# **第一部分 密码**

## **环游密码世界**

### 1.1密码学简介

密码学（在[西欧](https://baike.baidu.com/item/%E8%A5%BF%E6%AC%A7" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)语文中，源于[希腊语](https://baike.baidu.com/item/%E5%B8%8C%E8%85%8A%E8%AF%AD" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)kryptós“隐藏的”，和gráphein“书写”）是研究如何隐密地传递信息的学科。在现代特别指对信息以及其传输的数学性[研究](https://baike.baidu.com/item/%E7%A0%94%E7%A9%B6" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)，常被认为是[数学](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)和[计算机科学](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A7%91%E5%AD%A6" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)的分支，和[信息论](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E8%AE%BA" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)也密切相关。著名的密码学者Ron Rivest解释道：“密码学是关于如何在[敌人](https://baike.baidu.com/item/%E6%95%8C%E4%BA%BA" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)存在的[环境](https://baike.baidu.com/item/%E7%8E%AF%E5%A2%83" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)中通讯”，自[工程学](https://baike.baidu.com/item/%E5%B7%A5%E7%A8%8B%E5%AD%A6" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)的角度，这相当于密码学与纯数学的异同。密码学是[信息安全](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E5%AE%89%E5%85%A8" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)等相关议题，如认证、访问控制的核心。密码学的首要目的是隐藏信息的涵义，并不是隐藏信息的存在。密码学也促进了计算机科学，特别是在于电脑与[网络安全](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%AE%89%E5%85%A8" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)所使用的技术，如[访问控制](https://baike.baidu.com/item/%E8%AE%BF%E9%97%AE%E6%8E%A7%E5%88%B6" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)与信息的机密性。密码学已被应用在[日常生活](https://baike.baidu.com/item/%E6%97%A5%E5%B8%B8%E7%94%9F%E6%B4%BB" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)：包括[自动柜员机](https://baike.baidu.com/item/%E8%87%AA%E5%8A%A8%E6%9F%9C%E5%91%98%E6%9C%BA" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)的[芯片卡](https://baike.baidu.com/item/%E8%8A%AF%E7%89%87%E5%8D%A1" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)、电脑使用者存取[密码](https://baike.baidu.com/item/%E5%AF%86%E7%A0%81" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)、[电子商务](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)等等。

### 1.2什么事密码

[密码](https://baike.baidu.com/item/%E5%AF%86%E7%A0%81" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)是通信双方按约定的[法则](https://baike.baidu.com/item/%E6%B3%95%E5%88%99" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)进行信息特殊变换的一种重要[保密](https://baike.baidu.com/item/%E4%BF%9D%E5%AF%86" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)手段。依照这些法则，变明文为密文，称为加密变换；变密文为明文，称为脱密变换。密码在早期仅对文字或数码进行加、脱密变换，随着通信[技术](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)的[发展](https://baike.baidu.com/item/%E5%8F%91%E5%B1%95" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)，对[语音](https://baike.baidu.com/item/%E8%AF%AD%E9%9F%B3" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)、[图像](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)、[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE" \t "https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/_blank)等都可实施加、脱密变换。

## 历史上别人看不懂的文章

### 2.1凯撒密码

### 2.2替换密码

### 2.3 Enigma

（详情及更多加密方法见附录）

## 对称密码

### 3.1简介

对称加密(也叫[私钥](https://baike.baidu.com/item/%E7%A7%81%E9%92%A5" \t "https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95/_blank)加密)指加密和解密使用相同[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5" \t "https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95/_blank)的加密算法。有时又叫传统[密码算法](https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95/_blank)，就是加密密钥能够从解密密钥中推算出来，同时解密密钥也可以从加密密钥中推算出来。而在大多数的[对称算法](https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95/_blank)中，加密密钥和解密密钥是相同的，所以也称这种加密算法为秘密密钥算法或单密钥算法。它要求发送方和接收方在安全通信之前，商定一个密钥。对称算法的安全性依赖于密钥，泄漏密钥就意味着任何人都可以对他们发送或接收的消息解密，所以密钥的保密性对通信的安全性至关重要。

### 3.2 XOR运算

异或，英文为exclusive OR，或缩写成xor

异或（[xor](https://baike.baidu.com/item/xor" \t "https://baike.baidu.com/item/%E5%BC%82%E6%88%96/_blank)）是一个数学运算符。它应用于逻辑运算。异或的数学符号为“⊕”，计算机符号为“xor”。其运算法则为：

a⊕b = (¬a ∧ b) ∨ (a ∧¬b)

