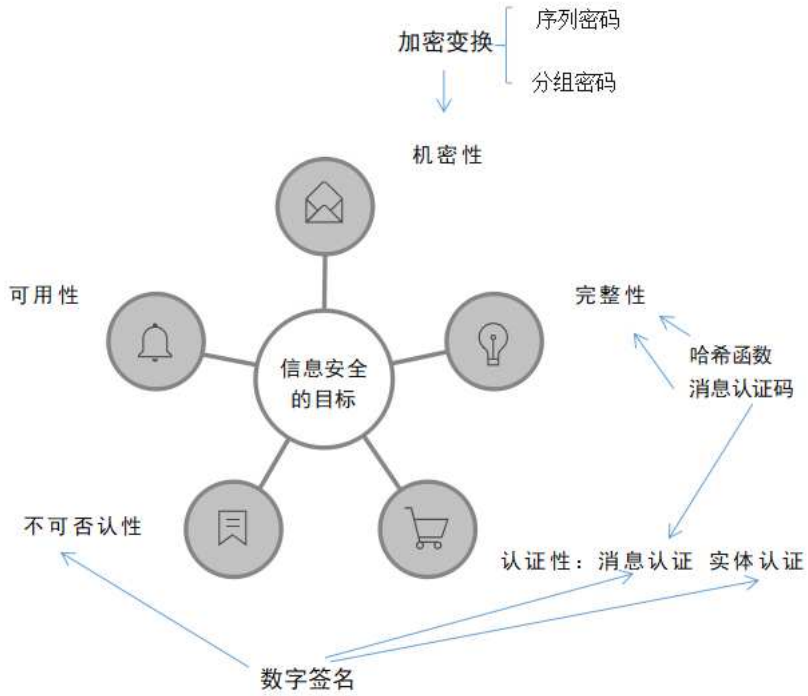
《现代密码学》

绪论

信息安全与密码学

- 经典的信息安全三要素（CIA）——机密性、完整性和认证性是信息安全的核心原则。
- 以密码学为基础的信息安全的五个方面：信息及信息系统的机密性、完整性、可用性、认证性和不可否认性。机密性可通过加密变换实现访问控制；完整性使用消息摘要算法防止篡改；认证性分为实体认证和消息认证。
- 攻击分为主动攻击（中断、篡改：对完整性的攻击、伪造：对认证性的攻击、重放：可使用时间戳进行预防）和被动攻击（截取：对机密性的攻击）。
- 密码学是保障信息安全的作用。



密码学发展史

- 发源：1949年香农发表一篇题为《保密系统的通信理论》的经典论文。
- 密码学的发展经历了两个阶段：传统密码学和现代密码学。
- 传统密码：古代密码学、近代密码。
- 现代密码学：1949年香农发表《保密系统的通信理论》标志着现代密码学的真正开始（第一次质的飞跃）。1976年，Diffie和Hellman发表《密码学的新方向》，标志着公钥密码体制的诞生（第二次质的飞跃）。1978年，Rivest、Shamir和Adleman提出了RSA公钥密码体制。

密码及法律法规

- 密码法规是社会信息化密码管理的依据。

密码学基础

密码学分类

- 密码学分为密码码编码学密码分析学。编码学主要分为保密体制和认证体制，从使用密钥的策略上分为:对称密码体制和非对称密码体制（亦称公钥密码体制）。
- 密码分析学中：设计和使用密码系统必须遵守：柯克霍夫准则，要求算法必须公开，对密钥进行保护。
- 保密体制模型：明文空间、密文空间、密钥空间、加密算法和解密算法。
- 保密体制的安全性：

1. 按照安全性递减的顺序划分：全部破解、全盘推导、实例推导和信息推导。

2.根据密码分析者可获得的密码分析的信息量把密码体制的攻击划分：

1. 向上取整和向下取整(1)
2. 信息抽取工具的学习和使用(1)

（1）唯密文攻击（仅知道一些密文）

（2）已知明文攻击（知道一些密文和相应明文）---------->密码系统至少应经受住的攻击；对流密码的攻击方式

（3）选择明文攻击（密码分析者可以选择一些明文并得到相应密文）

（4）选择密文攻击（密码分析者可以选择一些密文并得到相应明文）

（5）选择文本攻击

3.攻击方式的分类：

（1）穷举攻击（解决方法：增大密钥量）

（2）统计分析攻击（解决方法：使明文的统计特性和密文的统计特性不一样）

（3）数学分析攻击（解决方法：选用足够复杂的加密算法）

4.安全性级别：无条件安全性（H(P|C)）=H(P)）、计算安全性（计算出或估计出破译一个密码系统的计算量下限，利用已有的最好方法破译它所需要的代价超过了破译者的破译能力（时间、空间和资源等））和可证明安全性。

- 认证体制模型

  认证体制包括实体认证和消息认证。这里主要指消息认证。

  认证体制的安全性：按照攻击目标不同可分为完全摧毁、一般性伪造、选择性伪造和存在性伪造。

- 依照攻击者的资源，分为唯密钥攻击、已知消息攻击、一般的选择消息攻击、特殊的选择消息攻击、自适应的选择消息攻击。

香农理论

参看信息论与编码http://www.cnblogs.com/WittPeng/p/8988941.html

认证系统的信息理论

复杂度理论

算法的复杂度

- 度量要素：时间复杂度（计算复杂度）和空间复杂度
- 复杂度 O(*)

问题的复杂度

- P类问题：具有一个在多项式时间内求解的算法的问题
- NP类问题：不存在多项式时间求解算法的问题
- 量子计算机可以将NP类问题转化为P类问题

古典密码体制

1.置换密码

1.列置换密码

- 加密过程：（1）明文按照固定宽度m按行写出，不足部分按照双方约定方式填充，得到字符矩阵；

  （2）进行置换操作；

  （3）读出后即为密文

- 解密过程：将加密密钥逆置，按照加密过程操作，即可由密文得到明文。
- 如密钥e=(143)(56) 意思是1->4,4->3,3->1,5->6,6->5

2.周期置换密码

- 明文按照固定长度m分组，对字符串编号，重新排列位置从而得到密文；解密时将加密密钥逆置得到解密密钥，重新排列位置后得到明文。

2.代换密码

- 代换密码就是将明文中的字符替换为其他字符的密码体制。

单表代换密码

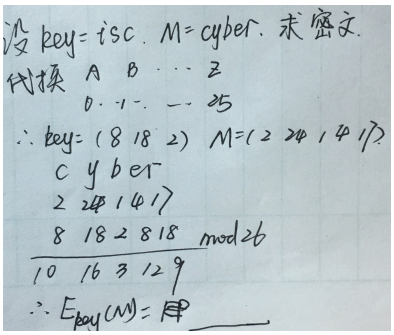
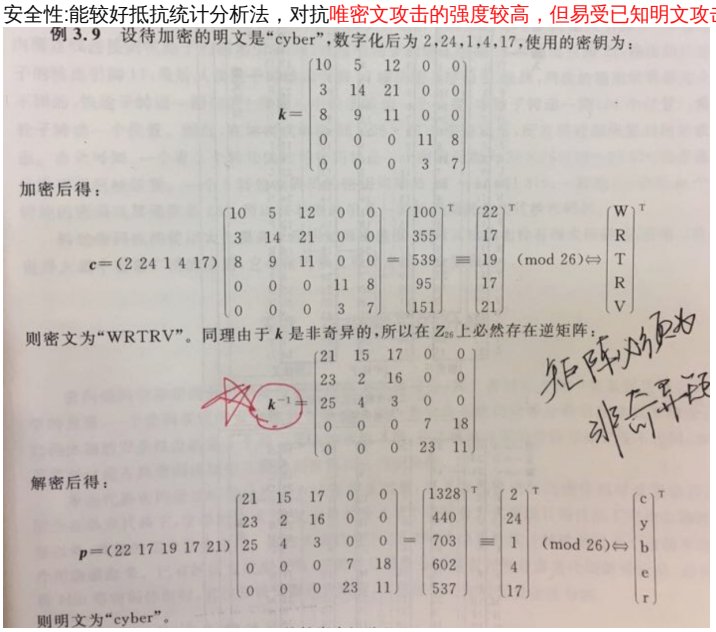
- 基于密钥的代换密码，明文字母对应的密文字母在密文中保持不变
- 仿射密码：e(x)=ax+b(mod26) (a,b属于Z_{26},gcd(a,26)=1)

