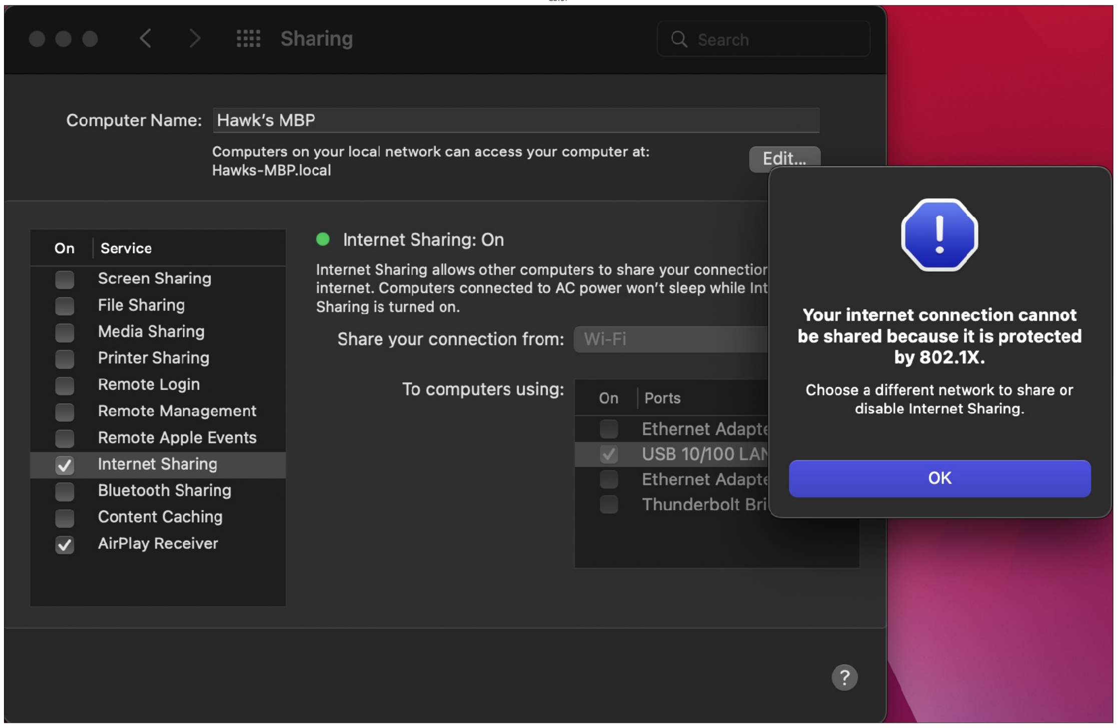
Setup

```
import cv2
from matplotlib import pyplot as plt

# This is a bit of magic to make matplotlib figures appear inline in the notebook
# rather than in a new window.
%matplotlib inline
plt.rcParams['figure.figsize'] = (100.0, 80.0) # set default size of plots
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'


def show_img(img):
    img = cv2.imread(img,-1)
    plt.subplot(131),plt.imshow(img),
    plt.title('Color'),plt.xticks([]), plt.yticks([])
    plt.show()
```

Task 1: Setup an Access Point

```
show_img('part1/Task 1/ap_setup_fail.png')
```

Color



Computer Name: Hawk's MBP

Computers on your local network can access your computer at: Hawks-MBP.local

On	Service
	Screen Sharing
	File Sharing
	Media Sharing
	Printer Sharing
	Remote Login
	Remote Management
	Remote Apple Events
✓	Internet Sharing
	Bluetooth Sharing
	Content Caching
✓	AirPlay Receiver

Internet Sharing: On

Internet Sharing allows other computers to share your connection internet. Computers connected to AC power won't sleep while Int Sharing is turned on.

Share your connection from: Wi-Fi

To computers using:

On	Ports
	Ethernet Adapte
✓	USB 10/100 LAN
	Ethernet Adapte
	Thunderbolt Bri

Your internet connection cannot be shared because it is protected by 802.1X.

Choose a different network to share or disable Internet Sharing.

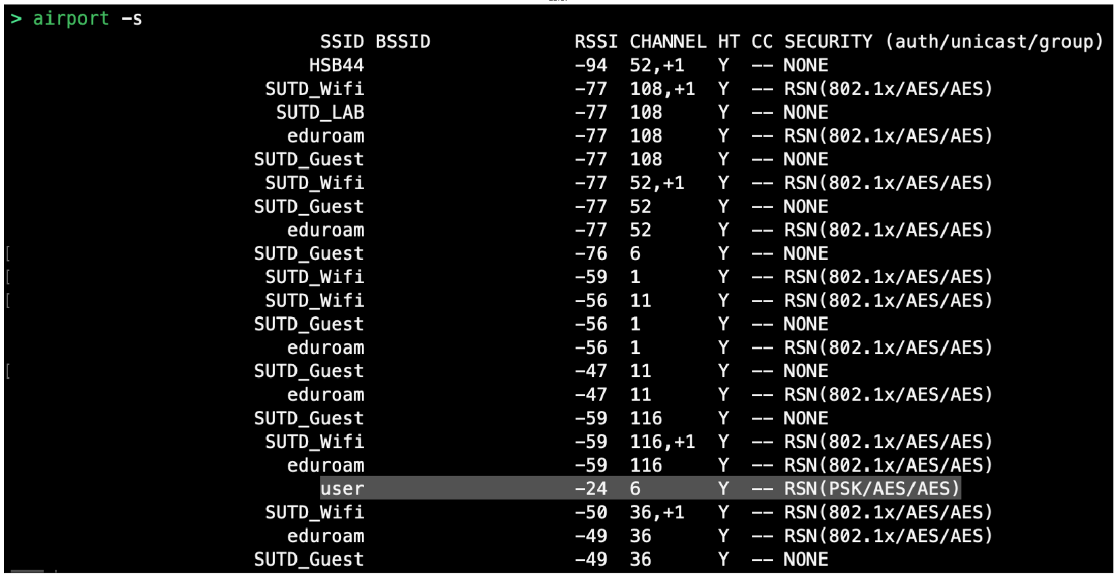
Internet sharing fails on Mac, and configuring wifi sharing is impossible due to the protection of 802.1X SUTD network. A possible workaround to this would be to use a Windows computer and connectify.

https://www.reddit.com/r/mac/comments/2jfdbe/bypass_8021x_to_enable_internet_sharing/

Task 2: Capturing Wireless Packets

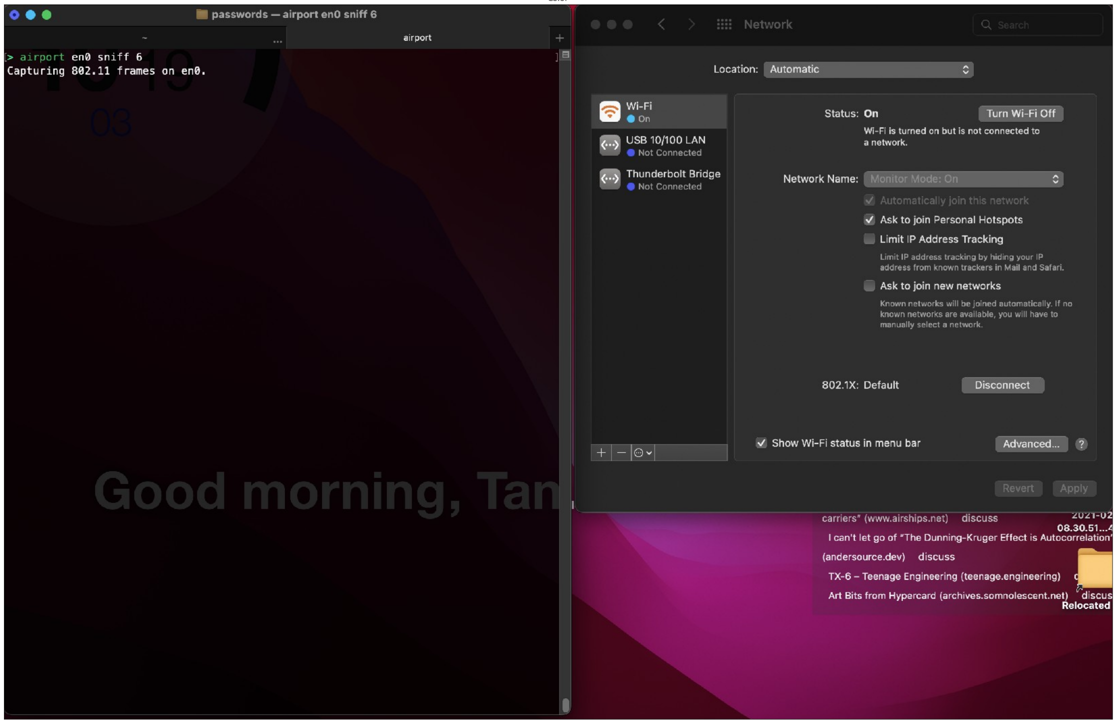
```
show_img('part1/Task 2/ap_targets.png')
show_img('part1/Task 2/monitor_mode.png')
```

