Enterprise



WIFI Security

WPA (WIFI Protected Access) Standard

WEP

WPA

WPA2

WPA3

WPA2-PSK/WPA2-Personal

WPA3-SAE/WPA3-Personal

WPA2-Enterprise

WPA3-Enterprise

EAP-TTLS/PAP

EAP-TLS

PEAP-MSCHAPv2

Certificated-Based Authentication (CBA)

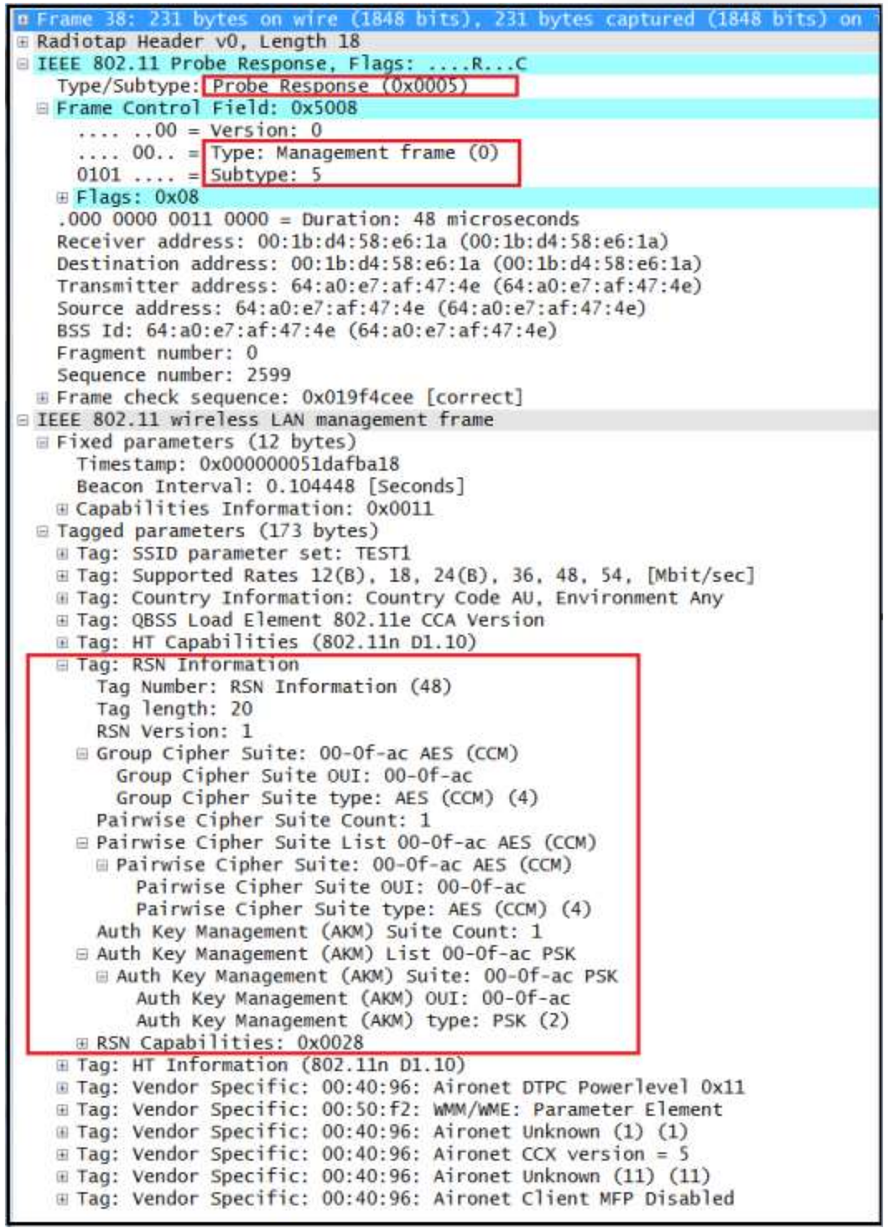
RSN Information Element

RSN or WPA (or both), it includes in its beacon and probe response an Information Element with the following information:

- Whether the access point is using Preshared key or authentication server (key management)
- What group security mechanism is operating
- A list of one or more pairwise key security mechanisms that are supported

	Element ID	Length	Version	Group Data Cipher Suite	Pairwise Cipher Suite Count	Pairwise Cipher Suite List
Octets:	1	1	2	4	2	4 × *m*

	AKM Suite Count	AKM Suite List	RSN Capabilities	PMKIDCount	PMKID List	Group Management Cipher Suite
Octets:	2	4 × *n*	2	2	16 × *s*	4



```
⊟ Frame 38: 231 bytes on wire (1848 bits), 231 bytes captured (1848 bits) on
⊞ Radiotap Header v0, Length 18
⊟ IEEE 802.11 Probe Response, Flags: ....R...C
    Type/Subtype: Probe Response (0x0005)
  ⊟ Frame Control Field: 0x5008
      .... ..00 = Version: 0
      .... 00.. = Type: Management frame (0)
      0101 .... = Subtype: 5
    ⊞ Flags: 0x08
    .000 0000 0011 0000 = Duration: 48 microseconds
    Receiver address: 00:1b:d4:58:e6:1a (00:1b:d4:58:e6:1a)
    Destination address: 00:1b:d4:58:e6:1a (00:1b:d4:58:e6:1a)
    Transmitter address: 64:a0:e7:af:47:4e (64:a0:e7:af:47:4e)
    Source address: 64:a0:e7:af:47:4e (64:a0:e7:af:47:4e)
    BSS Id: 64:a0:e7:af:47:4e (64:a0:e7:af:47:4e)
    Fragment number: 0
    Sequence number: 2599
  ⊞ Frame check sequence: 0x019f4cee [correct]
⊟ IEEE 802.11 wireless LAN management frame
  ⊟ Fixed parameters (12 bytes)
      Timestamp: 0x000000051dafba18
      Beacon Interval: 0.104448 [Seconds]
    ⊞ Capabilities Information: 0x0011
  ⊟ Tagged parameters (173 bytes)
    ⊞ Tag: SSID parameter set: TEST1
    ⊞ Tag: Supported Rates 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
    ⊞ Tag: Country Information: Country Code AU, Environment Any
    ⊞ Tag: QBSS Load Element 802.11e CCA Version
    ⊞ Tag: HT Capabilities (802.11n D1.10)
    ⊟ Tag: RSN Information
        Tag Number: RSN Information (48)
        Tag length: 20
        RSN Version: 1
      ⊟ Group Cipher Suite: 00-0f-ac AES (CCM)
          Group Cipher Suite OUI: 00-0f-ac
          Group Cipher Suite type: AES (CCM) (4)
        Pairwise Cipher Suite Count: 1
      ⊟ Pairwise Cipher Suite List 00-0f-ac AES (CCM)
        ⊟ Pairwise Cipher Suite: 00-0f-ac AES (CCM)
            Pairwise Cipher Suite OUI: 00-0f-ac
            Pairwise Cipher Suite type: AES (CCM) (4)
        Auth Key Management (AKM) Suite Count: 1
      ⊟ Auth Key Management (AKM) List 00-0f-ac PSK
        ⊟ Auth Key Management (AKM) Suite: 00-0f-ac PSK
            Auth Key Management (AKM) OUI: 00-0f-ac
            Auth Key Management (AKM) type: PSK (2)
      ⊞ RSN Capabilities: 0x0028
    ⊞ Tag: HT Information (802.11n D1.10)
    ⊞ Tag: Vendor Specific: 00:40:96: Aironet DTPC Powerlevel 0x11
    ⊞ Tag: Vendor Specific: 00:50:f2: WMM/WME: Parameter Element
    ⊞ Tag: Vendor Specific: 00:40:96: Aironet Unknown (1) (1)
    ⊞ Tag: Vendor Specific: 00:40:96: Aironet CCX version = 5
    ⊞ Tag: Vendor Specific: 00:40:96: Aironet Unknown (11) (11)
    ⊞ Tag: Vendor Specific: 00:40:96: Aironet Client MFP Disabled
```

From Passphrase to Key Generation - Personal

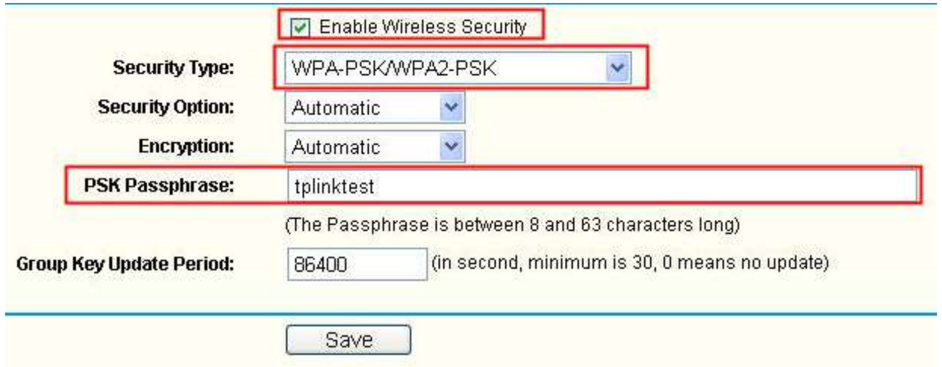
1) Passphrase is known to both AP and supplicant.