$$x=d(e(x))=a^{-1}(e(x)-b)(mod26)$$

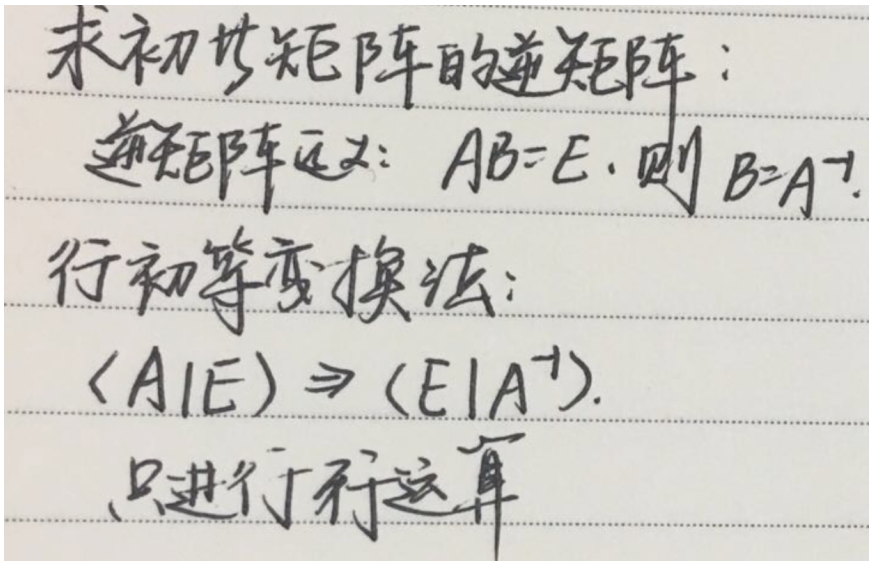
多表代换密码

- 明文中不同位置的同一明文字母在密文中对应的密文字母不同，能够很好地对抗统计密码分析
- 实例：

Playfair密码	加密步骤：a.在适当位置闯入一些特定字母，譬如q,使得明文字母串的长度为偶数，并且将明文字母串按两个字母一组进 b.明文m_1m_2对应的密文c1c2的确定：m_1和m_2同行或同列，则c_1为m_1后的字符，c_2为m_2后的字符；若m_1和m_2既不同行也 两个角上，c_1和m_1同行，c_2和m_2同行。
Vigenere(维吉尼亚密码)	

	设明文 $m=m_1m_2...m_n$，密钥 $k=k_1k_2...k_n$ 则密文：$c=E_k(m)=c_1c_2...c_n$， 其中 $c_i=(m_i+k_i)\bmod 26\ i=1,2,...n$ 当密钥长度比明文长度短时，密钥可周期性地重复利用。 例： 
Vernam(维尔姆密码)	设明文 $m=m_1m_2...m_i...$，密钥 $k=k_1k_2...k_i...$ 其中，$m_i,k_i\in GF(2)i\geq 1$ 则密文 $c=c_1c_2...c_i...$，其中 $c_i=m_i\oplus k_i\ i\geq 1$
Hill(希尔密码)	设明文 $m=(m_1m_2...m_n)\in Z_{26}^n$，密文 $c=(c_1c_2...c_n)\in Z_{26}^n$，密钥为 Z_{26} 上的 nXn 街可逆方阵 $K=(k_{ij})_{n\times n}$，则： $$c=mK\bmod 26$$ $$m=cK^{-1}\bmod 26$$ • $[p]_{1*n}=([c]_{1*n}*[k]^{-1}_{m*n})(\text{mod } 26)62333333$ • 所使用的矩阵必须为非奇异矩阵 • 安全性:能较好抵抗统计分析法，对抗唯密文攻击的强度较高，但易受已知明文攻击。 •  • 例 3.4 假设明文 friday 利用 $m=2$ 的希尔密码加密，得到密文为 PQCFKU。 首先我们有 $e_K(5,17)=(15,16),\ e_K(8,3)=(2,5),\ e_K(0,24)=(10,20)$ 使用头两个明一密文对，可得矩阵方程 $$\begin{pmatrix}15&16\\2&5\end{pmatrix}=\begin{pmatrix}5&17\\8&3\end{pmatrix}K$$ 利用引理 1.4，容易计算 • $\begin{pmatrix}5&17\\8&3\end{pmatrix}^{-1}=\begin{pmatrix}9&1\\2&15\end{pmatrix}$

- 逆矩阵的求法：



- 转轮密码机

古典密码的分析——统计分析法

单表代换密码分析

- 使用字母或汉字的统计规律

多表代换密码分析

- 确定密钥长度：

1. 卡西斯基（Kasisks）测试法：找相同字母间隔字母数的最大公因子，有可能是周期k
2. 重合指数法：计算概率分析重合指数IC=Σp_i^2 ，IC高的可能是单表代换，低的可能是多表代换。

- 确定密钥：拟重合指数测试法：$x=\Sigma r_i q_i$
- 恢复明文并验证

明文-密文对分析法

对称密码之分组密码

对称密码

- 特点：加密速度快、安全性好、基于标准化… …
- 应用：数据保密传输、加密存储… …

分组密码概述

- 将明文消息编码表示后的二进制序列，划分为固定大小的块
- 加密和解密是一一映射的
- 设计应满足要求

1. 分组足够长
2. 密钥长度足够长
3. 由密钥确定的置换算法足够复杂
4. 加密和解密运算简单
5. 一般无数据扩展

- 理想分组密码
- 分组密码的设计原则：扩散、混乱
- 乘积密码体制：在密钥控制下扩散和混乱两种密码操作的多次迭代
- 迭代结构

1. Feistel密码
2. SP网络

组成	S盒（代换）：混乱作用，P盒（置换）：扩散作用
效果	雪崩效应
设计原则	1. 分组长度 2. 密钥长度 3. 轮函数F的设计原则：基本原则：非线性、可逆性、雪崩效应，性能指标：安全性、速度、灵活性 4. 子密钥的生成方法 5. 迭代的轮数

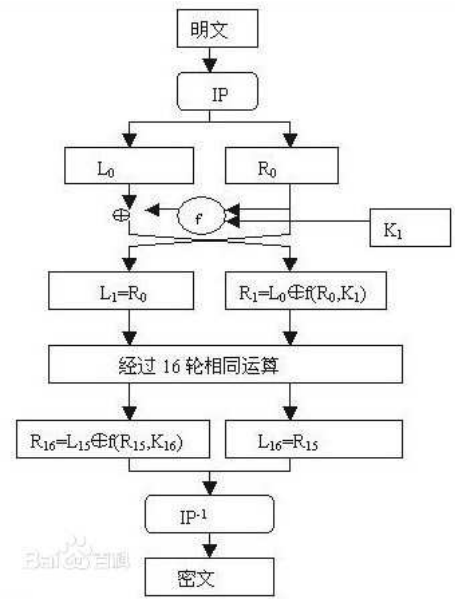
DES算法

- 特点

分组长度	64位
密码体制	对称密码体制，加密和密钥使用同一密钥，仅子密钥编排顺序不同

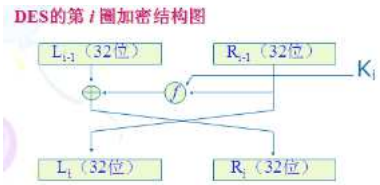
有效密钥长度	56位（原64位的每个第8位为奇偶校验位，可忽略）
迭代结构	SP网络结构，共16轮
优点	只使用了标准的算术和逻辑运算

- 流程



①64位明文->②IP置换->

③分为两部分L0和R0：L0为下一轮的R1；R0和第一次置换得到的子密钥混合经过F函数得到下一轮的L1



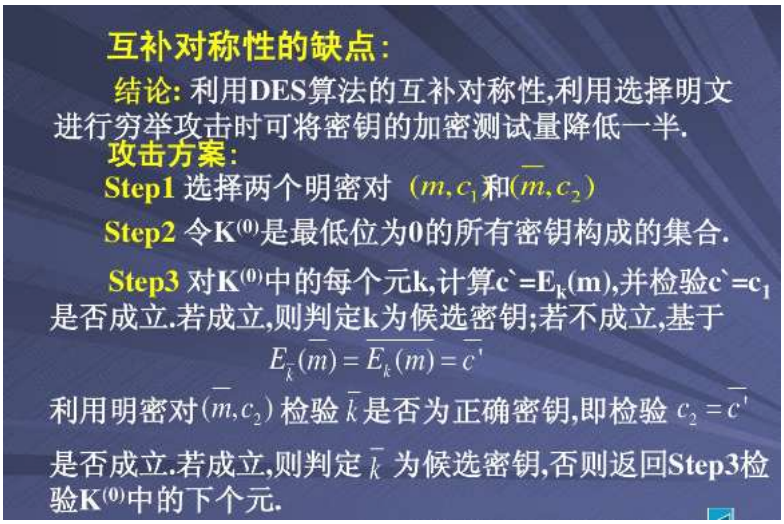
->④置换IP^{-1}->⑤得到64位密文

F函数：

扩展置换	8*4的矩阵 每行的头尾各补齐一位，变成8*6的矩阵
K_i子密钥的生成算法	将56位的有效的密钥压缩成48位
代换盒（S盒）	步骤说明：b_1b_6确定行，$b_2b_3b_4b_5$确定列，将48位变成32位 特点： 1. 非线性 2. 每一行包括所有16种四位二进制码 3. 两个输出相差1bit，输出至少相差2bit 4. … …
置换运算（P盒）	

- 安全性
  - 缺陷

1. 互补性：



2.弱密钥：4个弱密钥，12个半弱密钥

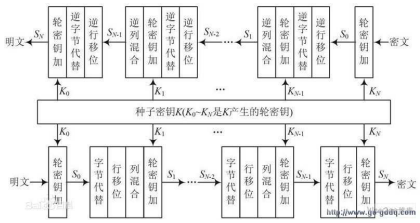
3.迭代轮数

4.密钥长度

- 应对方法：多重DES
  - 二重DES
  - 三重DES
- DES的分析方法：
  - 差分分析：由明文差和密文差求系数a，当轮数低于8轮时，个人计算机几分钟即可攻破
  - 线性分析

