实验三 分组密码算法 AES

一、 实验目的

通过用AES算法对实际的数据进行加密和解密来深刻了解AES的运行原理。

二、 实验环境

运行 Windows 操作系统的 PC 机，具有 VC 等语言编译环境

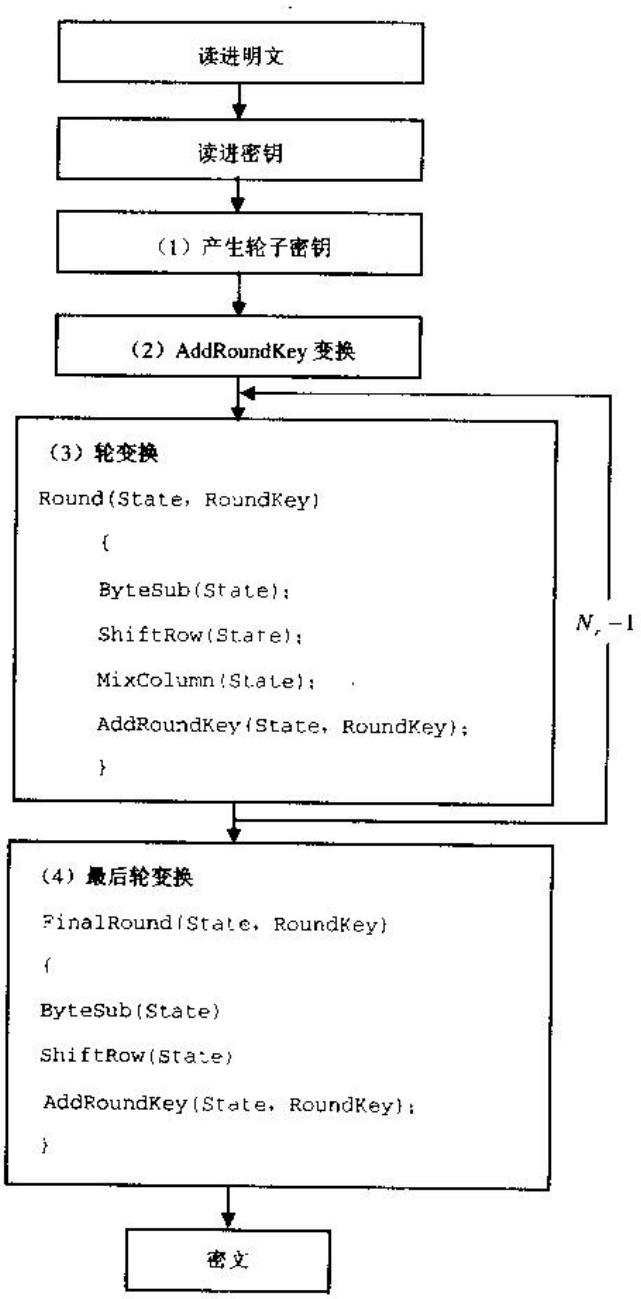

图 1 AES 加密/解密流程

三、 实验原理

AES 算法本质上是一种对称分组密码体制，采用代替/置换网络，每轮由三层组成：线性混合层确保多轮之上的高度扩散，非线性层由 16 个 S 盒并置起到混淆的作用，密钥加密层将子密钥异或到中间状态。Rijndael 是一个迭代分组密码，其分组长度和密钥长度都是可变的，只是为了满足 AES 的要求才限定处理的分组大小为 128 位，而密钥长度为 128 位、192 位或 256 位，相应的迭代轮数 N，为 10 轮、12 轮、14 轮。AES 汇聚了安全性能、效率、可实现性、灵活性等优点。最大的优点是可以给出算法的最佳差分特征的概率，并分析算法抵抗差分密码分析及线性密码分析的能力。其实现的加密流程图如图-1 所示。

加密的主要过程包括：对明文状态的一次密钥加，N_r-1 轮轮加密和末尾轮轮加密，最后得到密文。其中N_r-1 轮轮加密每一轮有四个部件，包括字节代换部件 ByteSub、行移位变换 ShiftRow、列混合变换 NixColumn 和一个密钥加 AddRoundKey 部件，末尾轮加密和前面轮加密类似，只是少了一个列混合变换 NixColumn 部件。下面具体介绍这几个部件的实现方法：

1、字节代换 ByteSub 部件

字节代换是非线形变换，独立地对状态的每个字节进行。代换表（即 S-盒）是可逆的，由以下两个变换的合成得到：

（1） 首先，将字节看作 GF(2^8)上的元素，映射到自己的乘法逆元，‘00’映射到自己。

（2）其次，对字节做如下的（GF(2)上的，可逆的）仿射变换：

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

该部件的逆运算部件就是先对自己做一个逆仿射变换，然后映射到自己的乘法逆元上。

2、行移位变换 ShiftRow

行移位是将状态阵列的各行进行循环移位，不同状态行的位移量不同。第 0 行不移动，第 1 行循环左移 C_1 个字节，第 2 行循环左移 C_2 个字节，第 3 行循环左移 C_3 个字节。位移量 C_1、C_2、C_3 的取值与 N_b 有关，由教材中表 3.10 给出。

