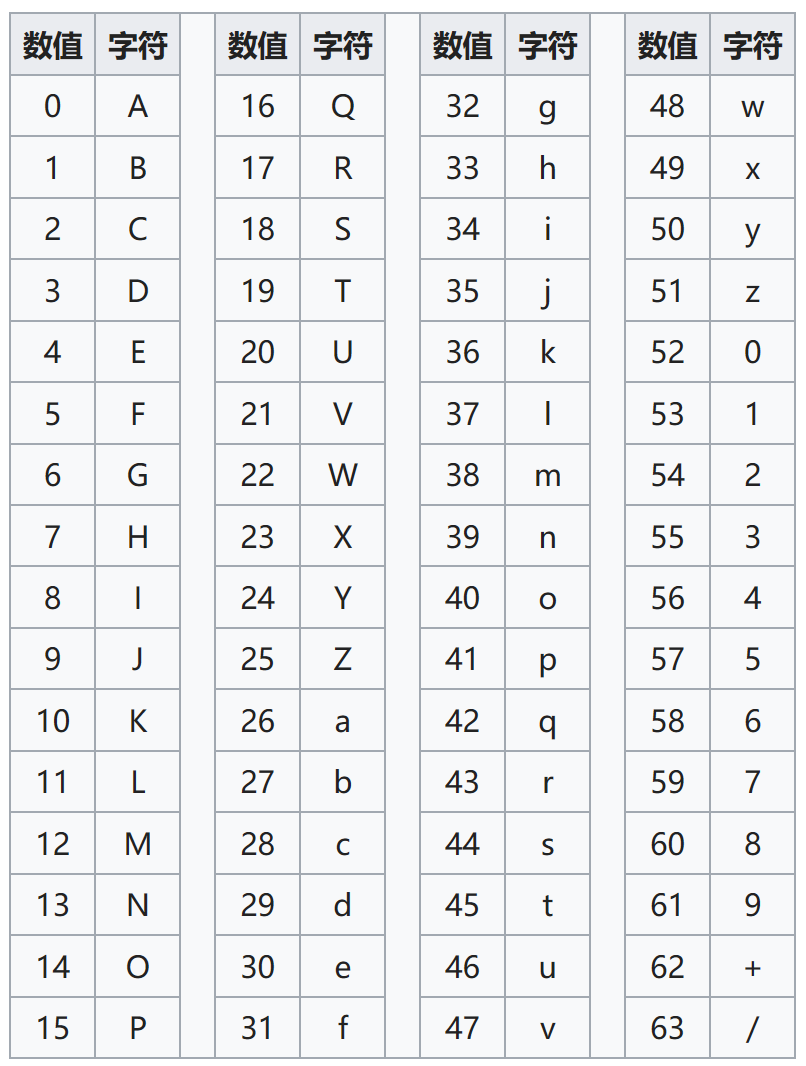
编码简介文档

# Base64编码

## 简介

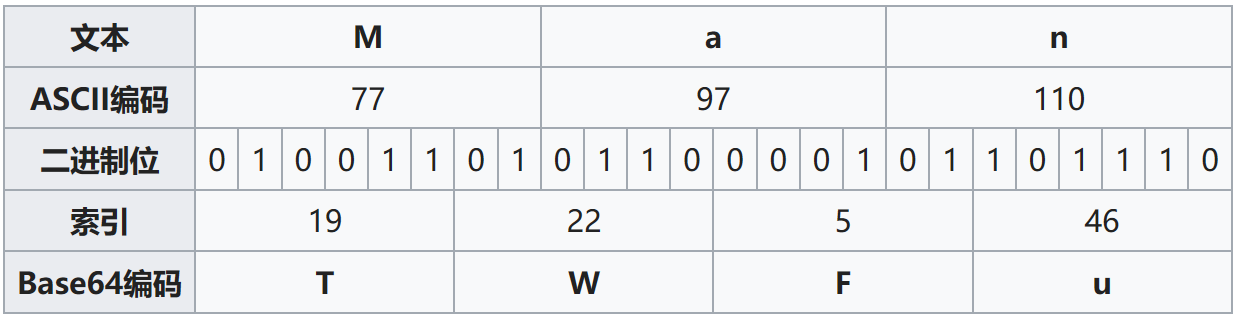
Base64是一种基于64个可打印字符来表示二进制数据的表示方法。由于 2^6 = 64 ，所以每6个位元为一个单元，对应某个可打印字符。3个字节有24个位元，对应于4个Base64单元，即3个字节可表示4个可打印字符。它可用来作为电子邮件的传输编码。在Base64中的可打印字符包括字母A-Z、a-z、数字0-9，这样共有62个字符，此外两个可打印符号在不同的系统中而不同。一些如uuencode的其他编码方法，和之后BinHex的版本使用不同的64字符集来代表6个二进制数字，但是不被称为Base64。Base64常用于在通常处理文本数据的场合，表示、传输、存储一些二进制数据，包括MIME的电子邮件及XML的一些复杂数据。