如果a、b两个值不相同，则异或结果为1。如果a、b两个值相同，异或结果为0。

注：如果把0看做偶数，1看做奇数，就可以将XOR和一般的加法运算等同起来。

### 3.3 一次性密码本——绝对不会被破译的密码

一次性密码本（One-time Pad；OTP）是密码学中的一种[加密算法](https://baike.baidu.com/item/%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/%E4%B8%80%E6%AC%A1%E6%80%A7%E5%AF%86%E7%A0%81%E6%9C%AC/_blank)。是以随机的金钥（key）组成[明文](https://baike.baidu.com/item/%E6%98%8E%E6%96%87" \t "https://baike.baidu.com/item/%E4%B8%80%E6%AC%A1%E6%80%A7%E5%AF%86%E7%A0%81%E6%9C%AC/_blank)，且只使用一次。这种方法在1882年被弗兰克·米勒（Frank Miller）发现。

加密方法为：首先手上要有一本一次性密码本用以加密文件，接着将一次性密码本里的字母，与被加密文件的字母给依序按某个事先约定的规定一一相混，其中一个相混的作法是将字母指定数字(如在英语中，将A至Z依序指定为0至25)然后将一次性密码文本上的字母所代表的数字和被加密文件上相对应的数字给相加，再除以该语言的字母数，假设是n(如英语为26)，若就此得出来的某个数字小于零，则将该小于零的数给加上n，如此便完成加密。

安全性：在理论上，此种密码是牢不可破的，而它的安全性已由克劳德艾尔伍德·香农（Claude Elwood Shannon）所证明。虽然它在理论上的安全性无庸置疑，但在实际操作上却有着以下的问题：

用以加密的文本，也就是一次性密码本，必须是无特定规律的。它可以是一串随机数字，一句话，或者一本英文名著。

它必须至少比被加密的文件等长。

用以加密的文本（密码本）只能用一次，且必须对非关系人小心保密，不再使用时，用以加密的文本应当要销毁，以防重复使用。

这样就出现了一个问题要保证一个密文在保存和传输过程中的安全，就必须保证一个等长的随机密钥序列的安全，这显然非常不实用。

### 3.4 DES

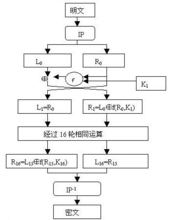
**DES算法入口参数**

DES算法的入口参数有三个：Key、Data、Mode。其中Key为7个字节共56位，是DES算法的工作密钥；Data为8个字节64位，是要被加密或被解密的数据；Mode为DES的工作方式,有两种:加密或解密。

**DES基本原则**

DES设计中使用了分组密码设计的两个原则：混淆（confusion）和扩散(diffusion)，其目的是抗击敌手对密码系统的统计分析。混淆是使密文的统计特性与密钥的取值之间的关系尽可能复杂化，以使密钥和明文以及密文之间的依赖性对密码分析者来说是无法利用的。扩散的作用就是将每一位明文的影响尽可能迅速地作用到较多的输出密文位中，以便在大量的密文中消除明文的统计结构，并且使每一位密钥的影响尽可能迅速地扩展到较多的密文位中，以防对密钥进行逐段破译。

**算法步骤**

[](https://baike.baidu.com/pic/DES/210508/0/9f510fb30f2442a754ef9a91d843ad4bd1130224?fr=lemma&ct=single)DES算法把64位的明文输入块变为64位的密文输出块,它所使用的密钥也是64位（实际用到了56位，第8、16、24、32、40、48、56、64位是校验位， 使得每个密钥都有奇数个1），其算法主要分为两步：

1）初始置换

其功能是把输入的64位数据块按位重新组合，并把输出分为L0、R0两部分，每部分各长32位，其置换规则为将输入的第58位换到第一位,第50位换到第2位……依此类推,最后一位是原来的第7位。L0、R0则是换位输出后的两部分，L0是输出的左32位，R0是右32位,例:设置换前的输入值为D1D2D3……D64,则经过初始置换后的结果为:L0=D58D50……D8;R0=D57D49……D7。

其置换规则见下表：

58,50,42,34,26,18,10,2,

60,52,44,36,28,20,12,4,

62,54,46,38,30,22,14,6,

64,56,48,40,32,24,16,8,

57,49,41,33,25,17,9,1,

59,51,43,35,27,19,11,3,

61,53,45,37,29,21,13,5,

63,55,47,39,31,23,15,7,

2）逆置换

经过16次迭代运算后,得到L16、R16,将此作为输入，进行逆置换，逆置换正好是初始置换的逆运算，由此即得到密文输出。

此算法是[对称加密算法](https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/DES/_blank)体系中的代表,在计算机网络系统中广泛使用。