2) PSK Gets generated from the Passphrase from the following function. We need passphrase and SSID to generate the PSK.

PSK = pbkdf2.pbkdf2(str.encode(passphrase), str.encode(SSID), 4096, 32)

3) PMK gets generated from the below function which uses HMAC-SHA1 to encode the data. If an 802.1X EAP exchange was carried out, the PMK is derived from the EAP parameters provided by the authentication server.

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



☑ Enable Wireless Security

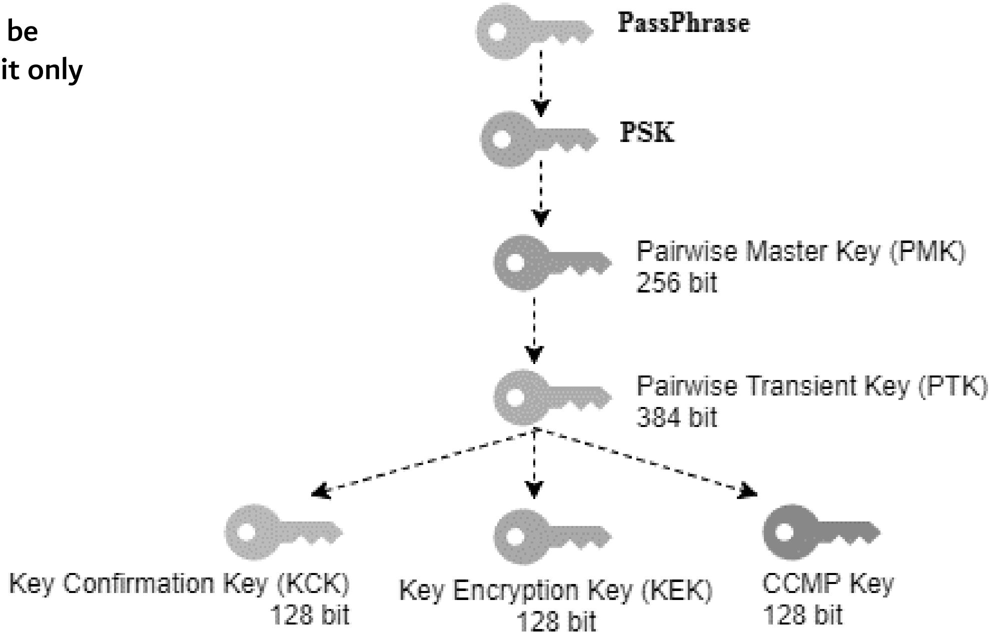
Security Type:	WPA-PSK/WPA2-PSK
Security Option:	Automatic
Encryption:	Automatic
PSK Passphrase:	tplinktest
	(The Passphrase is between 8 and 63 characters long)
Group Key Update Period:	86400 (in second, minimum is 30, 0 means no update)

4) PTK can be generated with a function (customPRF512) and this function expects few values to be passed as a arguments to regenerate the PTK which is the length of 384-bit, and additional 128-bit only for TKIP Configurations.

PTK = PRF (PMK + Anonce + SNonce + Mac (AA)+ Mac (SA))

5) PTK Consists of multiple keys they are

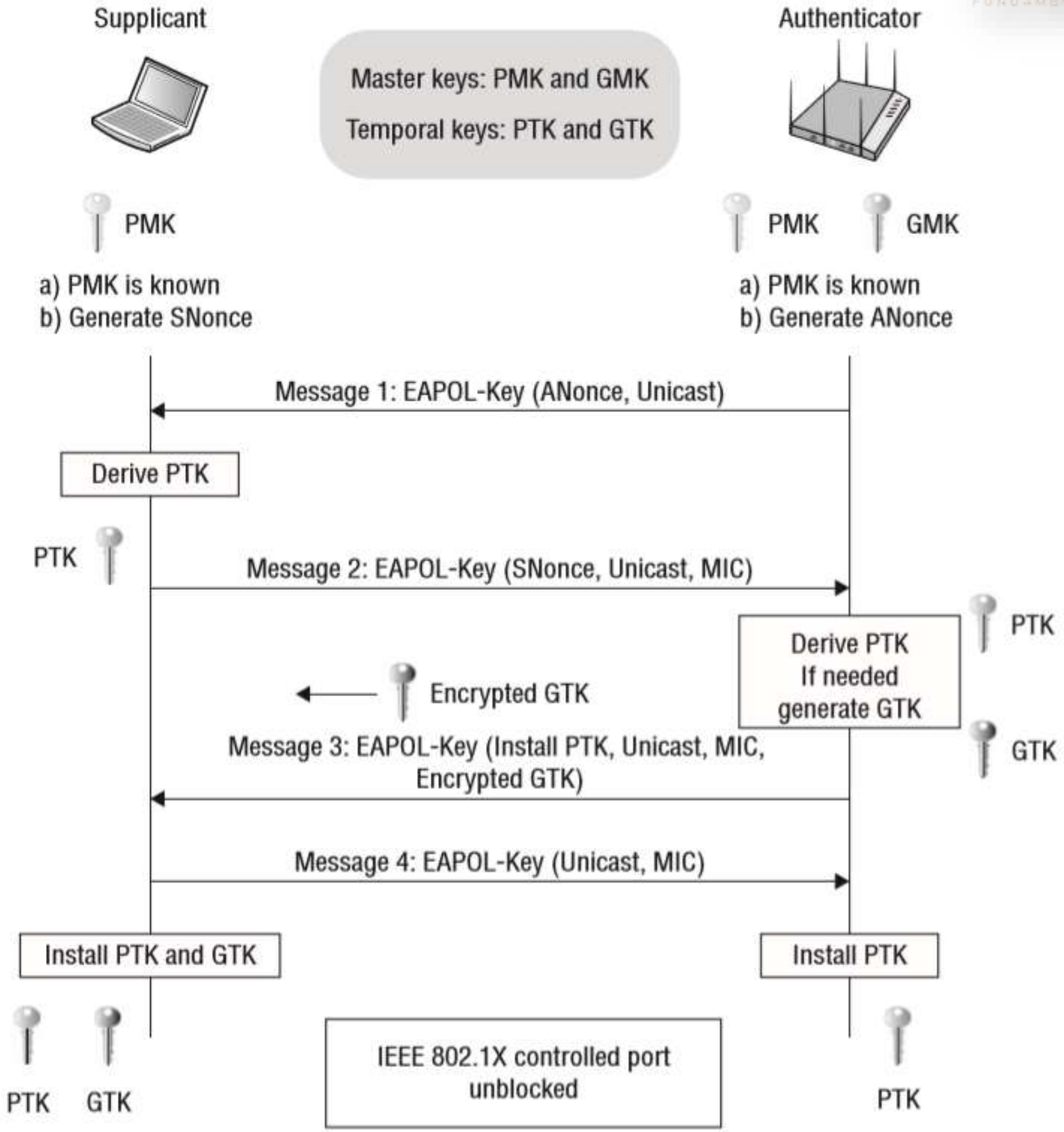
- KEK – Used to encrypt the keys such as GTKs
- KCK – Used during the creation of the MIC, Hash will be generated using KCK.
- TK – Encryption and decryption of unicast packets.
- MIC Tx – Only used with TKIP configurations for unicast packets sent by access points.
- MIC Rx – Only used with TKIP configurations for unicast packets sent by clients.



The 4-way Handshake Process

- Message 1: The authenticator sends its Anonce to the supplicant. The supplicant now has all the information needed to generate the PTK using the pseudo-random function. The PTK protects the unicast data traffic.
- Message 2: The supplicant will send its SNonce to the authenticator. The authenticator now has all the information needed to generate a matching PTK using the pseudo-random function.
- Message 3: The authenticator generates the GTK from the GMK and transfers the GTK to the supplicant. The GTK is encrypted using the PTK and a secure exchange takes place. The GTK protects the broadcast and multicast traffic.
- Message 4: An acknowledgement that the client has successfully installed the PTK and GTK.