## Base64的编码方式



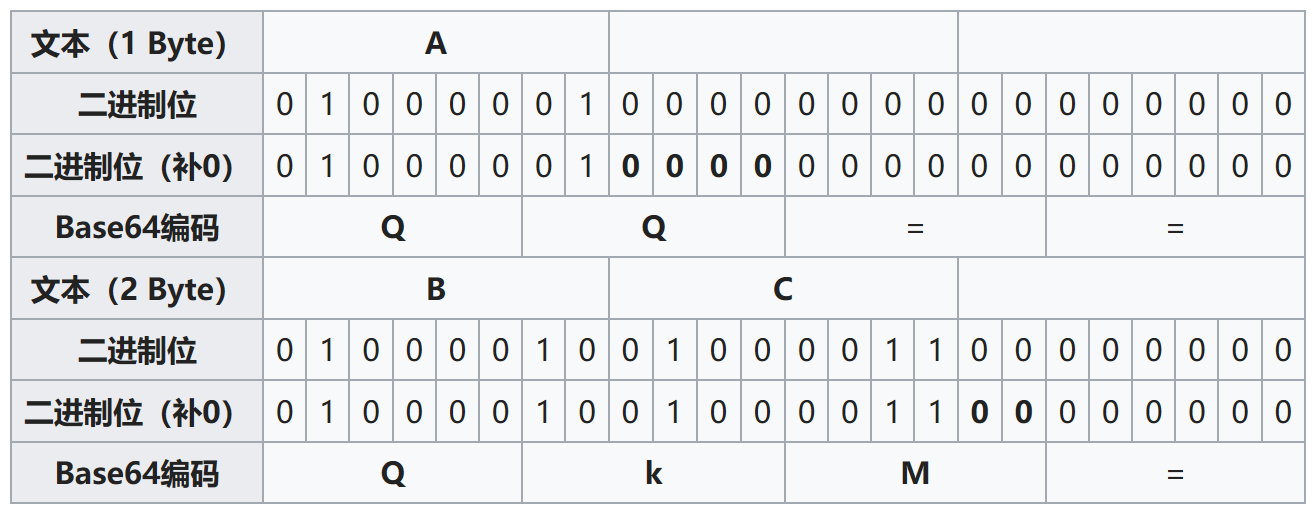
Base64索引表

Base64使用一组65个ASCII字符，约定6位代表一个可打印的字符（第65个字符为“=”，用于标记特殊的处理程序）。



使用Base64对Man进行编码

进行编码的时候，3个字符作为一组进行编码。以每个字符8位进行计算，编码后每个编码占六位，也就是一个三个字符的原文编码后成为四位的编码。编码时先将字符按ASCII转换成二进制编码，一组的所有成员看成一个整体，从左到右按六位进行分割，然后将得到的4个值，通过查找Base64的索引，转换位字符编码；



如果字符串的长度不是3（每组编码个数）的倍数，那就在字符串末尾补上0，凑足3的倍数，编码时结尾补上的0对应的编码变为=，补了几个0就会有几个=；

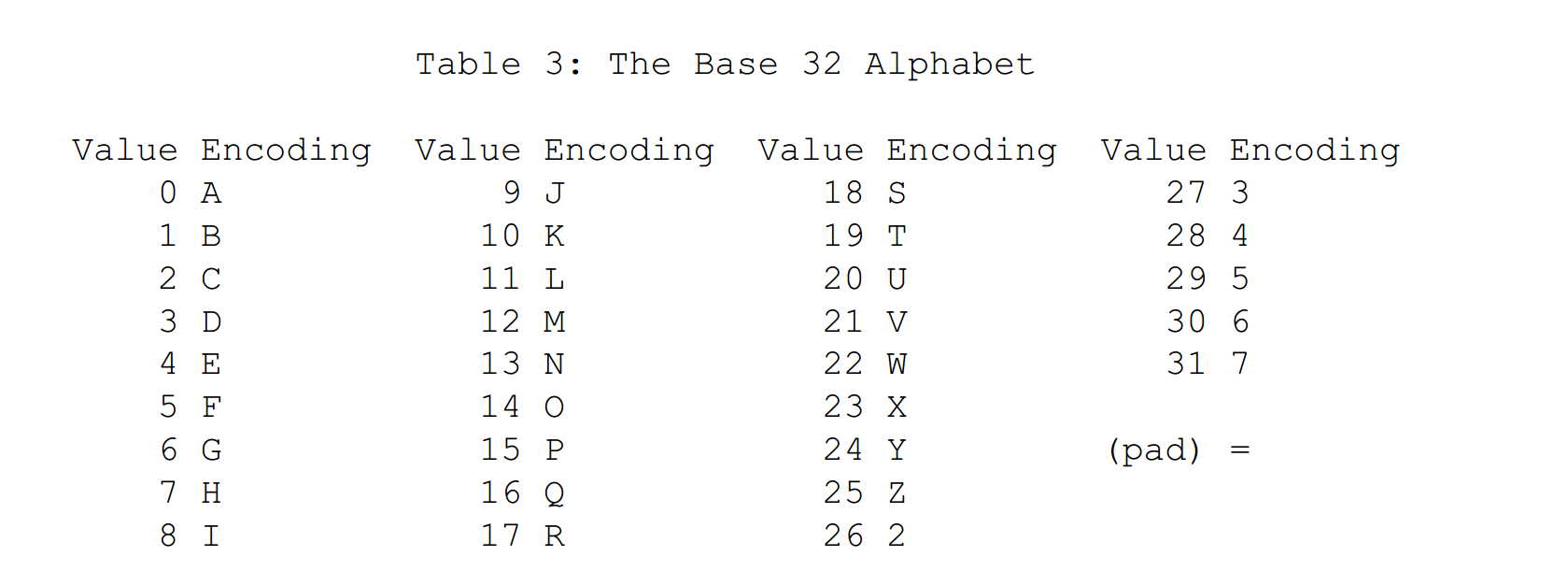
解码的时候，以四个编码作为一组，将编码通过索引转换为二进制，将一组作为一个整体，从左到右按八位进行分割，然后将得到的数值转换为ASCII码得到的就是解码后的字符串。

# Base32编码

## 简介

Base32和Base64一样，是一种将二进制数据转换为可打印字符的编码方式，不同点在于Base32使用32个可见字符来表示数据，这也是Base32名字的来源。

## Base32编码方式



Base32索引表

Base32的编码方式与Base64相似，编码时将5个字符做为一组进行编码，先将五个字符按照ASCII转换为二进制，然后将整个组看成一个整体，从左到右每5位作为一个编码，然后查找索引表，将每个字符转换为可打印字符。

当最后一组字符不足五个字符是，需要在编码最后加=号补齐，规则如下：

1. 当最后一组缺4个字符满一组时，需要在结尾补6个“=”；
2. 当最后一组缺3个字符满一组时，需要在结尾补4个“=”；
3. 当最后一组缺2个字符满一组时，需要在结尾补3个“=”；
4. 当最后一组缺1个字符满一组时，需要在结尾补1个“=”；

