**实验一 AES密码算法**

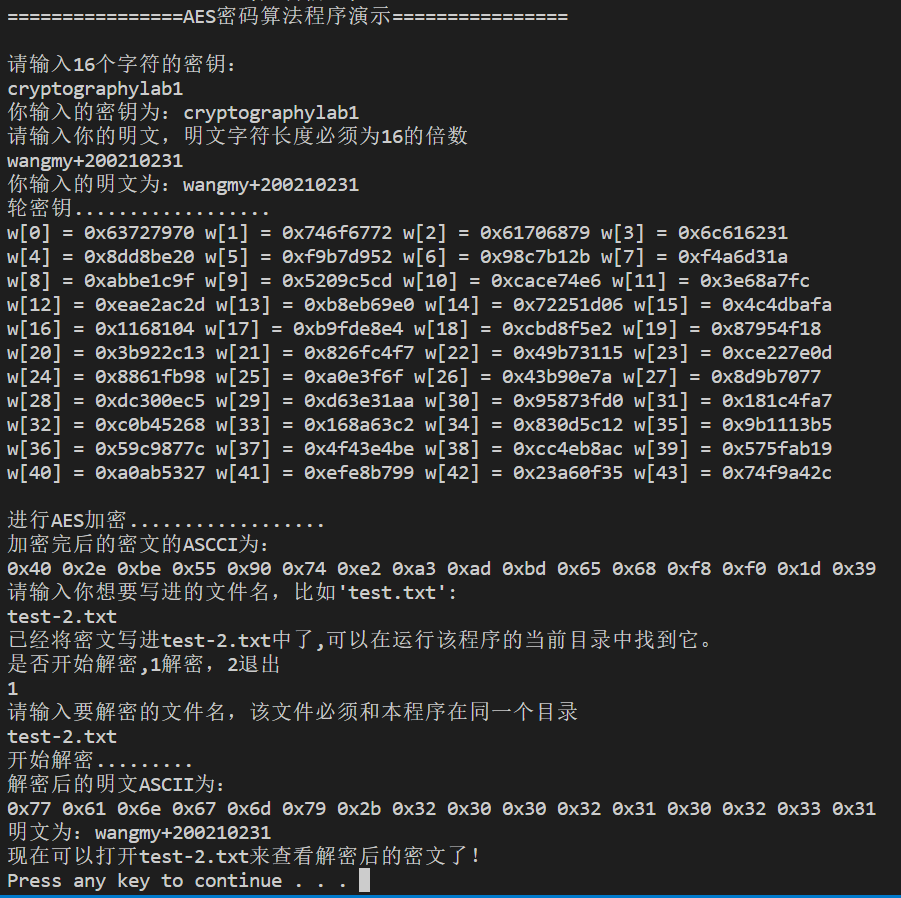
**姓名： 王木一 学号：\_\_200210231\_\_**

1. **运行截图**

**TEST-1**

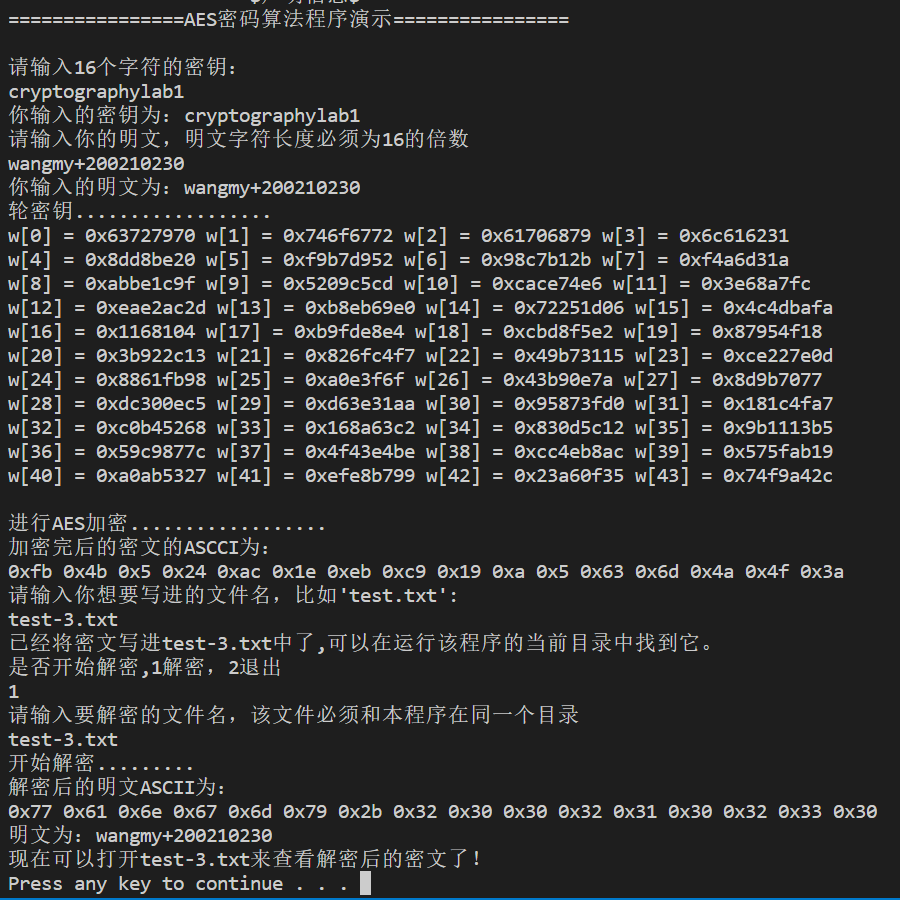
明文：thisisatestclass

密钥：securitysecurity

**TEST-2**

明文：wangmy+200210231

密钥： cryptographylab1

**TEST-3**

明文：wangmy+200210230

密钥：cryptographylab1

1. **实验过程中遇到的问题有哪些？你是怎么解决的。**
2. 列混淆计算

计算列混淆时要用到矩阵乘法，需设置tmp来存储部分积，即tmp = tmp ^ GFMul( colM[][], a[][])。算完后将tmp值赋给array[][]。每计算一个array矩阵元素的值都要将tmp重新置零。之前加密错误就出现在这里。