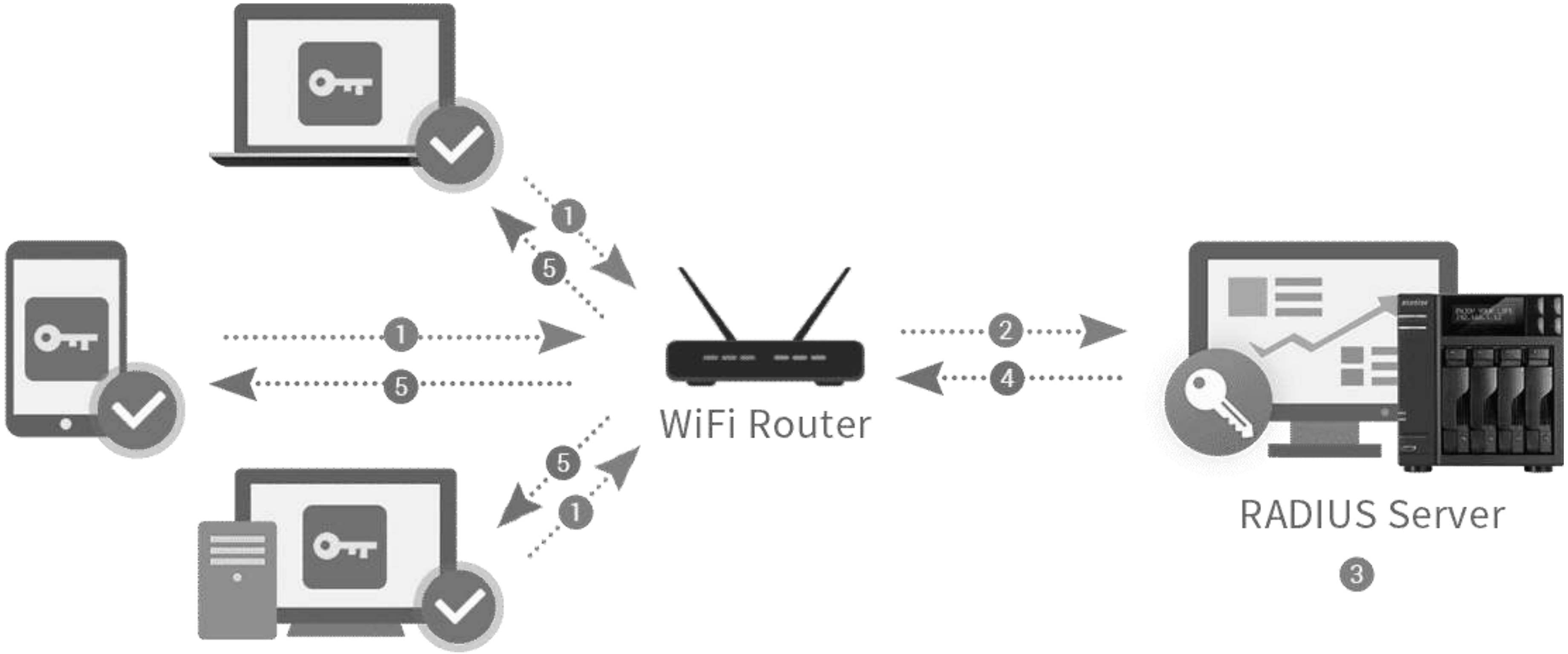
PTK = PRF (PMK + Anonce + SNonce + Mac (AA)+ Mac (SA))



Supplicant
Master keys: PMK and GMK
Temporal keys: PTK and GTK
Authenticator
PMK
PMK
GMK
a) PMK is known
b) Generate SNonce
a) PMK is known
b) Generate ANonce
Message 1: EAPOL-Key (ANonce, Unicast)
Derive PTK
PTK
Message 2: EAPOL-Key (SNonce, Unicast, MIC)
PTK
Derive PTK
If needed
generate GTK
Encrypted GTK
GTK
Message 3: EAPOL-Key (Install PTK, Unicast, MIC, Encrypted GTK)
Message 4: EAPOL-Key (Unicast, MIC)
Install PTK and GTK
Install PTK
PTK
GTK
IEEE 802.1X controlled port unblocked
PTK

Server-Based Authentication

- A possible solution for the security problem is maintaining centralized key servers like a RADIUS server for centralized key generation and distribution
- This would reduce the overhead of maintaining the key information of all the clients at the AP
- With RADIUS, authentication is user-based rather than device-based – for example, a stolen laptop does not necessarily imply a serious security breach
- RADIUS eliminates the need to store and manage authentication data on every AP on the WLAN, making security considerably easier to manage and scale



RADIUS Server

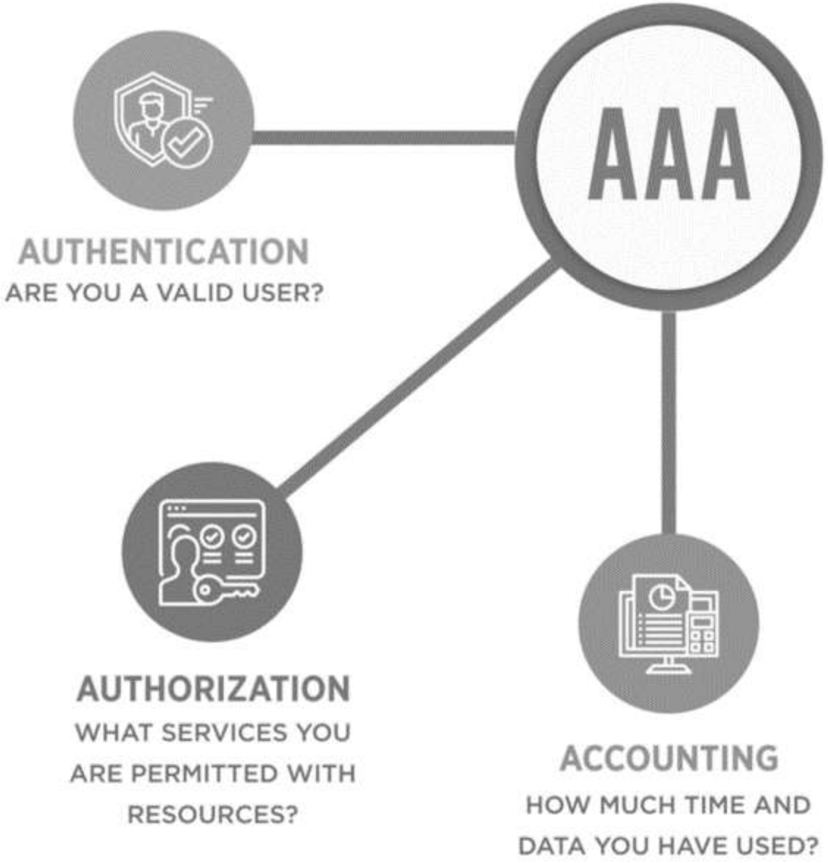
RADIUS (Remote Authentication Dial-In User Service) is a networking protocol used to manage Authentication, Authorization, and Accounting (AAA) for remote users who access a network service. It provides a centralized means of managing network access control and can be used to authenticate users connecting to a network through a variety of devices, including routers, firewalls, and VPNs.

The RADIUS protocol uses a RADIUS Server and RADIUS Clients.

Authentication - This refers to the confirmation of the user which can be accomplished via presenting identity and credentials (for example: username and password or OTP or digital certificates.)

Authorization - This refers to the granting of specific types of services or resources based on the authentication process of the user. This helps in giving restricted permissions to the users. These restrictions may be based on the physical location, IP address, or time of access.

Accounting - This refers to the tracking of consumption of resources by the users. This feature can be used independently of RADIUS authentication or authorization. This may be used for management, planning, billing, etc.