**DES与3DES**

3DES（即Triple DES）是DES向AES过渡的[加密算法](https://baike.baidu.com/item/%E5%8A%A0%E5%AF%86%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/DES/_blank)，它使用3条56位的密钥对数据进行三次加密。是DES的一个更安全的变形。它以DES为基本模块，通过组合分组方法设计出分组加密算法。比起最初的DES，3DES更为安全。

该方法使用两个密钥，执行三次DES算法，加密的过程是加密-解密-加密，解密的过程是解密-加密-解密。

3DES加密过程为：C=Ek3(Dk2(Ek1(P)))

3DES解密过程为：P=Dk1(EK2(Dk3(C)))

采用两个密钥进行三重加密的好处有：

①两个密钥合起来有效密钥长度有112bit，可以满足商业应用的需要，若采用总长为168bit的三个密钥，会产生不必要的开销。

②加密时采用加密-解密-加密，而不是加密-加密-加密的形式，这样有效的实现了与现有DES系统的向后兼容问题。因为当K1=K2时，三重DES的效果就和原来的DES一样，有助于逐渐推广三重DES。

③三重DES具有足够的安全性，目前还没有关于攻破3DES的报道。

### 3.5 AES——Rijindael

**高级加密标准**（英语：**Advanced Encryption Standard**，缩写：**AES**），在[密码学](https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6" \t "https://baike.baidu.com/item/aes/_blank)中又称**Rijndael加密法**，是[美国联邦政府](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD%E8%81%94%E9%82%A6%E6%94%BF%E5%BA%9C" \t "https://baike.baidu.com/item/aes/_blank)采用的一种区块加密标准。

**密码说明：**

严格地说，AES和Rijndael加密法并不完全一样（虽然在实际应用中二者可以互换），因为Rijndael加密法可以支持更大范围的[区块](https://baike.baidu.com/item/%E5%8C%BA%E5%9D%97" \t "https://baike.baidu.com/item/aes/_blank)和密钥长度：AES的区块长度固定为128 [比特](https://baike.baidu.com/item/%E6%AF%94%E7%89%B9" \t "https://baike.baidu.com/item/aes/_blank)，密钥长度则可以是128，192或256比特；而Rijndael使用的密钥和区块长度可以是32位的整数倍，以128位为下限，256比特为上限。加密过程中使用的密钥是由Rijndael密钥生成方案产生。

大多数AES计算是在一个特别的[有限域](https://baike.baidu.com/item/%E6%9C%89%E9%99%90%E5%9F%9F" \t "https://baike.baidu.com/item/aes/_blank)完成的。

AES加密过程是在一个4×4的[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82" \t "https://baike.baidu.com/item/aes/_blank)矩阵上运作，这个矩阵又称为“状态（state）”，其初值就是一个明文区块（矩阵中一个元素大小就是明文区块中的一个Byte）。（Rijndael加密法因支持更大的区块，其矩阵行数可视情况增加）加密时，各轮AES加密循环（除最后一轮外）均包含4个步骤：

AddRoundKey — 矩阵中的每一个字节都与该次轮秘钥（round key）做XOR运算；每个子密钥由密钥生成方案产生。

SubBytes — 通过非线性的替换函数，用[查找表](https://baike.baidu.com/item/%E6%9F%A5%E6%89%BE%E8%A1%A8" \t "https://baike.baidu.com/item/aes/_blank)的方式把每个字节替换成对应的字节。

