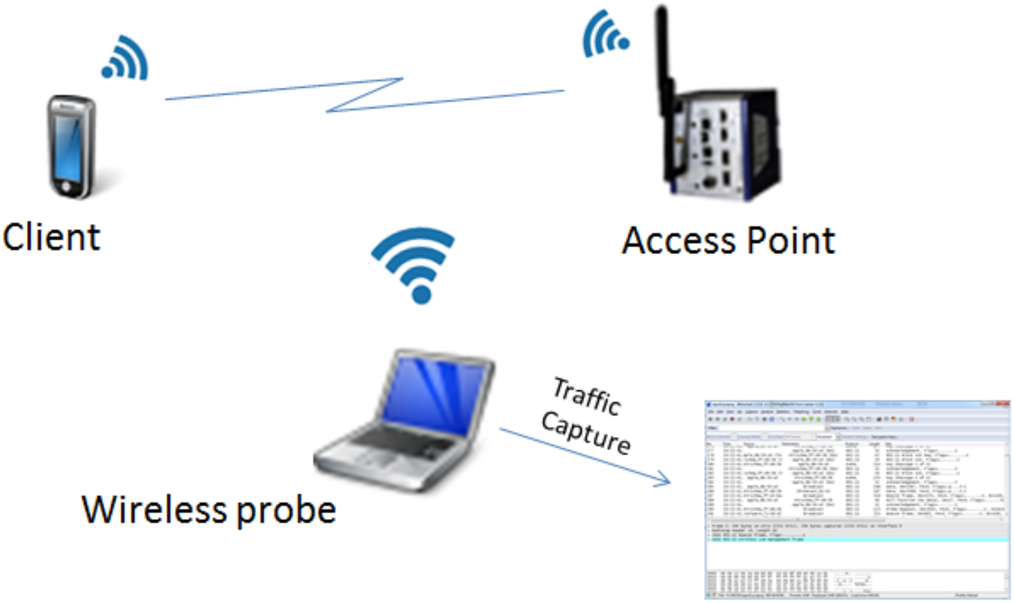
How to decrypt a WPA encrypted communication with Wireshark

This lesson describes step by step how to decrypt a WPA encrypted communication with Wireshark.
It suppose that a capture was previously done "in the air".
As this can be done for troubleshooting purpose it also suppose that the WPA key is known (it's not a hacking lesson).