Base32的解码规则也与Base64相似，8个编码作为一组进行解码，先将编码对照

索引表转换为二进制，然后将一组按8位作为一个字符解码。

# MD5编码

## 简介

MD5全称为Message Digest Algorithm 5（信息摘要算法，哈希算法），MD5的特点是不可逆和可以将任意长度的数据编码成固定长度的十六进制字符串。常用于确保信息传输完整一致。

## MD5的编码步骤

步骤一，位长扩展填充：

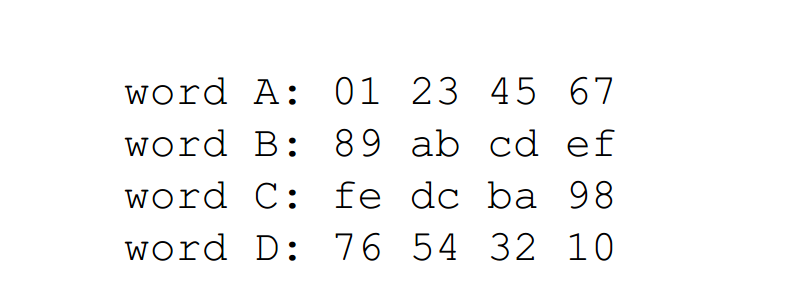
对数据进行填充使其长度对512求余的结果等于448；对数据填充的目的是使其刚好比512位的倍数少64位；位长填充是必须的，即使原数据长度已经满足条件。填充的方法为：先填充一个1，然后用0填充其他部分，直到满足条件。

步骤二，扩展长度：

用一个64位长度的数据记录之前填充的数据的长度，然后将64位的数据添加到前一步得到的数据中。在极少数情况下，如果这个数据的长度大于2^64，这种情况下只有数据的低64位会被使用。此时，得到一个刚好能被512整除的数据。

步骤三，初始化MD缓冲区：

初始化一个四字节的缓冲区，储存4个共32位的初始化链接变量A,B,C,D,这四个变量按大端字节序初始化为以下16进制值：



步骤四，分组处理数据：

首先定义四个方程，每个方程输入三个32位的数据，输出一个32位的数据：

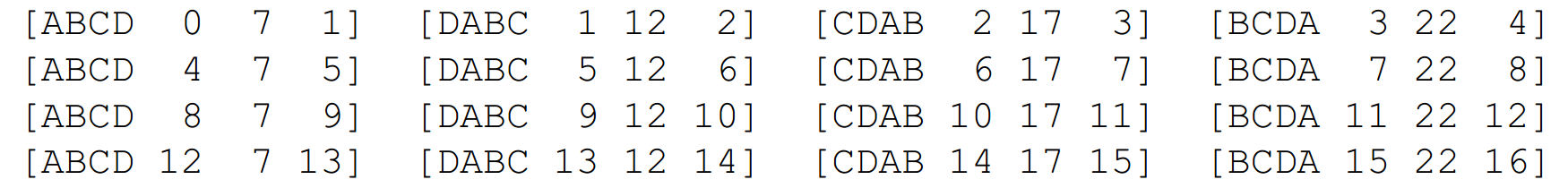
F(X,Y,Z) = XY v not(X) Z  
G(X,Y,Z) = XZ v Y not(Z)  
H(X,Y,Z) = X xor Y xor Z  
I(X,Y,Z) = Y xor (X v not(Z))

然后进行一下步骤

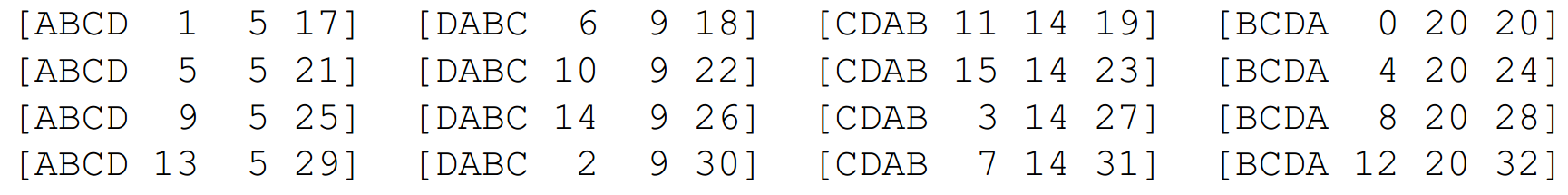
/\* Process each 16-word block. \*/  
For i = 0 to N/16-1 do  
/\* Copy block i into X. \*/  
For j = 0 to 15 do  
Set X[j] to M[i\*16+j].  
end /\* of loop on j \*/

然后把A,B,C,D保存为AA,BB,CC,DD

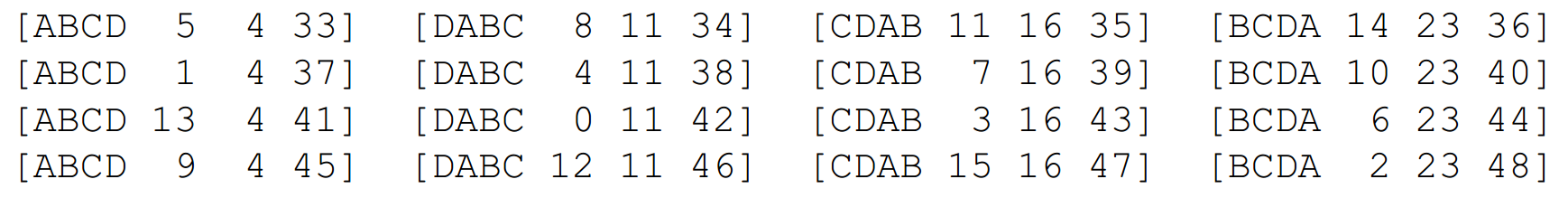
第一轮，用方程a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s处理数据（输入项为k,s,i，<<<表示循环左移位）

