信安数基研究报告

2212794 曹瑜 密码科学与技术

**Topic 1 [离散对数问题]**：（小组选题）

1. 数学问题：

离散对数：如果对于一个整数b,和质数p的一个原根a, 可以找到一个唯一的z指数i, 使得：,其中成立，那么指数i称为b的以a为基数的模p的离散对数；

离散对数难题：当已知一个质数𝑝和它的一个原根𝑎，如果给定一个𝑏，要计算𝑖的值是比较困难；给定群G，设g,h属于G，且有等式 ，那么求解x，就是离散对数问题（Discrete Logarithm Problem）。x称为h 以g 为底的离散对数，用符号表示为 ，或者 。

1. 算法介绍&特点：

Babystep-Giantstep方法：

**算法原理**：Shanks的Babystep-Giantstep方法是解决离散对数问题的一种有效算法；离散对数问题是指在给定元素g和h的情况下，在某个群 G 中找到一个整数，使得gx=h。这个问题在密码学中非常重要，尤其是在公钥加密和数字签名算法中；其核心思想是将可能的指数值分割成较小的区块，从而降低必须直接计算的指数的数量。算法主要分为两个阶段:Babysteps和Giantsteps。

**具体步骤**：

1. 选择整数 m:通常取m=⌈√n⌉，其中n是群G的阶；
2. Babysteps: 对于j=0到m-1，计算gj，并将结果与j存储在一个表中；
3. 计算g（-m）: 这是巨步的基础
4. Giantsteps: 对于i=0到m-1，计算h·g−im 并检查这个结果是否出现 在Babysteps的表中。如果找到匹配，假设对应的Babystep是gj，则 h=gim+j从而解为x=im +j；

**特点**：

相比简单枚举，大大减少了搜索时间，算法直观易理解，易于实现，在非常大的群中，即使是 O(√n)的复杂度也可能导致不实际的计算时间和资源需求，但需要大量内存存储Babysteps的结果

Pollard ρ算法：

**算法原理**：Pollard ρ算法利用了随机漫步和伪随机数生成的思想来寻找循环，从而实现对数值问题的求解。其核心概念是通过构造一个伪随机序列来寻找重复值，进而确定循环长度，并最终找到目标值。

**具体步骤**：

1. 选择初始值x0和伪随机函数f(x)；
2. 生成序列xn+1=f(xn)并计算序列的gcd值;
3. 使用Floyd判环算法（乌龟和兔子算法）检测序列中的循环;
4. 一旦检测到循环，计算两个数值的最大公约数，即可得到因子。

**特点**：

内存占用低，在许多实际应用中展现出高效性，但本质上是概率算法，不能保证在所有情况下都能迅速找到解，算法的性能在一定程度上依赖于伪随机函数的选择

Pohlig-Hellman 算法：

**算法原理**：Pohlig-Hellman 算法利用中国剩余定理和模数的素因子分解，将离散对数问题gx≡(mod p) 转化为多个小规模的离散对数问题。假设p-1的素因子分解为p-1=q1e1q2e2……qkek，则原问题可以分解为模qiei的子问题。

**具体步骤:**

1. 将模数字p-1分解为素因子乘积p-1=q1e1q2e2……qkek;
2. 对于每个素因子qi和对应的指数ei,求解模qkek的离散对数问题;
3. 利用中国剩余定理，将各个子问题的解合并，得到原问题的解;

**特点:**

适用于模数具有较小素因子的情况，能够显著降低计算复杂度;可扩展性：算法可以扩展到处理更大规模的离散对数问题;但算法性能依赖于模数p-1的素因子分解，如果p-1含有大素因子，算法效率会下降；但适用范围有限，主要适用于模数p为大质数的情况。

指数计算法：

**算法原理：**指数计算法(Index Calculus Method)是一种解决离散对数问题的算法，特别适用于大素数模数情况下的离散对数计算。该方法利用数论中的一些技巧，将离散对数问题转化为求解线性方程组，从而大大加快计算速度。

**具体步骤:**

1. 选择 B={p1,p2,…,pk}，这些素数通常选取较小的素数；
2. 随机选择整数a，计算ga(mod p),然后检查 ga(mod p) 是否能分解为因子基中的素数的乘积。如果可以，记录这个关系;
3. 将所有能分解的关系写成线性方程，形成矩阵 A 和向量 b，即 Ax=b.利用线性代数的方法解出x，即 logg pj；
4. 对于给定的 h，找到 a 使得 ga \*h−1能分解为因子基中的素数的乘积，通过前面的对数值计算出x。