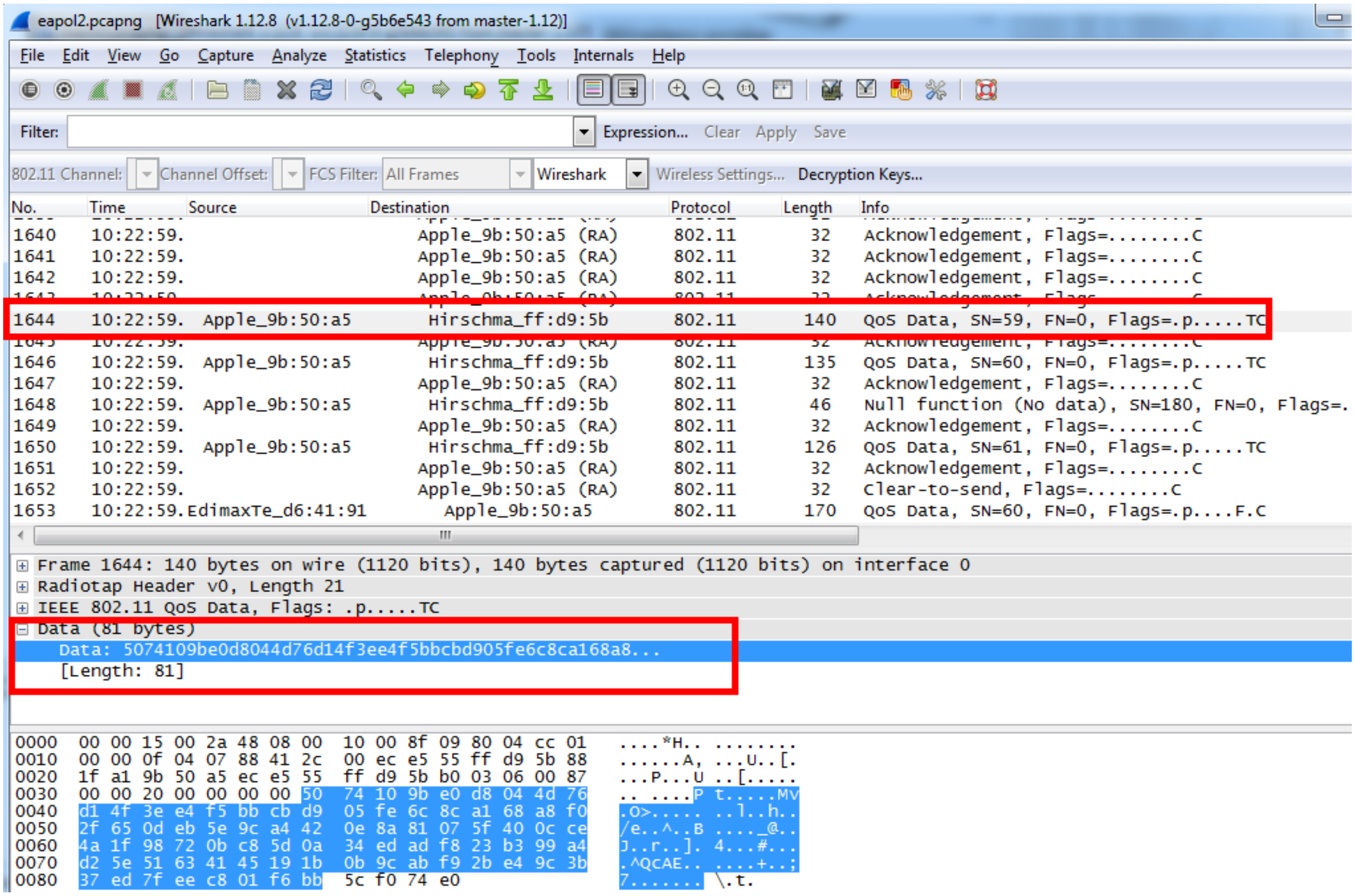
Capture the Wireless traffic between a client and an AP



As said previously and as represented above, a communication between a client and an AP must first be done.
Laptops running under Windows need most of the time to have specific hardware to be able to realise such a capture in "monitor mode". It means all the packets the Wifi NIC is able to see.
Linux systems with airmon-ng are able to do that and support most of the wiresless adapters.

In our example traffic was captured between an Open BAT and and Apple Iphone. The communication is encrypted with WPA2.
Capture was done on a laptop with Ubuntu, airmong-ng and Wireshark.