Color



```
> airport -s
                            SSID BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
                           HSB44                   -94  52,+1    Y  -- NONE
                       SUTD_Wifi                   -77  108,+1   Y  -- RSN(802.1x/AES/AES)
                        SUTD_LAB                   -77  108      Y  -- NONE
                         eduroam                   -77  108      Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -77  108      Y  -- NONE
                       SUTD_Wifi                   -77  52,+1    Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -77  52       Y  -- NONE
                         eduroam                   -77  52       Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -76  6        Y  -- NONE
                       SUTD_Wifi                   -59  1        Y  -- RSN(802.1x/AES/AES)
                       SUTD_Wifi                   -56  11       Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -56  1        Y  -- NONE
                         eduroam                   -56  1        Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -47  11       Y  -- NONE
                         eduroam                   -47  11       Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -59  116      Y  -- NONE
                       SUTD_Wifi                   -59  116,+1   Y  -- RSN(802.1x/AES/AES)
                         eduroam                   -59  116      Y  -- RSN(802.1x/AES/AES)
                            user                   -24  6        Y  -- RSN(PSK/AES/AES)
                       SUTD_Wifi                   -50  36,+1    Y  -- RSN(802.1x/AES/AES)
                         eduroam                   -49  36       Y  -- RSN(802.1x/AES/AES)
                      SUTD_Guest                   -49  36       Y  -- NONE
```

Color



```
> airport en0 sniff 6
Capturing 802.11 frames on en0.
```

Location: Automatic

Wi-Fi On
USB 10/100 LAN Not Connected
Thunderbolt Bridge Not Connected

Status: On
Wi-Fi is turned on but is not connected to a network.

Network Name: Monitor Mode: On
[✓] Automatically join this network
[✓] Ask to join Personal Hotspots
[ ] Limit IP Address Tracking
Limit IP address tracking by hiding your IP address from known trackers in Mail and Safari.
[ ] Ask to join new networks
Known networks will be joined automatically. If no known networks are available, you will have to manually select a network.

802.1X: Default

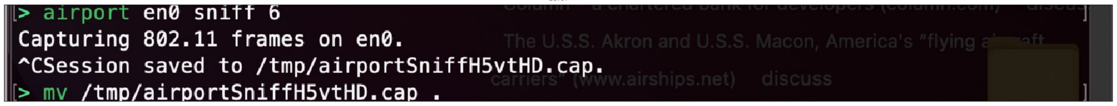
[✓] Show Wi-Fi status in menu bar

Since I will not be able to do the WPA2 wifi configuration, here is an attempt made at finding the Access Points, and the process of capturing the WiFi packets.

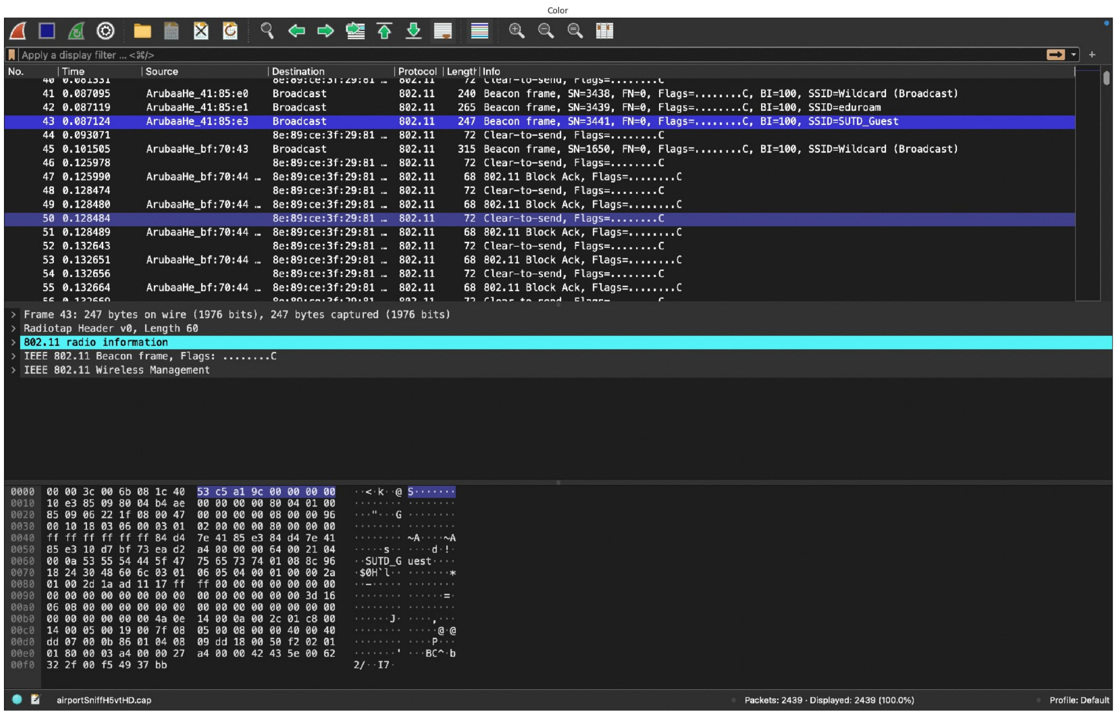
Task 3: Capturing the Four-way Handshake

```
show_img('part1/Task 3/sniff_cmd.png')
show_img('part1/Task 3/wpa.png')
show_img('part1/Task 3/eapol.png')
```