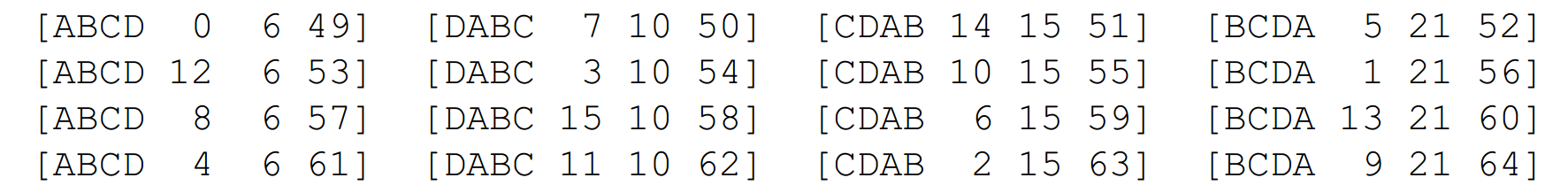


第二轮，用方程a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s处理数据



第三轮，用方程a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s处理数据

  
第四轮，用方程a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s

  
最后A = A + AA，B = B + BB，C = C + CC，D = D + DD；

步骤五，输出：

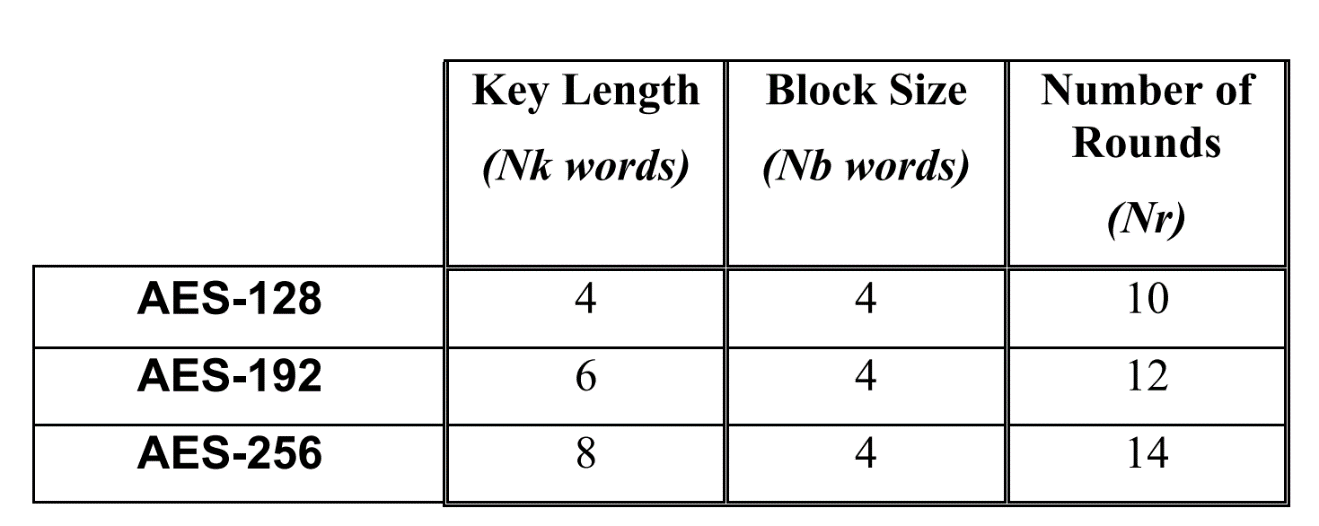
输出项为A,B,C,D四个变量，长度为32位，即该数据的MD5编码。

# AES加密

## 简介

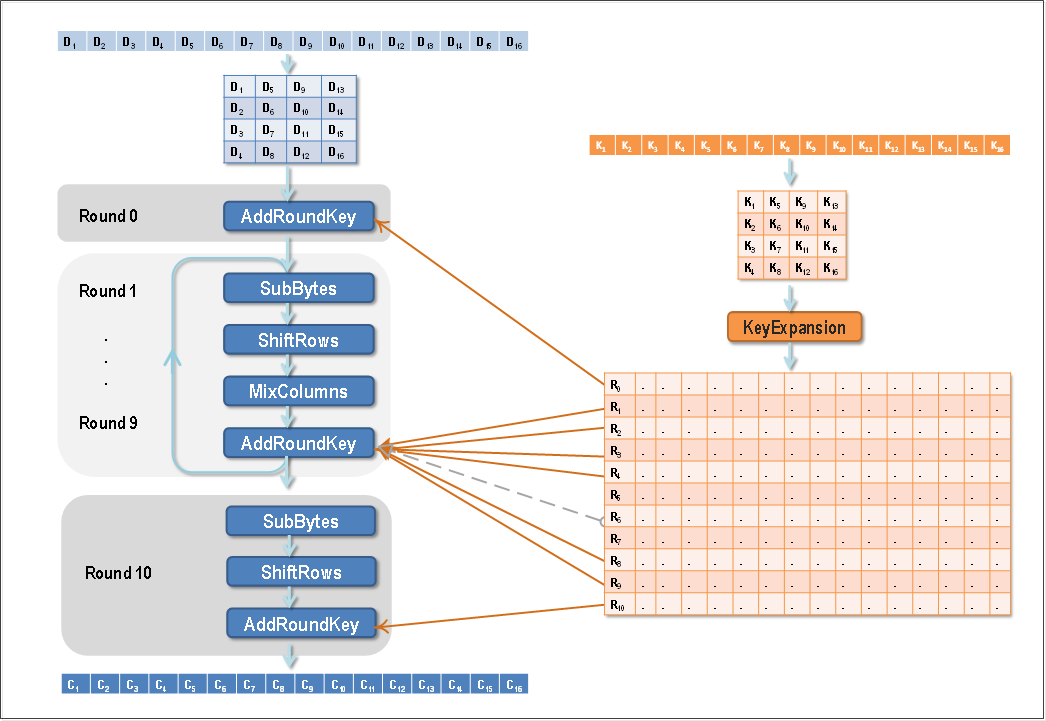
AES，全称为Advanced Encryption Standard（进阶加密标准），是美国联邦政府采用的一种区块加密标准。AES采用置换排列网络（SPN）,是一种被广泛使用的对称密钥加密算法。