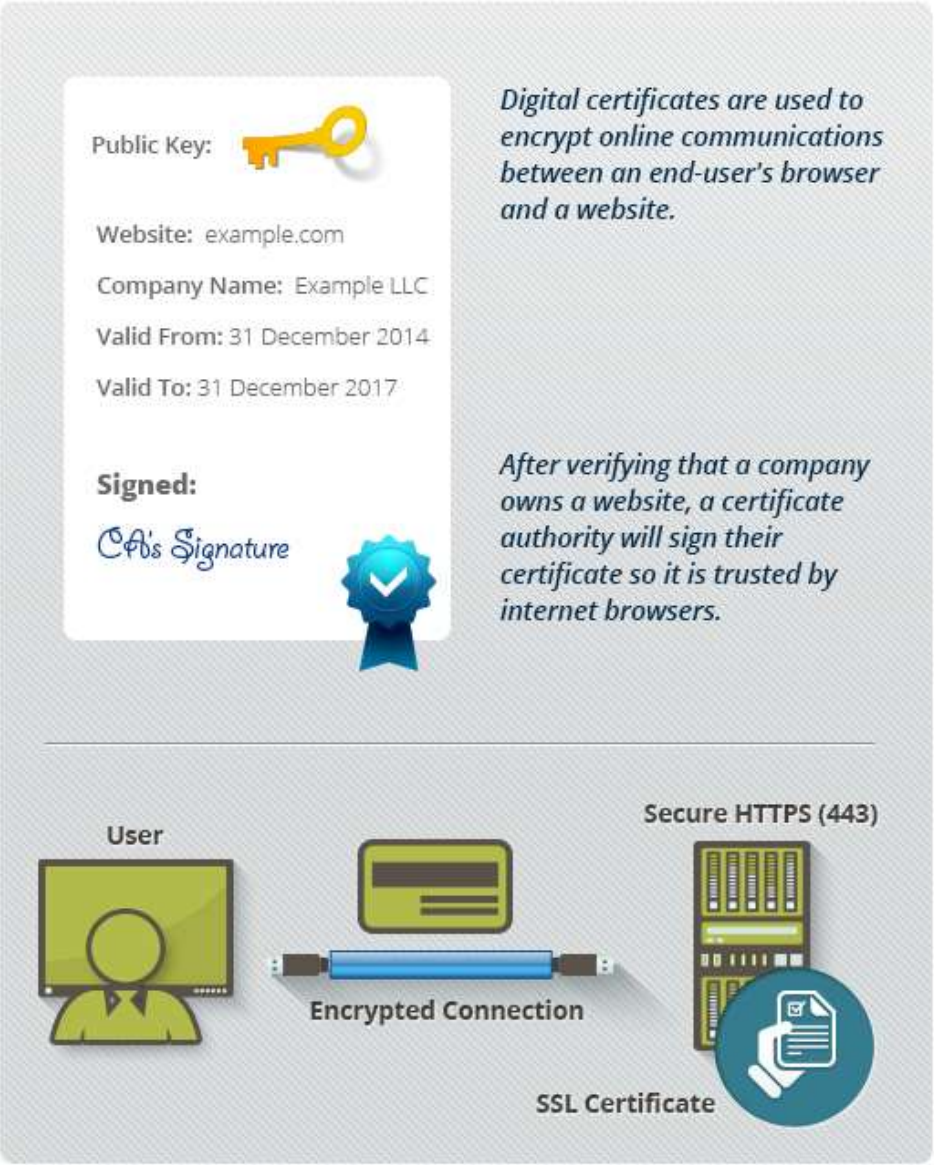

AAA
AUTHENTICATION
ARE YOU A VALID USER?
AUTHORIZATION
WHAT SERVICES YOU ARE PERMITTED WITH RESOURCES?
ACCOUNTING
HOW MUCH TIME AND DATA YOU HAVE USED?

Digital Certificate

A Digital Certificate is an electronic "password" that allows a person, organization to exchange data securely over the Internet using the public key infrastructure (PKI). Digital Certificate is also known as a public key certificate or identity certificate.

What does the certificate contain?

- Your organization’s name and information — The subject field shows that your organization is legitimate and owns the certificate.
- Your public key — This is the half of your public-private key pair that’s publicly known.
- The certificate issuer’s name — This is the name of the certificate authority that issues the certificate.
- The CA’s digital signature — This shows that the certificate was, in fact, issued by a reputable CA.
- A serial number — This is a code that’s unique to your individual SSL/TLS certificate.
- Your certificate’s issuance and expiration dates — These certificates are only valid for a set amount of time — up to 398 days starting Sept. 1, 2020).



Public Key:

Website: example.com

Company Name: Example LLC

Valid From: 31 December 2014

Valid To: 31 December 2017

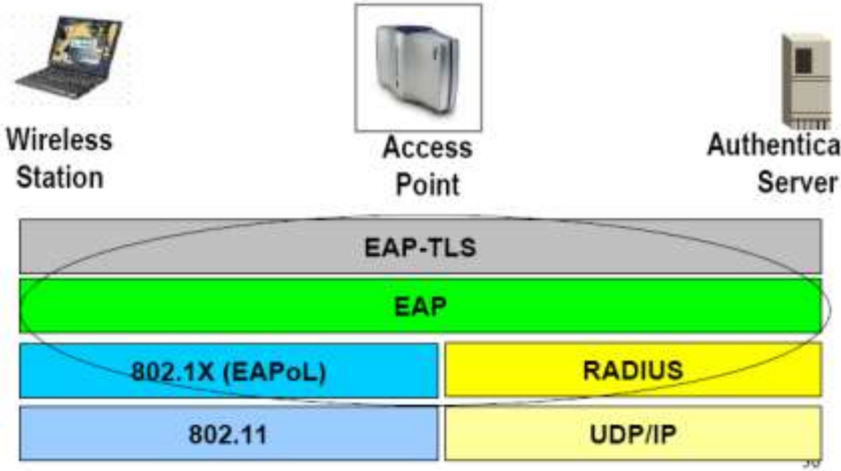
Signed:

CA's Signature

Digital certificates are used to encrypt online communications between an end-user's browser and a website.

After verifying that a company owns a website, a certificate authority will sign their certificate so it is trusted by internet browsers.

Server-Based Security: 802.1x / 802.11i



Wireless Station

Access Point

Authentication Server

EAP-TLS

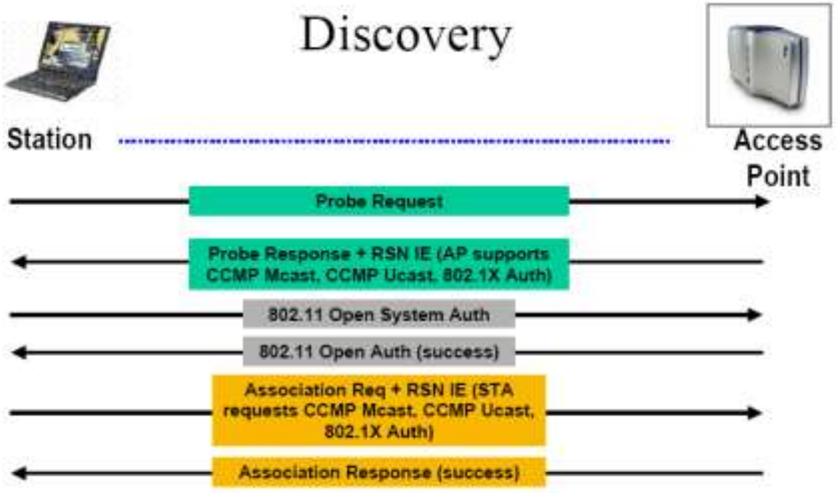
EAP

802.1X (EAPoL)

RADIUS

802.11

UDP/IP



Discovery

Station

Access Point

Probe Request

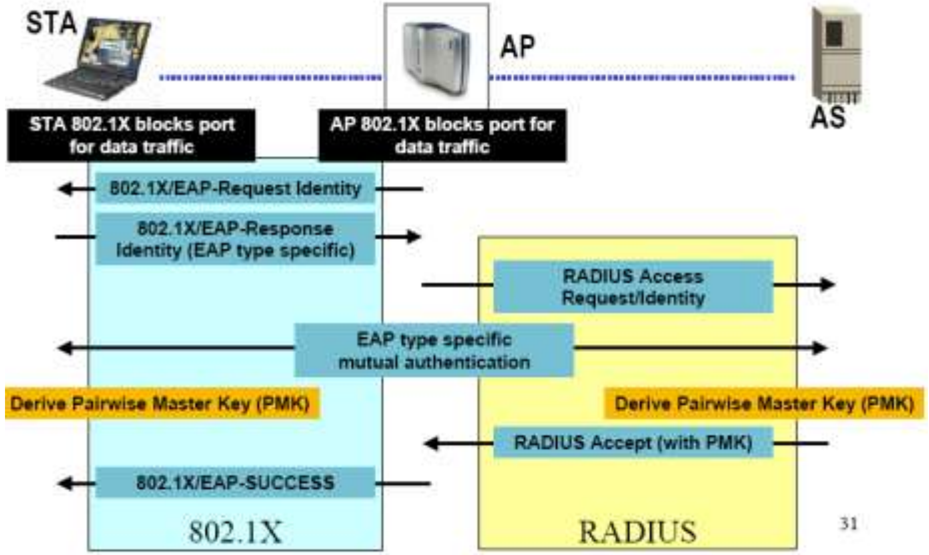
Probe Response + RSN IE (AP supports CCMP Mcast, CCMP Ucast, 802.1X Auth)

802.11 Open System Auth

802.11 Open Auth (success)

Association Req + RSN IE (STA requests CCMP Mcast, CCMP Ucast, 802.1X Auth)

Association Response (success)



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

Derive Pairwise Master Key (PMK)

Derive Pairwise Master Key (PMK)

