AES 加密算法简介

李丽娟

2013.11.12

AES（高级加密标准）为分组密码算法，明文密文长度为 128 位（16 个字节），密钥长度可以为 128 位、192 位或 256 位，对应的轮变换次数分别为 10 轮、12 轮、14 轮。我们以密钥长度为 128 位的 AES 算法为例进行介绍。128-AES 加密算法的伪代码如下：

```
CipherAES128(byte in[16], byte out[16], word w[44])
{
    byte state[4,4];
    state = in;
    AddRoundKey(state, w[0, 3]);
    for (round = 1; round <10; round ++)
    {
        SubBytes(state) ;
        ShiftRows(state) ;
        MixColumns(state);
        AddRoundKey(state, w[round*4, (round+1)*4-1]);
    }
    SubBytes(state);
    ShiftRows(state);
    AddRoundKey(state, w[40, 43]);
    out= state;
}
```

对于 128-AES，加密先经过一个初始的轮密钥加，然后经过 9 次轮变换，每轮变换包括：字节替换（SubBytes）、行移位（ShiftRows）、列混淆（MixColumns）和轮密钥加（AddRoundkey）。最后还要经过一轮变换，这轮变换只有三种运算，没有列混淆。

AES 直接解密算法的伪代码如下：

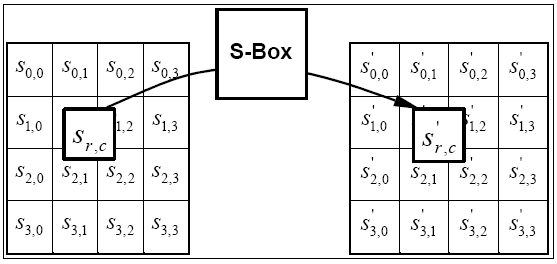
```
InvCipher128 (byte in[16], byte out[16], word w[44])
{
    byte state[4,4];
    state = in;
    AddRoundKey(state, w[40, 43]);
    for (round = 9; round >0; round --)
    {
        InvShiftRows(state) ;
        InvSubBytes(state) ;
        AddRoundKey(state, w[round*4, (round+1)*4-1]);
```

```
            InvMixColumns(state);
        }
        InvShiftRows(state);
        InvSubBytes(state);
        AddRoundKey(state, w[0, 3]);
        out= state;
    }
```

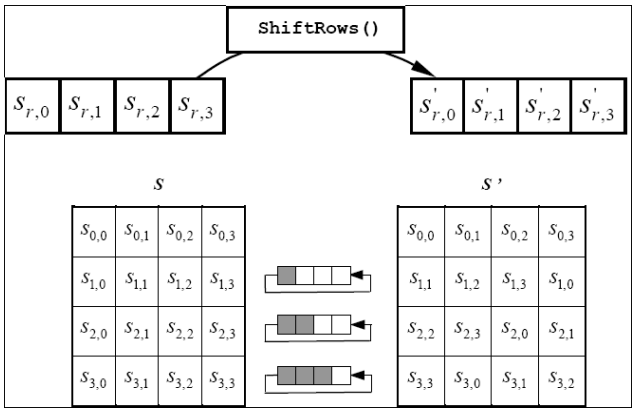
AES 解密变换是加密变换的逆过程，除了将所有变换转化为逆变换之外，各个运算的顺序也有所不同，密钥也需要逆序使用。解密先经过一个初始的轮密钥加，然后经过 9 次轮变换，每轮变换的顺序为：逆行移位（InvShiftRows）、逆字节替换（InvSubBytes）、轮密钥加（AddRoundkey）和逆列混淆（InvMixColumns），最后还要经过一轮变换，这轮变换只有三种运算，没有列混淆。

下面我们只简要介绍 128-AES 加密算法和密钥扩展算法，对 AES 其他部分感兴趣的可以阅读 FIPS-197 标准。AES 加密轮变换中的四种运算简介如下：