**具体步骤:**

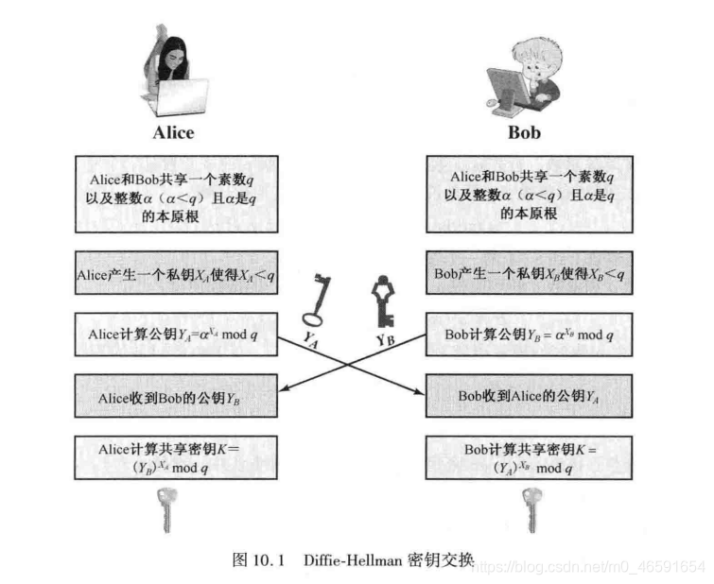
内存占用低，在许多实际应用中展现出高效性，但本质上是概率算法，不能保证在所有情况下都能迅速找到解，算法的性能在一定程度上依赖于伪随机函数的选择；

1. 在密码学中的应用：

Diffie-Hellman密钥交换协议：

Diffie-Hellman算法由Whitfield Diffie 和 Martin Hellman 提出，该算法的安全性也是基于一般有限域上的离散对数问题的难解性。该算法的目的是使两个用户能安全地交换密钥，以便在后续的通信中用该密钥对消息加密。该算法本身只限于进行密钥交换。

**通俗解释:**



**算法描述：**

Diffie-Hellman算法主要用于两个用户密钥的安全交换的场景下，利用一般有限域上的离散对数难解问题，使得敌手难以通过假定的已知信息推算得出密钥。

Diffie-Hellman算法描述如下：

首先选取素数和其本原根为两个公开的整数，用户Alice、Bob再分别选取一个随机整数并各自计算。在此过程中，用户只需保证各自所选取的X为私有的，计算所得出的Y则是可以公开访问的。在计算完毕各自的Y后，两用户通过信道交换得到对方的，再根据各自私有的随机整数X计算K：

两用户计算所得出的密钥K将是相等的:

**应用场景：**

Diffie-Hellman密钥交换协议广泛应用于网络通信和数据加密等领域。例如，HTTPS协议中就使用了这种算法来建立安全连接。通过使用Diffie-Hellman密钥交换协议，两个通信方可以在不直接交换密钥的情况下，建立起一个共同的密钥，从而确保通信内容的安全性。Diffie-Hellman算法也被广泛应用于VPN、安全通信等场景。

**Topic 2 [RSA问题]：**

1. 数学问题：

大整数因子分解问题：给定一个大合数n，找到它的两个质因子p和q是一个极其困难的数学问题；在RSA算法中，n是通过选择两个大质数p和q并计算它们的乘积得到的，即n = p \* q。这个n值将作为公钥和私钥的共同部分。由于大整数的因子分解困难，即使知道了n的值，也无法轻易得到p和q的值，从而保证了私钥的安全性；

求模的离散对数问题：

已知一个数a、模数n和余数b，求解指数x，使得ax mod n = b。在大整数的情况下，这个问题是非常困难的，因为没有有效的算法可以在多项式时间内解决。

在RSA算法中，这个问题与私钥的生成密切相关，具体来说，需要找到一个整数e（公钥的加密指数），使得1 < e < φ(n)且e与φ(n)互质（其中φ(n)是欧拉函数，表示小于n且与n互质的正整数的个数）。然后，需要计算e对于φ(n)的乘法逆元d（私钥的解密指数），满足ed ≡ 1 (mod φ(n))。这个d的求解过程就涉及到了求模的离散对数问题

1. 算法介绍&特点：

**算法介绍**：

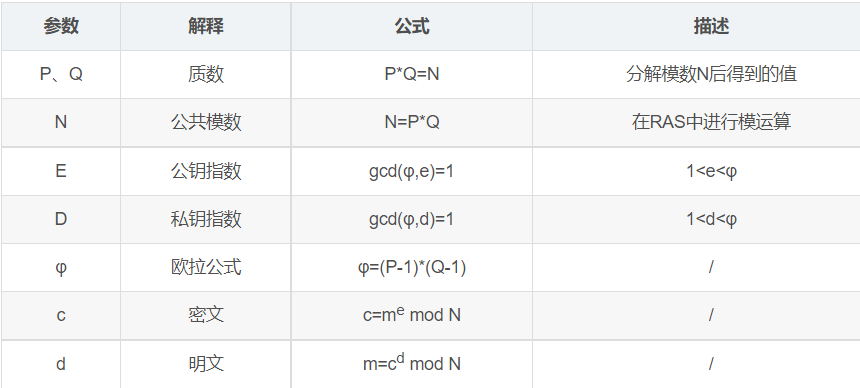
RSA算法由罗纳德·李维斯特（Ron Rivest）、阿迪·萨莫尔（Adi Shamir）和伦纳德·阿德曼（Leonard Adleman）在1977年提出。该算法使用一对密钥，即公钥和私钥，来进行加密和解密操作。公钥用于加密数据，而私钥用于解密数据。RSA算法的安全性基于以下两个事实：