AES算法

- 特点介绍
  - 分组长度：128位
  - 密钥长度和对应轮数：128位10轮，192位12轮，256位14轮
  - 过程：前9轮 字节代换、行移位、列混合和轮密钥加，第10轮 字节代换、行位移和轮密钥加



字节代换：S盒定义方法

（1）初始化S盒，将第m行n列的元素初始化为0xmn

（2）将S盒中的每个字节映射为它在有限域GF(2^8)中的逆，0x00映射为自身。AES使用$Z_2[x]$上的不可约多项式m(x)=$x^8+x^4+x^3+x+1$来构造GF(2^8)。求逆元素的方法是使用$Z_2[x]$上的扩展的欧几里得算法。

行位移：简单的左循环位移操作，第n行左移n位

列混合 ：通过矩阵相乘来实现

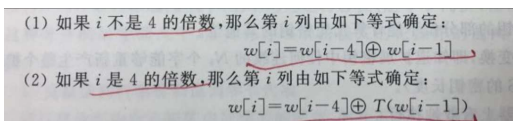
轮密钥加：

密码扩展算法：

1.将初始密钥输入到一个4*4的矩阵中，每列的四个字节组成一个字，依次命名为w[0],w[1],w[2],w[3]

2.扩充40个新列，构成总共44列的扩展密钥数组

产生方式为



其中，T的组成为：1.字循环（将一个字中的4个字节循环左移一个字节）

2.字节代换（使用S盒）

3.轮常量异或（将前两步的结果同轮常量Rcon[j]进行异或，其中j表示轮数）

  - AES的结构
    - AES结构的一个显著特征是它不是Feistel结构
    - 输入的密钥被扩展成由44个32位字节所组成的数组w[i]
    - AES结构由四个阶段组成
    - 仅仅在轮密钥加阶段使用密钥，并在算法的开始和结束都是用轮密钥加阶段
    - 每个阶段均可逆，解密算法和加密算法并不一样

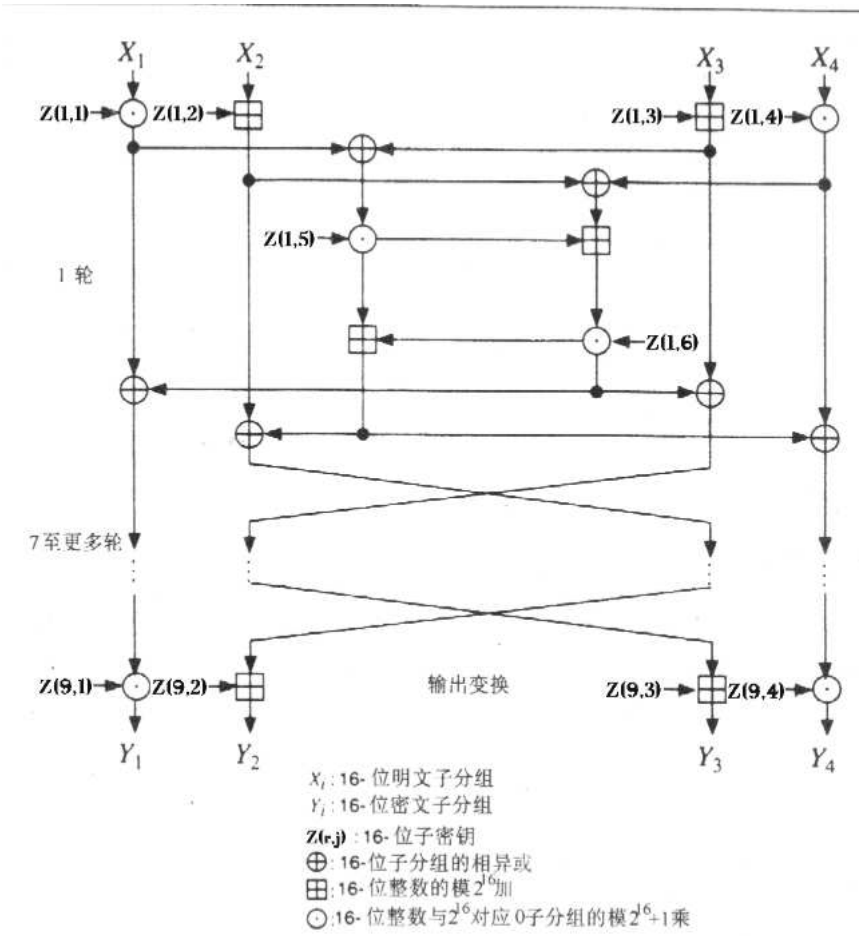
    - 加密和解密过程的最后一轮只包含3个阶段，这是由AES的特定结构所决定的，而且也是密码算法可逆性所要求的
- AES的安全性和可用性
    - AES和DES的对比

	相同	不同		
DES	1. 二者的轮函数都由3层构成，非线性结构、线性混合层、子密钥异或，只是顺序不同； 2. AES的子密钥加对应于DES的S盒之前的子密钥异或； 3. AES的列混合运算目的是让不同的字节相互影响，而DES中F函数的输出与左边一半数据相加也有类似的效果； 4. AES的非线性运算是字节代换，对应于DES中唯一的非线性运算S盒； 5. 行移位运算保证了每一行的字节不仅仅影响其他行对应的字节，而且影响其他行所有的字节，这与DES中的置换P相似	密钥长度固定56位	面向比特的运算	加密和解密运算一致
AES		密钥长度可以是128位、192位、256位	面向字节的运算	加密和解密运算不一致，加密器不能同时用做解密器

典型分组密码

国际数据加密算法（IDEA）

- 工作原理：明文64位，密钥128位
- 轮函数：分为8轮，每轮输入6个子密钥和4个状态块
- 输出变换



- 解密过程
- 子密钥生成

RC6

- 加密过程
- 解密过程
- 密钥扩展方案

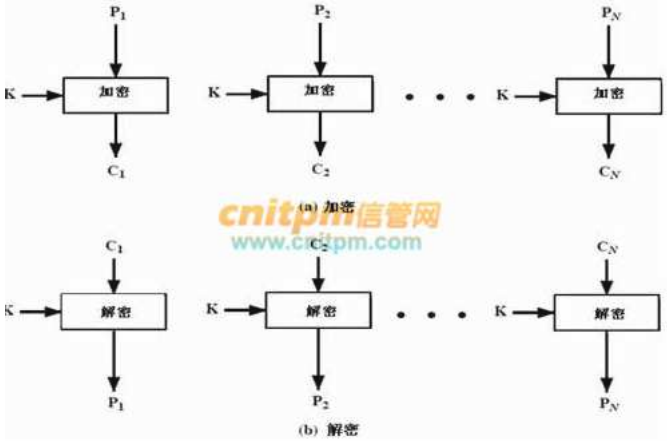
Skipjakc算法

Callmellia算法

工作模式

电子密码本（ECB）模式

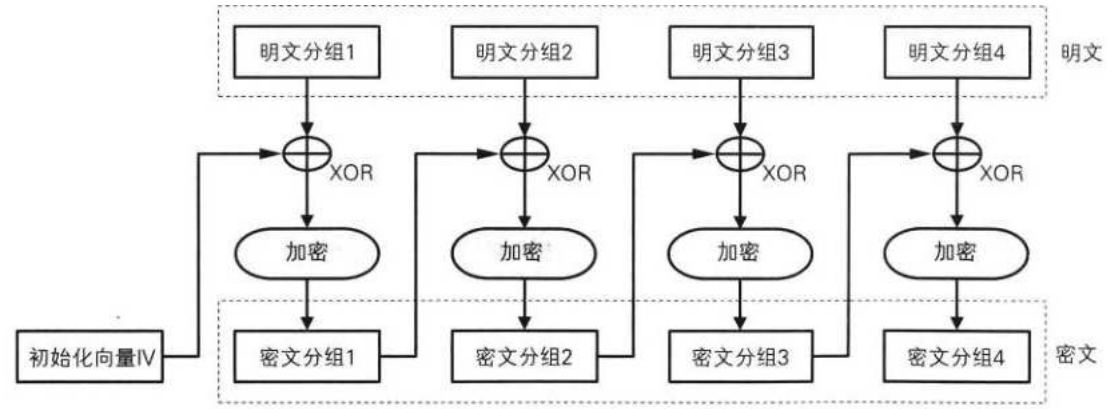
- 工作模式



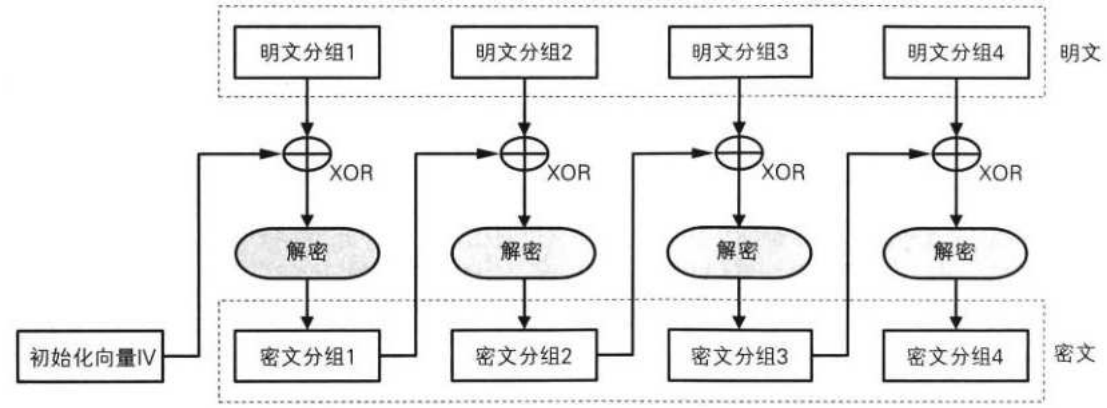
- 特点：
  - 分组数量庞大，易受统计分析攻击、分组重放攻击和代换攻击。
  - 明文或者密文出现一位的错误，只会影响一个分组，不会是错误扩散。
  - 是最快最简单的工作模式
- 应用：数据随机且较少的情况