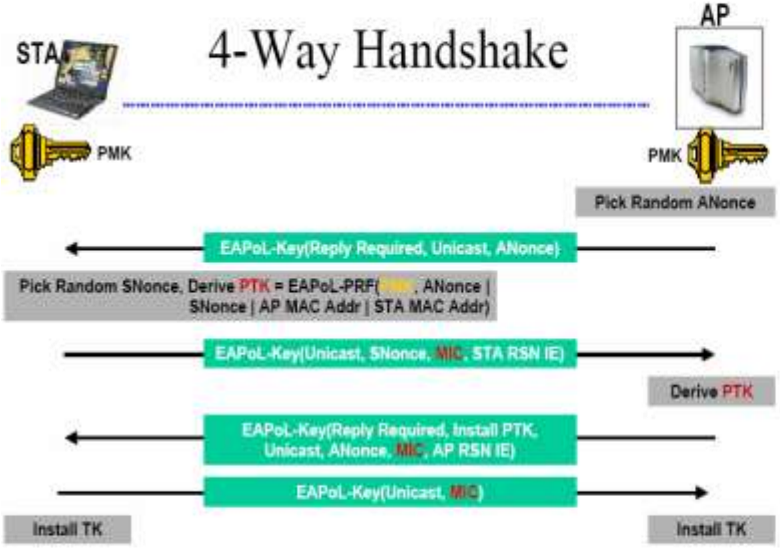
RADIUS Accept (with PMK)

802.1X/EAP-SUCCESS

802.1X

RADIUS

31



4-Way Handshake

STA

AP

PMK

PMK

Pick Random ANonce

EAPoL-Key(Reply Required, Unicast, ANonce)

Pick Random SNonce, Derive PTK = EAPoL-PRF(PMK, ANonce | SNonce | AP MAC Addr | STA MAC Addr)

EAPoL-Key(Unicast, SNonce, MIC, STA RSN IE)

Derive PTK

EAPoL-Key(Reply Required, Install PTK, Unicast, ANonce, MIC, AP RSN IE)

EAPoL-Key(Unicast, MIC)

Install TK

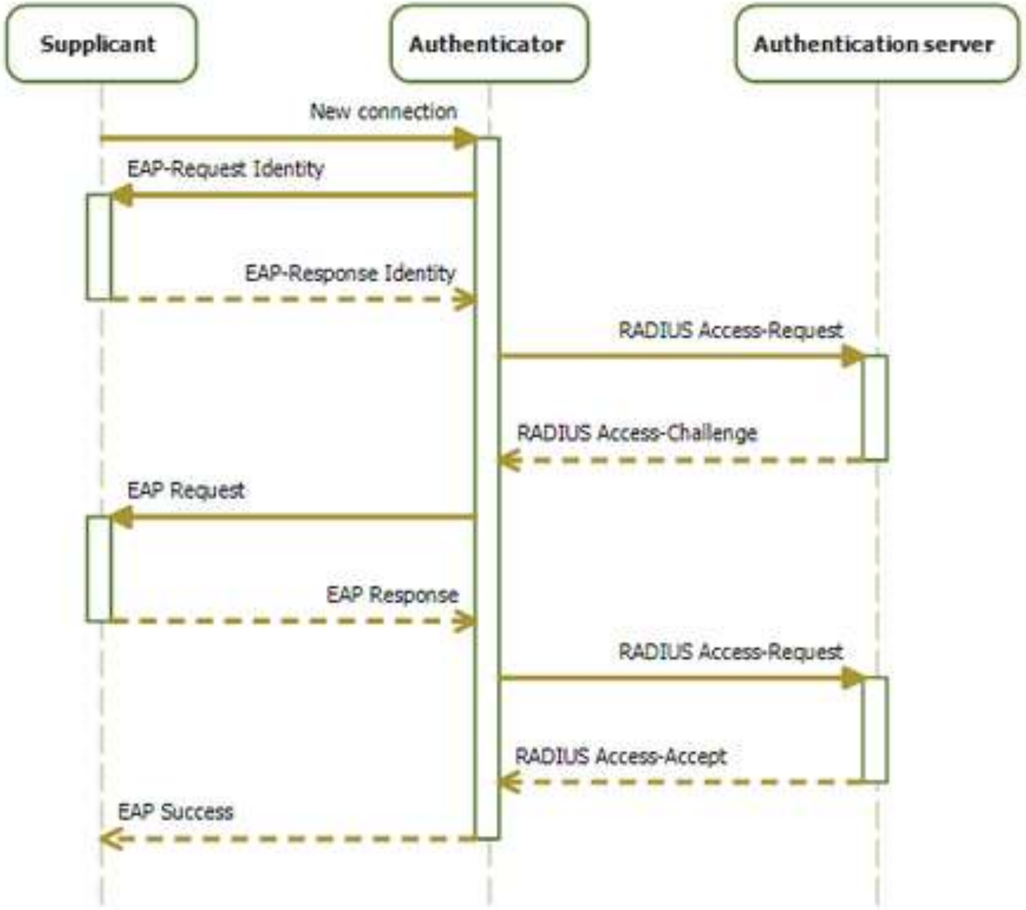
Install TK

802.1x Authentication

IEEE 802.1X defines the encapsulation of the Extensible Authentication Protocol (EAP) over wired IEEE 802 networks and over 802.11 wireless networks, which is known as "EAP over LAN" or EAPOL.

The authentication method begins when the client device requests to connect to the network. The authenticator receives the request and creates a virtual port with the supplicant. The authenticator acts as a proxy for the end user, passing authentication information to and from the authentication server on its behalf. The authenticator limits traffic to authentication data to the server. A negotiation takes place, which includes:

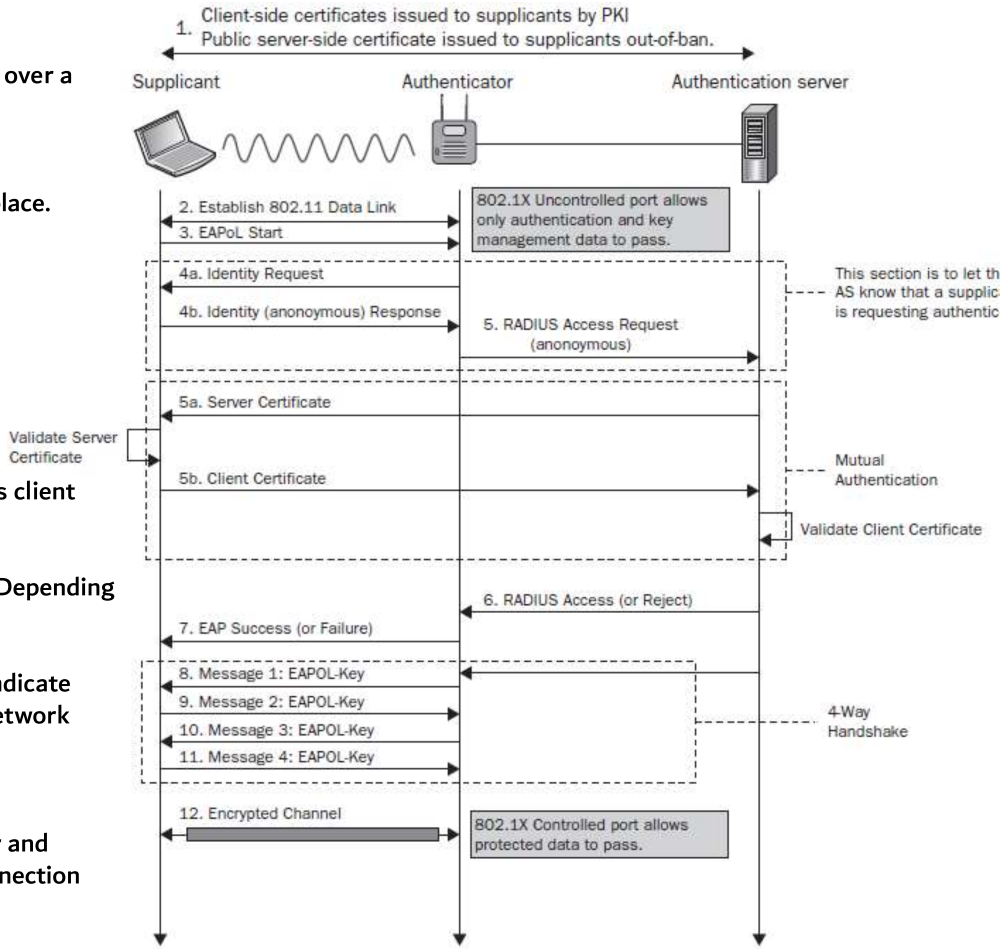
- The client may send an EAP-start message.
- The access point sends an EAP-request identity message.
- The client’s EAP-response packet with the client’s identity is “proxied” to the authentication server by the authenticator.
- The authentication server challenges the client to prove itself and may send its credentials to prove itself to the client (if using mutual authentication).
- The client checks the server’s credentials (if using mutual authentication) and then sends its credentials to the server to prove itself.
- The authentication server accepts or rejects the client’s request for connection.
- If the end user is accepted, the authenticator changes the virtual port with the end user to an authorized state allowing full network access to that end user.
- The client’s virtual port is changed back to the unauthorized state at log-off.