大数分解的困难性：给定一个大合数n，找到它的两个质因子p和q是一个极其困难的数学问题。RSA算法中，n是通过选择两个大质数p和q并计算它们的乘积得到的。

模幂运算和模逆元的存在性：RSA算法利用模幂运算和模逆元的概念来进行加密和解密操作。

**算法步骤：**

参数表：



加密算法：

c ≡ m e mod N

解密算法：

M ≡ c d mod N

密钥生成：

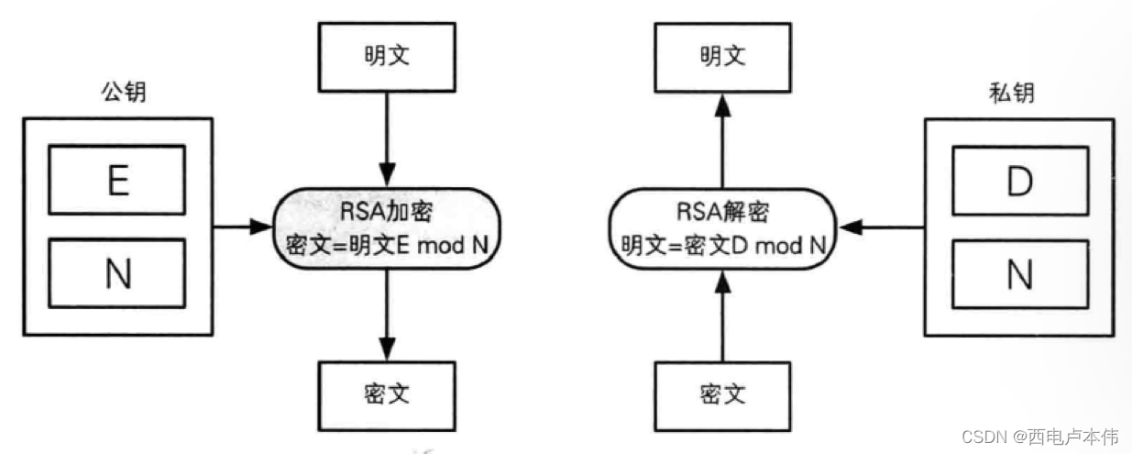
（1）选择两个大质数p和q，计算n = p \* q。

（2）计算欧拉函数φ(n) = (p - 1) \* (q - 1)。

（3）选择一个整数e（公钥的加密指数），满足1 < e < φ(n)且e与φ(n)互质。

（4）计算d（私钥的解密指数），满足ed ≡ 1 (mod φ(n))。

（5）公钥为(e, n)，私钥为(d, n)。



**算法特点**：

RSA是一种非对称加密，使用一对密钥，且基于大数分解问题的困难性，不仅可以用于加密和解密数据，还可以用于数字签名，允许使用不同长度的密钥，与其他公钥加密算法相比，RSA算法的实现相对容易理解，但在处理大数据时效率较低

3、在密码学中的应用：

（1）数据加密：

发送方可以使用接收方的公钥对数据进行加密，确保数据在传输过程中的机密性。只有拥有相应私钥的接收方才能解密数据，从而确保数据的安全性。这种加密方式在网络通信、云计算、移动设备等场景中得到了广泛应用，例如HTTPS协议就使用了RSA算法进行密钥交换和数据加密。

（2）数字签名：

RSA算法也可以用于生成数字签名，用于验证数据的完整性和认证发送方的身份。发送方使用自己的私钥对数据进行签名，接收方使用发送方的公钥验证签名的有效性。

（3）身份认证：

RSA算法还可以用于身份认证。例如，在网银等场景中，用户可以使用RSA算法生成一对公私钥，将公钥发送给银行。银行使用公钥对数据进行加密，只有用户拥有私钥才能解密，从而实现身份认证。

（4）密钥交换：

在网络通信中，RSA算法还可以用于密钥交换。通信双方可以首先使用RSA算法交换对称加密算法的密钥，然后使用对称加密算法进行后续的数据传输。这种方式结合了RSA算法的安全性和对称加密算法的高效性，使得网络通信更加安全可靠。

（5）数字证书：

数字证书是一种由权威机构颁发的电子文件，用于证明某个公钥与某个实体（如个人、组织或设备等）之间的绑定关系。数字证书中通常包含公钥、实体信息、证书颁发机构信息等内容，并使用RSA算法进行签名以保证其真实性和完整性。通过验证数字证书的有效性，可以确定通信对方的身份和公钥的可靠性。