How does the capture looks like



eapol2.pcapng [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

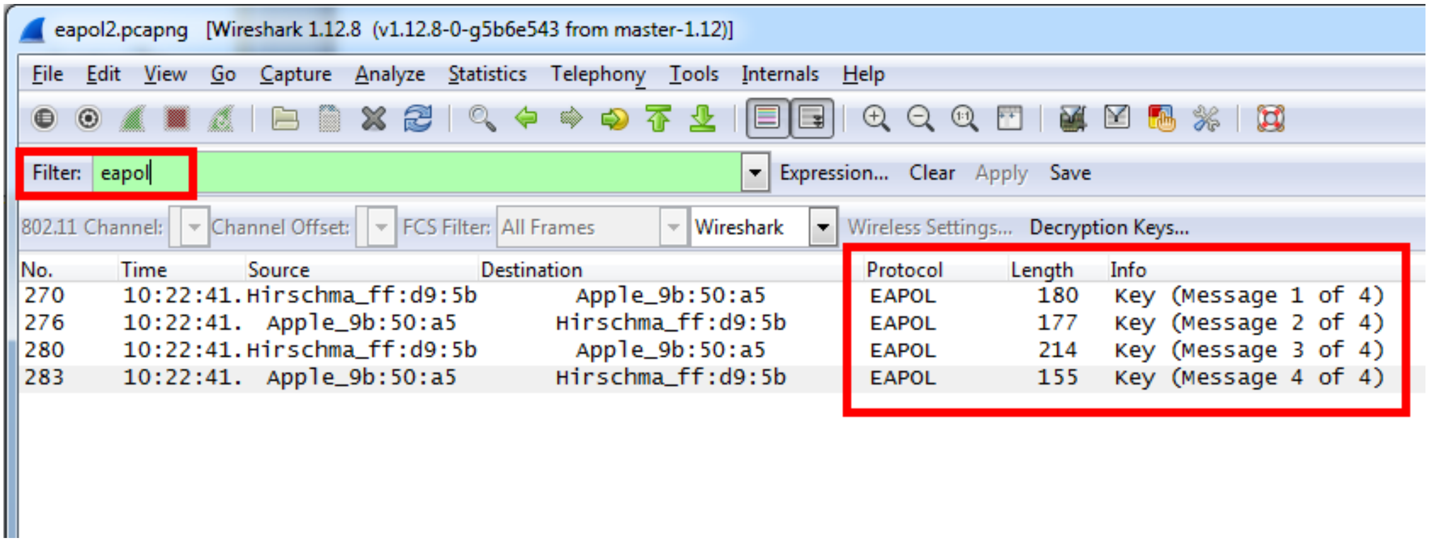
```
No.   Time        Source            Destination            Protocol  Length  Info
1640  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1641  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1642  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1643  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1644  10:22:59.  Apple_9b:50:a5      Hirschma_ff:d9:5b     802.11     140    QoS Data, SN=59, FN=0, Flags=.p.....TC
1645  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1646  10:22:59.  Apple_9b:50:a5      Hirschma_ff:d9:5b     802.11     135    QoS Data, SN=60, FN=0, Flags=.p.....TC
1647  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1648  10:22:59.  Apple_9b:50:a5      Hirschma_ff:d9:5b     802.11      46    Null function (No data), SN=180, FN=0, Flags=.
1649  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1650  10:22:59.  Apple_9b:50:a5      Hirschma_ff:d9:5b     802.11     126    QoS Data, SN=61, FN=0, Flags=.p.....TC
1651  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Acknowledgement, Flags=........C
1652  10:22:59.                     Apple_9b:50:a5 (RA)    802.11      32    Clear-to-send, Flags=........C
1653  10:22:59.EdimaxTe_d6:41:91      Apple_9b:50:a5       802.11     170    QoS Data, SN=60, FN=0, Flags=.p....F.C
```

```
⊞ Frame 1644: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface 0
⊞ Radiotap Header v0, Length 21
⊞ IEEE 802.11 QoS Data, Flags: .p.....TC
⊟ Data (81 bytes)
    Data: 5074109be0d8044d76d14f3ee4f5bbcbd905fe6c8ca168a8...
    [Length: 81]
```

```
0000  00 00 15 00 2a 48 08 00  10 00 8f 09 80 04 cc 01   ....*H.. ........
0010  00 00 0f 04 07 88 41 2c  00 ec e5 55 ff d9 5b 88   ......A, ...U..[.
0020  1f a1 9b 50 a5 ec e5 55  ff d9 5b b0 03 06 00 87   ...P...U ..[.....
0030  00 00 20 00 00 00 00 50  74 10 9b e0 d8 04 4d 76   .. ....P t.....Mv
0040  d1 4f 3e e4 f5 bb cb d9  05 fe 6c 8c a1 68 a8 f0   .O>..... ..l..h..
0050  2f 65 0d eb 5e 9c a4 42  0e 8a 81 07 5f 40 0c ce   /e..^..B ...._@..
0060  4a 1f 98 72 0b c8 5d 0a  34 ed ad f8 23 b3 99 a4   J..r..]. 4...#...
0070  d2 5e 51 63 41 45 19 1b  0b 9c ab f9 2b e4 9c 3b   .^QcAE.. ....+..;
0080  37 ed 7f ee c8 01 f6 bb  5c f0 74 e0               7....... \.t.
```

In the capture, the traffic between the iphone and the Access Point can be identified, but canno't be interpreted.