## AES加密过程

AES加密过程中，每个数据加密区块的长度固定为128位，加密密钥长度可以为128位，192位或者256位（AES-128，AES-192，AES-256）。用户密钥会通过密钥扩展算法（Key Expansion）扩展成轮密钥(Round Key)。每个明文数据加密区块会以特定的次序生成一个4X4的字节矩阵（state），然后经过轮函数加密生成密文。根据密钥的长度明文数据会经过多次轮函数加密，每一轮加密的结果作为下一轮的输入。

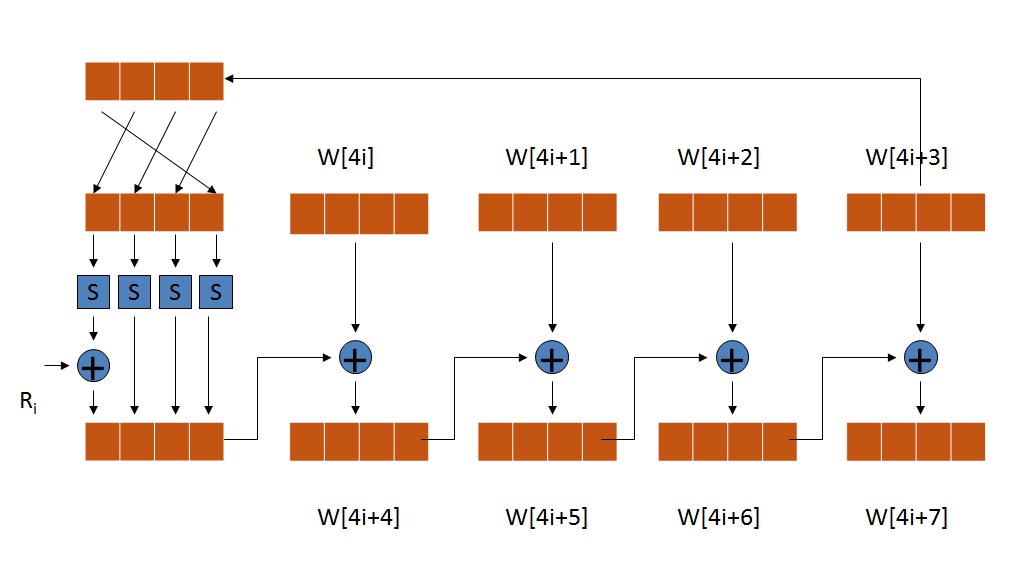
密钥长度与加密轮数

