# 实验三：AES密码算法

### 一．实验目的

通过编码掌握 AES 算法的密钥生成方法；掌握AES 算法的加解密过程；熟悉AES 算法的程序设计与操作方法。

### 二．实验内容和要求

### 1、参考模板代码ase-demo.c完成AES的加密和解密，也可以自行编写代码（请用demo.c中的S盒、逆S盒、轮常量Rcon）

### 2、完成明文和密钥为128位的AES-128加密解密算法；

### 考虑程序的健壮性

### 提交内容和截止时间

两周时间内提交（**2020-11-27 00:00**）；

平台链接 [http://10.249.182.83:8000/#/login](http://10.249.182.83:8000/)

将源码和截图文件打车zip包上传，以学号\_姓名命名。

### 实验原理

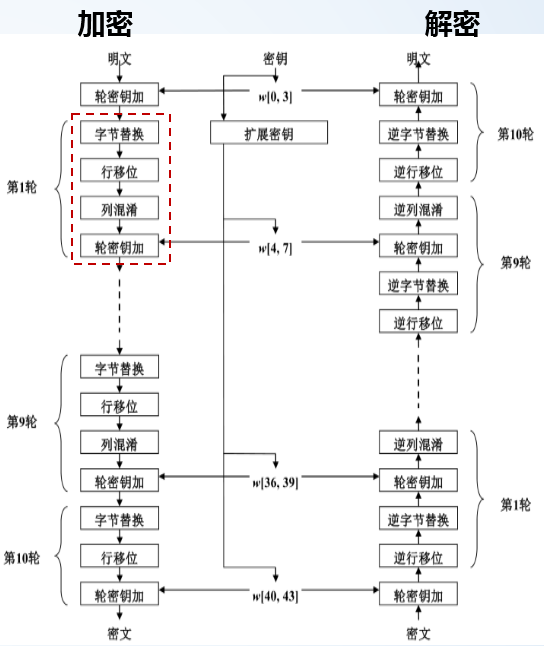
1. AES-128 算法的加密和解密过程

AES-128 分组长度为 128bit，共有10轮加密、解密过程。加密过程1-9轮迭代包括**字节代换**（ByteSubstitute）、**行移位**（ShiftRow）、**列混淆**（MixColumn）、**轮密钥加**（RoundKey）四个步骤，其中**第10轮不包含列混淆**；解密过程1-9轮迭代包括**行移位求逆、字节代换求逆、轮密钥加求逆、列混淆求逆**四个步骤，其中**第10轮不包含列混淆求逆**。

AES 算法在 GF(28 )域上进行运算，该域中的加法定义为简单的按位异或，乘法运算相对复杂一些。AES 中的运算一部分是按字节（有限域GF(28 )的元素）来定义的，另一部分是按 4 个字节组成的字来定义的，一个字为 32 位(密钥部分)，可以看成是系数在有限域 GF(28 )上的次数小于 4 的多项式。有限域GF(28 )是由不可约多项式（对应二进制 100011011，十六进制 11B）来定义的，这里所选的 m(x)是所有次数为 8 的不可约多项式列表中的其中一个，其中的元素可以表示成GF(28 )上的多项式形式，也可以表示成字节或十六进制的形式。

3.1加解密流程

AES 算法的加密与解密流程如图所示。



AES 加密的第 1 轮到第 9 轮的轮函数是一样，包括 4 个操作：字节代换、行位移、列混淆和轮密钥加，最后一轮迭代不执行列混淆。另外，在第一轮迭代之前，先将明文和原始密钥进行一次加密操作。

AES 的解密过程仍为 10 轮，每一轮的操作是加密操作的逆操作。由于 AES 算法的 4 个轮操作（字节代换、行移位、列混淆和轮密钥加）都是可逆的，因而解密操作的一轮就是顺序执行逆行移位，逆字节代换、逆轮密钥加和逆列混淆。同加密操作类似，最后一轮不执行逆列混淆，在第一轮解密之前，要执行 1 次逆轮密钥加操作。