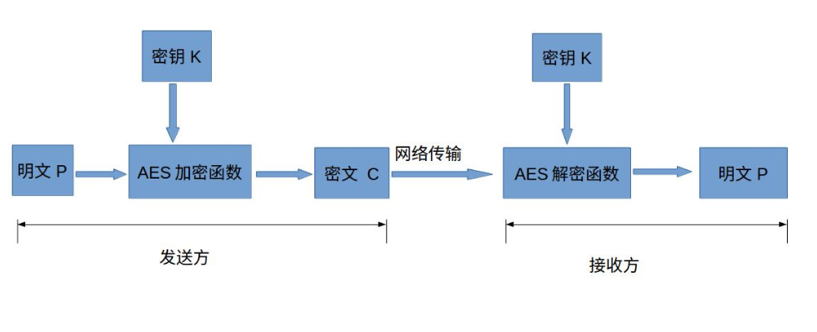
**Topic 3 [AES加密]：**

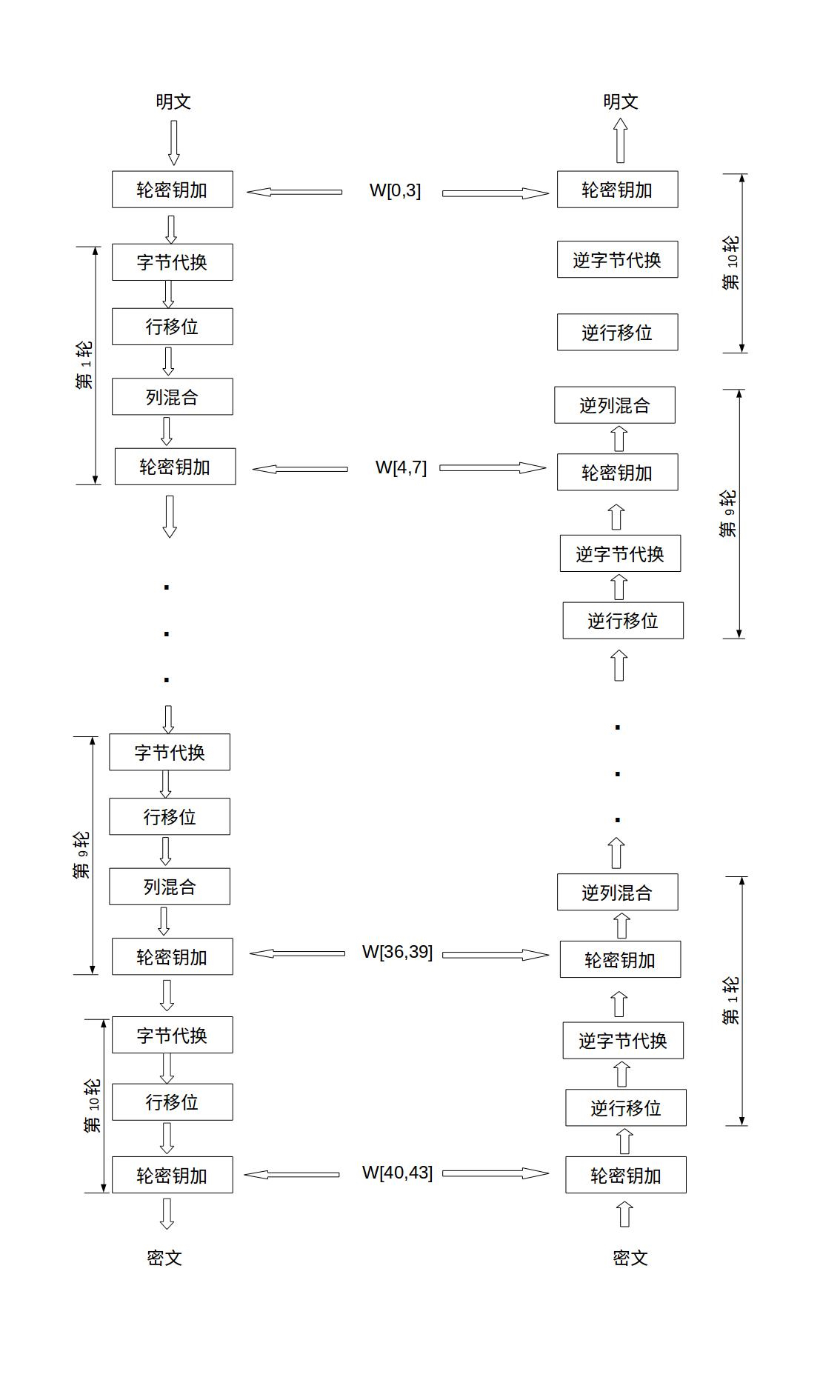
1. 密码原理：

AES 加密算法是一种分组密码体制，其将明文按照固定大小进行分组，然后对每一分组进行加密。在加密过程中采用了多轮加密的方式，每一轮加密都包含了四种操作：SubBytes、ShiftRows、MixColumns 和 AddRoundKey。