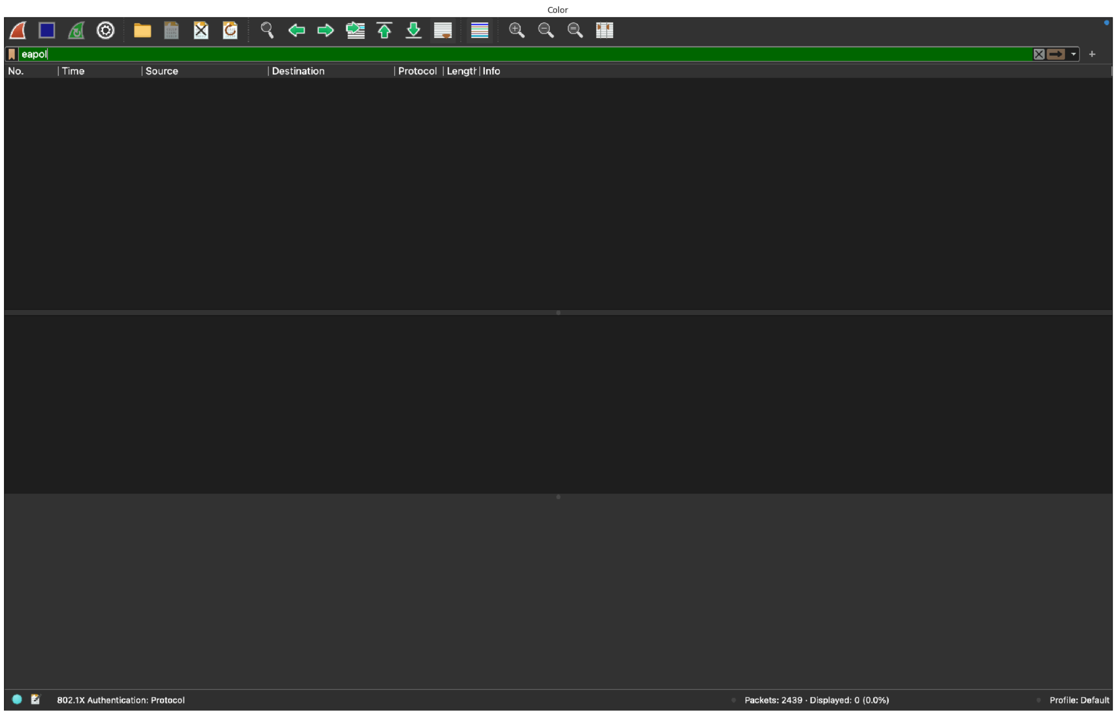
Color



```
> airport en0 sniff 6
Capturing 802.11 frames on en0.
^CSession saved to /tmp/airportSniffH5vtHD.cap.
> mv /tmp/airportSniffH5vtHD.cap .
```



```
No. Time      Source                Destination           Protocol Length Info
 41 0.087095  ArubaaHe_41:85:e0     Broadcast             802.11   240 Beacon frame, SN=3438, FN=0, Flags=........C, BI=100, SSID=Wildcard (Broadcast)
 42 0.087119  ArubaaHe_41:85:e1     Broadcast             802.11   265 Beacon frame, SN=3439, FN=0, Flags=........C, BI=100, SSID=eduroam
 43 0.087124  ArubaaHe_41:85:e3     Broadcast             802.11   247 Beacon frame, SN=3441, FN=0, Flags=........C, BI=100, SSID=SUTD_Guest
 44 0.093071                        8e:89:ce:3f:29:81 …   802.11    72 Clear-to-send, Flags=........C
 45 0.101505  ArubaaHe_bf:70:43     Broadcast             802.11   315 Beacon frame, SN=1650, FN=0, Flags=........C, BI=100, SSID=Wildcard (Broadcast)
 46 0.125978                        8e:89:ce:3f:29:81 …   802.11    72 Clear-to-send, Flags=........C
 47 0.125990  ArubaaHe_bf:70:44 …   8e:89:ce:3f:29:81 …   802.11    68 802.11 Block Ack, Flags=........C
 48 0.128474                        8e:89:ce:3f:29:81 …   802.11    72 Clear-to-send, Flags=........C
 49 0.128480  ArubaaHe_bf:70:44 …   8e:89:ce:3f:29:81 …   802.11    68 802.11 Block Ack, Flags=........C
 50 0.128484                        8e:89:ce:3f:29:81 …   802.11    72 Clear-to-send, Flags=........C
 51 0.128489  ArubaaHe_bf:70:44 …   8e:89:ce:3f:29:81 …   802.11    68 802.11 Block Ack, Flags=........C
 52 0.132643                        8e:89:ce:3f:29:81 …   802.11    72 Clear-to-send, Flags=........C
 53 0.132651  ArubaaHe_bf:70:44 …   8e:89:ce:3f:29:81 …   802.11    68 802.11 Block Ack, Flags=........C
 54 0.132656                        8e:89:ce:3f:29:81 …   802.11    72 Clear-to-send, Flags=........C
 55 0.132664  ArubaaHe_bf:70:44 …   8e:89:ce:3f:29:81 …   802.11    68 802.11 Block Ack, Flags=........C

> Frame 43: 247 bytes on wire (1976 bits), 247 bytes captured (1976 bits)
> Radiotap Header v0, Length 60
> 802.11 radio information
> IEEE 802.11 Beacon frame, Flags: ........C
> IEEE 802.11 Wireless Management

0000  00 00 3c 00 6b 08 1c 40  53 c5 a1 9c 00 00 00 00   ··<·k··@ S·······
0010  10 e3 85 09 80 04 b4 ae  00 00 00 00 80 04 01 00   ········ ········
0020  85 09 06 22 1f 08 00 47  00 00 00 00 08 00 00 96   ···"···G ········
0030  00 10 18 03 06 00 03 01  02 00 00 00 80 00 00 00   ········ ········
0040  ff ff ff ff ff ff 84 d4  7e 41 85 e3 84 d4 7e 41   ········ ~A····~A
0050  85 e3 10 d7 bf 73 ea d2  a4 00 00 00 64 00 21 04   ·····s·· ····d·!·
0060  00 0a 53 55 54 44 5f 47  75 65 73 74 01 08 8c 96   ··SUTD_G uest····
0070  18 24 30 48 60 6c 03 01  06 05 04 00 01 00 00 2a   ·$0H`l·· ·······*
0080  01 00 2d 1a ad 11 17 ff  ff 00 00 00 00 00 00 00   ··-····· ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 3d 16   ········ ······=·
00a0  06 08 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 00 00 00 00 4a 0e  14 00 0a 00 2c 01 c8 00   ······J· ····,···
00c0  14 00 05 00 19 00 7f 08  05 00 08 00 00 40 00 40   ········ ·····@·@
00d0  dd 07 00 0b 86 01 04 08  09 dd 18 00 50 f2 02 01   ········ ····P···
00e0  01 80 00 03 a4 00 00 27  a4 00 00 42 43 5e 00 62   ·······' ···BC^·b
00f0  32 2f 00 f5 49 37 bb                               2/··I7·

airportSniffH5vtHD.cap        Packets: 2439 · Displayed: 2439 (100.0%)        Profile: Default
```



```
eapol
No. | Time | Source | Destination | Protocol | Length | Info

802.1X Authentication: Protocol        Packets: 2439 · Displayed: 0 (0.0%)        Profile: Default
```

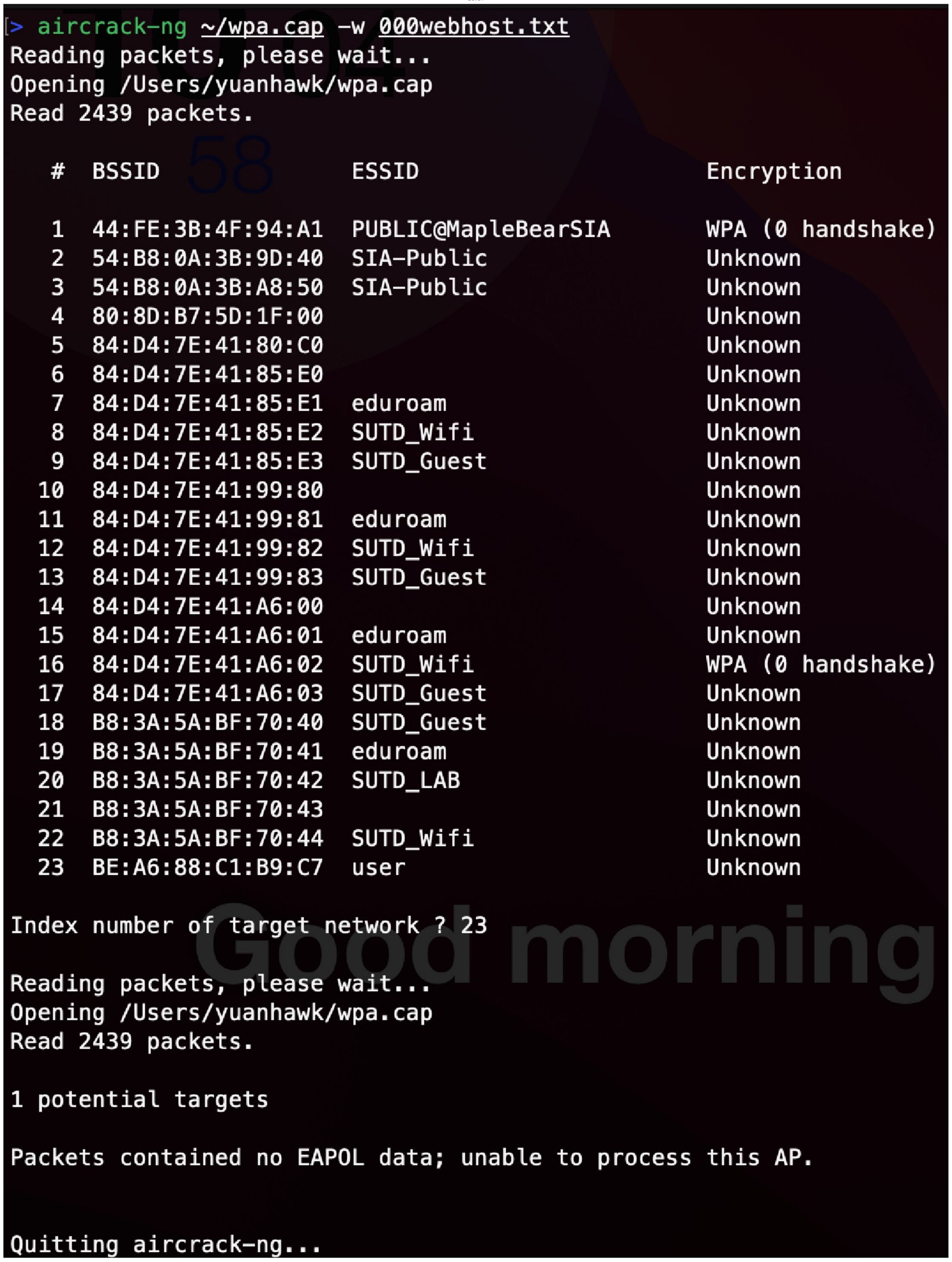
I was able to configure monitor mode on Mac M1 Macbook Pro, but was not successful in the capturing 4-way handshake packets.

It is expected to have no EAPOL data since the hotspot that I created on my Android phone uses WPA2-PSK which is not a WPA2 network.