Source: https://study-ccna.com/802-1x-authentication/

EAP-TLS Method

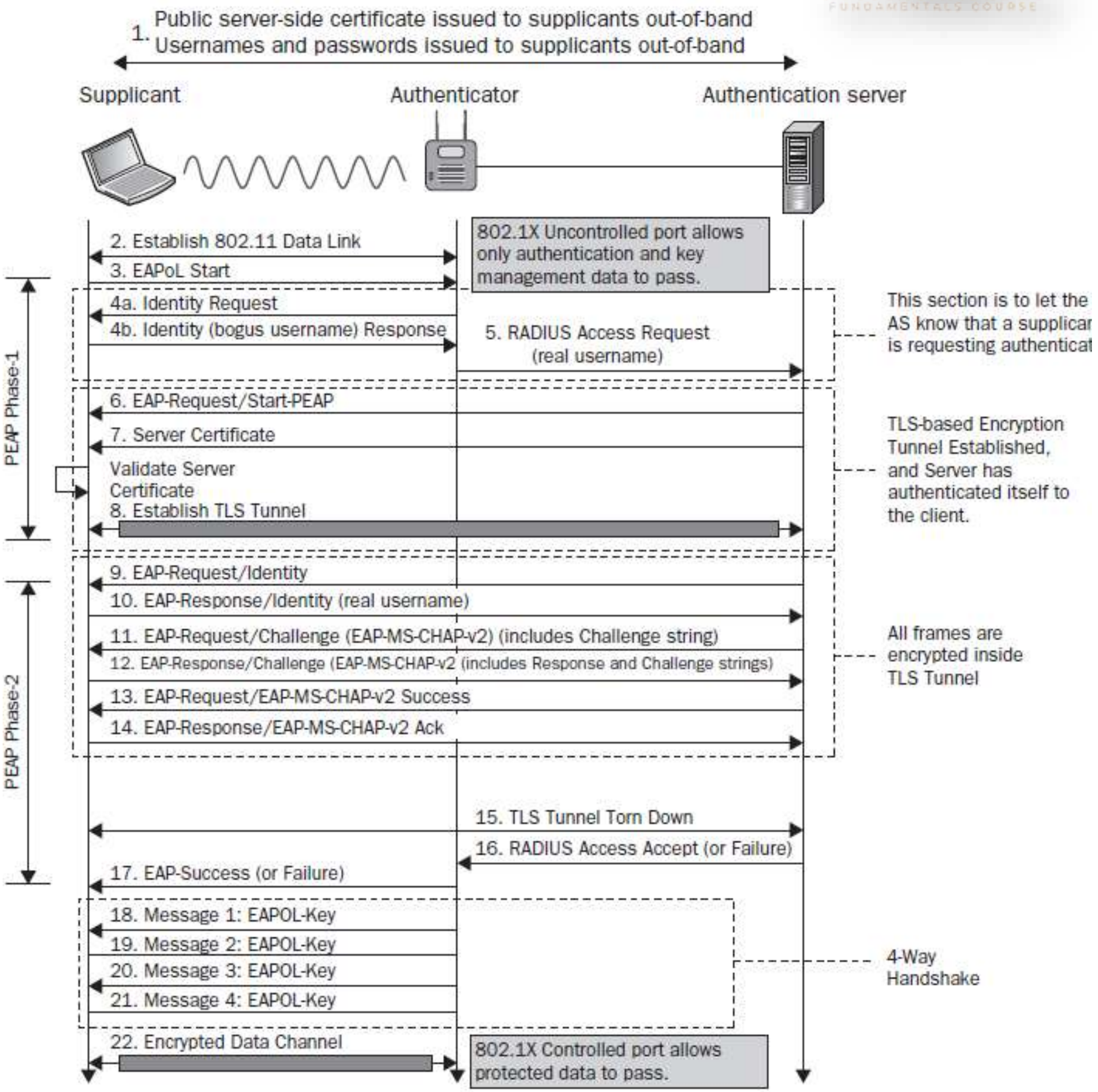
1. Client-side certificates issued to supplicants by PKI, Public server-side certificate issued to supplicants out-of-band
    - The supplicant and the authentication server begin by saying “hello” and prepare their certificates for authentication to establish a trusted connection.
2. Establish 802.11 Data Link
    - The supplicant establishes a connection to the authenticator. This will allow for a secure exchange of information between the two parties.
3. EAPoL Start
    - EAPoL (Extensible Authentication Protocol over LAN) indicates that information can be exchanged between all three parties over a secured LAN channel. Additionally, this is where the authentication method is determined – in this case, EAP-TLS.
4. Identity Section
    - 4a. Identity Request :
        - The supplicant requests the identity of the authenticator to ensure it is sending the client certificate to the correct place.
    - 4b. Identity (anonymous) Response
        - The authenticator requests that the supplicant identify itself.
5. RADIUS Access Request (anonymous)
    - The information that identifies the supplicant and authenticator is sent to the RADIUS to confirm their identity and allow for authenticating information to be sent.
    - 5a. Server Certificate
        - The RADIUS sends its server certificate to confirm its identity through server certificate validation
    - 5b. Client Certificate
        - The supplicant validates the identity of the authentication server certificate. After validation, the supplicant sends its client certificate.
6. RADIUS Access (or Reject)
    - The RADIUS authentication server receives the client certificate and authenticates its identity as an approved network user. Depending on the user’s certificate, the RADIUS sends an Access or Reject message to the authenticator.
7. EAP Success (or Failure)
    - Based on the RADIUS Access or Reject message, the authenticator sends a Success or Failure message to the supplicant to indicate whether they have been approved or denied network access. If the message is Success, the switch port is opened for direct network communication between the supplicant and authentication server.
8. Message 1/2/3: EAPOL-Key
9. Message 4: EAPOL-Key
    - The next step is a series of messages known as the EAPOL-Key exchange. It is a 4 step handshake between the authenticator and supplicant that generates encryption keys. These keys are used to encrypt information that will be sent over the wireless connection and ensures that all ongoing network communications are encrypted and cannot be read by outside parties.
    - Linked here is a detailed list of keys that are generated during this handshake.
10. Encrypted Channel
    - The end result of EAP-TLS authentication is an encrypted channel of communication. The user is ready to access the secure network and utilize all resources available to them.


1. Client-side certificates issued to supplicants by PKI
Public server-side certificate issued to supplicants out-of-ban.
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 (anonoymous) Response
5. RADIUS Access Request (anonoymous)
This section is to let th
AS know that a supplic
is requesting authentic
5a. Server Certificate
Validate Server Certificate
5b. Client Certificate
Mutual Authentication
Validate Client Certificate
6. RADIUS Access (or Reject)
7. EAP Success (or Failure)
8. Message 1: EAPOL-Key
9. Message 2: EAPOL-Key
10. Message 3: EAPOL-Key
11. Message 4: EAPOL-Key
4-Way Handshake
12. Encrypted Channel
802.1X Controlled port allows protected data to pass.

Source: https://www.securew2.com/blog/802-1x-eap-tls-authentication-flow-explained

EAP-PEAP

- Developed by Microsoft, Cisco & RSA Security.
- Referred as EAP within EAP.
- 3 major versions of PEAP:
  - EAP-PEAPv0(EAP-MSCHAPv2) => most widely used
  - EAP-PEAPv0(EAP-TLS)
  - EAP-PEAPv1(EAP-GTC)
- PEAPv0 & PEAPv1 refer to the outer authentication method and are the mechanism that create the secure TLS tunnel to protect subsequent authentication transaction.
- EAP protocol inside parenthesis (i.e. MSCHAPv2, TLS & GTC) is the Inner Authentication/EAP Protocol.
- Identity (client's username) should not be sent in cleartext, only an "anonymous" identity should be sent to server before TLS tunnel establishment.