ShiftRows — 将矩阵中的每个横列进行循环式移位。

MixColumns — 为了充分混合矩阵中各个直行的操作。这个步骤使用线性转换来混合每列的四个字节。

最后一个加密循环中省略MixColumns步骤，而以另一个AddRoundKey取代。

## 分组密码

### 4.1分组密码与流密码

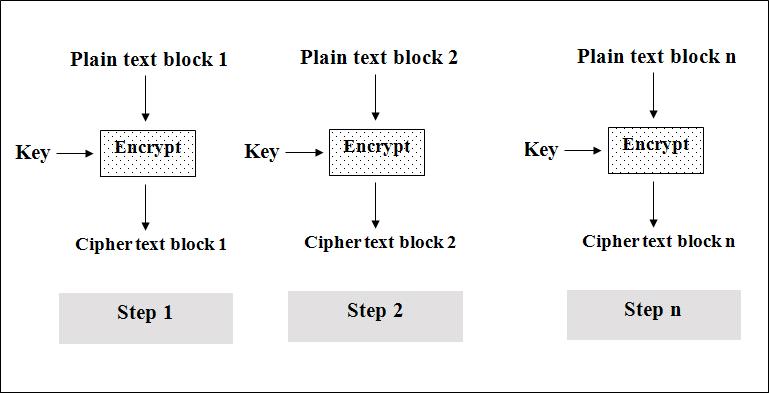
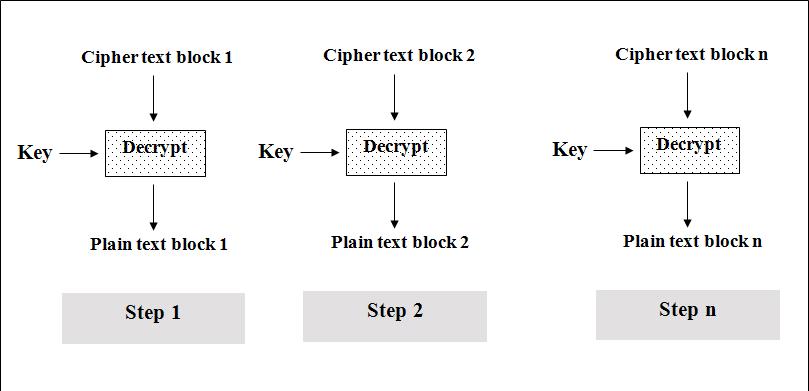
**分组密码**：[分组](https://baike.baidu.com/item/%E5%88%86%E7%BB%84" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)[密码](https://baike.baidu.com/item/%E5%AF%86%E7%A0%81/65553" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)是将[明文](https://baike.baidu.com/item/%E6%98%8E%E6%96%87" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)[消息](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF/1619218" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)[编码](https://baike.baidu.com/item/%E7%BC%96%E7%A0%81/80092" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)[表示](https://baike.baidu.com/item/%E8%A1%A8%E7%A4%BA" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)后的[数字](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)（简称[明文](https://baike.baidu.com/item/%E6%98%8E%E6%96%87" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)数字）[序列](https://baike.baidu.com/item/%E5%BA%8F%E5%88%97/1302588" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)，划分成[长度](https://baike.baidu.com/item/%E9%95%BF%E5%BA%A6" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)为[n](https://baike.baidu.com/item/n/1915" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)的组（可看成长度为n的[矢量](https://baike.baidu.com/item/%E7%9F%A2%E9%87%8F/1400417" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)），每组分别在[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5" \t "https://baike.baidu.com/item/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81/_blank)的控制下变换成等长的输出数字（简称密文数字）序列。

**流密码**：利用密钥产生一个密钥流Z=Z1Z2Z3…，然后利用此密钥流依次对明文X=X0X1X2...进行加密，这样产生的密码就是序列密码，也称流密码。密钥流由密钥流发生器f产生：zi=f(k,si)，这里的si是加密器中存储器（记忆元件）在i时刻的状态，k是密钥。序列密码方案的发展是模仿“一次一密”系统的尝试。

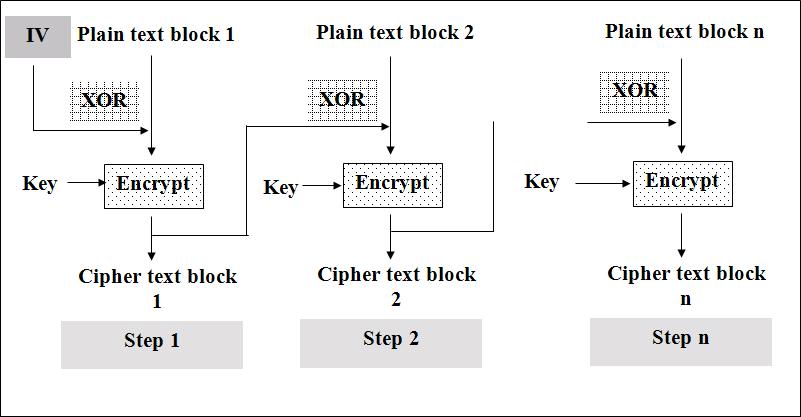
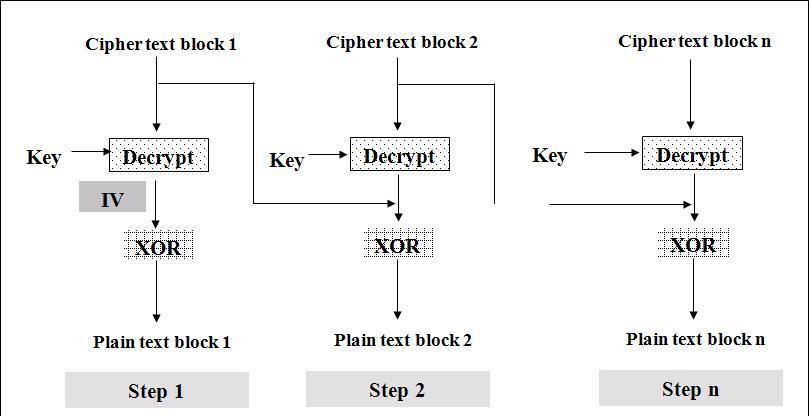
### 4.2分组密码的模式