Task 4: Cracking WPA2 WiFi Passphrase Using Aircrack-ng

```
show_img('part1/Task 4/crack_attempt.png')
```

Color



```
[> aircrack-ng ~/wpa.cap -w 000webhost.txt
Reading packets, please wait...
Opening /Users/yuanhawk/wpa.cap
Read 2439 packets.

   #  BSSID              ESSID                     Encryption

   1  44:FE:3B:4F:94:A1  PUBLIC@MapleBearSIA       WPA (0 handshake)
   2  54:B8:0A:3B:9D:40  SIA-Public                Unknown
   3  54:B8:0A:3B:A8:50  SIA-Public                Unknown
   4  80:8D:B7:5D:1F:00                            Unknown
   5  84:D4:7E:41:80:C0                            Unknown
   6  84:D4:7E:41:85:E0                            Unknown
   7  84:D4:7E:41:85:E1  eduroam                   Unknown
   8  84:D4:7E:41:85:E2  SUTD_Wifi                 Unknown
   9  84:D4:7E:41:85:E3  SUTD_Guest                Unknown
  10  84:D4:7E:41:99:80                            Unknown
  11  84:D4:7E:41:99:81  eduroam                   Unknown
  12  84:D4:7E:41:99:82  SUTD_Wifi                 Unknown
  13  84:D4:7E:41:99:83  SUTD_Guest                Unknown
  14  84:D4:7E:41:A6:00                            Unknown
  15  84:D4:7E:41:A6:01  eduroam                   Unknown
  16  84:D4:7E:41:A6:02  SUTD_Wifi                 WPA (0 handshake)
  17  84:D4:7E:41:A6:03  SUTD_Guest                Unknown
  18  B8:3A:5A:BF:70:40  SUTD_Guest                Unknown
  19  B8:3A:5A:BF:70:41  eduroam                   Unknown
  20  B8:3A:5A:BF:70:42  SUTD_LAB                  Unknown
  21  B8:3A:5A:BF:70:43                            Unknown
  22  B8:3A:5A:BF:70:44  SUTD_Wifi                 Unknown
  23  BE:A6:88:C1:B9:C7  user                      Unknown

Index number of target network ? 23

Reading packets, please wait...
Opening /Users/yuanhawk/wpa.cap
Read 2439 packets.

1 potential targets

Packets contained no EAPOL data; unable to process this AP.


Quitting aircrack-ng...
```

If the network provided is WPA2, the aircrack-ng will provide the key which is also the simple password.

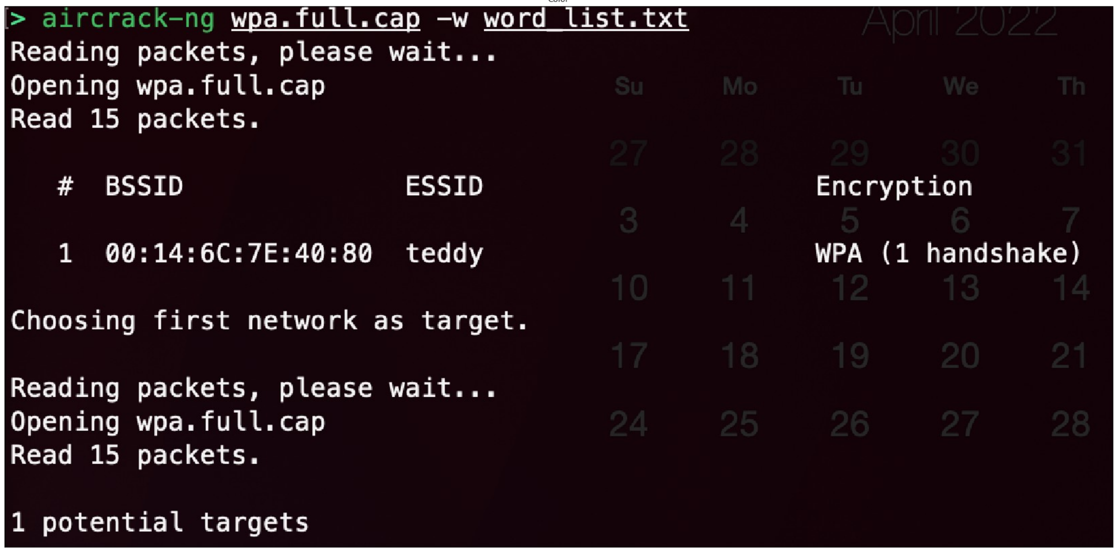
2a. Monitoring mode is used for capturing packets from all surrounding 802.11 devices, and outputs ethernet packets attached with 802.11 header.

Promiscuous mode is used for outputting standard ethernet frames belonging to one wireless network associated with the BSSID. The device will dump packets from AP to wireless devices.

2b. The attacker should capture the four-way handshake of the existing connection, after forcing the deassociation of the client from the AP. The attacker can then use aircrack-ng to launch a dictionary attack on the captured pcap file to retrieve the key.

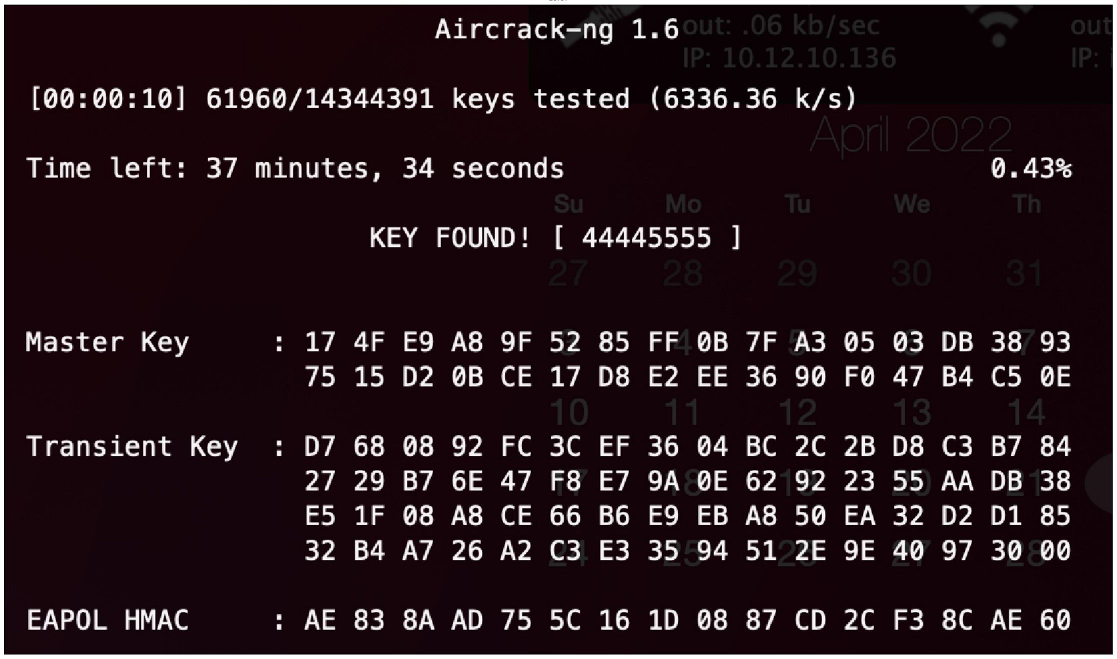
Cracking WPA pcap files provided on eDimension

```
show_img('part1/Edim/crack_full_wpa.png')
show_img('part1/Edim/key_found.png')
```



```
> aircrack-ng wpa.full.cap -w word_list.txt
Reading packets, please wait...
Opening wpa.full.cap
Read 15 packets.

   #  BSSID              ESSID                     Encryption

   1  00:14:6C:7E:40:80  teddy                     WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening wpa.full.cap
Read 15 packets.

1 potential targets
```

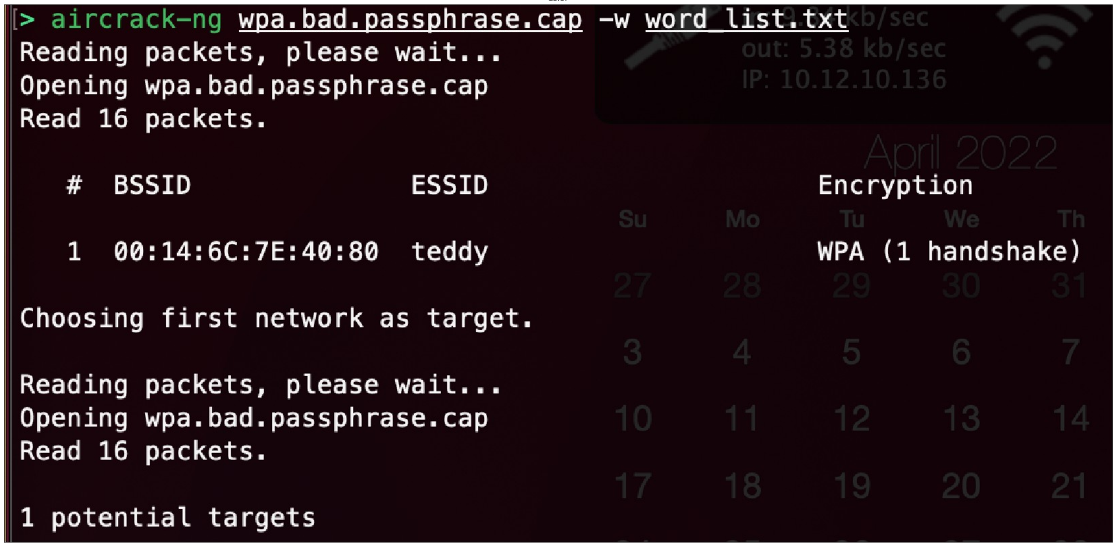
Color



```
                               Aircrack-ng 1.6

      [00:00:10] 61960/14344391 keys tested (6336.36 k/s)

      Time left: 37 minutes, 34 seconds                          0.43%

                           KEY FOUND! [ 44445555 ]


      Master Key     : 17 4F E9 A8 9F 52 85 FF 0B 7F A3 05 03 DB 38 93
                       75 15 D2 0B CE 17 D8 E2 EE 36 90 F0 47 B4 C5 0E

      Transient Key  : D7 68 08 92 FC 3C EF 36 04 BC 2C 2B D8 C3 B7 84
                       27 29 B7 6E 47 F8 E7 9A 0E 62 92 23 55 AA DB 38
                       E5 1F 08 A8 CE 66 B6 E9 EB A8 50 EA 32 D2 D1 85
                       32 B4 A7 26 A2 C3 E3 35 94 51 2E 9E 40 97 30 00

      EAPOL HMAC     : AE 83 8A AD 75 5C 16 1D 08 87 CD 2C F3 8C AE 60
```