AES 加密算法也是一种对称加密算法，加密和解密操作是相反的过程，因此需要使用相同的密钥作为加密和解密的关键参数。AES 算法支持三种密钥长度：128 比特、192 比特和 256 比特，对于不同长度的密钥，AES 算法也会采用不同的轮数进行加密。

以128比特为例，一共进行10轮轮变换，前九轮依次进行SubBytes、ShiftRows、MixColumns 和 AddRoundKey；最后一轮则只进行SubBytes、ShiftRows和 AddRoundKey；

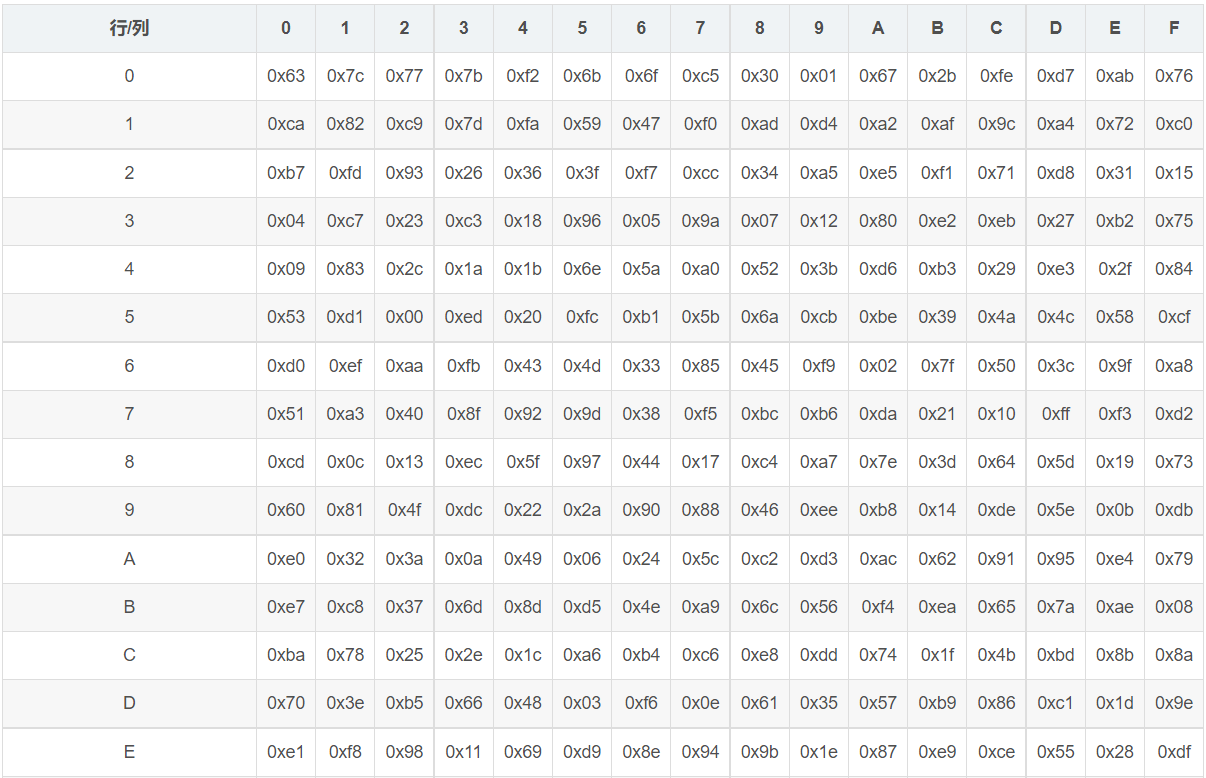




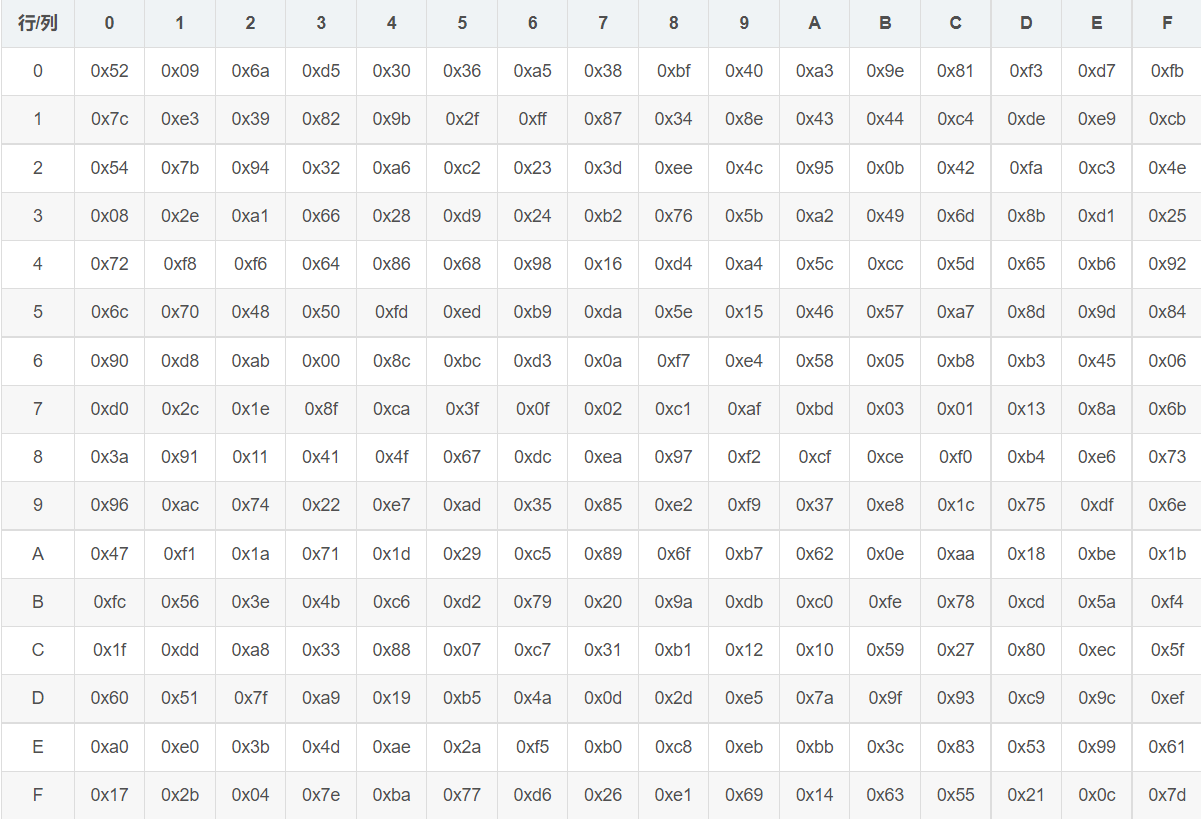
SubBytes：字节代换