1. CBC

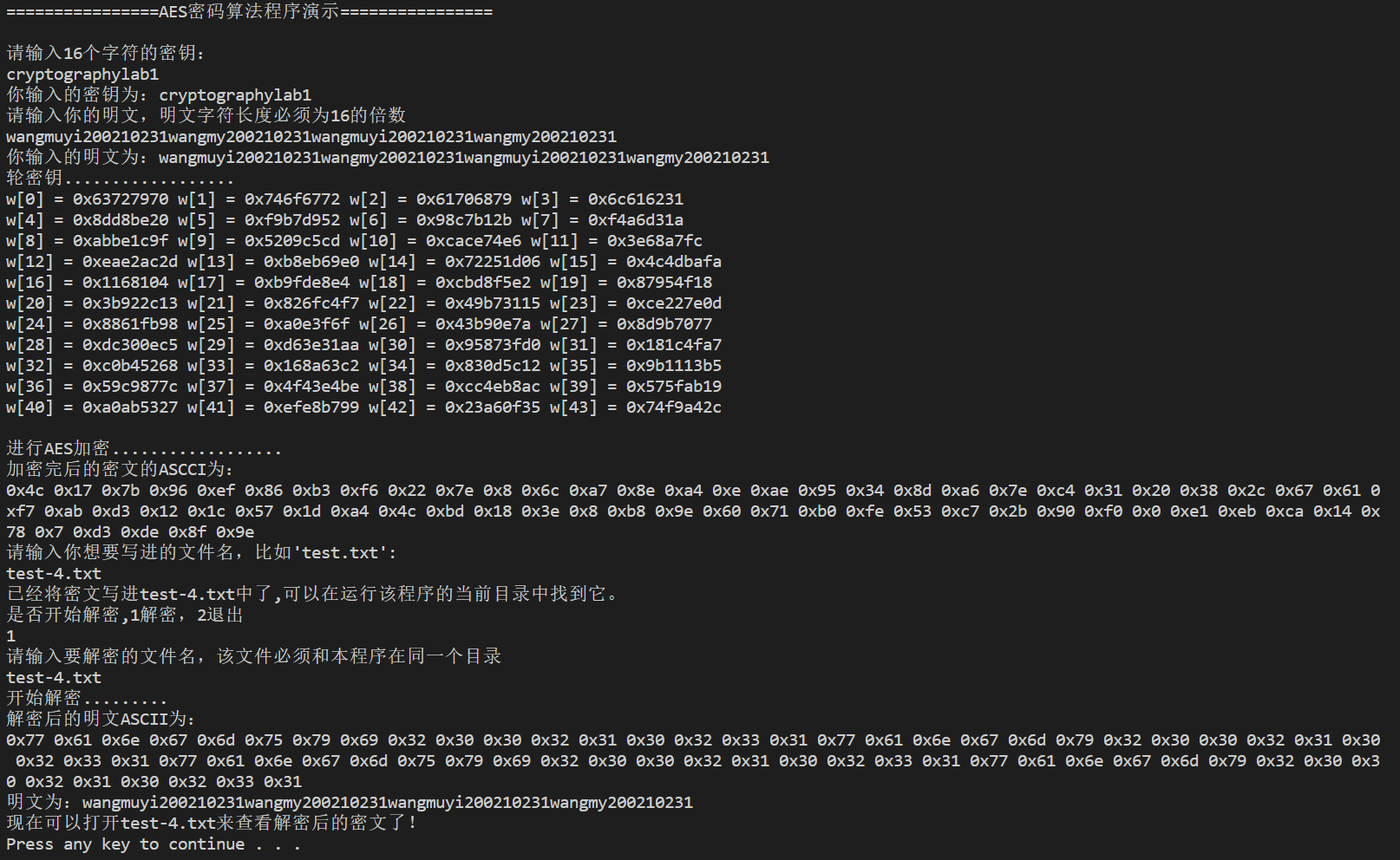
实现CBC模式时，每次AES加密前需要在原有分组明文中加上一个向量，此向量来自IV或前一组加密的结果。相反的，解密时deAES的结果还要加上上一组的密文（或IV）才能成为本组明文。由于代码函数参数传入的是指针，解密后的明文直接覆盖之前的密文，就不能直接把它加给下一组的解密结果。故需要使用单独的变量提前复制一份当前组的密文。之前CBC解密错误就在这里。

1. **如果不用lab1-aes.c代码框架或者实现了CBC模式，请说明。**

在原有代码框架下，实现了CBC模式。

1. 将加解密功能分别从aes()和deAes()方法中提取出，分别为void AES(char \*p)和void deAES(char \*c)，单独完成加解密任务。
2. 在原有aes()和deAes()方法中增加对明文/密文长度判断。若长度为16，就执行一般加解密操作，不涉及IV（初始向量）的加入；若长度为大于16且为16倍数，就执行CBC模式。
3. 增加了新的方法来辅助实现CBC模式。void addVector(char \*p, char \*v)用于将向量v加到p中。void copyVector(char \*c, char \*out)用于将c串复制给out串（解决上面的问题2）

测试结果：

明文：wangmuyi200210231wangmy200210231wangmuyi200210231wangmy200210231密钥：cryptographylab1

注：CBC初始向量IV设置为“abcdefghijklmnop”，具体实现详见lab1-aes.c