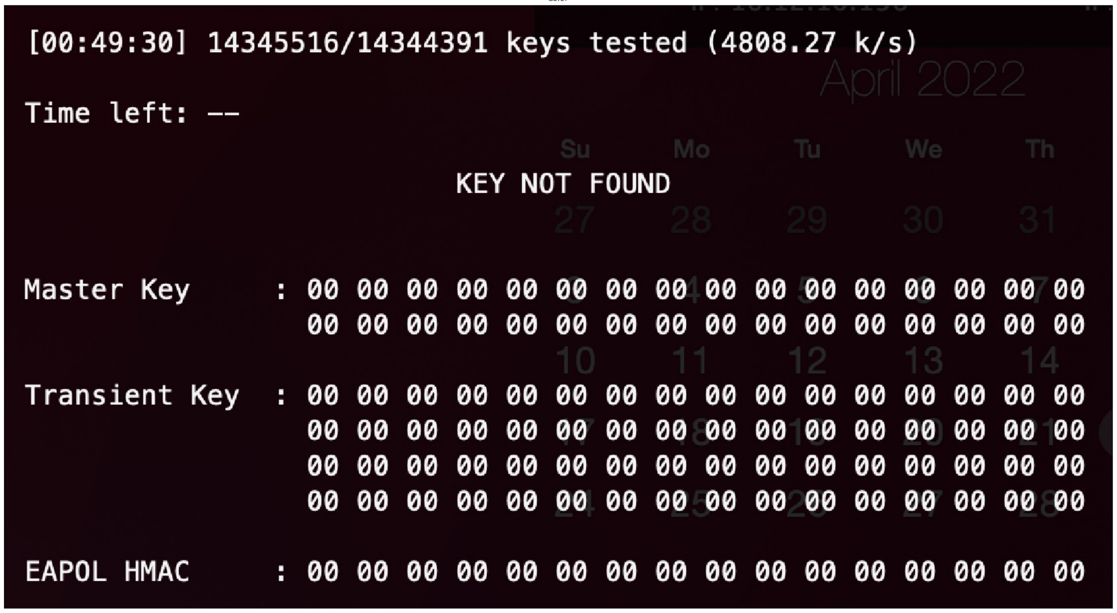
```
show_img('part1/Edim/crack_bad_wpa.png')
show_img('part1/Edim/key_not_found.png')
```

Color



```
> aircrack-ng wpa.bad.passphrase.cap -w word_list.txt
Reading packets, please wait...
Opening wpa.bad.passphrase.cap
Read 16 packets.

   #  BSSID              ESSID                     Encryption

   1  00:14:6C:7E:40:80  teddy                     WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening wpa.bad.passphrase.cap
Read 16 packets.

1 potential targets
```

Color



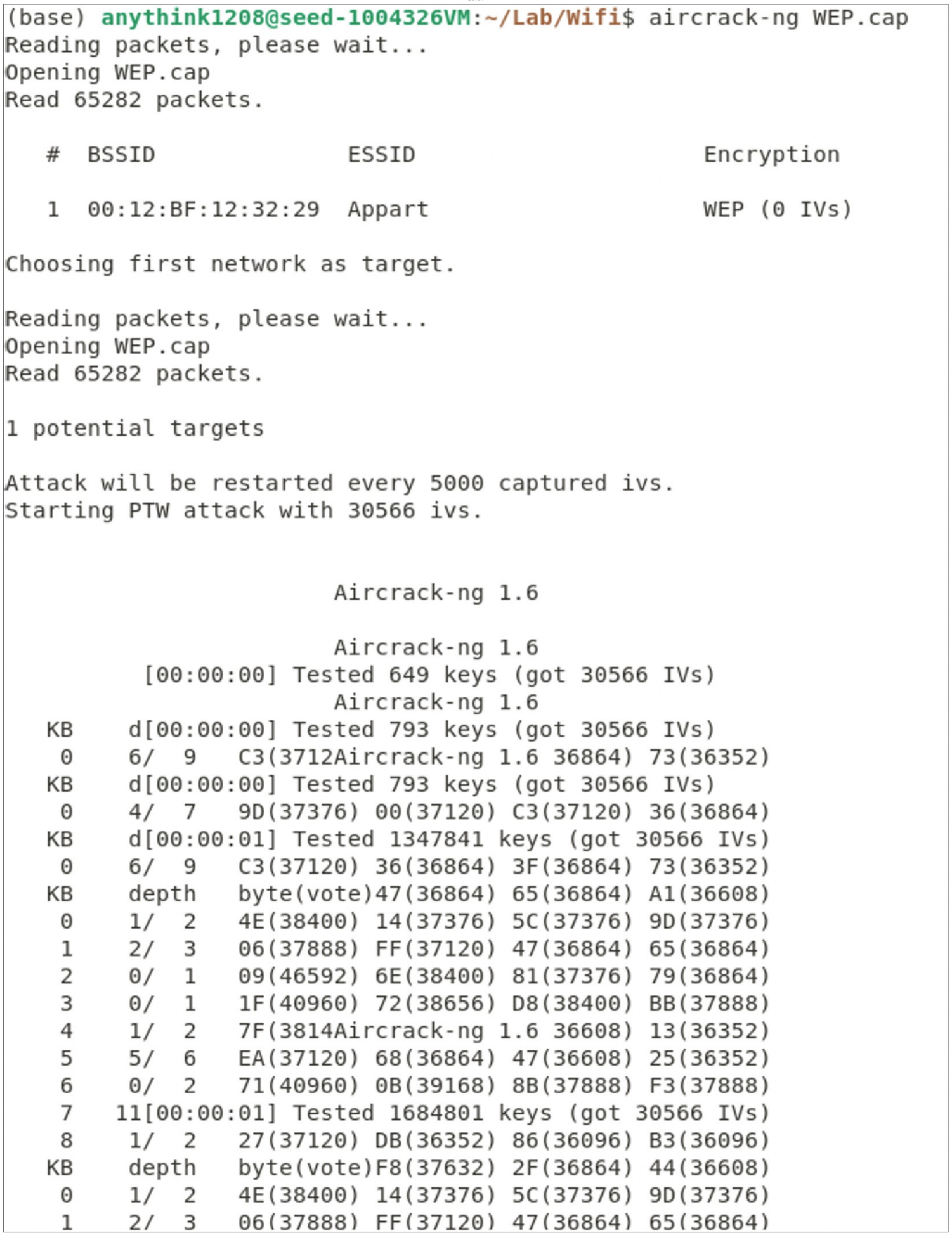
```
[00:49:30] 14345516/14344391 keys tested (4808.27 k/s)

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
```