根据字节内容进行查表代换

S盒：

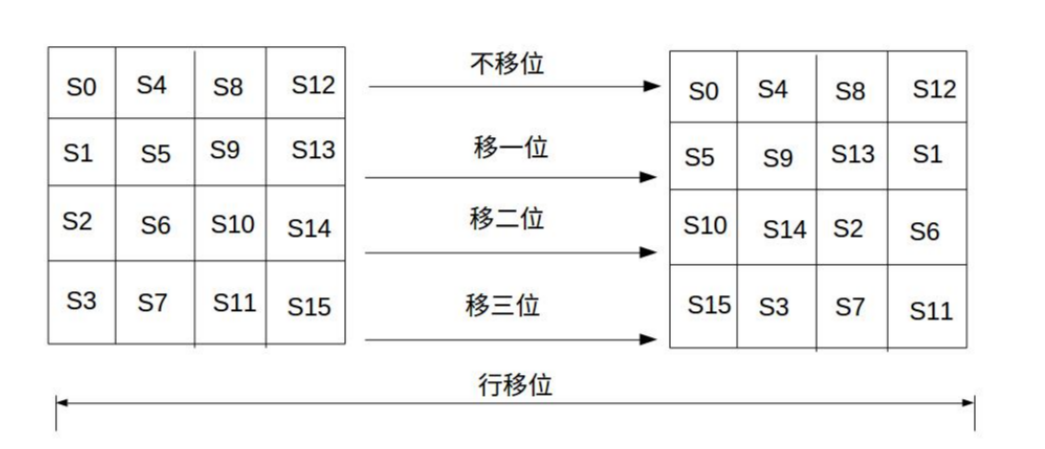


逆S盒：



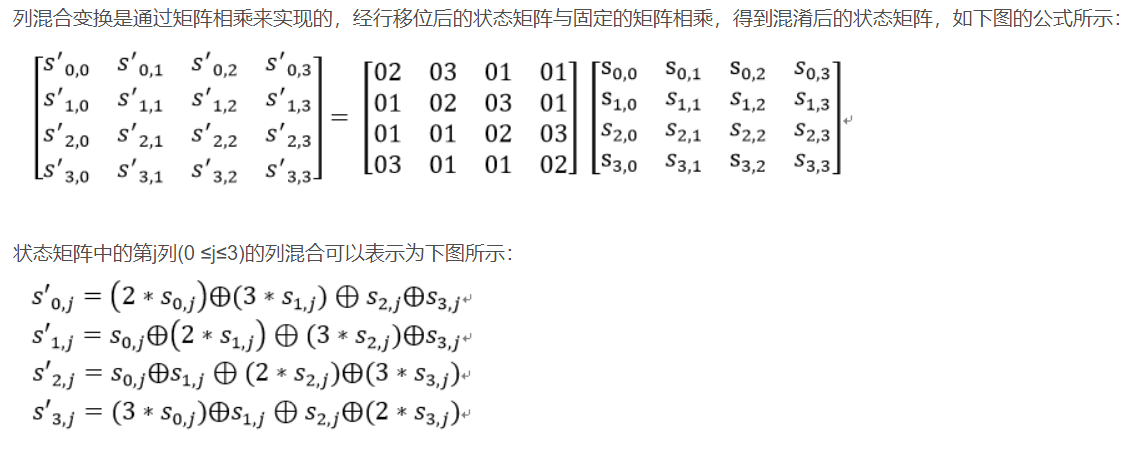
ShiftRows：行移位

将明文分组（一个4x4的字节矩阵）的每一行进行循环移位；第一行保持不变，第二行向左循环移位1个字节，第三行向左循环移位2个字节，第四行向左循环移位3个字节；

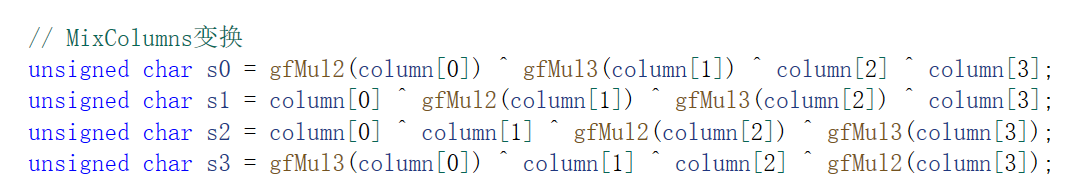


MixColumns：列混淆

列混淆操作使用了一个固定的矩阵（在AES标准中定义）与每一列进行乘法运算。这里的乘法不是常规的十进制乘法，而是基于伽罗瓦域（Galois Field）GF(2^8)上的运算