在分组加密算法中，有ECB,CBC,CFB,OFB这几种算法模式。

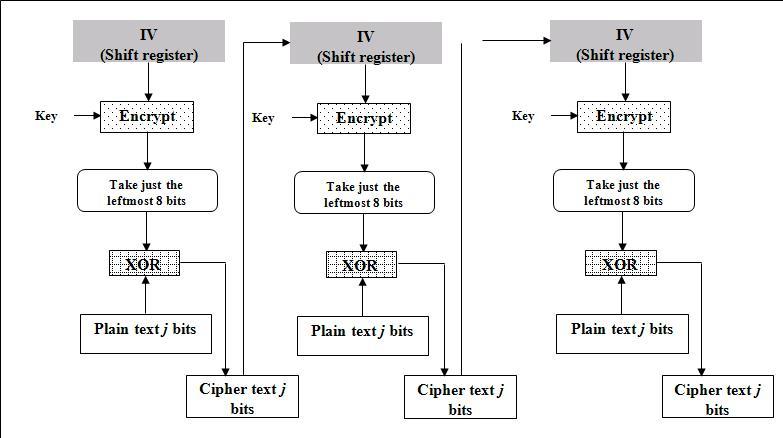
#### 1)ECB(Electronic Code Book)/电码本模式

  
  
1.简单，有利于并行计算，误差不会被传送；  
2.不能隐藏明文的模式；  
repetitions in message may show in cipher text/在密文中出现明文消息的重复   
3.可能对明文进行主动攻击；  
加密消息块相互独立成为被攻击的弱点/weakness due to encrypted message blocks being independent

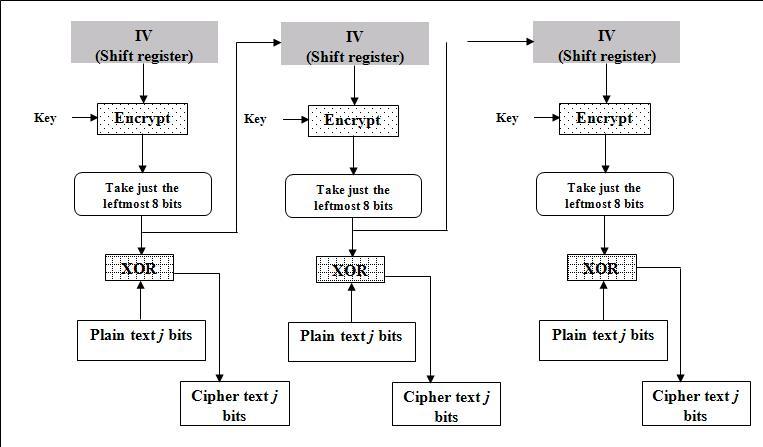
#### 2)CBC(Cipher Block Chaining)

  
  
这个词在分组密码中经常会用到，它是指一个明文分组在被加密之前要与前一个的密文分组进行异或运算。当加密算法用于此模式的时候除密钥外，还需协商一个初始化向量（IV），这个IV没有实际意义，只是在第一次计算的时候需要用到而已。采用这种模式的话安全性会有所提高。  
1. 不容易主动攻击,安全性好于ECB,适合传输长度长的报文,是SSL、IPSec的标准。  
each ciphertext block depends on all message blocks/每个密文块依赖于所有的信息块  
thus a change in the message affects all ciphertext blocks/明文消息中一个改变会影响所有密文块  
2. need Initial Vector (IV) known to sender & receiver/发送方和接收方都需要知道初始化向量   
3.加密过程是串行的，无法被并行化(在解密时，从两个邻接的密文块中即可得到一个平文块。因此，解密过程可以被并行化)；误差传递

