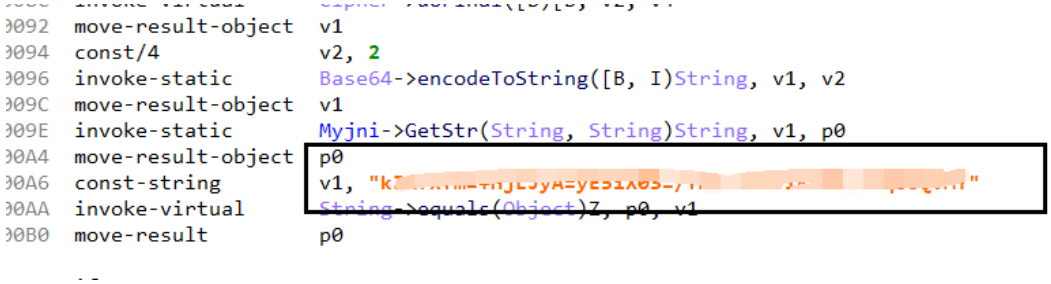
mobileC

反编译看出是AES/CBC/pad5的AES算法

然后动调找出密钥和iv

```
key=QERAPG9dPyZfTC5f
iv = aUBTJjg4Q2NDLg==
```

然后找到密文



通过查看so文件可以发现这是一个6组8位栅栏

需要进行排序

每8个字符一组。

例如

kJFSDfm=+HAXJyA=yE5CX03=/fMFGa2=JlDSRV/=dq6CXVMY

可以分成

kJFSDfm=

+HAXJyA=

yE5CX03=

/fMFGa2=

JlDSRV/=

dq6CXVMY

一共6组，但是由于栅栏分组位数不足补=号，则第一组一定不可能有 = 号。（有些人是这种情况，如果你的第一组没有=号，则正常分组，如果第一组有=号则逆序排列）实际上分组为

dq6CXVMY

JlDSRV/=

/fMFGa2=

yE5CX03=

+HAXJyA=

kJFSDfm=

然后对6组进行排列组合

第一位一定是第一组，第二位一定是第六组，也就是

dq6CXVMYkJFSDfm=

然后后面的四位需要进行排列组合后爆破，一共是A43=24种排序

可以用这个脚本

```
str1 = list()
str1 = ["xxxx","xxxx","xxxx","xxxxx","xxxx","xxxx"] #8个字符一组
fp = open("giao.txt","w")
arr = list()

arr  = [
(1, 6, 2, 3, 4, 5), (1, 6, 2, 3, 5, 4), (1, 6, 2, 4, 3, 5), (1, 6, 2, 4, 5, 3),
(1, 6, 2, 5, 3, 4), (1, 6, 2, 5, 4, 3), (1, 6, 3, 2, 4, 5), (1, 6, 3, 2, 5, 4),
(1, 6, 3, 4, 2, 5), (1, 6, 3, 4, 5, 2), (1, 6, 3, 5, 2, 4), (1, 6, 3, 5, 4, 2),
(1, 6, 4, 2, 3, 5), (1, 6, 4, 2, 5, 3), (1, 6, 4, 3, 2, 5), (1, 6, 4, 3, 5, 2),
(1, 6, 4, 5, 2, 3), (1, 6, 4, 5, 3, 2), (1, 6, 5, 2, 3, 4), (1, 6, 5, 2, 4, 3),
(1, 6, 5, 3, 2, 4), (1, 6, 5, 3, 4, 2), (1, 6, 5, 4, 2, 3), (1, 6, 5, 4, 3, 2)]

for i in arr:

fp.write(str1[i[0]-1]+str1[i[1]-1]+str1[i[2]-1]+str1[i[3]-1]+str1[i[4]-1]+str1[i
[5]-1]+'\n')
```

然后把算出来的密文进行栅栏解密

每组字数为6

然后把解密出来的24组密文，进行AES解密就行