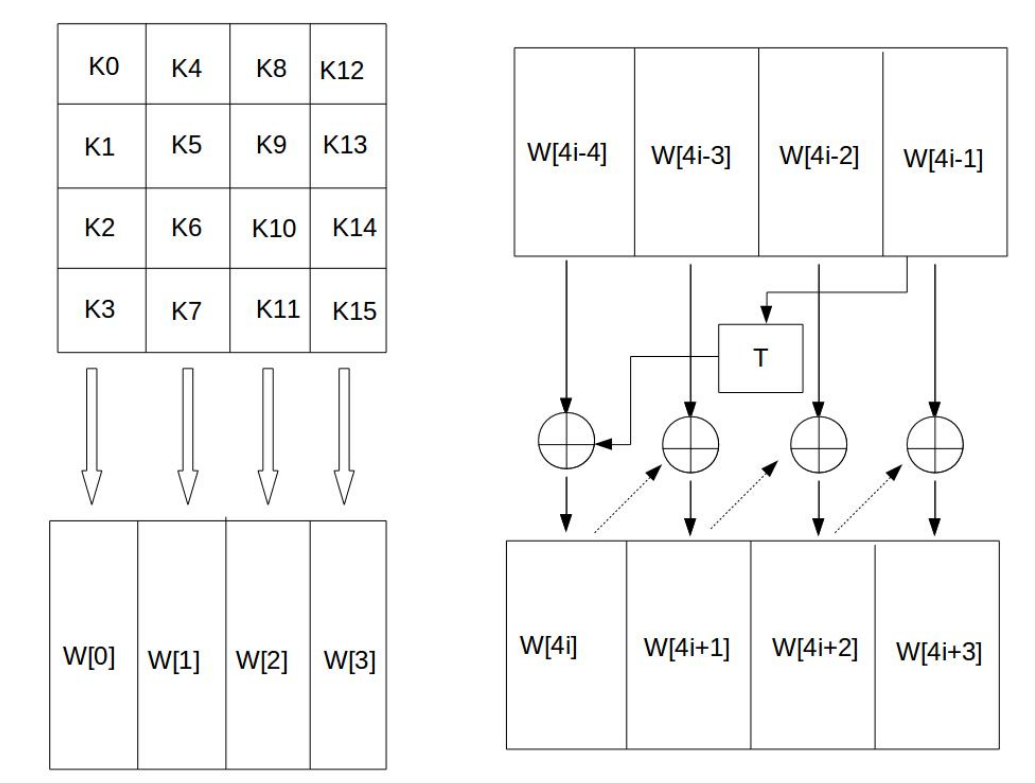
代码实现：



AddRoundKey：轮密钥加

当前分组和对应的该轮扩展密钥进行按位异或，将输入或中间态S的每一列与一个密钥字ki进行按位异或，即将128位轮密钥 Ki 同状态矩阵中的数据进行逐位异或操作。

KeyExtend：轮密钥扩展



（1）扩展密钥数组

扩展过程：为了进行多轮加密，需要对W数组进行扩展，增加40个新的列，形成总共44列的扩展密钥数组。

新列的生成：

如果i（新列的索引）不是4的倍数，则第i列由前一个4的倍数列（W[i-4]）和前一个列（W[i-1]）进行异或（⨁）运算得到：W[i] = W[i-4] ⨁ W[i-1]；