密码分组链接（CBC）模式

- 工作模式



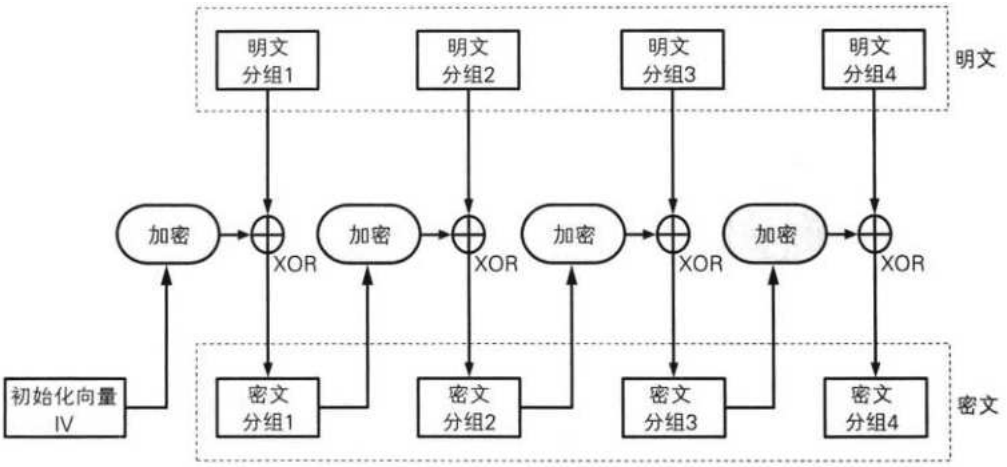
CBC模式的解密



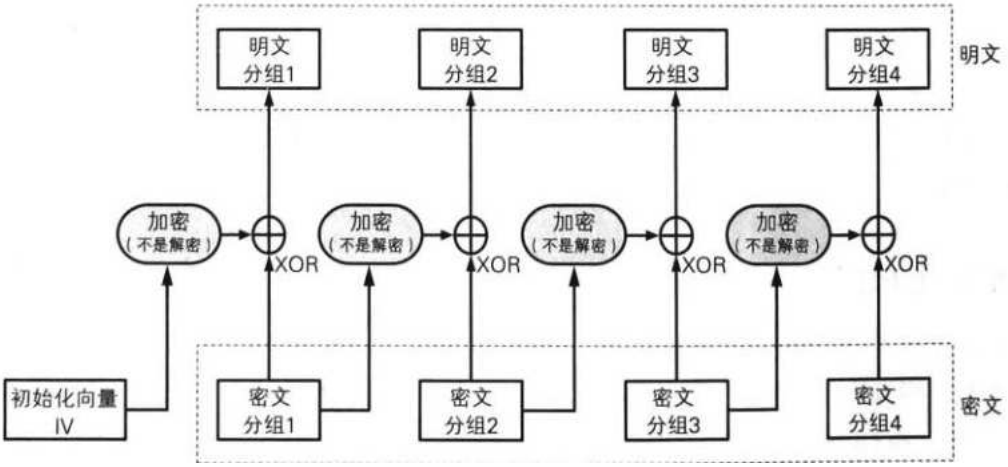
- 特点
  - 克服了ECB模式的缺点
  - 虽然加密会使错误扩散，但解密的过程又进行了抵消，最后出错的仍是一个分组（密文的错误会由一组变成两组）
  - 若文档中的一个分组和他前面的一个分组和另一个文档相同，则这个分组会加密出相同的结果，所以引进了初始化向量IV，使头文件不同（IV不用加密，可以和明文一起传递）
- 应用：大型文件的加密，是软件加密的最好选择

密码反馈（CFB）模式

- 工作模式



CFB模式的解密

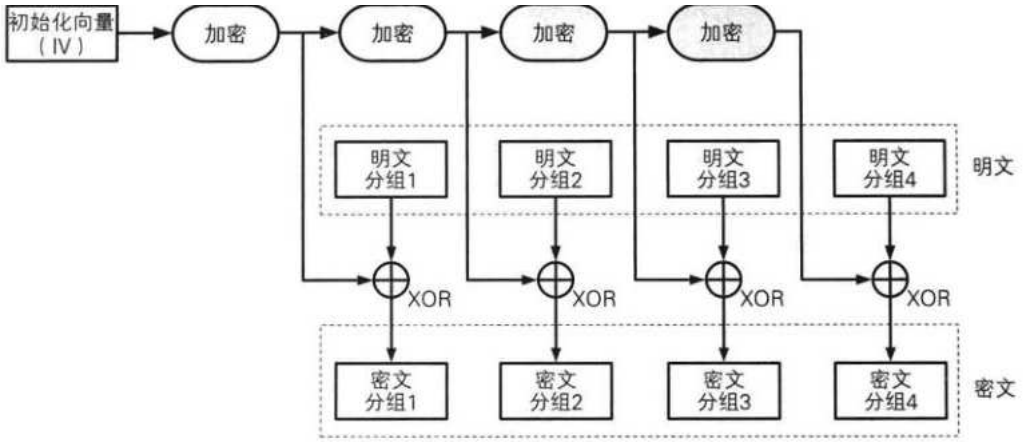


其中，加密算法也能用于解密。加密是对移位寄存器的操作，不对明文加密

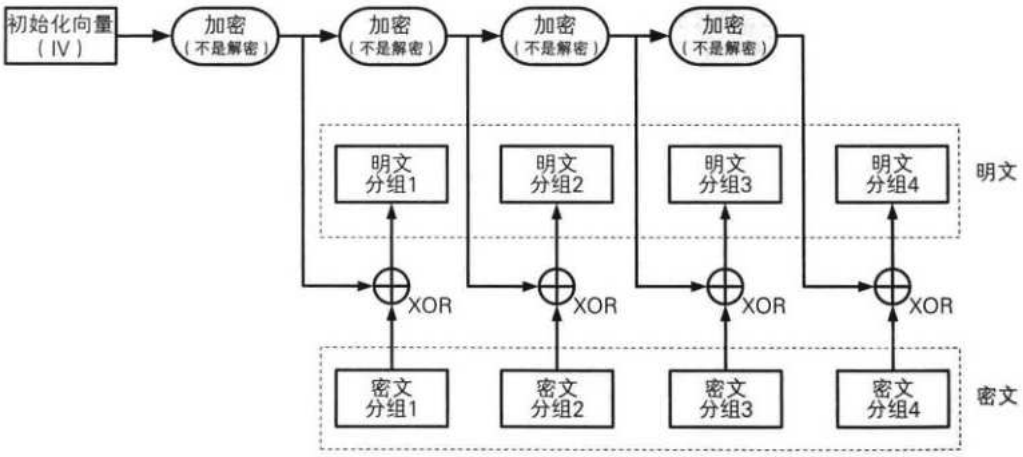
- 特点
  - 面向比特流进行操作
  - 可用于同步序列密码，加密和解密可同时进行
  - 有CBC的优点
  - 对信道错误较敏感且会进行传播，但解密后会纠正。对明文只会影响一个分组，对密文的错误影响只有寄存器推出错误密文后，才能阻止扩散
  - 数据加密速率低
- 应用:加密字符序列

输出反馈（OFB）模式

- 工作模式



OFB模式的解密



- 特点
  - 改进了CFB，错误不会传播，但密文的错误难以发现
  - 不具有自同步的能力
  - 初始向量IV不需要保密
- 应用：在极易出错的环境选用的模式，但需要有高速同步机制

