密码学第二次大作业

DES

des文件运行方式，第一个参数如果是0，那么第二个参数就是输入的字符串；

```
key is: aaaaaaaa
input string is: aaaaaaaa
Encrypt hex string is: 78aa3eb42832f885
Decrypt string is: aaaaaaaa
```

在github上找到了一个Encryption验证正确；

对于规模进行计算，第一个参数如果不是0，那么就代表生成字符串的长度；输出的是秒数：

```
./des.o 1000000

1.10938
```

这里面生成的密钥可以是随机的，需要将`generateRandomKey()`中的使能改为`true`；

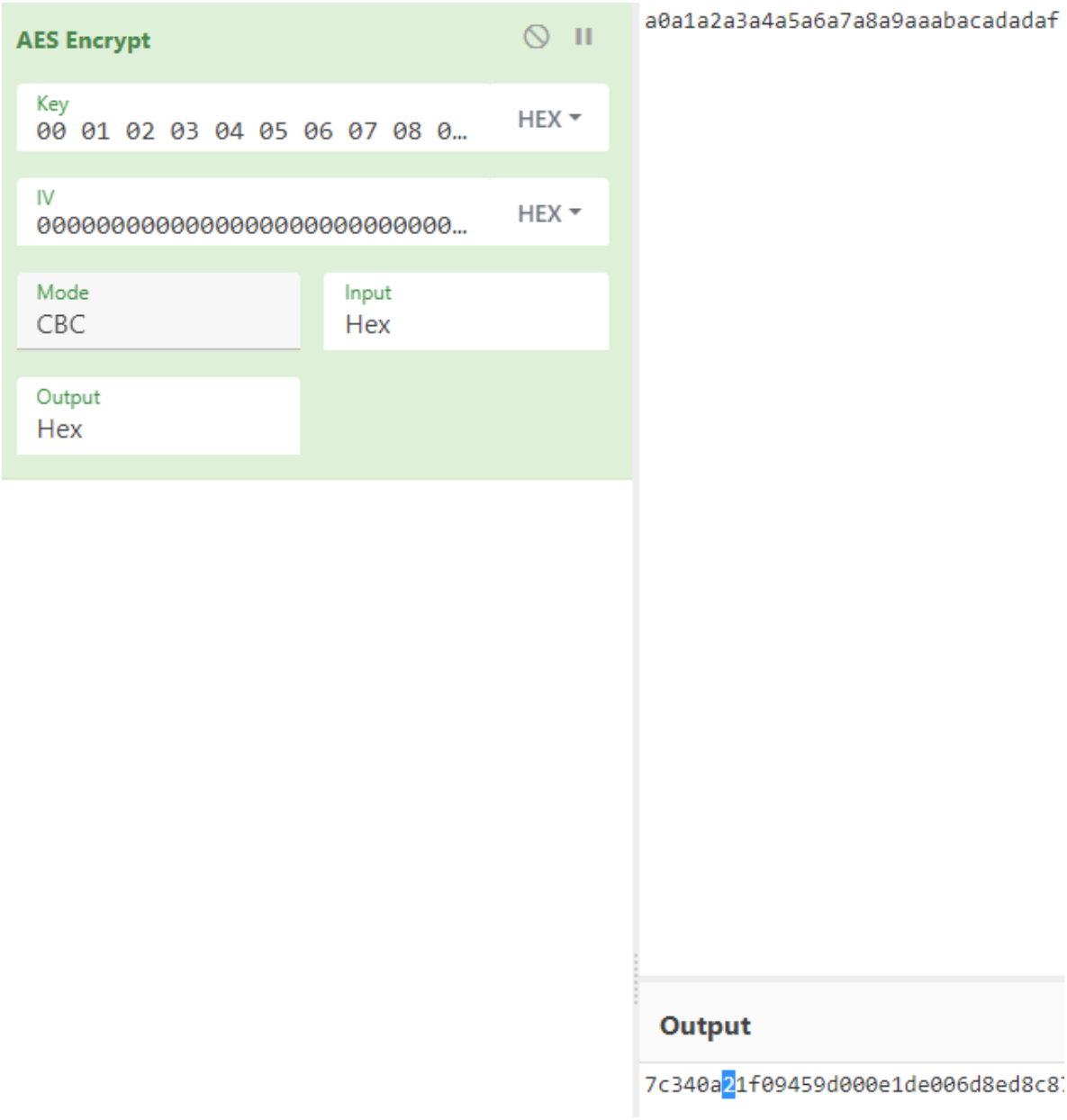
这里面的算法速度较为慢；但是能力有限，没有继续实现高性能的算法；

aes-128

类似的输入格式，如果第一个参数是0，则输出测试的内容；

```
origin: a0a1a2a3a4a5a6a7a8a9aaabacadadaf
encrypt: 7c340a21f09459d000e1de006d8ed8c8
Decrypt: a0a1a2a3a4a5a6a7a8a9aaabacadadaf
```