如果i是4的倍数，则第i列由前一个4的倍数列（W[i-4]）和经过函数T处理的前一个列（W[i-1]）进行异或运算得到：W[i] = W[i-4] ⨁ T(W[i-1])；

（2）函数T

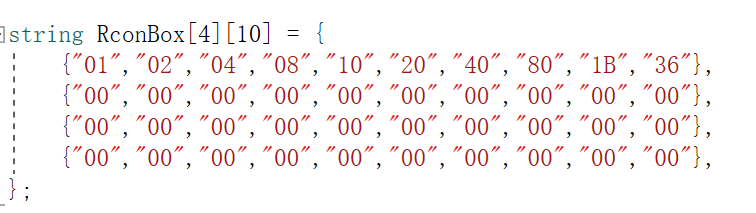
函数T用于生成4的倍数列时的特殊处理，它由以下三部分组成：

字循环（Word Shift）：将输入的32位字中的4个字节循环左移1个字节。即，如果输入字是[b0, b1, b2, b3]，则输出将是[b1, b2, b3, b0]。

字节代换（Byte Substitution）：对字循环的结果使用AES的S盒（Substitution-box）进行字节级的代换。S盒是一个预定义的替换表，用于将每个输入字节映射到一个输出字节。

轮常量异或（Rcon XOR）：将前两步的结果（即字循环和字节代换后的字）与一个轮常量Rcon[j]进行异或运算。这里的j表示当前的轮数（round number）。

轮常量Rcon[j]：



1. 应用场景：

数据保护：AES算法被广泛用于保护敏感数据的机密性。这包括个人身份信息、商业机密、政府数据等。通过使用AES加密，可以确保数据在存储、传输或处理过程中不被未经授权的第三方访问或泄露。

网络通信安全：在网络通信中，AES用于确保数据的完整性和保密性。无论是电子邮件、即时消息、文件传输还是在线交易，AES都可以提供强大的加密保护，防止数据在传输过程中被截获或篡改。

软件安全：AES也常用于软件安全领域，如加密软件、安全通信协议等。通过集成AES加密算法，软件可以提供更高级别的数据保护和身份验证功能。

数字内容保护：对于数字内容（如电影、音乐、电子书等），AES加密算法可以用于防止未经授权的复制和分发。通过加密内容，可以确保只有拥有正确密钥的用户才能访问和使用这些内容。

硬件安全：在硬件领域，AES加密算法也扮演着重要角色。例如，智能卡、安全芯片和加密设备等常常使用AES来保护存储在其中的敏感数据。

1. 包含的数学问题：

有限域（GF(2^8)）

在AES中，使用的是GF(2^8)，即包含256个元素的有限域。这些元素对应于一个字节（8位）的所有可能值（从00到FF的十六进制数）。

**加法与减法**：在GF(2^8)中，加法和减法实际上是相同的操作，因为它们是在二进制（模2）下进行的。加法通过异或（XOR）操作实现，即如果两个对应的二进制位不同，则结果为1，否则为0。

**乘法**：乘法在GF(2^8)中稍微复杂一些，因为它涉及到多项式乘法和模多项式除法。AES使用一个固定的模多项式（称为不可约多项式）来定义乘法操作。

**逆元**：在GF(2^8)中，每个非零元素都有一个乘法逆元，即存在一个元素使得与该元素相乘的结果为1（模多项式的余数为1）。逆元在AES的列混合步骤中用于实现矩阵的逆运算。

有限域的使用确保了AES算法中的运算都是可逆的，并且运算结果始终在有限的范围内。这为算法的加密和解密过程提供了数学基础。

线性代数（Linear Algebra）

线性代数在AES算法中主要体现在列混合步骤中。列混合是一个线性变换过程，它通过一个固定的4x4矩阵与状态数组（一个4x1的列向量）相乘来实现。在列混合步骤中，状态数组的每一列都被视为一个4x1的列向量，并与一个固定的4x4矩阵相乘。这个矩阵乘法操作是在GF(2^8)上进行的，因此涉及到有限域中的乘法运算；线性代数的应用使得AES算法在加密过程中具有更高的复杂性和安全性。通过矩阵乘法和扩散性的实现，AES能够抵御各种密码分析攻击，并保护数据的机密性和完整性。