计数器（CTR）模式

对称密码之序列密码

简介

- 定义：指明文消息按字符逐字符地加密的一类密码算法

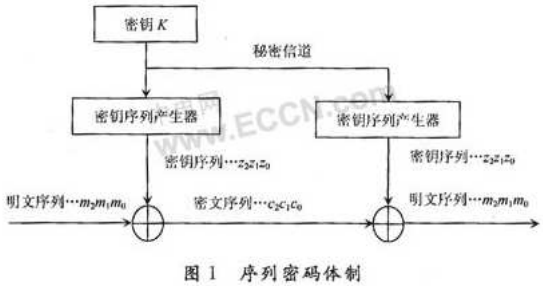
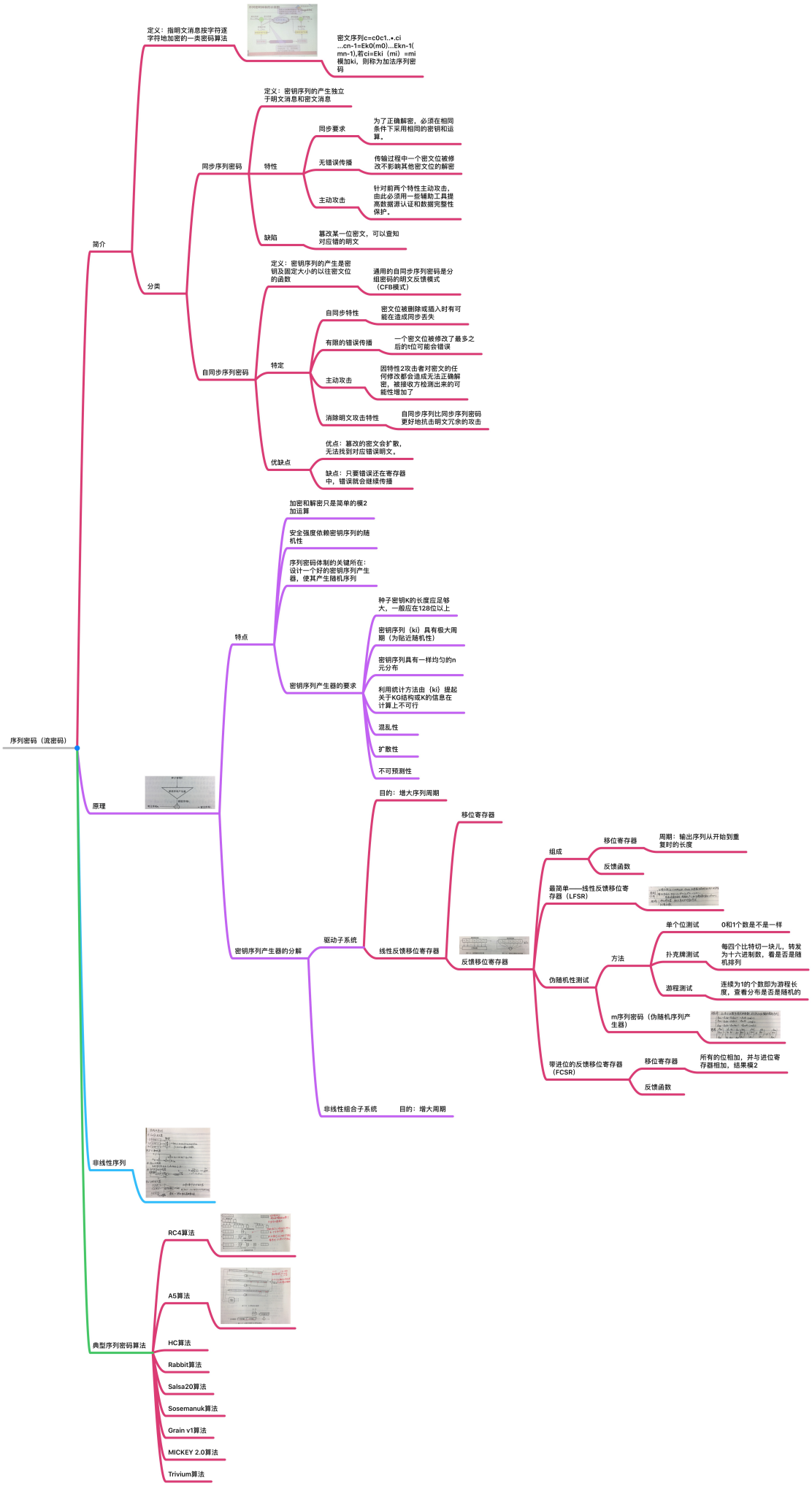


图 1　序列密码体制

密文序列c=c0c1…*…cn-1=Ek0(m0）…Ekn-1(mn-1),若ci=Eki(mi)=mi模加ki，称为加法序列密码。

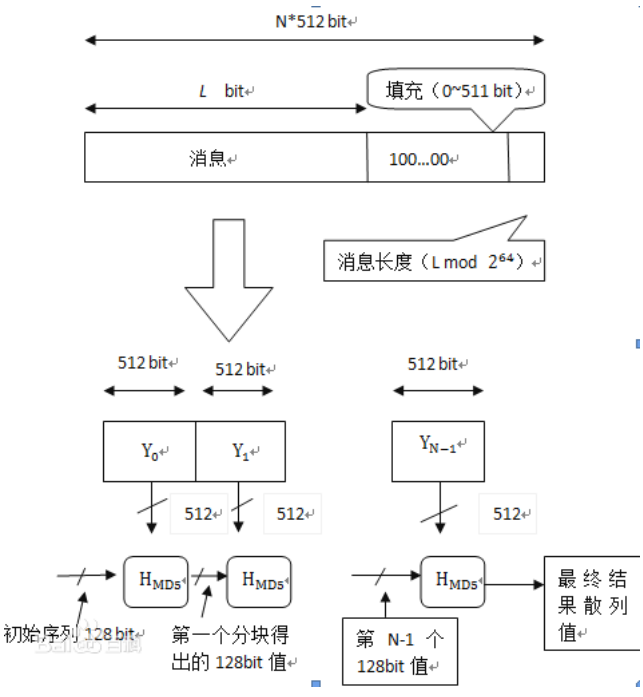


Hash函数和消息认证

Hash函数

定义	是一个从消息空间到像空间不可逆映射，同时是一种具有压缩性的单向函数	
散列值的生成	h=H(M) h是定长的散列值，H是Hash函数运算，M是一个变成消息	
应用	数字签名	
	消息认证	生成程序或文档的“数字指纹” 用于安全运输和存储口令
性质	生成任意长度的消息	
	产生定长的输出	
	计算任意给定的消息比较容易	
	安全性	单向性：找到H(x)=h的x是不可行的
		抗弱碰撞性：对于给定的消息M_1，要发现另一个消息M_2，满足$H(M_1)=H(M_2)$在计算上是不可行的
		抗强碰撞性：找任意一对不同的消息M_1M_2，使$H(M_1)=H(M_2)$在计算上是不可行的
	雪崩效应	
	单向性	
用途	生成数字签名Sign(H(M))	
	对程序或者文件生成摘要（完整性认证）H(M)	
	口令H（Password）	
函数结构	核心技术：设计无碰撞的压缩函数f	
	迭代结构： 1.将输入消息分成L个固定长度的分组，每个分组为b位，最后一个分组包含输入消息的总长度，若不足b位需要填充， 2.压缩函数f:有两个输入，一个是前一次迭代的n位输出，成为链接变量；一个是消息的b位分组，并产生一个n位的输出，即散列值。	

Hash算法

MD5（128位）	结构	
	生成过程	
	主循环	每一分组的算法流程如下： 第一分组需要将上面四个链接变量复制到另外四个变量中：A到a，B到b，C到c，D到d。从第二分组开始的变量为上一分组的运算结果，即A = a， B = b， C = c， D = d。