Task 1: Cracking the WEP Password

1. Cracking broadcast WEP packet, e.g., SN=2000, for the lab.

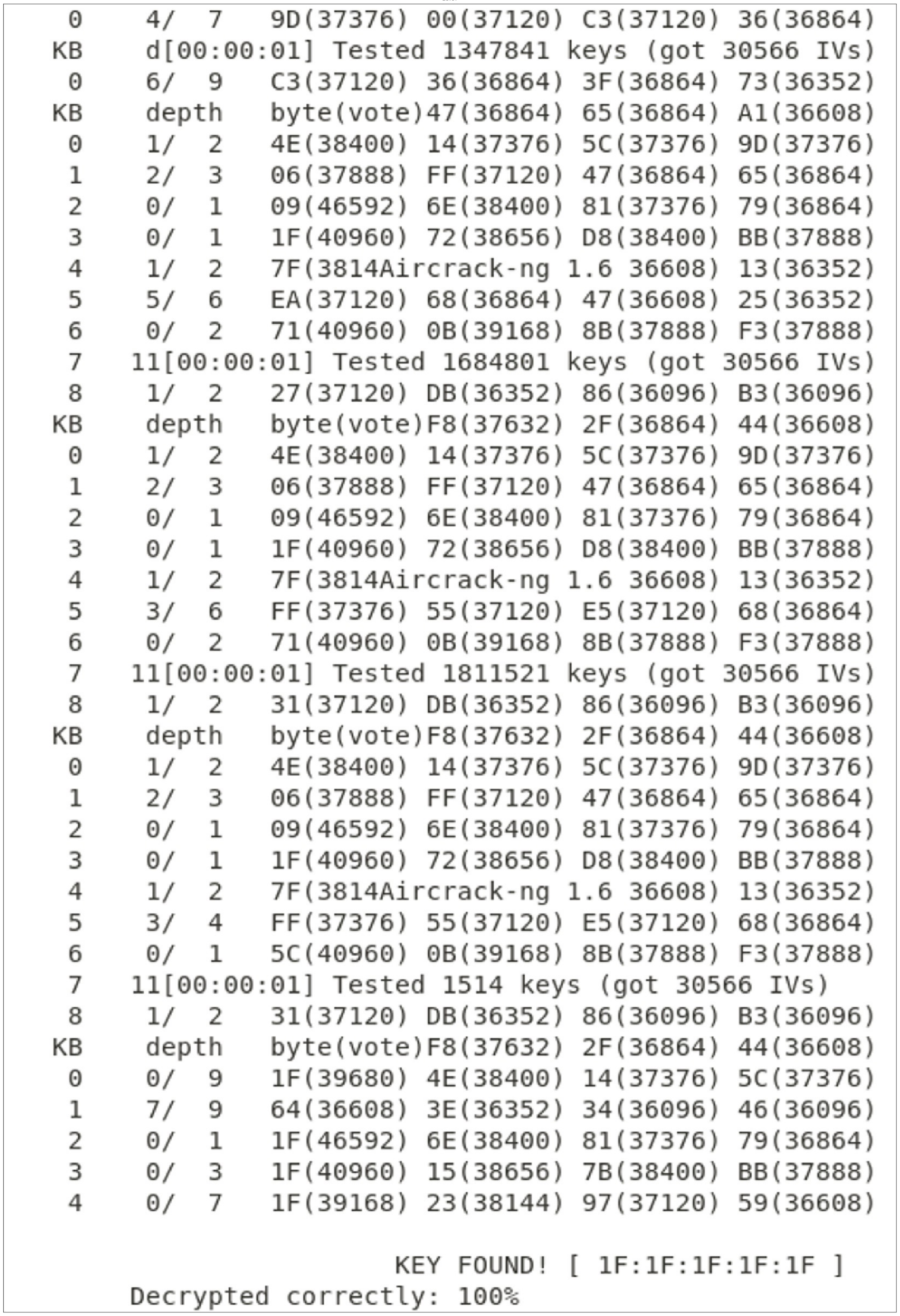
```
show_img('Task 1/aircrack_cmd.png')
show_img('Task 1/aircrack_out.png')
```

Color



```
(base) anythink1208@seed-1004326VM:~/Lab/Wifi$ aircrack-ng WEP.cap
Reading packets, please wait...
Opening WEP.cap
Read 65282 packets.

   #  BSSID              ESSID                     Encryption

   1  00:12:BF:12:32:29  Appart                    WEP (0 IVs)

Choosing first network as target.

Reading packets, please wait...
Opening WEP.cap
Read 65282 packets.

1 potential targets

Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 30566 ivs.


                        Aircrack-ng 1.6

                        Aircrack-ng 1.6
          [00:00:00] Tested 649 keys (got 30566 IVs)
                        Aircrack-ng 1.6
   KB    d[00:00:00] Tested 793 keys (got 30566 IVs)
    0    6/  9   C3(3712Aircrack-ng 1.6 36864) 73(36352)
   KB    d[00:00:00] Tested 793 keys (got 30566 IVs)
    0    4/  7   9D(37376) 00(37120) C3(37120) 36(36864)
   KB    d[00:00:01] Tested 1347841 keys (got 30566 IVs)
    0    6/  9   C3(37120) 36(36864) 3F(36864) 73(36352)
   KB    depth   byte(vote)47(36864) 65(36864) A1(36608)
    0    1/  2   4E(38400) 14(37376) 5C(37376) 9D(37376)
    1    2/  3   06(37888) FF(37120) 47(36864) 65(36864)
    2    0/  1   09(46592) 6E(38400) 81(37376) 79(36864)
    3    0/  1   1F(40960) 72(38656) D8(38400) BB(37888)
    4    1/  2   7F(3814Aircrack-ng 1.6 36608) 13(36352)
    5    5/  6   EA(37120) 68(36864) 47(36608) 25(36352)
    6    0/  2   71(40960) 0B(39168) 8B(37888) F3(37888)
    7   11[00:00:01] Tested 1684801 keys (got 30566 IVs)
    8    1/  2   27(37120) DB(36352) 86(36096) B3(36096)
   KB    depth   byte(vote)F8(37632) 2F(36864) 44(36608)
    0    1/  2   4E(38400) 14(37376) 5C(37376) 9D(37376)
    1    2/  3   06(37888) FF(37120) 47(36864) 65(36864)
```

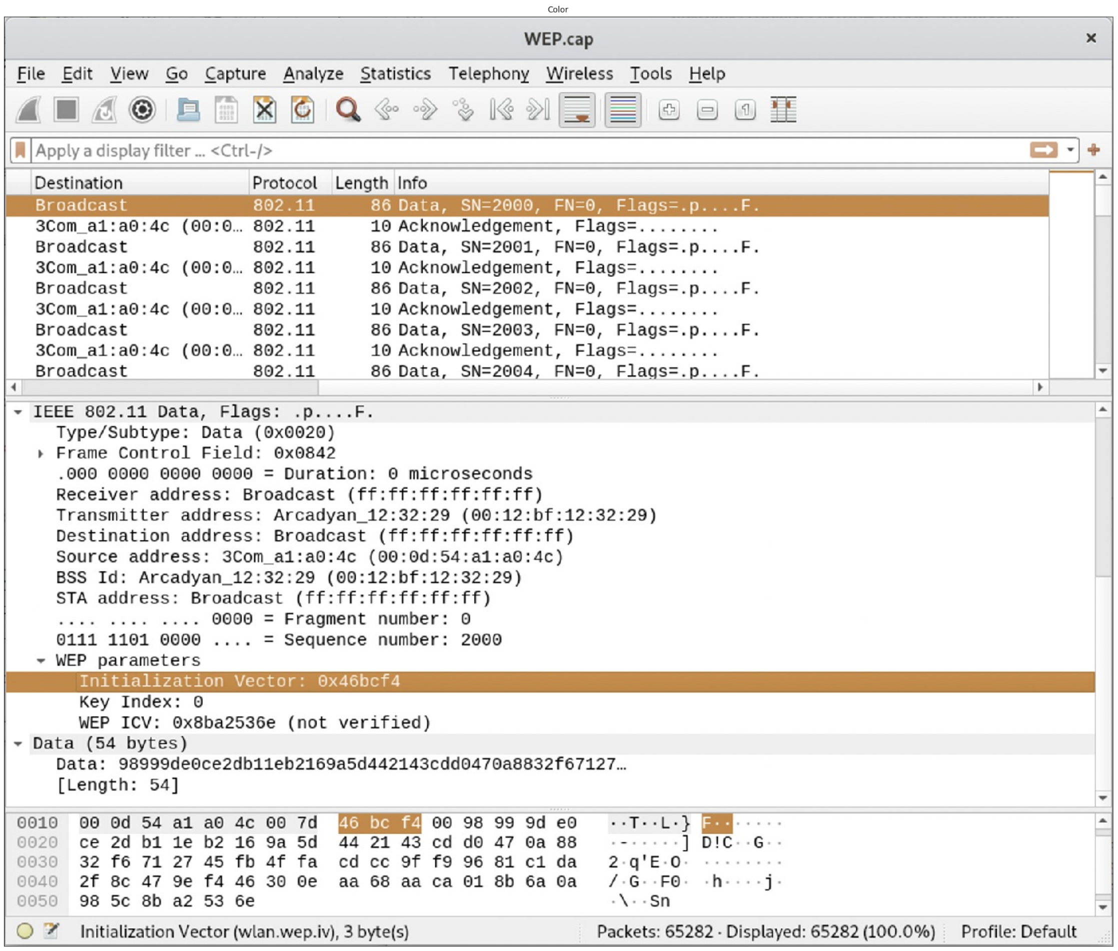
Color



```
  0    4/  7   9D(37376) 00(37120) C3(37120) 36(36864)
 KB    d[00:00:01] Tested 1347841 keys (got 30566 IVs)
  0    6/  9   C3(37120) 36(36864) 3F(36864) 73(36352)
 KB    depth   byte(vote)47(36864) 65(36864) A1(36608)
  0    1/  2   4E(38400) 14(37376) 5C(37376) 9D(37376)
  1    2/  3   06(37888) FF(37120) 47(36864) 65(36864)
  2    0/  1   09(46592) 6E(38400) 81(37376) 79(36864)
  3    0/  1   1F(40960) 72(38656) D8(38400) BB(37888)
  4    1/  2   7F(3814Aircrack-ng 1.6 36608) 13(36352)
  5    5/  6   EA(37120) 68(36864) 47(36608) 25(36352)
  6    0/  2   71(40960) 0B(39168) 8B(37888) F3(37888)
  7   11[00:00:01] Tested 1684801 keys (got 30566 IVs)
  8    1/  2   27(37120) DB(36352) 86(36096) B3(36096)
 KB    depth   byte(vote)F8(37632) 2F(36864) 44(36608)
  0    1/  2   4E(38400) 14(37376) 5C(37376) 9D(37376)
  1    2/  3   06(37888) FF(37120) 47(36864) 65(36864)
  2    0/  1   09(46592) 6E(38400) 81(37376) 79(36864)
  3    0/  1   1F(40960) 72(38656) D8(38400) BB(37888)
  4    1/  2   7F(3814Aircrack-ng 1.6 36608) 13(36352)
  5    3/  6   FF(37376) 55(37120) E5(37120) 68(36864)
  6    0/  2   71(40960) 0B(39168) 8B(37888) F3(37888)
  7   11[00:00:01] Tested 1811521 keys (got 30566 IVs)
  8    1/  2   31(37120) DB(36352) 86(36096) B3(36096)
 KB    depth   byte(vote)F8(37632) 2F(36864) 44(36608)
  0    1/  2   4E(38400) 14(37376) 5C(37376) 9D(37376)
  1    2/  3   06(37888) FF(37120) 47(36864) 65(36864)
  2    0/  1   09(46592) 6E(38400) 81(37376) 79(36864)
  3    0/  1   1F(40960) 72(38656) D8(38400) BB(37888)
  4    1/  2   7F(3814Aircrack-ng 1.6 36608) 13(36352)
  5    3/  4   FF(37376) 55(37120) E5(37120) 68(36864)
  6    0/  1   5C(40960) 0B(39168) 8B(37888) F3(37888)
  7   11[00:00:01] Tested 1514 keys (got 30566 IVs)
  8    1/  2   31(37120) DB(36352) 86(36096) B3(36096)
 KB    depth   byte(vote)F8(37632) 2F(36864) 44(36608)
  0    0/  9   1F(39680) 4E(38400) 14(37376) 5C(37376)
  1    7/  9   64(36608) 3E(36352) 34(36096) 46(36096)
  2    0/  1   1F(46592) 6E(38400) 81(37376) 79(36864)
  3    0/  3   1F(40960) 15(38656) 7B(38400) BB(37888)
  4    0/  7   1F(39168) 23(38144) 97(37120) 59(36608)


                      KEY FOUND! [ 1F:1F:1F:1F:1F ]
      Decrypted correctly: 100%
```