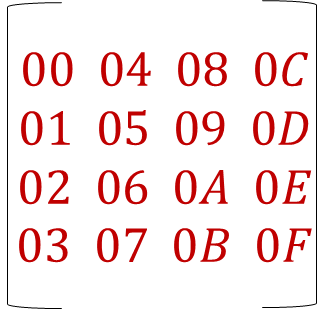
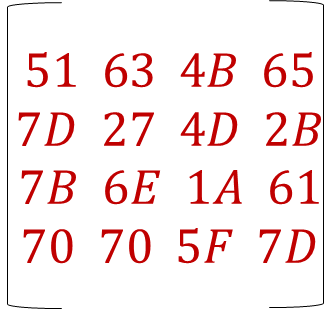
**预处理：**

4个轮操作（字节代换、行移位、列混淆和轮密钥加）对应的都是矩阵的方式，那么对于读入的信息要进行初始化的处理，在读出显示时也做个处理。

例如(16进制)

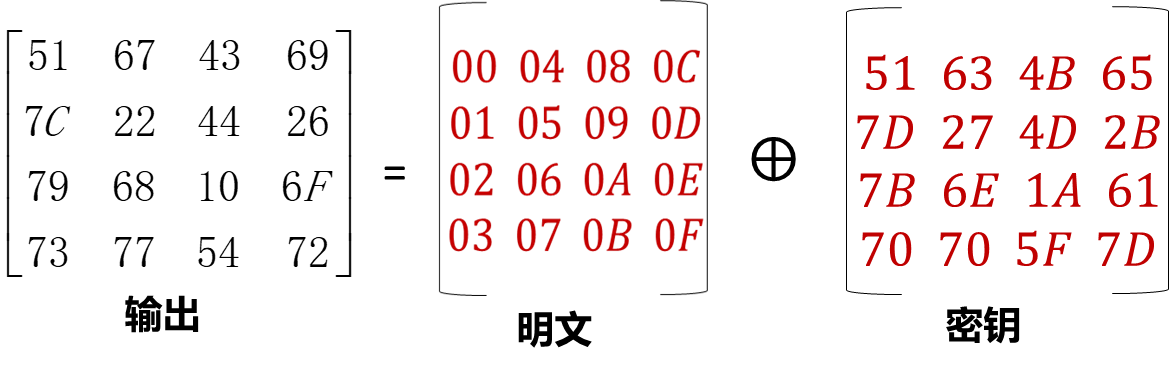
明文：00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

密钥：51 7D 7B 70 63 27 6E 70 4B 4D 1A 5F 65 2B 61 7D

明文 密钥

开始10轮之前的轮密钥加



3.2字节代换ByteSub ()

字节代换ByteSub (State)是一个关于字节的非线性变换，可以使用S盒查表得到输出。

AES定义了一个S盒，是由16×16个字节组成的方形表，包含了8位值所能表示的256种可能的变换。对于已知的某一字节作为S盒的输入，把该字节的高4位作为行号，低4位作为列号，查附录表1取出S盒中对应行列交叉点的元素作为输出。

设当前的State为:



查附录表1经变换后得到:



3.3行移位ShifRow()

在行移位变换中，状态矩阵State中的每一行将以字节为单位，循环左移不同的位移量。State的第一行保持不变，第二行循环左移一个字节，第三行循环左移两个字节，第四行循环左移三个字节。