在网站https://gchq.github.io/CyberChef/中进行验证，得到的结果如下：



如果第二个参数不是0，则代表的测试长度；如果输入长度为10485760，输出结果如下：

```
./aes.o 10485760

0.625
```

算法效率较高；其中的IV可以随机生成，需要将generateRandomVector中的参数改为true；

SM4

有同样的输入格式，如果第一个参数为0，则进行样例输入输出；

```
plaintext: 01234567 89abcdef fedcba98 76543210
ciphertext: 681edf34 d206965e 86b3e94f 536e4246
deciphertext: 01234567 89abcdef fedcba98 76543210
```

如果输入的长度为10485760，那么输出的时间为：

```
./sm4.o 10485760

test mode is on
1.79688
```

RC4

输入的密钥size为128字节，按照算法生成密钥；输入的第一个参数为16384，为16k大小，产生的时间为0；不能用clock()进行准确计时；

BM 算法

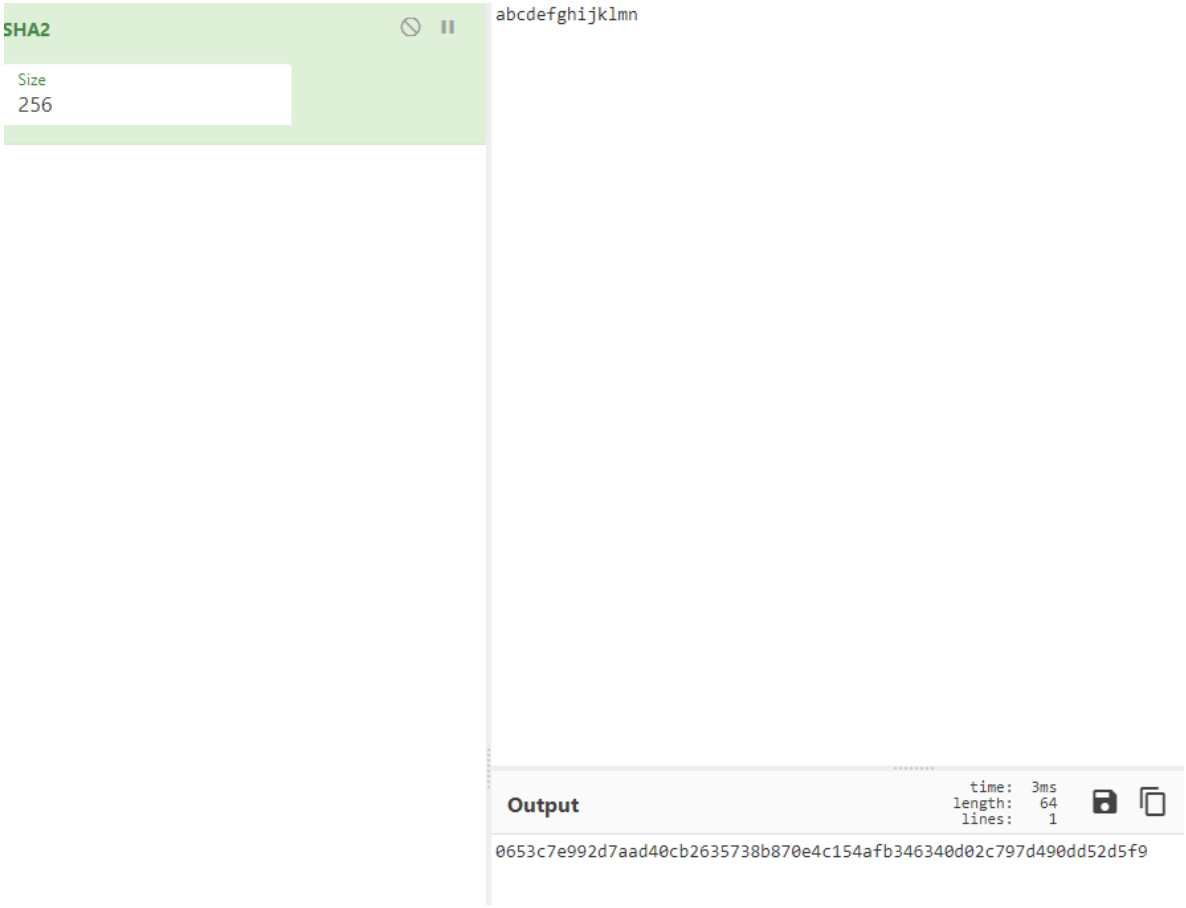
应用课上作业的例子，这样生成的内容如下：

```
./bm.o 00101010010001
00101010010001
a91
f(x) = 1 + x^1 + x^2 + x^4 + x^5
```