Task 2: Cracking the WEP Packet

```
show_img('Task 2/wireshark_out.png')
```



```
Destination              Protocol  Length Info
Broadcast                802.11       86 Data, SN=2000, FN=0, Flags=.p....F.
3Com_a1:a0:4c (00:0…     802.11       10 Acknowledgement, Flags=........
Broadcast                802.11       86 Data, SN=2001, FN=0, Flags=.p....F.
3Com_a1:a0:4c (00:0…     802.11       10 Acknowledgement, Flags=........
Broadcast                802.11       86 Data, SN=2002, FN=0, Flags=.p....F.
3Com_a1:a0:4c (00:0…     802.11       10 Acknowledgement, Flags=........
Broadcast                802.11       86 Data, SN=2003, FN=0, Flags=.p....F.
3Com_a1:a0:4c (00:0…     802.11       10 Acknowledgement, Flags=........
Broadcast                802.11       86 Data, SN=2004, FN=0, Flags=.p....F.
```

```
IEEE 802.11 Data, Flags: .p....F.
  Type/Subtype: Data (0x0020)
  Frame Control Field: 0x0842
  .000 0000 0000 0000 = Duration: 0 microseconds
  Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
  Transmitter address: Arcadyan_12:32:29 (00:12:bf:12:32:29)
  Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
  Source address: 3Com_a1:a0:4c (00:0d:54:a1:a0:4c)
  BSS Id: Arcadyan_12:32:29 (00:12:bf:12:32:29)
  STA address: Broadcast (ff:ff:ff:ff:ff:ff)
  .... .... .... 0000 = Fragment number: 0
  0111 1101 0000 .... = Sequence number: 2000
  WEP parameters
    Initialization Vector: 0x46bcf4
    Key Index: 0
    WEP ICV: 0x8ba2536e (not verified)
Data (54 bytes)
  Data: 98999de0ce2db11eb2169a5d442143cdd0470a8832f67127…
  [Length: 54]
```

```
0010  00 0d 54 a1 a0 4c 00 7d  46 bc f4 00 98 99 9d e0   ··T··L·} F·······
0020  ce 2d b1 1e b2 16 9a 5d  44 21 43 cd d0 47 0a 88   ·-·····] D!C··G··
0030  32 f6 71 27 45 fb 4f fa  cd cc 9f f9 96 81 c1 da   2·q'E·O· ········
0040  2f 8c 47 9e f4 46 30 0e  aa 68 aa ca 01 8b 6a 0a   /·G··F0· ·h····j·
0050  98 5c 8b a2 53 6e                                   ·\··Sn
```

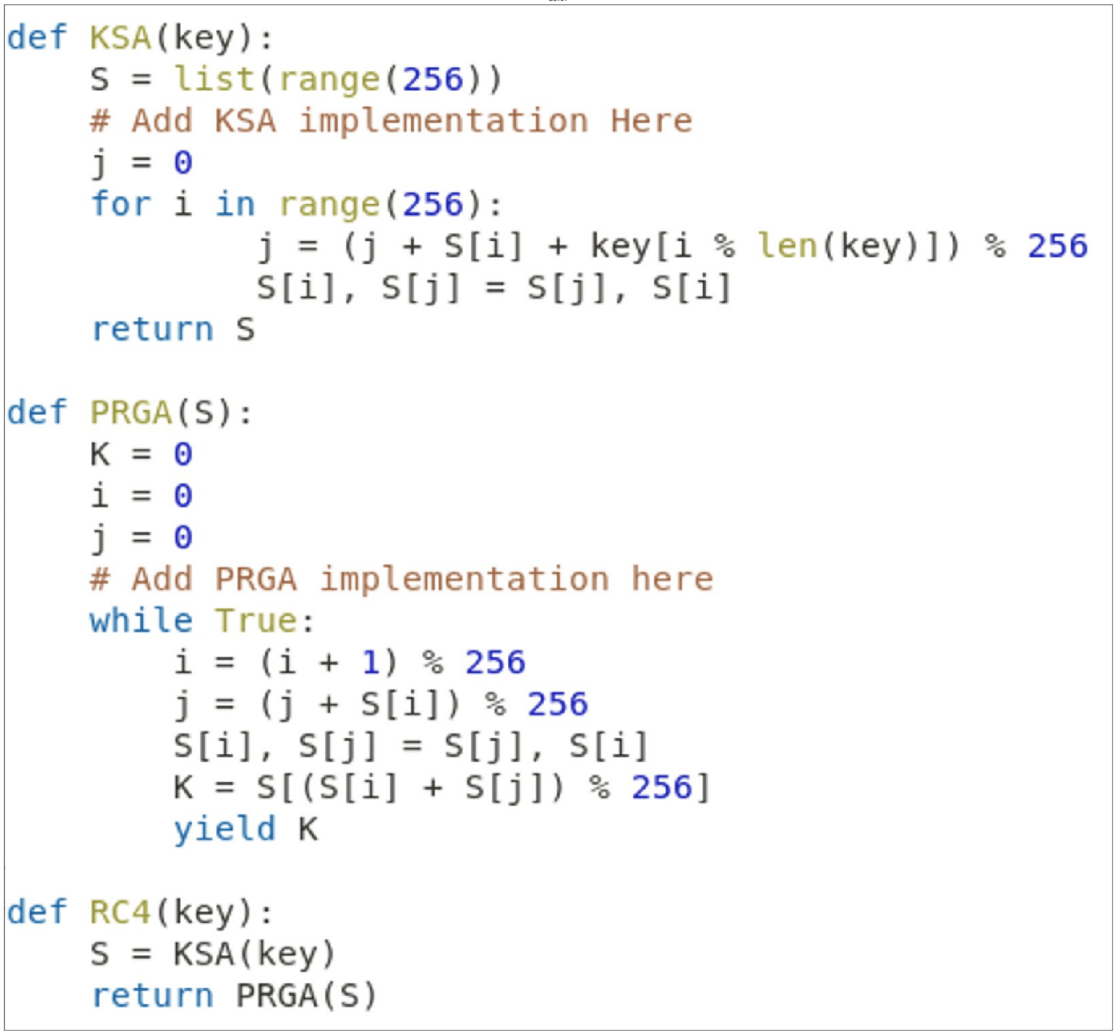
Initialization Vector (wlan.wep.iv), 3 byte(s)   Packets: 65282 · Displayed: 65282 (100.0%)   Profile: Default