#### 3)CFB(Cipher Feedback)

  
密文反馈（CFB，Cipher feedback）模式类似于CBC，可以将块密码变为自同步的流密码；工作过程亦非常相似，CFB的解密过程几乎就是颠倒的CBC的加密过程：  
需要使用一个与块的大小相同的移位寄存器，并用IV将寄存器初始化。然后，将寄存器内容使用块密码加密，然后将结果的最高x位与平文的x进行异或，以产生密文的x位。下一步将生成的x位密文移入寄存器中，并对下面的x位平文重复这一过程。解密过程与加密过程相似，以IV开始，对寄存器加密，将结果的高x与密文异或，产生x位平文，再将密文的下面x位移入寄存器。  
与CBC相似，平文的改变会影响接下来所有的密文，因此加密过程不能并行化；而同样的，与CBC类似，解密过程是可以并行化的。

#### 4)OFB(Output Feedback)

  
输出反馈模式（Output feedback, OFB）可以将块密码变成同步的流密码。它产生密钥流的块，然后将其与平文块进行异或，得到密文。与其它流密码一样，密文中一个位的翻转会使平文中同样位置的位也产生翻转。这种特性使得许多错误校正码，例如奇偶校验位，即使在加密前计算而在加密后进行校验也可以得出正确结果。  
每个使用OFB的输出块与其前面所有的输出块相关，因此不能并行化处理。然而，由于平文和密文只在最终的异或过程中使用，因此可以事先对IV进行加密，最后并行的将平文或密文进行并行的异或处理。  
可以利用输入全0的CBC模式产生OFB模式的密钥流。这种方法十分实用，因为可以利用快速的CBC硬件实现来加速OFB模式的加密过程。

## 公钥密码

### 5.1 密钥配送问题

解决密钥配送问题的方法有以下几种：

通过事先共享密钥来解决

通过密钥配送中心来解决

通过Diffie-Hellman密钥交换来解决

通过公钥密码来解决

### 5.2 Diffie-Hellman密钥交换

**步骤**：

1. Alice向Bob发送两个质数P和G

P必须是一个非常大的质数，而G则是一个和P相关的数，称为**生成元**。G可以是一个较小的数。

**P和G不需要保密**，被窃听者Eve获取也无所谓。

此外，P和G可以有Alice和Bob中的任意一方生成。

1. Alice和Bob各自生成一个随机数A和B。

A和B是1~P-2之间的整数，且由各自**秘密保存。**

1. Alice将这个数发给Bob，Bob将这个数发给Alice。
2. Alice用Bob发过来的书计算A次方并求 mod P

Alice的密钥=

=

Bob用Alice发过来的书计算B次方并求 mod P

Bob的密钥=

=

最终Alice计算的密钥=Bob计算的密钥

**Eve能计算出密钥吗？**

在以上的各个步骤中公开的信息有：P、G、、在知道以上信息的情况下能计算出A吗？如果是的话计算A并不难但是根据计算出A的有效算法到现在还没有出现。

**生成元的意义**

举个栗子：已知P是质数，我们假设以13为栗。计算

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G  A | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 4 | 8 | 3 | 6 | 12 | 11 | 9 | 5 | 10 | 7 | 1 |
| 3 | 3 | 9 | 1 | 3 | 9 | 1 | 3 | 9 | 1 | 3 | 9 | 1 |
| 4 | 4 | 3 | 12 | 9 | 10 | 1 | 4 | 3 | 12 | 9 | 10 | 1 |
| 5 | 5 | 12 | 8 | 1 | 5 | 12 | 8 | 1 | 5 | 12 | 8 | 1 |
| 6 | 6 | 10 | 8 | 9 | 2 | 12 | 7 | 3 | 5 | 4 | 11 | 1 |
| 7 | 7 | 10 | 5 | 9 | 11 | 12 | 6 | 3 | 8 | 4 | 2 | 1 |
| 8 | 8 | 12 | 5 | 1 | 8 | 12 | 5 | 1 | 8 | 12 | 5 | 1 |
| 9 | 9 | 3 | 1 | 9 | 3 | 1 | 9 | 3 | 1 | 9 | 3 | 1 |
| 10 | 10 | 9 | 12 | 3 | 4 | 1 | 10 | 9 | 12 | 3 | 4 | 1 |
| 11 | 11 | 4 | 5 | 3 | 7 | 12 | 2 | 9 | 8 | 10 | 6 | 1 |
| 12 | 12 | 1 | 12 | 1 | 12 | 1 | 12 | 1 | 12 | 1 | 12 | 1 |