和当时计算的内容是一样的，这里面输入的就是bit序列；

SHA-2

```
./sha2.o 0 abcdefghijklmn
input string is: abcdefghijklmn
output hash is: 0653c7e992d7aad40cb2635738b870e4c154afb346340d02c797d490dd52d5f9
```



同样利用的是上面的网站；

如果压缩的是16k也就是16384byte的数据，应用的时间是0，不能用clock精确计时；

测试的方法就是第一个参数是长度；

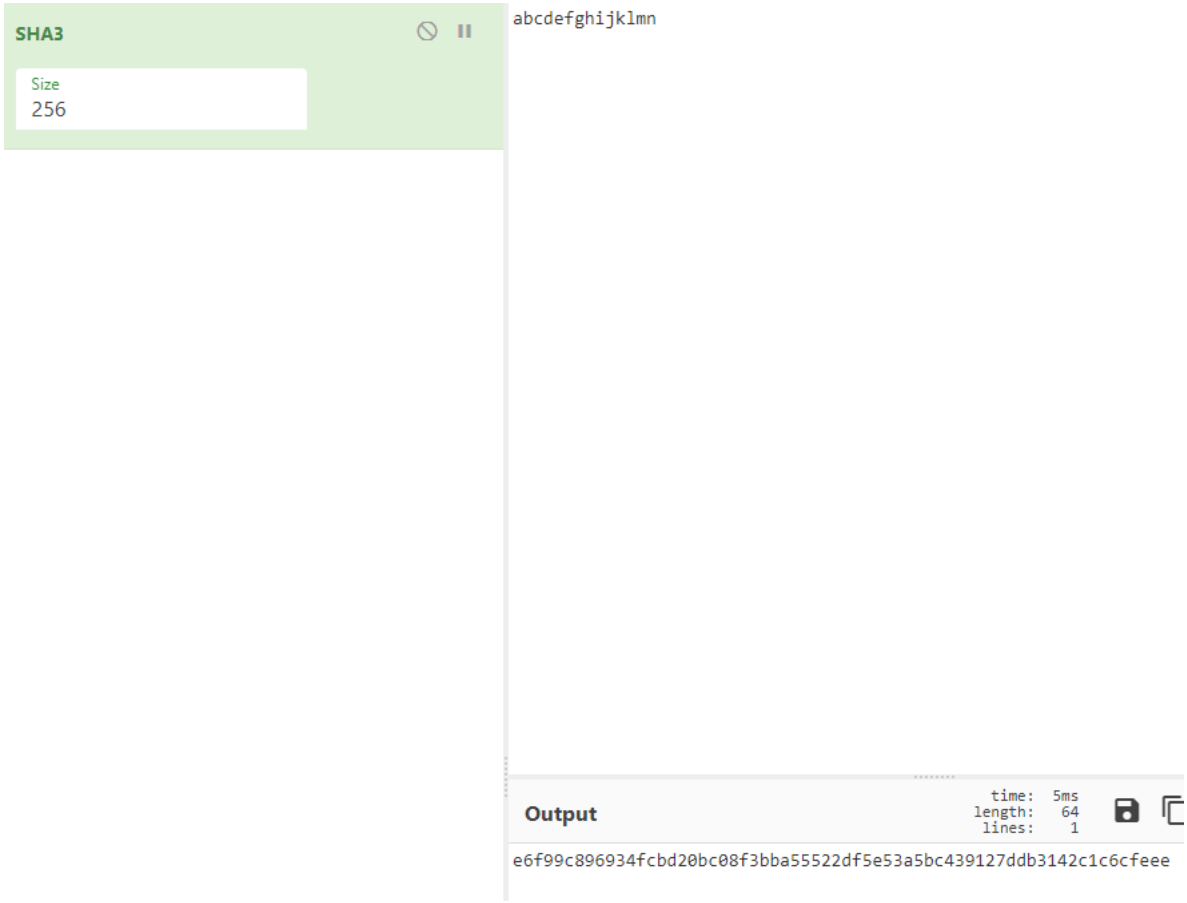
SM-3

```
input string is: abcdefghijklm
output hash is: d435b2f1d39b2d3fe3f20700789c6018250dcac08c8ee58fc266d1e054e8d505
```

如果输入的是18k也就是16384byte数据，应用的时间是0，不能够用clock精确计时；

SHA-3

```
input string is: abcdefghijklmn
hash string is: e6f99c896934fcbd20bc08f3bba55522df5e53a5bc439127ddb3142c1c6cfeee
```



和之前打印的内容相同；如果压缩的是16k也就是16384byte数据，用的时间为0.015625s，速度很快，但是较sha2更慢一些；

所有的明文在plain.txt中，然后生成的内容在以算法命名的txt文件中；