根据3.2中的State，进行行移位变换后为：

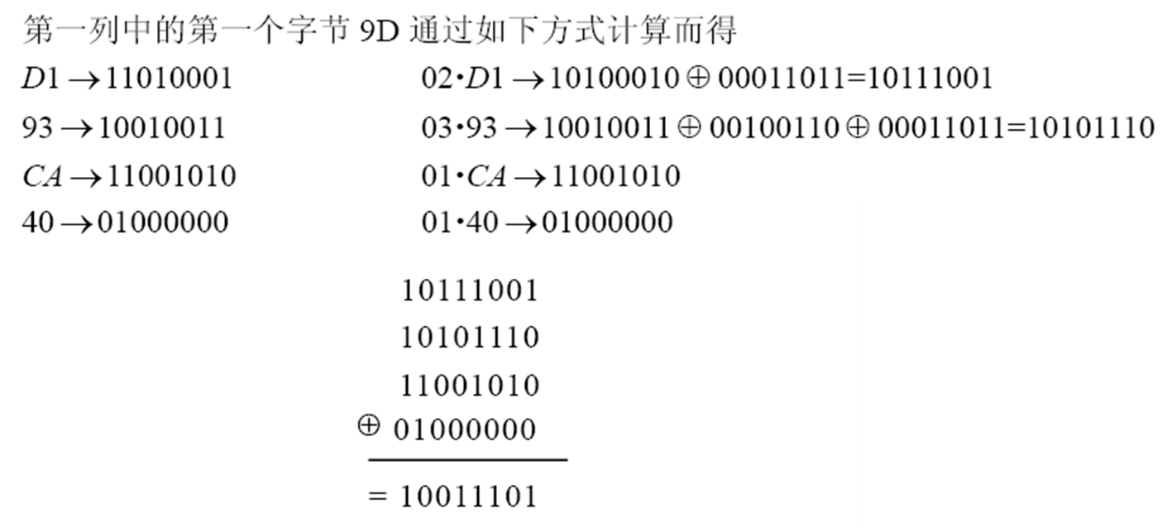


3.4列混淆MixColumn( )

列混淆变换将State乘以一个固定的矩阵A，对State逐列进行变换，每一列中的每个字节被变换成一个新值，直到4列都变换完毕。相乘后得到的乘积矩阵，其中每个元素均是一行和一列中所对应元素的乘积之和。这里的乘法和加法都是定义在有限域GF(28 )上的。

将3.2中的State进行列混淆后为：

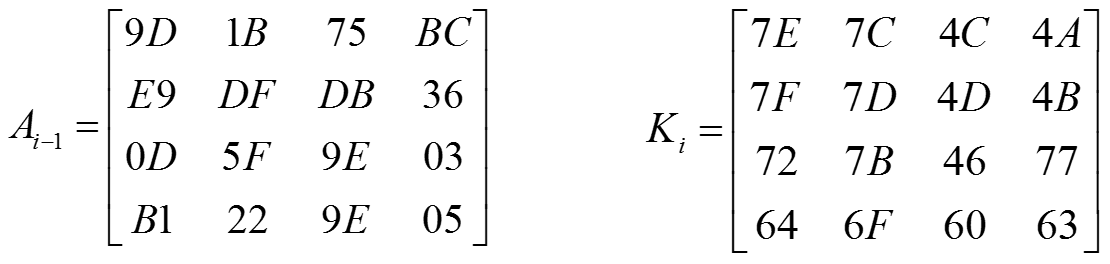




其他字节可用类似方法得出。

3.5轮密钥加

轮密钥加是将轮密钥Ki简单地与状态State进行逐比特异或。



求轮密钥加后的结果Ci。



3.6密钥扩展（以16字节密钥扩展为176字节为例）

3.6.1由密钥扩展算法将种子密钥扩展成为扩展密钥的计算过程如下：

1、当i<4时

W[0]=(k00k10k20k30)

W[1]=(k01k11k21k31)

W[2]=(k02k12k22k32)

W[3]=(k03k13k23k33)

2、当i>=4时



这里的*Nk*＝4，其中：

*temp*=SubByte (RotByte (*W*[*i*-1])) Rcon[j]