主循环有四轮（MD4只有三轮），每轮循环都很相似。第一轮进行16次操作。每次操作对a、b、c和d中的其中三个作一次非线性函数运算，然后将所得结果加上第四个变量，
文本的一个子分组和一个常数。再将所得结果向左环移一个不定的数，并加上a、b、c或d中之一。最后用该结果取代a、b、c或d中之一。
以下是每次操作中用到的四个非线性函数（每轮一个）。
F(X ,Y ,Z) = (X & Y) | ((~X) & Z)
G(X ,Y ,Z) = (X & Z) | (Y & (~Z))
H(X ,Y ,Z) =X ^ Y ^ Z
I(X ,Y ,Z) =Y ^ (X | (~Z))
（&是与（And），|是或（Or），~是非（Not），^是异或（Xor））
这四个函数的说明：如果X、Y和Z的对应位是独立和均匀的，那么结果的每一位也应是独立和均匀的。
F是一个逐位运算的函数。即，如果X，那么Y，否则Z。函数H是逐位奇偶操作符。
假设Mj表示消息的第j个子分组（从0到15），常数ti是4294967296*abs(sin(i) ）的整数部分，i 取值从1到64，单位是弧度。（4294967296=232）
现定义：
FF(a ,b ,c ,d ,Mj ,s ,ti) 操作为 a = b + ((a + F(b,c,d) + Mj + ti) << s)
GG(a ,b ,c ,d ,Mj ,s ,ti) 操作为 a = b + ((a + G(b,c,d) + Mj + ti) << s)
HH(a ,b ,c ,d ,Mj ,s ,ti) 操作为 a = b + ((a + H(b,c,d) + Mj + ti) << s)
II(a ,b ,c ,d ,Mj ,s ,ti) 操作为 a = b + ((a + I(b,c,d) + Mj + ti) << s)
注意：“<<”表示循环左移位，不是左移位。
这四轮（共64步）是：
第一轮
FF(a ,b ,c ,d ,M0 ,7 ,0xd76aa478)
FF(d ,a ,b ,c ,M1 ,12 ,0xe8c7b756)
FF(c ,d ,a ,b ,M2 ,17 ,0x242070db)
FF(b ,c ,d ,a ,M3 ,22 ,0xc1bdceee)
FF(a ,b ,c ,d ,M4 ,7 ,0xf57c0faf)
FF(d ,a ,b ,c ,M5 ,12 ,0x4787c62a)
FF(c ,d ,a ,b ,M6 ,17 ,0xa8304613)
FF(b ,c ,d ,a ,M7 ,22 ,0xfd469501)
FF(a ,b ,c ,d ,M8 ,7 ,0x698098d8)
FF(d ,a ,b ,c ,M9 ,12 ,0x8b44f7af)
FF(c ,d ,a ,b ,M10 ,17 ,0xffff5bb1)
FF(b ,c ,d ,a ,M11 ,22 ,0x895cd7be)
FF(a ,b ,c ,d ,M12 ,7 ,0x6b901122)
FF(d ,a ,b ,c ,M13 ,12 ,0xfd987193)
FF(c ,d ,a ,b ,M14 ,17 ,0xa679438e)
FF(b ,c ,d ,a ,M15 ,22 ,0x49b40821)
第二轮
GG(a ,b ,c ,d ,M1 ,5 ,0xf61e2562)
GG(d ,a ,b ,c ,M6 ,9 ,0xc040b340)
GG(c ,d ,a ,b ,M11 ,14 ,0x265e5a51)
GG(b ,c ,d ,a ,M0 ,20 ,0xe9b6c7aa)
GG(a ,b ,c ,d ,M5 ,5 ,0xd62f105d)
GG(d ,a ,b ,c ,M10 ,9 ,0x02441453)
GG(c ,d ,a ,b ,M15 ,14 ,0xd8a1e681)
GG(b ,c ,d ,a ,M4 ,20 ,0xe7d3fbc8)
GG(a ,b ,c ,d ,M9 ,5 ,0x21e1cde6)
GG(d ,a ,b ,c ,M14 ,9 ,0xc33707d6)
GG(c ,d ,a ,b ,M3 ,14 ,0xf4d50d87)
GG(b ,c ,d ,a ,M8 ,20 ,0x455a14ed)
GG(a ,b ,c ,d ,M13 ,5 ,0xa9e3e905)
GG(d ,a ,b ,c ,M2 ,9 ,0xfcefa3f8)
GG(c ,d ,a ,b ,M7 ,14 ,0x676f02d9)
GG(b ,c ,d ,a ,M12 ,20 ,0x8d2a4c8a)
第三轮
HH(a ,b ,c ,d ,M5 ,4 ,0xfffa3942)
HH(d ,a ,b ,c ,M8 ,11 ,0x8771f681)
HH(c ,d ,a ,b ,M11 ,16 ,0x6d9d6122)
HH(b ,c ,d ,a ,M14 ,23 ,0xfde5380c)
HH(a ,b ,c ,d ,M1 ,4 ,0xa4beea44)
HH(d ,a ,b ,c ,M4 ,11 ,0x4bdecfa9)
HH(c ,d ,a ,b ,M7 ,16 ,0xf6bb4b60)
HH(b ,c ,d ,a ,M10 ,23 ,0xbebfbc70)
HH(a ,b ,c ,d ,M13 ,4 ,0x289b7ec6)
HH(d ,a ,b ,c ,M0 ,11 ,0xeaa127fa)
HH(c ,d ,a ,b ,M3 ,16 ,0xd4ef3085)
HH(b ,c ,d ,a ,M6 ,23 ,0x04881d05)
HH(a ,b ,c ,d ,M9 ,4 ,0xd9d4d039)
HH(d ,a ,b ,c ,M12 ,11 ,0xe6db99e5)
HH(c ,d ,a ,b ,M15 ,16 ,0x1fa27cf8)
HH(b ,c ,d ,a ,M2 ,23 ,0xc4ac5665)
第四轮
II(a ,b ,c ,d ,M0 ,6 ,0xf4292244)
II(d ,a ,b ,c ,M7 ,10 ,0x432aff97)
II(c ,d ,a ,b ,M14 ,15 ,0xab9423a7)
II(b ,c ,d ,a ,M5 ,21 ,0xfc93a039)
II(a ,b ,c ,d ,M12 ,6 ,0x655b59c3)
II(d ,a ,b ,c ,M3 ,10 ,0x8f0ccc92)
II(c ,d ,a ,b ,M10 ,15 ,0xffeff47d)
II(b ,c ,d ,a ,M1 ,21 ,0x85845dd1)
II(a ,b ,c ,d ,M8 ,6 ,0x6fa87e4f)
II(d ,a ,b ,c ,M15 ,10 ,0xfe2ce6e0)
II(c ,d ,a ,b ,M6 ,15 ,0xa3014314)
II(b ,c ,d ,a ,M13 ,21 ,0x4e0811a1)
II(a ,b ,c ,d ,M4 ,6 ,0xf7537e82)
II(d ,a ,b ,c ,M11 ,10 ,0xbd3af235)
II(c ,d ,a ,b ,M2 ,15 ,0x2ad7d2bb)
II(b ,c ,d ,a ,M9 ,21 ,0xeb86d391)
所有这些完成之后，将a、b、c、d分别在原来基础上再加上A、B、C、D。
即a = a + A，b = b + B，c = c + C，d = d + D
然后用下一分组数据继续运行以上算法。

SHA-1（160位）	美国国家标准技术研究所NIST开发	1、将消息摘要转换成位字符串 因为在Sha-1算法中，它的输入必须为位，所以我们首先要将其转化为位字符串，我们以“abc”字符串来说明问题，因为'a'=97, 'b'=98, 'c'=99，所以将其转换为位串后为： 01100001 01100010 01100011 2、对转换后的位字符串进行补位操作 Sha-1算法标准规定，必须对消息摘要进行补位操作，即将输入的数据进行填充，使得数据长度对512求余的结果为448，填充比特位的最高位补一个1，其余的位补0，如果在补位之前已经满足对512取模余数为448，也要进行补位，在其后补一位1即可。总之，补位是至少补一位，最多补512位，我们依然以“abc”为例，其补位过程如下： 初始的信息摘要：01100001 01100010 01100011 第一步补位： 01100001 01100010 01100011 1 .... 补位最后一位： 01100001 01100010 01100011 10.....0(后面补了423个0) 而后我们将补位操作后的信息摘要转换为十六进制，如下所示： 61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 3、附加长度值 在信息摘要后面附加64bit的信息，用来表示原始信息摘要的长度，在这步操作之后，信息报文便是512bit的倍数。通常来说用一个64位的数据表示原始消息的长度，如果消息长度不大于2^64，那么前32bit就为0，在进行附加长度值操作后，其“abc”数据报文即变成如下形式： 61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000018 因为“abc”占3个字节，即24位 ，换算为十六进制即为0x18。 4、初始化缓存 一个160位MD缓冲区用以保存中间和最终散列函数的结果。它可以表示为5个32位的寄存器(H0,H1,H2,H3,H4)。初始化为： H0 = 0x67452301 H1 = 0xEFCDAB89 H2 = 0x98BADCFE H3 = 0x10325476 H4 = 0xC3D2E1F0 如果大家对MD-5不陌生的话，会发现一个重要的现象，其前四个与MD-5一样，但不同之处为存储为big-endien format. 5、计算消息摘要 在计算报文之前我们还要做一些基本的工作，就是在我们计算过程中要用到的方法，或定义。 (1)、循环左移操作符Sn(x),x是一个字，也就是32bit大小的变量，n是一个整数且0<=n<=32。Sn(X) = (X<<n)OR(X>>32-n) (2)、在程序中所要用到的常量，这一系列常量字k(0)、k(1)、...k(79)，将其以十六进制表示如下： Kt = 0x5A827999 (0 <= t <= 19) Kt = 0x6ED9EBA1 (20 <= t <= 39) Kt = 0x8F1BBCDC (40 <= t <= 59) Kt = 0xCA62C1D6 (60 <= t <= 79) (3)、所要用到的一系列函数 Ft(b,c,d) ((b&c)|((~b)&d)) (0 <= t <= 19) Ft(b,c,d) (b^c^d) (20 <= t <= 39) Ft(b,c,d) ((b&c)|(b&d)|(c&d)) (40 <= t <= 59) Ft(b,c,d) (b^c^d) (60 <= t <= 79) (4)、计算 计算需要一个缓冲区，由5个32位的字组成，还需要一个80个32位字的缓冲区。第一个5个字的缓冲区被标识为A，B，C，D，E。80个字的缓冲区被标识为W0, W1,..., W79 另外还需要一个一个字的TEMP缓冲区。 为了产生消息摘要，在第4部分中定义的16个字的数据块M1, M2,..., Mn 会依次进行处理，处理每个数据块Mi 包含80个步骤。 现在开始处理M1, M2, ... , Mn。为了处理 Mi,需要进行下面的步骤 (1). 将 Mi 分成 16 个字 W0, W1, ... , W15, W0 是最左边的字 (2). 对于 t = 16 到 79 令 Wt = S1(Wt-3 XOR Wt-8 XOR Wt- 14 XOR Wt-16). (3). 令 A = H0, B = H1, C = H2, D = H3, E = H4. (4) 对于 t = 0 到 79，执行下面的循环 TEMP = S5(A) + ft(B,C,D) + E + Wt + Kt; E = D; D = C; C = S30(B); B = A; A = TEMP; (5). 令 H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E. 在处理完所有的 Mn, 后，消息摘要是一个160位的字符串，以下面的顺序标识 H0 H1 H2 H3 H4. 对于SHA256,SHA384,SHA512。你也可以用相似的办法来计算消息摘要。对消息进行补位的算法完全是一样的。