Check that the "4 way handshake" was captured



eapol2.pcapng [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]

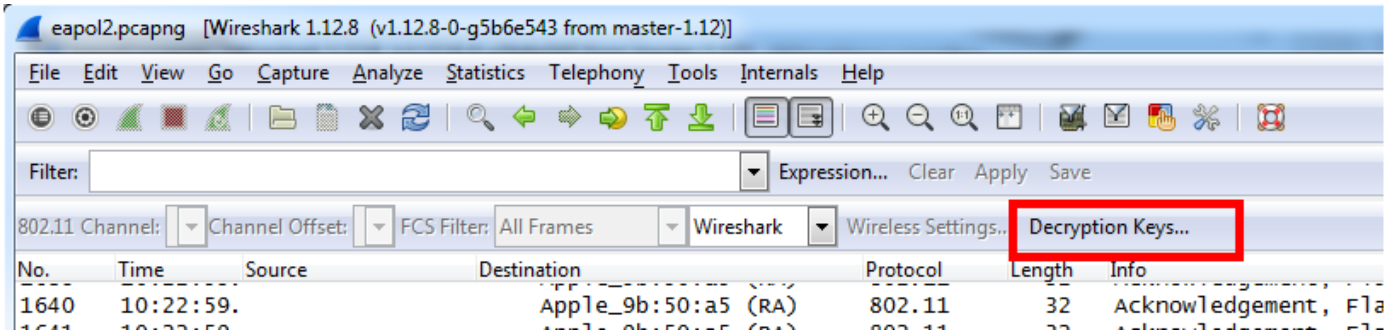
Filter: eapol

```
No.   Time         Source              Destination          Protocol     Length   Info
270   10:22:41.Hirschma_ff:d9:5b        Apple_9b:50:a5       EAPOL         180    Key (Message 1 of 4)
276   10:22:41.  Apple_9b:50:a5       Hirschma_ff:d9:5b      EAPOL         177    Key (Message 2 of 4)
280   10:22:41.Hirschma_ff:d9:5b        Apple_9b:50:a5       EAPOL         214    Key (Message 3 of 4)
283   10:22:41.  Apple_9b:50:a5       Hirschma_ff:d9:5b      EAPOL         155    Key (Message 4 of 4)
```

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

The capture must include the association and authentication process.
The EAPOL 4 way handshake is part of it. If it isn't included in the capture then a decryption as described in the following steps won't be possible.
It means of course that the capture must be started before the client associate with the AP.
To make sure that the 4 way handshake is included in the capture "eapol" can be used as display filter in wireshark.
The 4 way handhake can be then easily be identified. The 4 messages of the handshake must be present in the capture.