RotByte ( )表示循环左移一个字节；

SubByte( )是S盒的字节代换；

Rcon[j]为轮常数，其中j是轮数。

**3.6.2**假定种子密钥K0值如下，计算第一轮的子密钥K1的值的过程如下：

K0= 06 07 08 09 0A 0B 0C 0D 0E 0F 00 01 02 03 04 05



其中

*W*0 = 06 07 08 09

*W*1 = 0*A* 0*B* 0*C* 0*D*

*W*2 = 0*E* 0*F* 00 01

*W*3 = 02 03 04 05

根据密钥扩展算法可计算出，其中Rcon[1]的值可查附录表1

***W*4 = SubByte (RotByte (*W*3))** ⊕ **Rcon[1]** ⊕ ***W*0**

**= SubByte(03 04 05 02)** ⊕ **Rcon[1]** ⊕***W*0**

**= (7*B* *F*2 6*B* 77)** ⊕**(01 00 00 00)** ⊕ **(06 07 08 09)**

**=7*C* *F*5 63 7*E***

***W*5 = *W*1** ⊕ ***W*4 = (0*A* 0*B* 0*C* 0*D*)** ⊕ **(7*C* *F*5 63 7*E*) = 76 *FE* 6*F* 73**

***W*6 = *W*2** ⊕ ***W*5 = (0*E* 0*F* 00 01)** ⊕ **(76 *FE* 6*F* 73) = 78 *F*1 6*F* 72**

***W*7 = *W*3** ⊕ ***W*6 = (02 03 04 05)** ⊕ **( 78 *F*1 6*F* 72) = 7*A* *F*2 6*B* 77**

从而得出第一轮的子密钥K1的值如下：