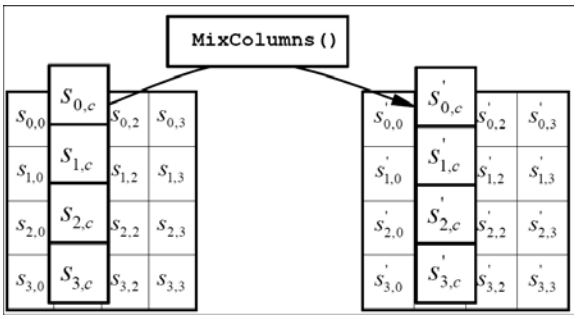
- 字节替换（SubBytes）：用一个 S 盒完成分组的字节到字节的代替。可用查找表法实现，即以 8 位输入为地址，查表得到 8 位输出。



- 行移位（ShiftRows）：state 数组的第一行保持不变，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。



- 列混淆（MixColumns）：对 state 数组的每一列单独进行操作，可由（1）式所示的矩阵乘法表示。



$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad (1)$$

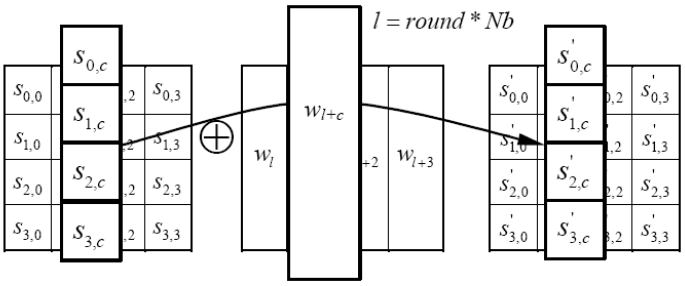
$$\begin{aligned} s'_{0,c} &= (02 \cdot s_{0,c}) \oplus (03 \cdot s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (02 \cdot s_{1,c}) \oplus (03 \cdot s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (02 \cdot s_{2,c}) \oplus (03 \cdot s_{3,c}) \\ s'_{3,c} &= (03 \cdot s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (02 \cdot s_{3,c}) \end{aligned} \quad (2)$$

（2）式中包含有限域 $GF(2^8)$的加法（⊕ ）和乘法（· ）操作。这里只给出计算方法。加法 ⊕即为按位异或操作。AES 中定义的乘法规则如下，设$s = (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)$，则

$$02 \cdot s = \begin{cases} (b_6 b_5 b_4 b_3 b_2 b_1 b_0 0), & b_7 = 0 \\ (b_6 b_5 b_4 b_3 b_2 b_1 b_0 0) \oplus (00011011), & b_7 = 1 \end{cases}$$

$03 \cdot s = (02 \cdot s) \oplus s$

- 轮密钥加（AddRoundKey）：将 state 的每列和对应的轮密钥的列进行异或，即 128 位的异或运算。



- 密钥扩展算法：AES 还需要密钥扩展算法来为每轮变换提供密钥。128-AES 由 128 位初始密钥（4 word）扩展为 4x(10+1)=44 word 的密钥。密钥扩展的伪代码如下,其中⊕为异或操作。

```
KeyExpansion128 (byte key[16], word w[44])
{
    word temp
    for (i=0;i<4;i++)
        w[i] = (key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);
    for (i=4;i<44;i++)
    {
```

```
            temp = w[i-1];
            if (i mod 4 = 0)     temp = SubByte(Rotword(temp))⊕Rcon[i/4];
            w[i]=w[i-4]⊕temp;
        }
}
```

Rotword：将一个字中的 4 个字节循环左移一个字节，即将输入字$[B_0, B_1, B_2, B_3]$变换成$[B_1, B_2, B_3, B_0]$。

SubByte：用 S 盒对输入字中的每个字节进行字节代替。

Rcon: 轮常量是一个字，这个字最右边三个字节总为 0，因此字与 Rcon 相异或，其结果只是与该字最左的字节相异或。定义 Rcon[j]=(RC[j],0,0,0)。RC[j]的值按十六进制表示为：

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

RC[j]也可由 RC[j+1]<=02·RC[j]计算生成。