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

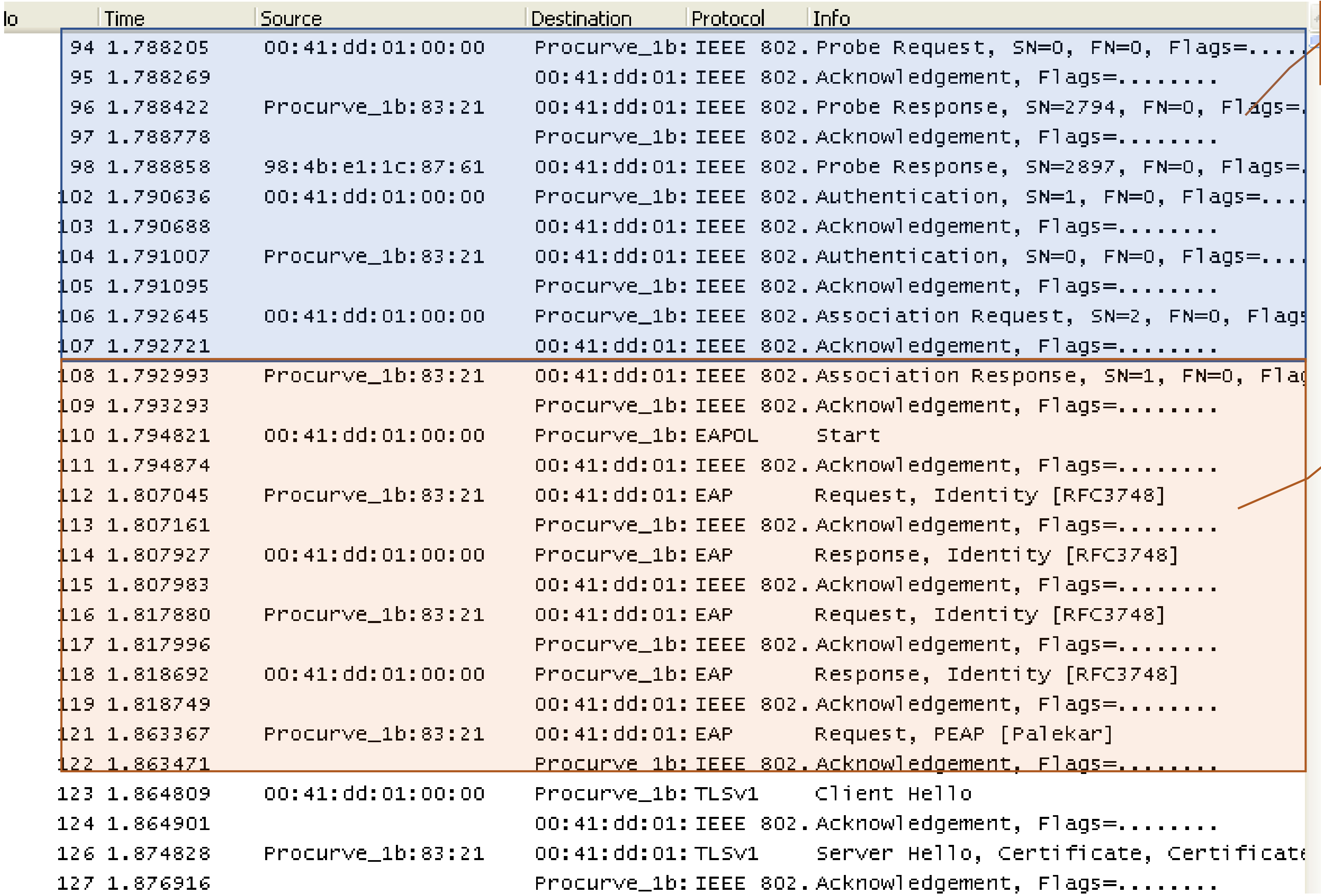
21. Message 4: EAPOL-Key

4-Way Handshake

22. Encrypted Data Channel

802.1X Controlled port allows protected data to pass.

802.1x connection handshakes

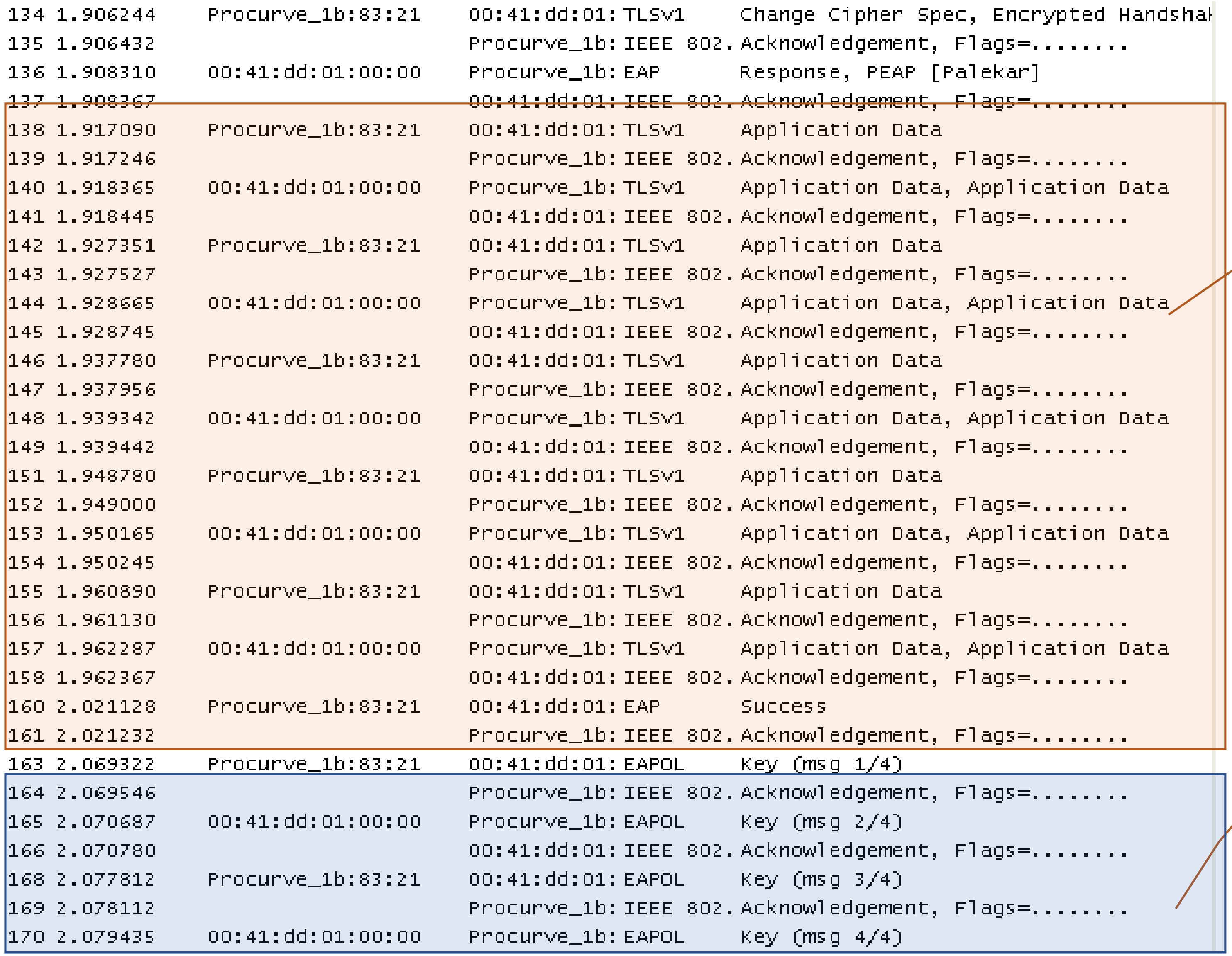


```
lo   Time      Source             Destination  Protocol  Info
 94 1.788205   00:41:dd:01:00:00  Procurve_1b: IEEE 802. Probe Request, SN=0, FN=0, Flags=....
 95 1.788269                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
 96 1.788422   Procurve_1b:83:21  00:41:dd:01: IEEE 802. Probe Response, SN=2794, FN=0, Flags=.
 97 1.788778                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
 98 1.788858   98:4b:e1:1c:87:61  00:41:dd:01: IEEE 802. Probe Response, SN=2897, FN=0, Flags=.
102 1.790636   00:41:dd:01:00:00  Procurve_1b: IEEE 802. Authentication, SN=1, FN=0, Flags=....
103 1.790688                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
104 1.791007   Procurve_1b:83:21  00:41:dd:01: IEEE 802. Authentication, SN=0, FN=0, Flags=....
105 1.791095                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
106 1.792645   00:41:dd:01:00:00  Procurve_1b: IEEE 802. Association Request, SN=2, FN=0, Flags
107 1.792721                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
108 1.792993   Procurve_1b:83:21  00:41:dd:01: IEEE 802. Association Response, SN=1, FN=0, Flag
109 1.793293                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
110 1.794821   00:41:dd:01:00:00  Procurve_1b: EAPOL     Start
111 1.794874                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
112 1.807045   Procurve_1b:83:21  00:41:dd:01: EAP       Request, Identity [RFC3748]
113 1.807161                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
114 1.807927   00:41:dd:01:00:00  Procurve_1b: EAP       Response, Identity [RFC3748]
115 1.807983                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
116 1.817880   Procurve_1b:83:21  00:41:dd:01: EAP       Request, Identity [RFC3748]
117 1.817996                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
118 1.818692   00:41:dd:01:00:00  Procurve_1b: EAP       Response, Identity [RFC3748]
119 1.818749                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
121 1.863367   Procurve_1b:83:21  00:41:dd:01: EAP       Request, PEAP [Palekar]
122 1.863471                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
123 1.864809   00:41:dd:01:00:00  Procurve_1b: TLSv1     Client Hello
124 1.864901                      00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
126 1.874828   Procurve_1b:83:21  00:41:dd:01: TLSv1     Server Hello, Certificate, Certificate
127 1.876916                      Procurve_1b: IEEE 802. Acknowledgement, Flags=........
```