Add the Wireless toolbar in Wireshark



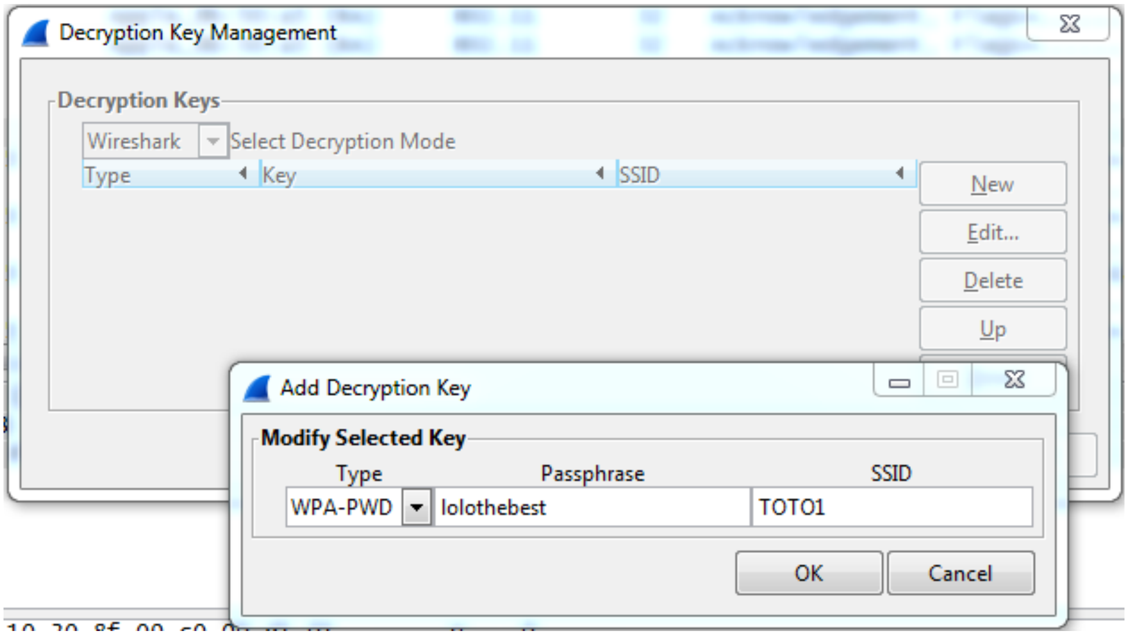
eapol2.pcapng [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]

```
No.    Time          Source          Destination            Protocol     Length   Info
1640   10:22:59.                     Apple_9b:50:a5 (RA)    802.11         32     Acknowledgement, Fla
```

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

Go in view and select "Wireless toolbar".
Make sure that the "Decryption Keys" button is included in the toolbar.
If not (Wireless for Linux), you can access the Decryption Keys and procede to the following step in:
Edit > Preferences > Protocols > IEEE 802.11 > Decryptions Keys [Edit]

Enter the WPA password



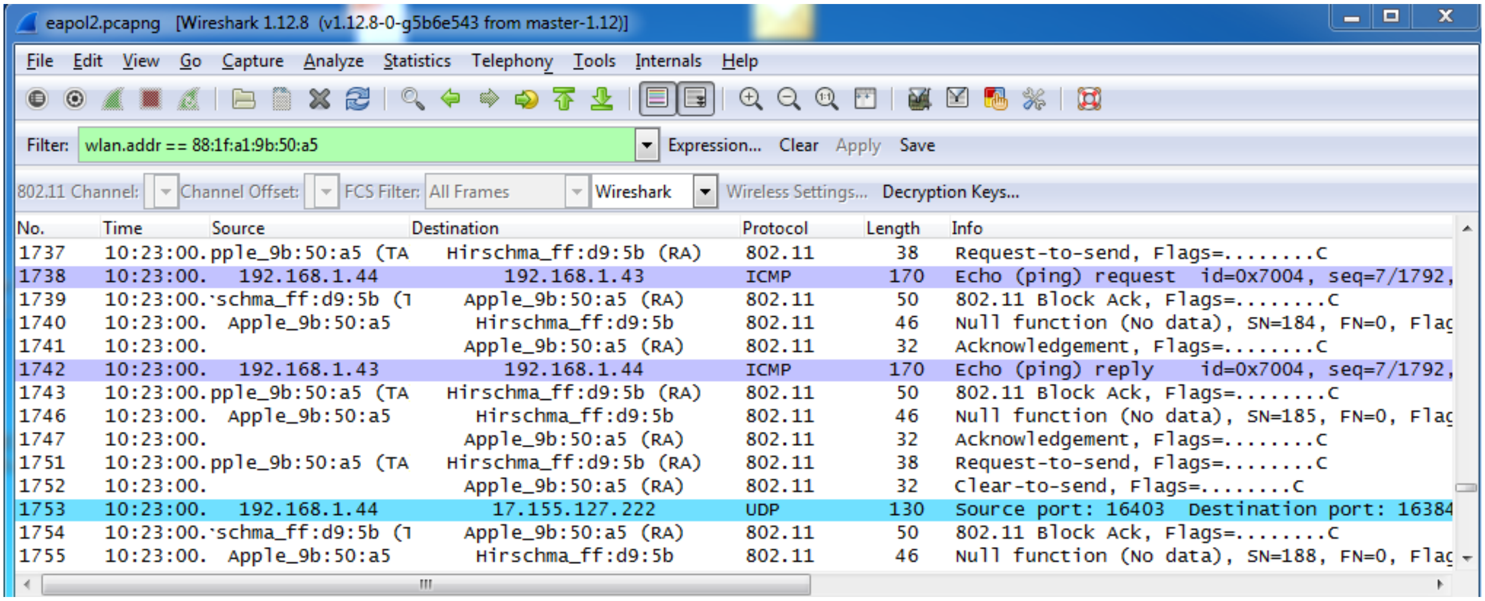
Type	Passphrase	SSID
WPA-PWD	lolothebest	TOTO1