轮函数包括4种运算操作：字节代换（SubByte），行移位（ShiftRow），列混合（MixColumn）和轮密钥相加（AddRoundKey）。



AES加密流程

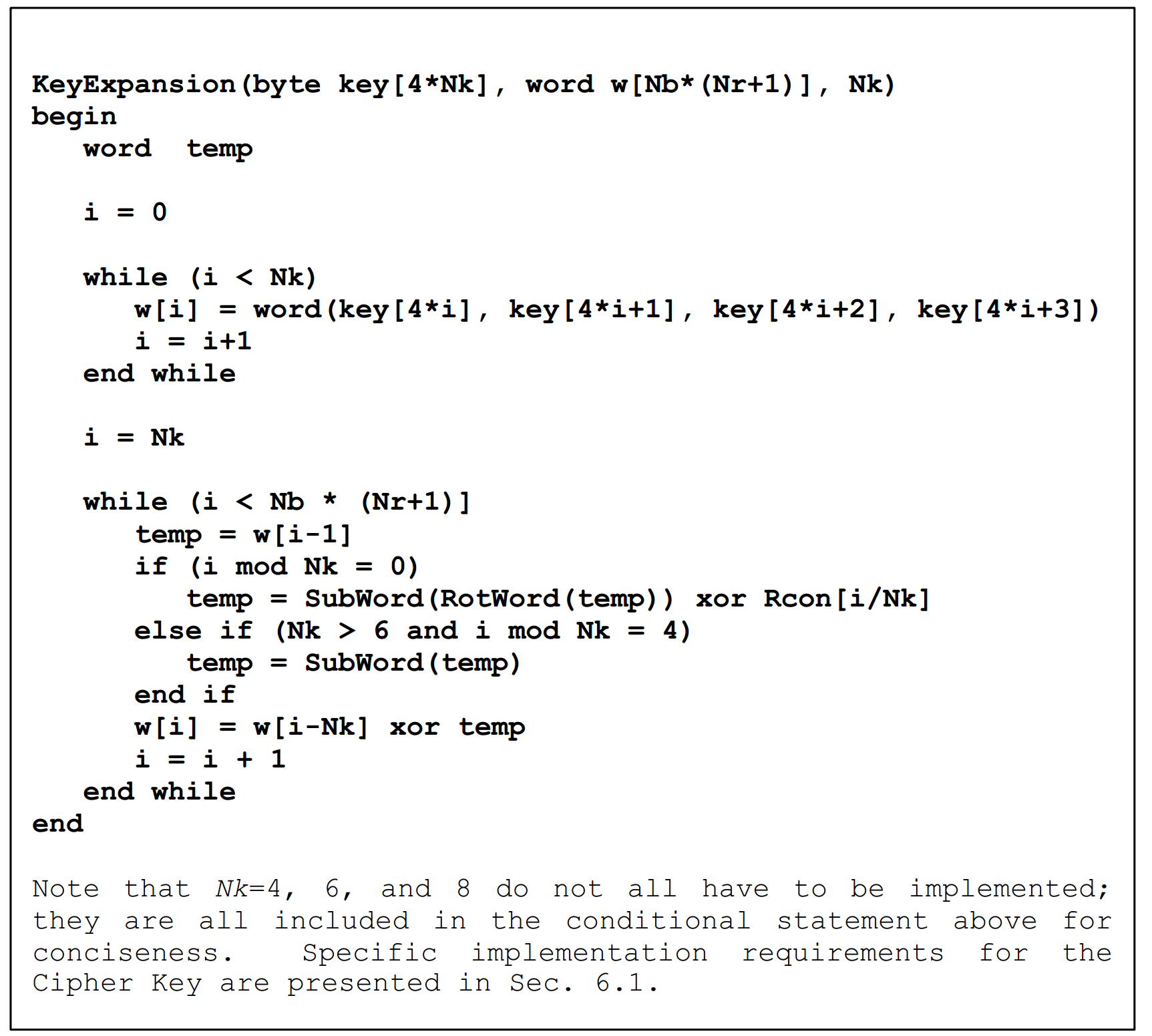
## 密钥扩展算法（Key Expansion）



密钥扩展算法将用户密钥扩展为多组128位的轮密钥。

先把用户密钥放入一个4(字节)x4的矩阵中，然后每次取矩阵每一行的一个字节合成一个4字节的字，得到一个16字节的数组；

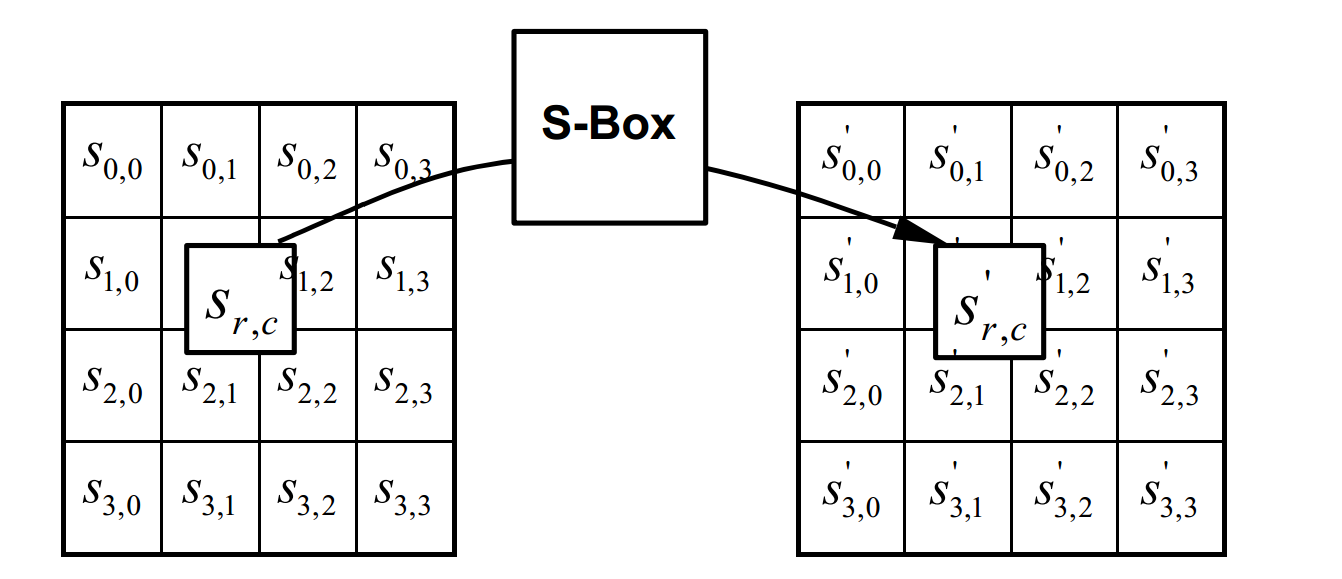
然后对再循环40次，每次取出



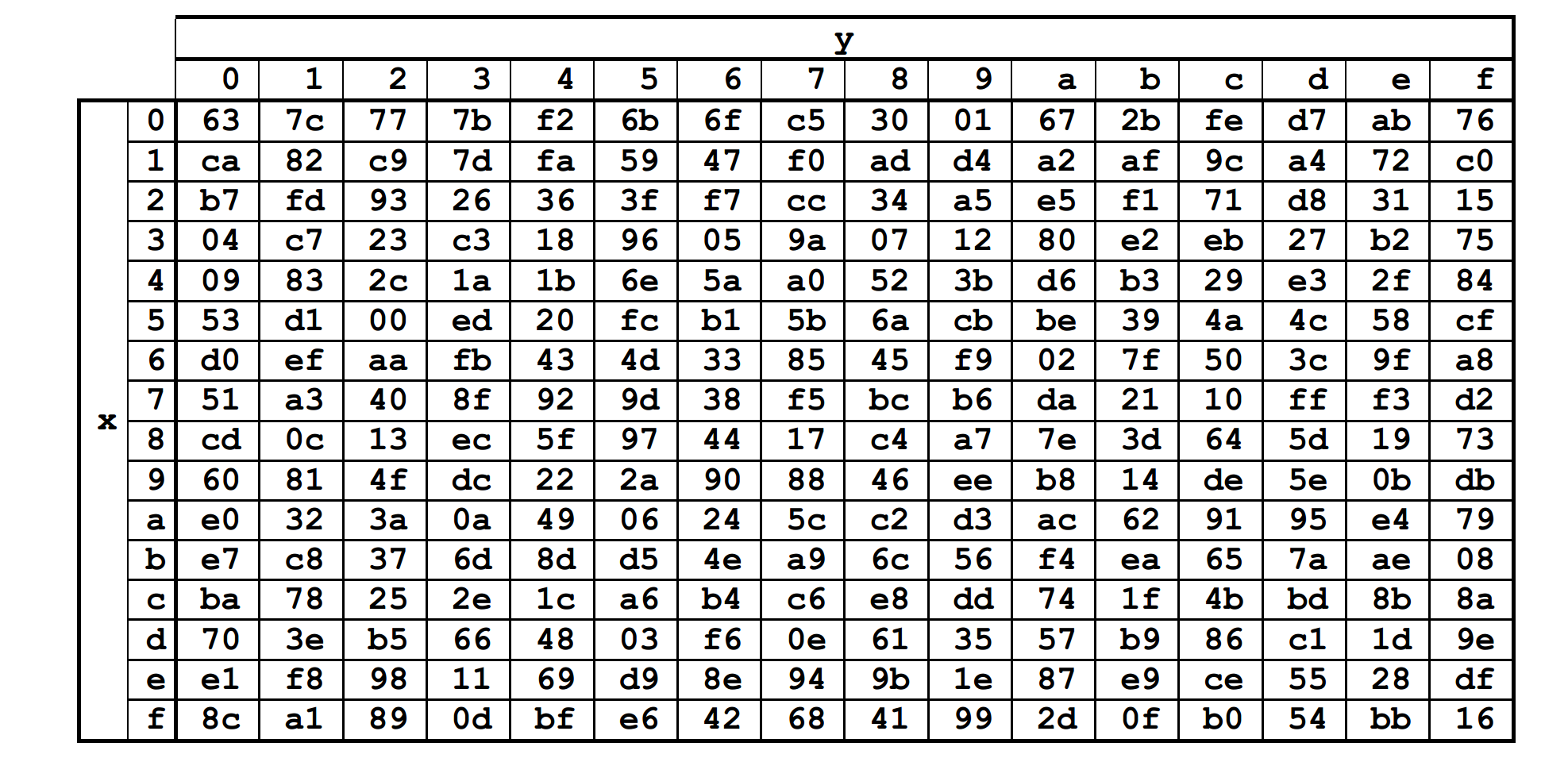