Hash函数的攻击

生日攻击：$p(k)=1-p_{365}^{k}/365^{k}$;k=23,p(k)=0.5073;k=100,p(k)=0.99999997。可以减少一半的尝试量

两个集合的相交问题。

(1) 攻击者 A 准备好合法的消息 x，再拟定一个准备替换消息 x 的假消息 x'。

(2) 攻击者对消息 x 产生 $2^{m/2}$ 个变形的消息(含义不变)，同时产生 x' 的 $2^{m/2}$ 个变形消息，计算所有这些消息的散列值。

(3) 比较这两个散列函数值集合，以便发现具有相同散列值(匹配)的消息。根据生日悖论，“两个相交集合”问题成功的概率大于 1/2。如果没发现，则再产生其他一批合法消息和假消息，直到出现一个匹配为止。

(4) 攻击者将消息 x 发送给签名者 A，得到其签名 $\mathrm{Sign}_k(H(x))$。

(5) 用匹配的假消息 x' 代替 x，后面仍然附加签名 $\mathrm{Sign}_k(H(x))$ 送给接收方 B。

Hash函数的应用

消息认证：

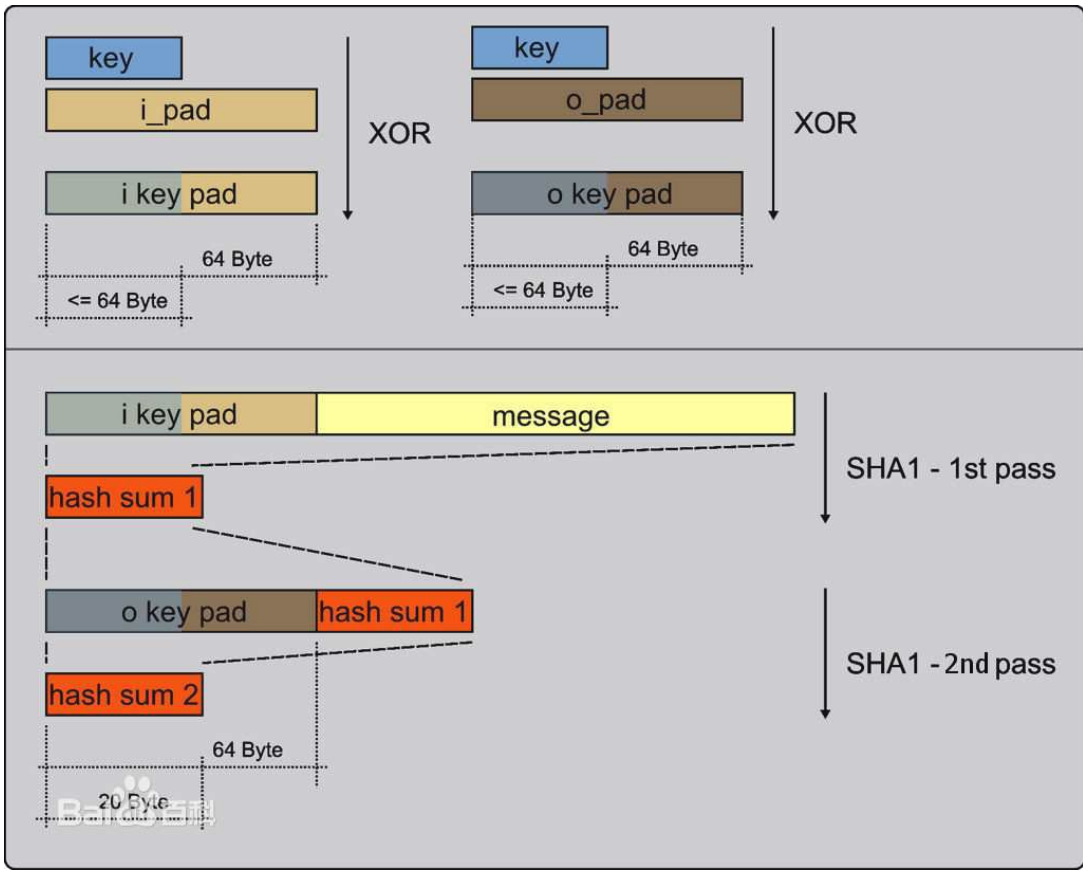
目的：

1. 验证是否真实
2. 验证完整性

MDC和加密：仅能验证完整性

消息认证码MAC

- 用于消息源认证和消息完整性的认证
- 构造MAC的方法
  - 使用DES分组加密算法的MAC
  - 基于Hash的认证码——HMAC



- 安全性：包里搜索密钥需要2^k（k为密钥长度）；攻击MAC需要2^n次

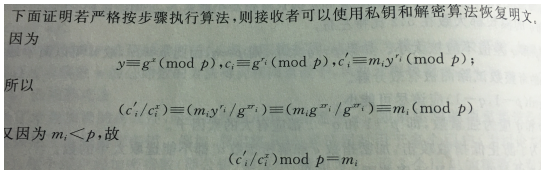
第七章 公钥密钥体制

公钥密码体制概述

- 对称密码体制的局限性：
  - 密钥分发问题
  - 密钥管理问题
  - 数字签名问题
- 公钥加密体制的思想
  - 公钥和私钥
  - 基于陷门单向函数的困难问题
- 公钥密码体制的分类

公钥加密体制介绍

ElGamal	应用	1.加密；2.数字签名
	密钥生成	1. 随机选择一个满足安全要求的1024位的大素数p，并生成有限域Z_p的一个生成元$g \in Z_p^*$；

		2. 选择一个随机数x(1<x<p-1)，计算y≡g^x(mod p),则公钥为(y,g,p)，私钥为S_k=x。 （随机数的选取可以使同一明文在不同时间加密成不同密文）
	加解密算法	1. 加密过程：将明文分组比特串分组，是分组长度小于$\log_2$p,，然后对每个明文分组分别加密 1. 得到公钥P_k（y,g,p）； 2. 消息m分组$m_1m_2m_3m_4$…… 3. 对第i块消息随机选择整数r_i(1<ri<p-1); 4. 计算c_i≡g^{r_i}(mod p),c_i'≡$m_i y^{r_i}$(mod p)(1≤i≤t) 5. 将密文C=(c_1,c_1')()()()····发送给接收方 （可知，密文是明文长度的2倍） 2. 解密过程 1. 接收方接收密文C 2. 使用密钥x和解密算法m_i≡(c_i'/c_i^x)(mod p)(1≤i≤t)进行计算 3. 得到明文 3. 正确性  下面证明若严格按步骤执行算法，则接收者可以使用私钥和解密算法恢复明文， 因为 $y \equiv g^x \pmod p$，$c_i \equiv g^{r_i} \pmod p$，$c_i' \equiv m_i y^{r_i} \pmod p$； 所以 $(c_i'/c_i^x) \equiv (m_i y^{r_i}/g^{xr_i}) \equiv (m_i g^{xr_i}/g^{xr_i}) \equiv m_i \pmod p$ 又因为 $m_i < p$，故 $(c_i'/c_i^x) \bmod p = m_i$
	安全性分析	1. 穷举法和列表法（二分查找算法） O（p） 2. 小步大步算法 来源于生日攻击的思想， 小步为 序列1：g^1,···,g^j,···,g^m(1≤j≤m) 大步为 序列2：y,y*g^{-m},···，y*g^{-im} 找到g^j≡y*g^{-im}(mod p),即找到y=g^{j+im}(mod p),即x=j+im私钥被找到 时间复杂度：O($p^{1/2}$) 3.指数积分法 1. 选取因子基S 2. 建构同余方程组：对若干随机整数k(0≤k≤p)，计算g^k，尝试将g^k写成S中的元素幂次的乘积，即g^k=∏$p_i^{e_i}$(mod p),式子两边取离散对数k≡∑$e_i\log_g(p_i)$(mod (p-1)),重复这个过程，直到有超过m个方程 3. 求$\log_g(p_i)$ 4. 计算x:随机取整数r，计算yg^rmod p，使得其值可表示为S中元素幂次的乘积，即yg^r=∏$p_i^{d_i}$(mod p),取离散对数可得x≡$\log_g(y)$≡-r+∑$d_i\log_g(p_i)$(mod (p-1)),如果成功，即求得此解x。
MH背包公钥加密体制	难题来源	背包问题：∑a_ix_i=b，这是一个NP完全类问题 特例：超递增序列（每一个元素都比先前的元素和大） 可将背包问题转化为P类问题
	公私钥对的生成	1. 选取素数p 和u，且bi为超递增序列 2. 利用uv=1（mod p）可求得v 3. a=ub_k(mod p) 4. {a_i}和p作为公钥，{b_i}和v作为私钥
	加密和解密	1. 明文消息分组 2. 密文c=a_1m_1+···+a_nm_n 1. 利用{bi}求v c的解，可得消息m
	安全性分析	1. NP类问题，至今没有好的求解方法，能经受住穷举攻击 2. 隐蔽性不够，此公钥密码是不安全的
	地位	第一个公钥算法
RSA公钥密码	理论基础	数论中的欧拉定理，安全性依赖于大整数的素因子分解的困难性 欧拉定理：若a和n互素，则$a^{\Phi(n)}$≡1(mod n)
	密钥生成算法 加密和解密	**（1）生成公私密钥** • 选取两个大素数p和q，至少要1024位 • 计算n=p*q • 随机选取整数e在(1≤e≤Φ(n))作为公钥，要求满足gcd(e,Φ(n))=1 • 用Euclid扩展算法计算私钥d，满足d*e≡1(mod Φ(n)),则e和n是公钥，d是私钥