ShiftRow 的逆变换是对状态阵列的后 3 列分别以位移量 N_b-C_1、N_b-C_2、N_b-C_3 进行循环移位，使得第 i 行第 j 列的字节移位到$(j+N_b-C_i) \bmod N_b$。

3、列混合变换 NixColumn

在列混合变换中，将状态阵列的每个列视为系数为 GF(2^8)上的多项式，再与一个固定的多项式 c(x)进行模 x^4+1 乘法。当然要求 c(x)是模 x^4+1 可逆的多项式，否则列混合变换就是不可逆的，因而会使不同的输入分组对应的输出分组可能相同。Rijndael 的设计者给出的 c(x)为（系数用十六进制数表示）：

$$c(x)=\text{‘03’}x^3+\text{‘01’}x^2+\text{‘01’}x+\text{‘02’}$$

c(x)是与 x^4+1 互素的，因此是模 x^4+1 可逆的。列混合运算也可写为矩阵乘法。设 b(x)= c(x)⊗a(x)，则

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

这个运算需要做 GF(2^8)上的乘法，但由于所乘的因子是 3 个固定的元素 02、03、01，所以这些乘法运算仍然是比较简单的。

列混合运算的逆运算是类似的，即每列都用一个特定的多项式 d(x)相乘。d(x)满足

$$(\text{‘03’}x^3+\text{‘01’}x^2+\text{‘01’}x+\text{‘02’}))\otimes d(x)=\text{‘01’}$$

由此可得

$$d(x)=\text{‘0B’}x^3+\text{‘0D’}x^2+\text{‘09’}x+\text{‘0E’}$$

4、密钥加 AddRoundKey 部件

密钥加是将轮密钥简单地与状态进行逐比特异或。轮密钥由种子密钥通过密钥编排算法得到，轮密钥长度等于分组长度 N_b。密钥加运算的逆运算是其自身。

5、密钥编排

密钥编排指从种子密钥得到轮密钥的过程，它由密钥扩展和轮密钥选取两部分组成。其基本原则如下：

（1） 轮密钥的字数（4 比特 32 位的数）等于分组长度乘以轮数加 1；

（2） 种子密钥被扩展成为扩展密钥；

（3） 轮密钥从扩展密钥中取，其中第 1 轮轮密钥取扩展密钥的前 N_b 个字，第 2 轮轮密钥取接下来的 N_b 个字，如此下去。

密钥扩展的方法和密钥的取法具体请参考教材和 ppt。

6、解密过程

AES 算法的解密过程和加密过程是相似的，也是先经过一个密钥加，然后进行$N_r - 1$ 轮轮解密和末尾轮轮解密，最后得到明文。和加密不同的是$N_r - 1$ 轮轮解密每一轮四个部件都需要用到它们的逆运算部件，包括字节代换部件的逆运算、行移位变换的逆变换、逆列混合变换和一个密钥加部件，末尾轮加密和前面轮加密类似，只是少了一个逆列混合变换部件。

在解密的时候，还要注意轮密钥和加密密钥的区别，设加密算法的初始密钥加、第 1 轮、第 2 轮、…、第 N_r 轮的子密钥依次为

$k(0)$, $k(1)$, $k(2)$, …, $k(N_{r-1})$, $k(N_r)$

则解密算法的初始密钥加、第 1 轮、第 2 轮、…、第 Nr 轮的子密钥依次为

$k(N_r)$, InvMixColumn ($k(N_{r-1})$), InvMixColumn ($k(N_{r-2})$), …,

InvMixColumn ($k(1)$), $k(0)$。

四、 实验环境

运行 Windows 操作系统的 PC 机，具有 VC 等语言编译环境

五、 实验内容和步骤，

1．算法分析：

对课本中 AES 算法进行深入分析，对其中用到的基本数学算法、字节代换、行移位变换、列混合变换原理进行详细的分析，并考虑如何进行编程实现。对轮函数、密钥生成等环节要有清晰的了解，并考虑其每一个环节的实现过程。

2．AES 实现程序的总体设计：

在第一步的基础上，对整个 AES 加密函数的实现进行总体设计，考虑数据的存储格式，参数的传递格式，程序实现的总体层次等，画出程序实现的流程图。

3．在总体设计完成后，开始具体的编码，在编码过程中，注意要尽量使用高效的编码方式。

4．利用 3 中实现的程序，对 AES 的密文进行雪崩效应检验。即固定密钥，仅改变明文中的一位，统计密文改变的位数；固定明文，仅改变密钥中的一位，统计密文改变的位数。

六、 实验报告和要求

（1） 实现 AES 的加密和解密，提交程序代码和执行结果。

（2） 在检验雪崩效应中，要求至少改变明文和密文中各八位，给出统计结果并计算出平均值。