（2为生成元）

（6为生成元）

（7为生成元）

（11为生成元）

我们发现2的乘方对应的值（共12个）都是不一样的。也就是说2的乘方的结果出现了1到12的全部整数。具有这种性质的数称为13的**生成元**。

### 5.3 公钥密码——RSA

目的：为解决密钥配送问题

Mod运算（加法，减法，乘法，除法，乘方，对数）求倒数尤为关键

注：判断一个数在 modx 下是否存在倒数<=>这个数与x的最大公约数是否为1（即互质）

RSA：明文，密钥，密文都是数字

加密：密文=明文E mod N () 公钥（E,N）  
 解密：明文=密文D mod N () 私钥（D,N）

密钥对的生成：

（1）求N。 N=p×q p,q为很大的质数（利用仿随机数生成器生成再利用费马测试或米勒·拉宾测试其是否为质数）

（2）求L。 L=lcm(p-1,q-1) p-1和q-1的最小公倍数。

（3）求E。 1<E<L 且 gcd(E,L)=1 最大公约数（辗转相除法）

（4）求D。 1<D<L 且 E×D mod L=1

### 5.4 对RSA的攻击

此处省略

## 混合密码系统

在大多数实际的实现中，公开密钥密码用来保护和分发会话密钥(session key)。这些会话密钥用在对称算法中，对通信消息进行保密。有时称这种系统为混合密码系统(hybrid cryptosystem)。

1. Bob将他的公开密钥发给Alice。
2. Alice产生随机会话密钥K，用Bob的公开密钥加密，并把加密的密钥EB(K)送给Bob。
3. Bob用他的私人密钥解密Alice的消息，恢复出会话密钥：DB(EB(K))=K。
4. 他们两人用同一个会话密钥对他们的通信消息进行加密。

# 第二部分 认证

## 单向散列函数——消息的指纹

**目的**：保持信息的完整性，也称一致性，及可以检查信息在传输的过程中有没有被篡改。

**单向[散列函数](https://baike.baidu.com/item/%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)**，又称单向[Hash函数](https://baike.baidu.com/item/Hash%E5%87%BD%E6%95%B0" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)、杂凑函数，就是把任意长的输入消息串变化成固定长的输出串且由输出串难以得到输入串的一种函数。这个输出串称为该消息的散列值。一般用于产生[消息摘要](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E6%91%98%E8%A6%81" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)，[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)加密等.

**常见单向散列函数(Hash函数)**

[MD5](https://baike.baidu.com/item/MD5" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)（Message Digest Algorithm 5）：是RSA[数据安全](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AE%89%E5%85%A8" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)公司开发的一种[单向散列算法](https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)，MD5被广泛使用，可以用来把不同长度的[数据块](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%9D%97" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)进行暗码运算成一个128位的数值。

[SHA](https://baike.baidu.com/item/SHA" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)（Secure Hash Algorithm）这是一种较新的散列算法，可以对任意长度的数据运算生成一个160位的数值。

[MAC](https://baike.baidu.com/item/MAC/329741" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)（Message Authentication Code）：[消息认证](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E8%AE%A4%E8%AF%81" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)代码，是一种使用[密钥](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)的单向函数，可以用它们在系统上或用户之间认证文件或消息。HMAC（用于消息认证的密钥[散列法](https://baike.baidu.com/item/%E6%95%A3%E5%88%97%E6%B3%95" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)）就是这种函数的一个例子。

CRC（Cyclic Redundancy Check）：[循环冗余校验码](https://baike.baidu.com/item/%E5%BE%AA%E7%8E%AF%E5%86%97%E4%BD%99%E6%A0%A1%E9%AA%8C%E7%A0%81" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)，CRC校验由于实现简单，检错能力强，被广泛使用在各种[数据校验](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%A0%A1%E9%AA%8C" \t "https://baike.baidu.com/item/%E5%8D%95%E5%90%91%E6%95%A3%E5%88%97%E5%87%BD%E6%95%B0/_blank)应用中。占用系统资源少，用软硬件均能实现，是进行数据传输差错检测地一种很好的手段（CRC 并不是严格意义上的散列算法，但它的作用与散列算法大致相同，所以归于此类）。