密钥扩展算法伪代码

## 字节代换（SubBytes）

字节代换是对state矩阵中每个字节独立查找s盒中元素然后进行替换的非线性运算。字节代换是AES中唯一一项非线性运算。字节代换中使用到的S盒是一组事先设计好的256个元素的查询表。



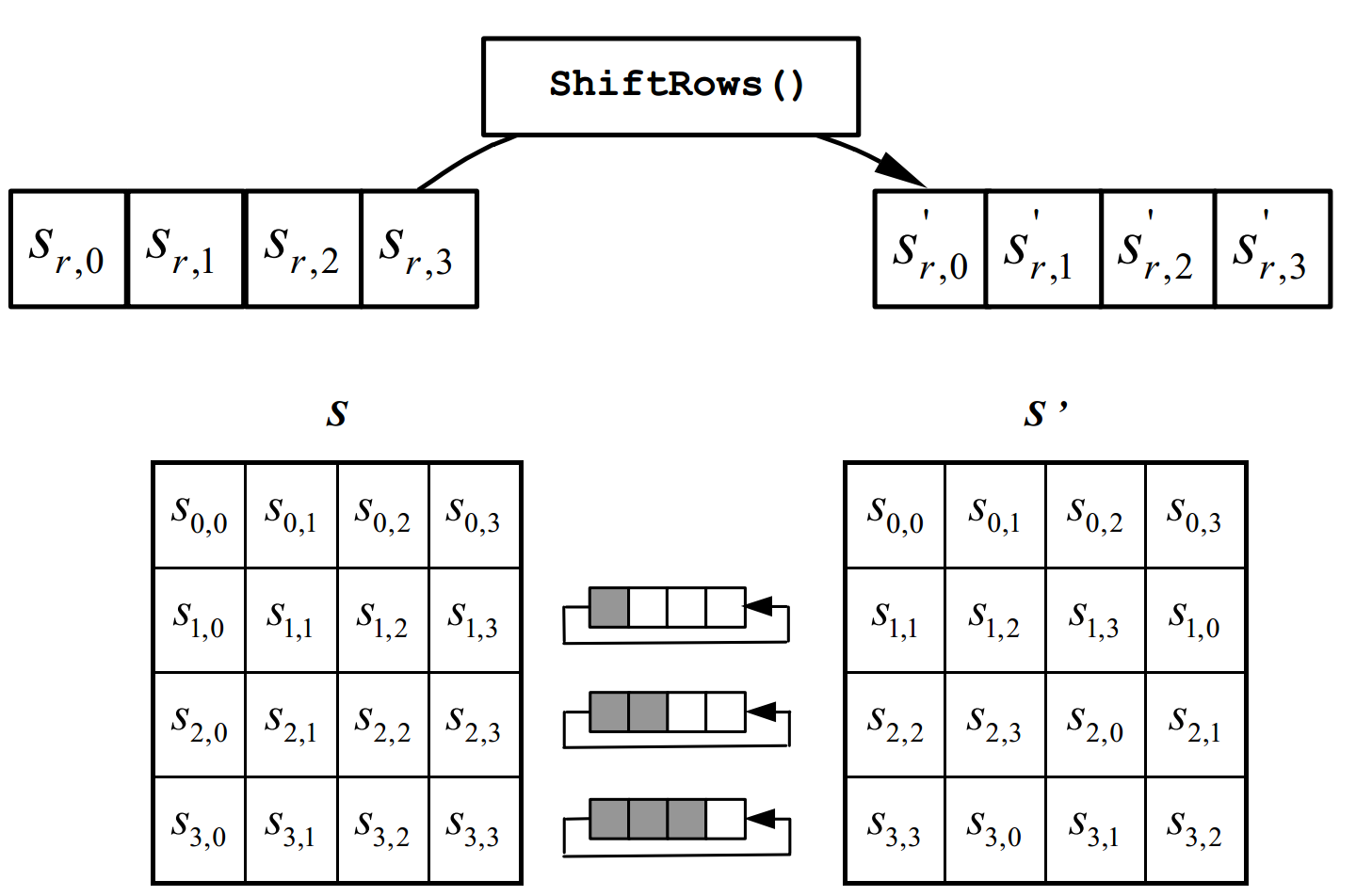
对state矩阵逐字节查找S盒替换元素



S盒

## 行移位（ShiftRow）

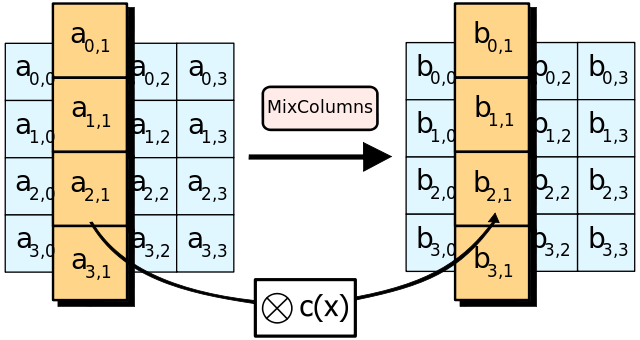
对State矩阵中第n行进行右移n位，第0行不进行移位。行移位主要目的是实现字节在每一行的扩散，属于线性变换。