After viewing the wireshark cap file, and sniffing SN=2000, 2000th packet, the IV is 0x46bcf4, WEP_ICV = 0x8ba2536e, Data = 98999de0ce2db11eb2169a5d442143cdd0470a8832f6712745fb4ffacdcc9ff99681c1da2f8c479ef446300eaa68aaca018b6a0a985c

Step 2: Implement the RC4 Algorithm

```
show_img('Task 2/ksa_prga.png')
```

Color

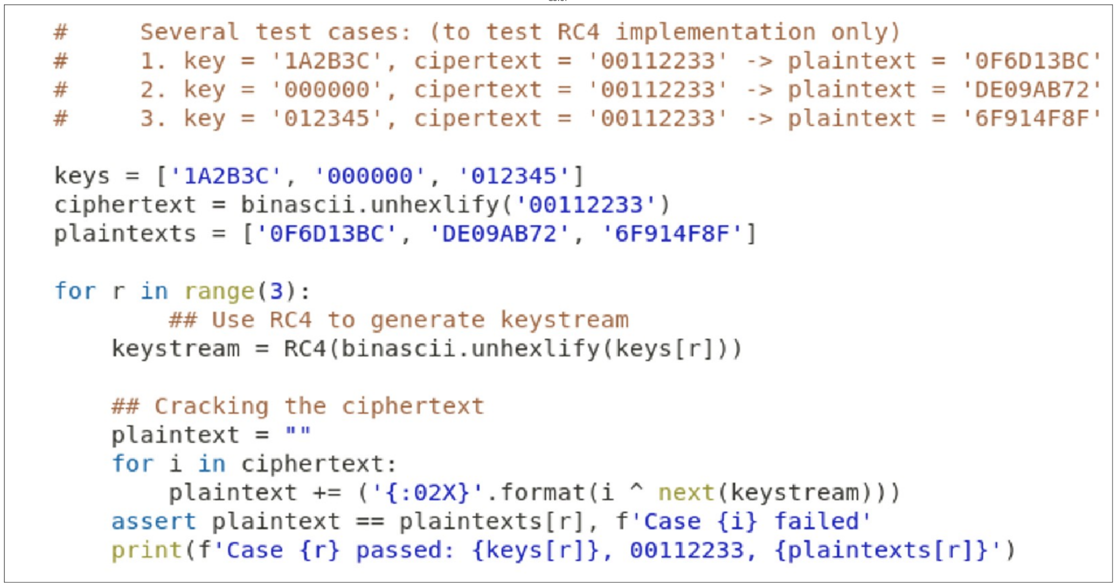


```python
def KSA(key):
    S = list(range(256))
    # Add KSA implementation Here
    j = 0
    for i in range(256):
            j = (j + S[i] + key[i % len(key)]) % 256
            S[i], S[j] = S[j], S[i]
    return S

def PRGA(S):
    K = 0
    i = 0
    j = 0
    # Add PRGA implementation here
    while True:
        i = (i + 1) % 256
        j = (j + S[i]) % 256
        S[i], S[j] = S[j], S[i]
        K = S[(S[i] + S[j]) % 256]
        yield K

def RC4(key):
    S = KSA(key)
    return PRGA(S)
```

Step 3: Verify Your Results

```python
show_img('Task 2/verify.png')
```

Color



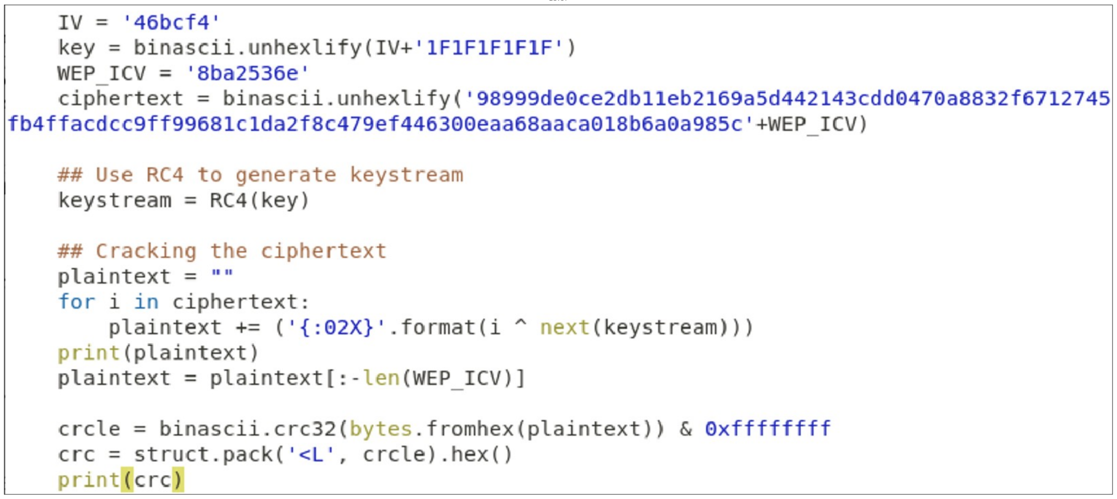
```
#     Several test cases: (to test RC4 implementation only)
#     1. key = '1A2B3C', cipertext = '00112233' -> plaintext = '0F6D13BC'
#     2. key = '000000', cipertext = '00112233' -> plaintext = 'DE09AB72'
#     3. key = '012345', cipertext = '00112233' -> plaintext = '6F914F8F'

keys = ['1A2B3C', '000000', '012345']
ciphertext = binascii.unhexlify('00112233')
plaintexts = ['0F6D13BC', 'DE09AB72', '6F914F8F']

for r in range(3):
        ## Use RC4 to generate keystream
    keystream = RC4(binascii.unhexlify(keys[r]))

    ## Cracking the ciphertext
    plaintext = ""
    for i in ciphertext:
        plaintext += ('{:02X}'.format(i ^ next(keystream)))
    assert plaintext == plaintexts[r], f'Case {i} failed'
    print(f'Case {r} passed: {keys[r]}, 00112233, {plaintexts[r]}')
```

The keys for testing consists of IV||Key, ciphertext consists of message||icv.

Step 1: convert hex keystream to byte keystream Step 2: convert hex ciphertext to byte ciphertext Step 3: zor the ciphertext with the keystream to get plaintext

```
show_img('Task 2/rc4_cap.png')
```

Color



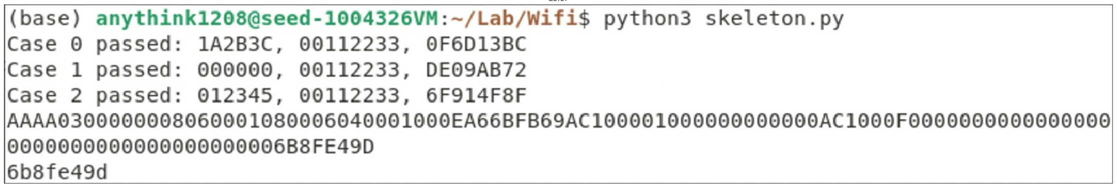
```
    IV = '46bcf4'
    key = binascii.unhexlify(IV+'1F1F1F1F1F')
    WEP_ICV = '8ba2536e'
    ciphertext = binascii.unhexlify('98999de0ce2db11eb2169a5d442143cdd0470a8832f6712745
fb4ffacdcc9ff99681c1da2f8c479ef446300eaa68aaca018b6a0a985c'+WEP_ICV)

    ## Use RC4 to generate keystream
    keystream = RC4(key)

    ## Cracking the ciphertext
    plaintext = ""
    for i in ciphertext:
        plaintext += ('{:02X}'.format(i ^ next(keystream)))
    print(plaintext)
    plaintext = plaintext[:-len(WEP_ICV)]

    crcle = binascii.crc32(bytes.fromhex(plaintext)) & 0xffffffff
    crc = struct.pack('<L', crcle).hex()
    print(crc)
```

Follow the 3 steps after using the testing case.

Plaintext is removed from decrypted ciphertext, leaving the remaining 8 characters to be the decrypted ICV. We then use the crc32 function to compute the unsigned 32-bit checksum of the plaintext. In python2, the output could return a signed integer, so the 0xffffffff mask would convert the output into unsigned integer. The mask does not change the output of an unsigned integer. The struct.pack converts the little endian crcle to the big endian crc.

```
show_img('Task 2/rc4_out.png')
```

Color



```
(base) anythink1208@seed-1004326VM:~/Lab/Wifi$ python3 skeleton.py
Case 0 passed: 1A2B3C, 00112233, 0F6D13BC
Case 1 passed: 000000, 00112233, DE09AB72
Case 2 passed: 012345, 00112233, 6F914F8F
AAAA0300000008060001080006040001000EA66BFB69AC10000100000000000000AC1000F0000000000000000
0000000000000000000006B8FE49D
6b8fe49d
```