**函数性质**

Hash[1]  是作用于一任意长度的消息M，返回一固定长度的散列值h:h=H(m)。其中h的长度为m。Hash函数主要用于封装或者数字签名的过程当中，它必须具有如下几个性质:

1.给定h，根据H(M)=h计算M在计算上是不可行的;

2.给定M，要找到另一消息M’。并满足H(m)=H(m’)在计算上是不可行的。

上述特性中的任何弱点都有可能破坏使用Hash函数进行封装或者签名的各种协议的安全性，如生日攻击。Hash函数的重要之处就是赋予M唯一的“指纹”。如果用户A用数字签名算法H(m)进行签名，而B能产生满足H(m)=H(m’)的另一消息M’，那么B就可以声称A对M进行了签名。

Hash函数除了需要上述性质外还需要的性质有：

3.给定M，很容易计算h;

4.抗碰撞性。即随机找到两个消息M和M’，使H(m)=H(m’)在计算上不可行。

## 消息认证——传送的消息准确吗

目的:保持信息的完整性，检验上一章的指纹是否正确

**消息认证（message authentication）**就是验证消息的完整性，当接收方收到发送方的报文时，接收方能够验证收到的报文是真实的和未被篡改的。它包含两层含义：一是验证信息的发送者是真正的而不是冒充的，即数据起源认证；二是验证信息在传送过程中未被篡改、重放或延迟等。

**消息内容认证常用的方法**：消息发送者在消息中加入一个鉴别码（MAC、MDC等）并经加密后发送给接受者（有时只需加密鉴别码即可）。接受者利用约定的算法对解密后的消息进行鉴别运算，将得到的鉴别码与收到的鉴别码进行比较，若二者相等，则接收，否则拒绝接收。

## 数字签名——谁写的消息

消息认证使用的是一对对称密钥，可以计算MAC值的不只有信息的发送者，所以无法防止否认，而数字签名则利用公钥密码，签名的只有发送者本人，这样就防止了事后否认。

数字签名（又称[公钥](https://baike.baidu.com/item/%E5%85%AC%E9%92%A5" \t "https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D/_blank)数字签名、[电子签章](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E7%AD%BE%E7%AB%A0" \t "https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D/_blank)）是一种类似写在纸上的普通的物理签名，但是使用了公钥加密领域的技术实现，用于鉴别数字信息的方法。一套数字签名通常定义两种互补的运算，一个用于签名，另一个用于验证。

数字签名，就是只有信息的发送者才能产生的别人无法伪造的一段数字串，这段数字串同时也是对信息的发送者发送信息真实性的一个有效证明。

**签名过程：**“发送报文时，发送方用一个哈希函数从报文文本中生成[报文摘要](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87%E6%91%98%E8%A6%81" \t "https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D/_blank),然后用自己的私人密钥对这个摘要进行加密，这个加密后的摘要将作为报文的数字签名和报文一起发送给接收方，接收方首先用与发送方一样的哈希函数从接收到的原始报文中计算出报文摘要，接着再用发送方的公用密钥来对报文附加的数字签名进行解密，如果这两个摘要相同、那么接收方就能确认该数字签名是发送方的。

## 证书——给公钥加上签名

目的：为了判断自己手里的公钥是否合法（即是否属于自己或者通讯方）

**公钥证书**，通常简称为证书，是一种[数字签名](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E7%AD%BE%E5%90%8D" \t "https://baike.baidu.com/item/%E5%85%AC%E9%92%A5%E8%AF%81%E4%B9%A6/_blank)的声明，它将[公钥](https://baike.baidu.com/item/%E5%85%AC%E9%92%A5/6447788" \t "https://baike.baidu.com/item/%E5%85%AC%E9%92%A5%E8%AF%81%E4%B9%A6/_blank)的值绑定到持有对应[私钥](https://baike.baidu.com/item/%E7%A7%81%E9%92%A5" \t "https://baike.baidu.com/item/%E5%85%AC%E9%92%A5%E8%AF%81%E4%B9%A6/_blank)的个人、设备或服务的身份。

# 第三部分 附录

**在线工具**

**在线计算器http://www.99cankao.com/**

**在线转换破解**

**<http://www.bejson.com/>**

**<http://www.mxcz.net/tools/Hex.aspx>**

**<http://www.atool.org/>**

**<http://www.qqxiuzi.cn/bianma/wenbenjiami.php>**

**<http://www.mokuge.com/>**

**<http://emn178.github.io/online-tools/base32_decode.html>**

**常用加密和编码**

**<http://www.evil0x.com/posts/27318.html>**