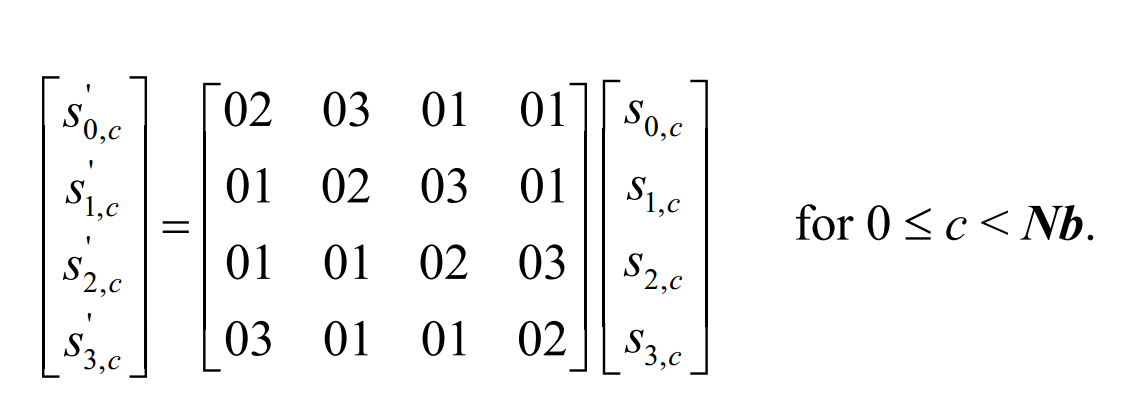
行移位

## 列混合（MixColumn）

列混合是将state矩阵逐列与常矩阵C相乘。列混合的目的是实现列的扩散。



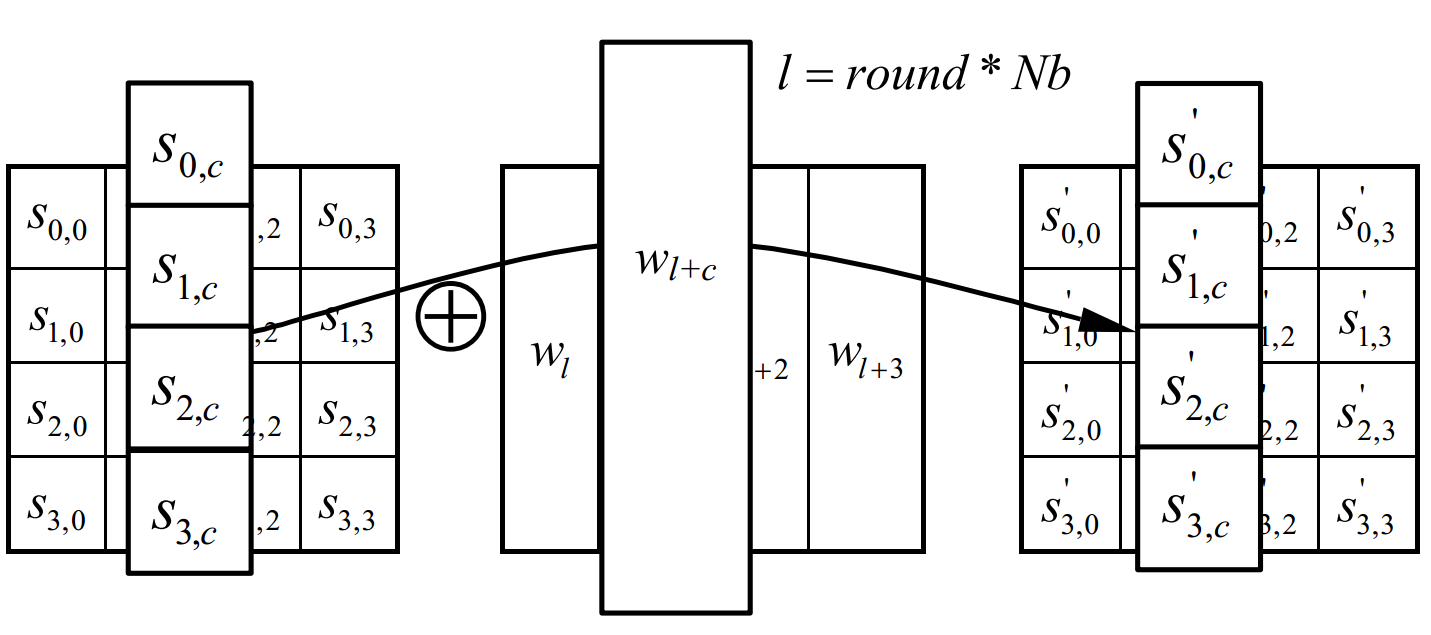
列混合



逐列与常矩阵C相乘

## 轮密钥相加（AddRoundKey）

轮密钥相加是将轮密钥与矩阵state逐位异或。



轮密钥相加

## 解密过程

解密时密钥扩展算法与加密相同,解密轮函数与加密轮函数相反,从最后一轮开始一直循环到第0轮,最后一轮与第0轮解密步骤是直接按步骤执行逆运算就可以,中间的几轮的轮密钥要先执行列混合之后才执行轮密钥相加。