In the Wireless Toolbar, select "Decryption Keys"
In the Decryption Key Management Window, select "New"
Enter the WPApassord selecting "WPA-PWD" as Type.

If you accessed thsi menu via the "Preferences" (in the case you don't have "Decryption Keys" in the Wireless toolbar), password and SSID must be entered as follow:
Key Type = wpa-pwd
Key = password:SSID (in our case: "lolothebest:TOTO1")

> OK

Result


eapol2.pcapng [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

Filter: wlan.addr == 88:1f:a1:9b:50:a5

No.	Time	Source	Destination	Protocol	Length	Info
1737	10:23:00.	pple_9b:50:a5 (TA	Hirschma_ff:d9:5b (RA)	802.11	38	Request-to-send, Flags=........C
1738	10:23:00.	192.168.1.44	192.168.1.43	ICMP	170	Echo (ping) request id=0x7004, seq=7/1792,
1739	10:23:00.	schma_ff:d9:5b (T	Apple_9b:50:a5 (RA)	802.11	50	802.11 Block Ack, Flags=........C
1740	10:23:00.	Apple_9b:50:a5	Hirschma_ff:d9:5b	802.11	46	Null function (No data), SN=184, FN=0, Flag
1741	10:23:00.		Apple_9b:50:a5 (RA)	802.11	32	Acknowledgement, Flags=........C
1742	10:23:00.	192.168.1.43	192.168.1.44	ICMP	170	Echo (ping) reply id=0x7004, seq=7/1792,
1743	10:23:00.	pple_9b:50:a5 (TA	Hirschma_ff:d9:5b (RA)	802.11	50	802.11 Block Ack, Flags=........C
1746	10:23:00.	Apple_9b:50:a5	Hirschma_ff:d9:5b	802.11	46	Null function (No data), SN=185, FN=0, Flag
1747	10:23:00.		Apple_9b:50:a5 (RA)	802.11	32	Acknowledgement, Flags=........C
1751	10:23:00.	pple_9b:50:a5 (TA	Hirschma_ff:d9:5b (RA)	802.11	38	Request-to-send, Flags=........C
1752	10:23:00.		Apple_9b:50:a5 (RA)	802.11	32	Clear-to-send, Flags=........C
1753	10:23:00.	192.168.1.44	17.155.127.222	UDP	130	Source port: 16403 Destination port: 16384
1754	10:23:00.	schma_ff:d9:5b (T	Apple_9b:50:a5 (RA)	802.11	50	802.11 Block Ack, Flags=........C
1755	10:23:00.	Apple_9b:50:a5	Hirschma_ff:d9:5b	802.11	46	Null function (No data), SN=188, FN=0, Flag

The packets containing data exchanged between the AP and the Iphone can now be interpreted by Wireshark.

NB: In some cases I already had to restart Wireshark after entering the WPA password.