802.11 Connection Handshakes

802.1x Connection Handshakes

802.1x connection handshakes contd



```
134 1.906244    Procurve_1b:83:21    00:41:dd:01: TLSv1     Change Cipher Spec, Encrypted Handshak
135 1.906432                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
136 1.908310    00:41:dd:01:00:00    Procurve_1b: EAP       Response, PEAP [Palekar]
137 1.908367                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
138 1.917090    Procurve_1b:83:21    00:41:dd:01: TLSv1     Application Data
139 1.917246                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
140 1.918365    00:41:dd:01:00:00    Procurve_1b: TLSv1     Application Data, Application Data
141 1.918445                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
142 1.927351    Procurve_1b:83:21    00:41:dd:01: TLSv1     Application Data
143 1.927527                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
144 1.928665    00:41:dd:01:00:00    Procurve_1b: TLSv1     Application Data, Application Data
145 1.928745                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
146 1.937780    Procurve_1b:83:21    00:41:dd:01: TLSv1     Application Data
147 1.937956                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
148 1.939342    00:41:dd:01:00:00    Procurve_1b: TLSv1     Application Data, Application Data
149 1.939442                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
151 1.948780    Procurve_1b:83:21    00:41:dd:01: TLSv1     Application Data
152 1.949000                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
153 1.950165    00:41:dd:01:00:00    Procurve_1b: TLSv1     Application Data, Application Data
154 1.950245                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
155 1.960890    Procurve_1b:83:21    00:41:dd:01: TLSv1     Application Data
156 1.961130                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
157 1.962287    00:41:dd:01:00:00    Procurve_1b: TLSv1     Application Data, Application Data
158 1.962367                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
160 2.021128    Procurve_1b:83:21    00:41:dd:01: EAP       Success
161 2.021232                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
163 2.069322    Procurve_1b:83:21    00:41:dd:01: EAPOL     Key (msg 1/4)
164 2.069546                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
165 2.070687    00:41:dd:01:00:00    Procurve_1b: EAPOL     Key (msg 2/4)
166 2.070780                         00:41:dd:01: IEEE 802. Acknowledgement, Flags=........
168 2.077812    Procurve_1b:83:21    00:41:dd:01: EAPOL     Key (msg 3/4)
169 2.078112                         Procurve_1b: IEEE 802. Acknowledgement, Flags=........
170 2.079435    00:41:dd:01:00:00    Procurve_1b: EAPOL     Key (msg 4/4)
```

802.1x Connection Handshakes contd.. To generate PMK

4-way handshake to generate session keys from PMK

Comparison of Various EAP Methods

Feature	EAP-MD5	LEAP	EAP-TLS	EAP-FAST	EAP-TTLS	PEAPv0 (EAP-MSCHAPv2)	PEAPv0 (EAP-TLS)	PEAPv1 (EAP-GTC)
Server Certificate	No	No	Yes	Optional (can use PAC instead)	Yes	Yes	Yes	Yes
Client Certificate	No	No	Yes (also supports smartcard)	No	Optional	No	Yes	Optional
Supported Client Authentication	MD5 hash challenge response	MSCHAPv2 challenge/response	Via Certificate/Smart card	MSCHAPv2, GTC	PAP, CHAP, MSCHAPv2, GTC, Certif.	MSCHAPv2	Certificate	GTC
Mutual Authentication	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User Identity Protection	No	No	Yes (anonymous)	Yes (bogus username)	Yes (TLS encryption)	Yes (TLS encryption)	Yes (TLS encryption)	Yes (TLS encryption)
Client Auth in cleartext	Yes (sniffing possible)	Yes (sniffing possible)	No	No	No	No	No	No
Client Auth Handhshake offline cracking	Tool eapmd5pass	Tool Asleap	No	Tool Asleap (for MSCHAPv2)	Tool Asleap (for MSCHAPv2)	Tool Asleap	No	Cleartext (inside TLS tunnel)
Evil Twin Attack Possible ?	Yes	Yes	No	Yes if no server's PAC validation	Yes if no server's certif validation	Yes if no server's certif validation	No	Yes if no server's certificate validation

References

EAP Methods

https://github.com/koutto/pi-pwnbox-rogueap/wiki/07.-WPA-WPA2-Enterprise-%28MGT%29

4-way handshake keys generation and MIC Verification

https://praneethwifi.in/2019/11/09/4-way-hand-shake-keys-generation-and-mic-verification/

4-way handshake

https://wlan1nde.wordpress.com/2014/10/27/4-way-handshake/

802.1x Authentication

https://study-ccna.com/802-1x-authentication/

EAP-TLS Method

https://www.securew2.com/blog/802-1x-eap-tls-authentication-flow-explained

DEMO

Remaining Sessions:

Tue Nov 28th – Session 4c

Tue Dec 05th – Session 4d

Tue Dec 12th – General Q/A and interactive session

Tue Dec 19th – Online Exam

Thu Dec 21st – Certificate Presentation

Q&A

?
QUIZ!
TIME

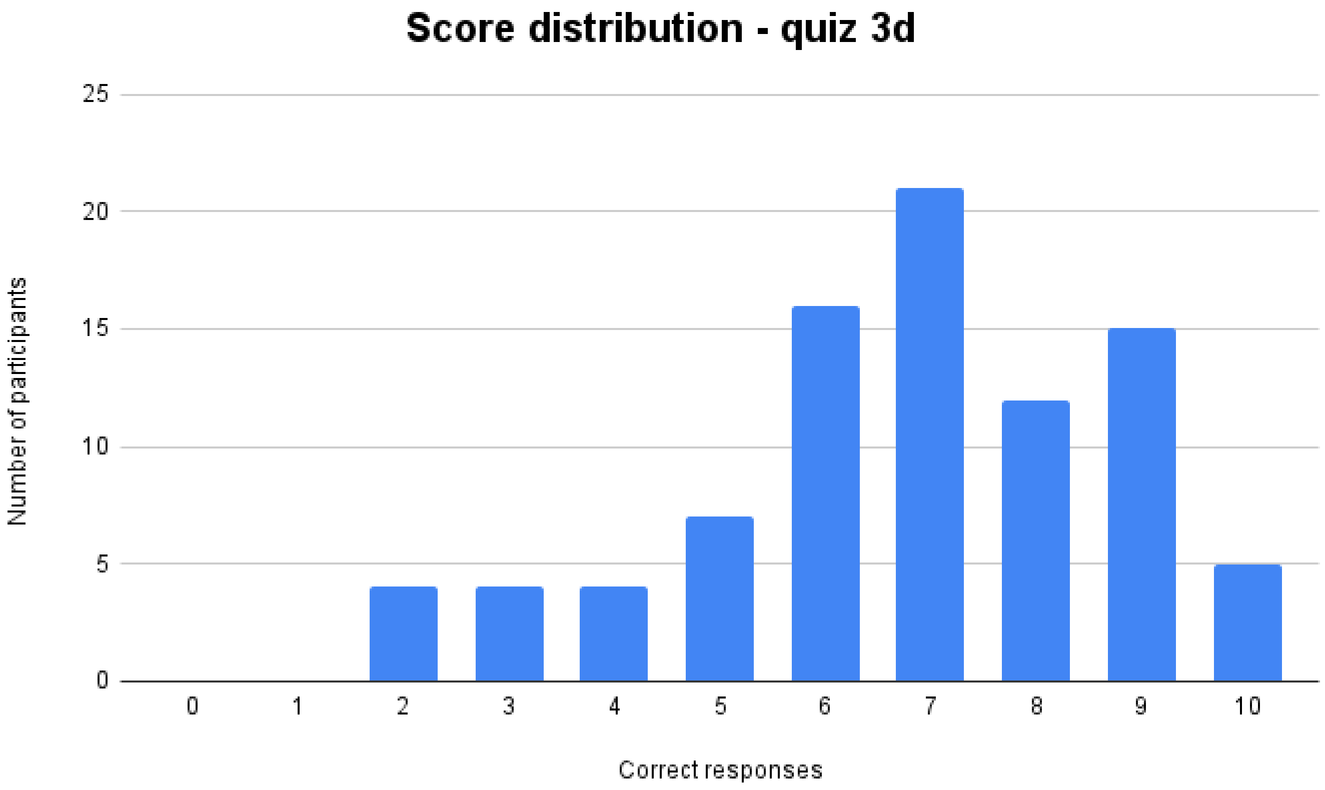
Quiz 3d Results

Winner

Vivekananthan

INDIA

Number of participants - 88


Score distribution - quiz 3d
Number of participants
Correct responses