3.7解密过程

AES算法的整个解密过程与加密过程类似，要依次完成：初始密钥加，Nr-1轮解密逆变换，最后一轮逆变换操作。

解密轮变换使用前面加密轮变换的逆变换函数，解密算法中的一轮与加密算法的一轮有类似的结构，依次进行逆字节代换，逆行移位，逆列混淆和轮密钥加。

InvByteSub ( )是ByteSub ()的逆变换，通过查附录表3逆S盒来实现。

InvShifRow( )是ShifRow()的逆变换，对State的各行进行一定量的循环移位（Nb=4）。

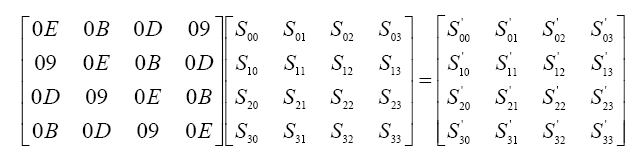
第0行不移位；

第1行循环右移1个字节；

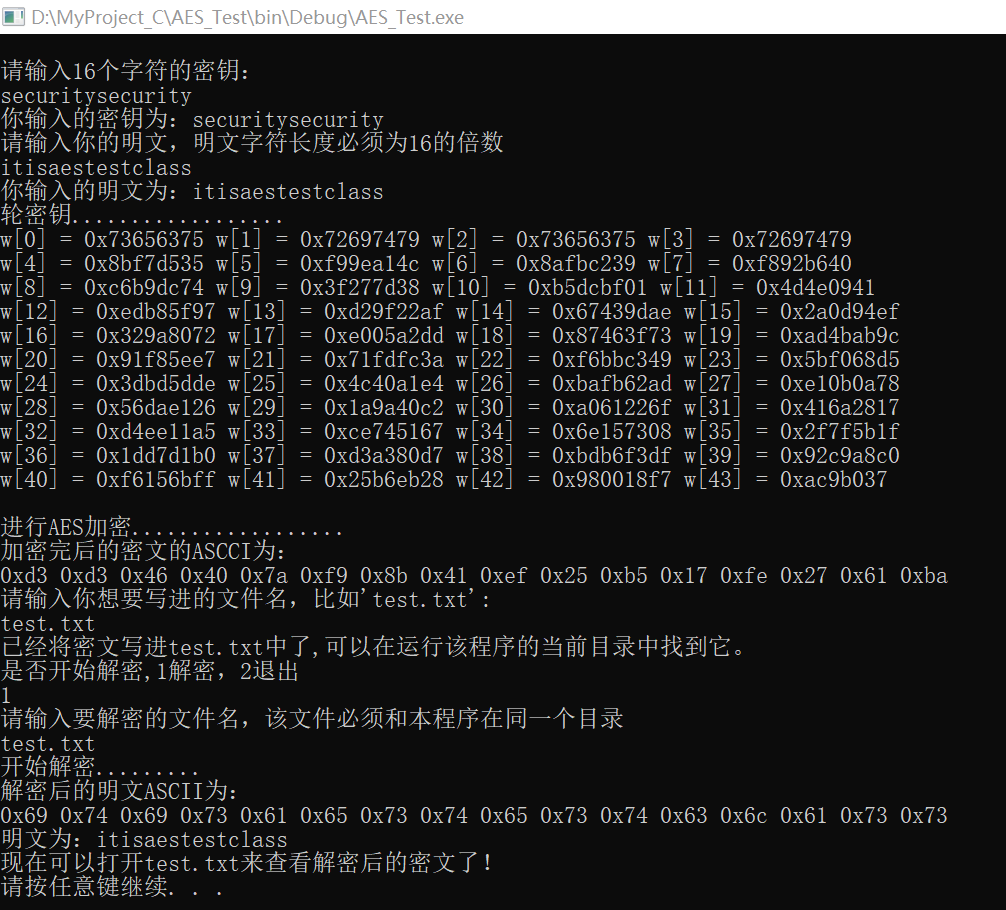
第2行循环右移2个字节；

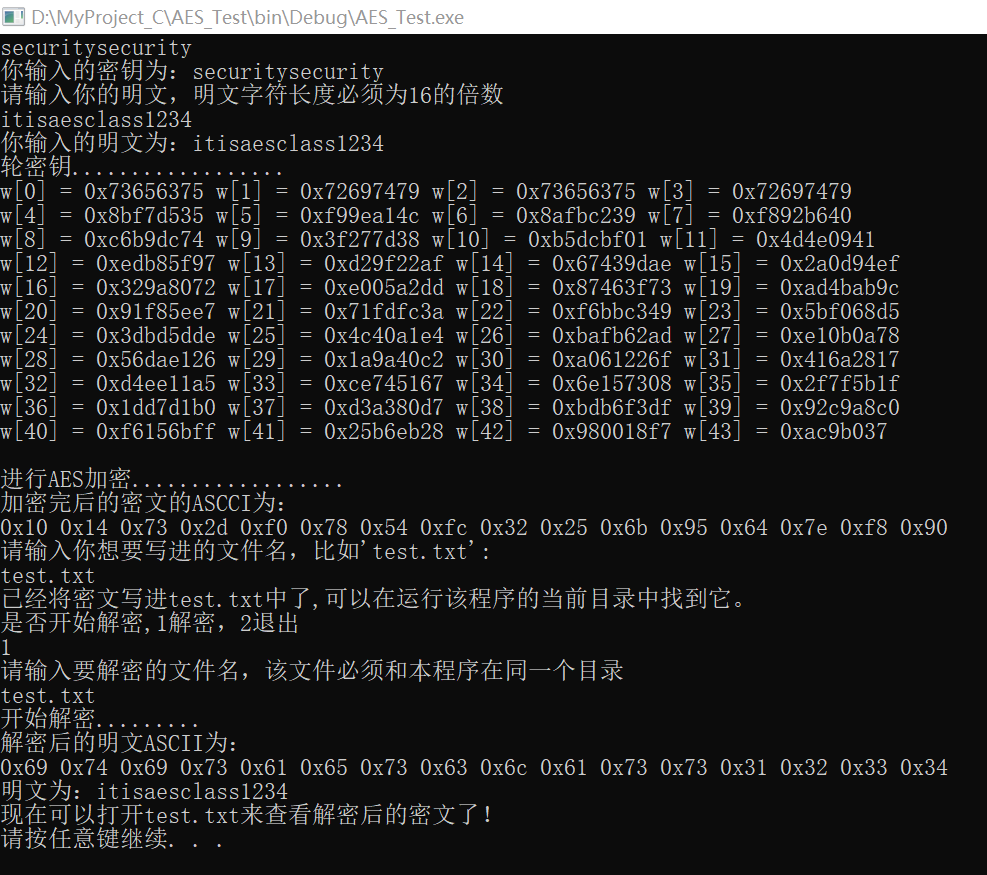
第3行循环右移3个字节。

InvMixColumn( )是MixColumn( )的逆变换，可由如下矩阵乘法定义



### 测试用例（部分用例供参考）





### 实验报告

1、请把电子版实验报告及源代码打包成一个压缩包上传到系统中（截止日期5月15日24点），命名格式如下：

实验报告：“学号\_姓名\_实验3”

压缩包：“学号\_姓名\_实验3”

### 附录