（3）明文加密

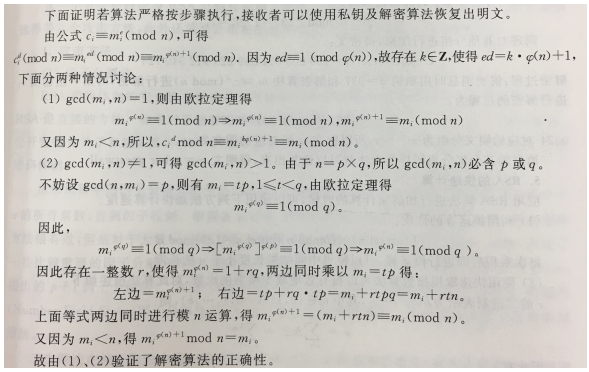
$c=E(m)=m^e(\text{mod } n)$

（4）密文解密。

$m=D(c)=cd(\text{mod } n)$

安全性

1.算法正确性的证明



下面证明若算法严格按步骤执行，接收者可以使用私钥及解密算法恢复出明文。

由公式 $c_i \equiv m_i^e (\text{mod } n)$，可得

$c_i^d (\text{mod } n) \equiv m_i^{ed} (\text{mod } n) \equiv m_i^{k\varphi(n)+1} (\text{mod } n)$. 因为 $ed \equiv 1 (\text{mod } \varphi(n))$，故存在 $k \in \mathbf{Z}$，使得 $ed = k \cdot \varphi(n) + 1$，

下面分两种情况讨论：

(1) $\gcd(m_i, n) = 1$，则由欧拉定理得

$$m_i^{\varphi(n)} \equiv 1 (\text{mod } n) \Rightarrow m_i^{\varphi(n)} \equiv 1 (\text{mod } n), m_i^{\varphi(n)+1} \equiv m_i (\text{mod } n)$$

又因为 $m_i < n$，所以，$c_i^d \bmod n \equiv m_i^{k\varphi(n)+1} \equiv m_i (\text{mod } n)$。

(2) $\gcd(m_i, n) \neq 1$，可得 $\gcd(m_i, n) > 1$。由于 $n = p \times q$，所以 $\gcd(m_i, n)$ 必含 p 或 q。

不妨设 $\gcd(n, m_i) = p$，则有 $m_i = tp, 1 \leqslant t < q$，由欧拉定理得

$$m_i^{\varphi(q)} \equiv 1 (\text{mod } q)。$$

因此，

$$m_i^{\varphi(q)} \equiv 1 (\text{mod } q) \Rightarrow [m_i^{\varphi(q)}]^{\varphi(p)} \equiv 1 (\text{mod } q) \Rightarrow m_i^{\varphi(n)} \equiv 1 (\text{mod } q)。$$

因此存在一整数 r，使得 $m_i^{\varphi(n)} = 1 + rq$，两边同时乘以 $m_i = tp$ 得：

$$\text{左边} = m_i^{\varphi(n)+1}； \quad \text{右边} = tp + rq \cdot tp = m_i + rtpq = m_i + rtn。$$

上面等式两边同时进行模 n 运算，得 $m_i^{\varphi(n)+1} = (m_i + rtn) \equiv m_i (\text{mod } n)$。

又因为 $m_i < n$，得 $m_i^{\varphi(n)+1} \bmod n = m_i$。

故由(1)、(2)验证了解密算法的正确性。

2.攻击

1. 针对n分解的攻击
    1. 试除法
    2. 因子分解分析法：二次筛因子分析法
    3. 侧信道攻击

2.针对算法参数的攻击

1.对素数p和q选取时的限制；p和q长度相差不大，大小相差要大，否则难以抵御除法的攻击；p-1和q-1都应有大的素因子。

2.共模攻击（因此不同用户不用使用相同的p和q）

3.低指数攻击

椭圆曲线公钥加密体制

椭圆曲线

韦尔斯特拉方程 ：$E: y^2+axy+by=x^3+cx^2+dx+e$。密码学中，常采用的椭圆曲线为： $E: y^2=x^3+ax+b$，并要求 $4a^3+27b^2 \neq 0$

Hasse定理：如果E是有限域GF(p)上的椭圆曲线，N是E上的点(x,y)（其中x,yξGF(p)）的个数，则：$|N-(p+1)| \leq 2(p)^{1/2}$

椭圆曲线上的点集合E_p（a,b）对于如下定义的加法规则构成一个Abel群：

1. O+O=O;(O是单位元)
2. 椭圆上的点P，P+O=P；
3. P的逆元是-P；

令P=(x1,y1) ∈ $E_p(a, b)$， Q=(x2, y2) ∈ $E_p(a, b)$, 则P+Q=R=(x3,y3)

其中：

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{若} P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{若} P = Q \text{(倍点规则)} \end{cases}$$

P

4.

5. 满足交换律
6. 满足结合律

点乘规则：

- 如果k为整数，kP=P+…+P （k个P相加）
- 如果s和t为整数，(s+t)P=sP+tP,s(tP)=t(sP)

椭圆曲线点的计算：

		Step1: 对x=0,1,…, p-1计算$x^3+ax+b(\text{mod } p)$ **Step2**: 对step1得到的每一结果确定它是否有一个模p的平方根，如果没有，则$E_p(a,b)$中没有以该结果相应的x为横坐标的点；如果有，就有两个平方根y和p-y，从而点(x, y)和(x, p-y)都是$E_p(a,b)$上的点。
	ECC密钥生成算法	1. 选择一个椭圆曲线E，构造一个椭圆群$E_p(a,b)$。 2. 在椭圆群中挑选生成元点$G=(x_0,y_0)$,需满足nG=O的最小的n是一个非常大的素数。 3. 选择一个小于n的整数n_B作为私钥，然后利用$P_B=n_BG$算出P_B。 公钥为(E,n,G,P_B)，私钥为n_B。
	加密过程	1. A将明文消息编码成一个数m<p,并在椭圆群$E_p(a,b)$中任选一点$P_t=(x_t,y_t)$; 2. 在区间[1,n-1]内，A选取一个随机数k，计算P_1：$P_1=(x_1,y_1)=kG$; 3. 依据接收方B的公钥P_B，A计算点P_2：$P_2=(x_2,y_2)=kP_B$； 4. A计算密文$C=mx_t+y_t$; 5. A传送加密数据$C_m=\{kG,P_t+kP_B,C\}$给接收方B。
	解密过程	1. 接收方B收到Cm; 2. B使用私钥n_B做运算：$P_t+kP_B-n_B(kG)=P_t+k(n_BG)-n_B(kG)=P_t$; 3. B计算$m=(C-y_t)/x_t$,得明文m。
	安全性和优势	安全性基于椭圆曲线上的离散对数问题
		应用前景好，尤其是在移动通信和无线设备上的应用，计算量小，处理速度快，存储空间占用小，带宽要求低。
		160位的ECC密钥和1024位的RSA和1024位的ElGamal的安全性等同。
		可用于加密、数字签名。
		未申请专利
Rabin公钥加密体制	前言（学习意义）	具有很好的参考价值
	特点	不是以一一对应的陷门单向函数为基础，同一密文可能有多种明文；
		破译该体制等价于对大整数的因子分解。
	密钥生成算法	随机选取两个大素数p和q，并且p≡q≡3mod4，将p和q作为私钥，n=pq作为公钥
	加密算法	设明文块为m(m<n)，运用公式$c=m^2 \text{mod} n$ 进行加密，c为密文。
	解密算法	

解密就是求c模n的平方根，即解$x^2 \equiv c \bmod n$，由中国剩余定理知解该方程等价于解方程组

$$\begin{cases} x^2 \equiv c \bmod p \\ x^2 \equiv c \bmod q \end{cases}$$

由于$p \equiv q \equiv 3 \bmod 4$，下面将看到，方程组的解可容易地求出，其中每个方程都有两个解，即

$$x \equiv m \bmod p, x \equiv -m \bmod p$$

$$x \equiv m \bmod q, x \equiv -m \bmod q$$

第八章 数字签名技术

数字签名概述

基本概念

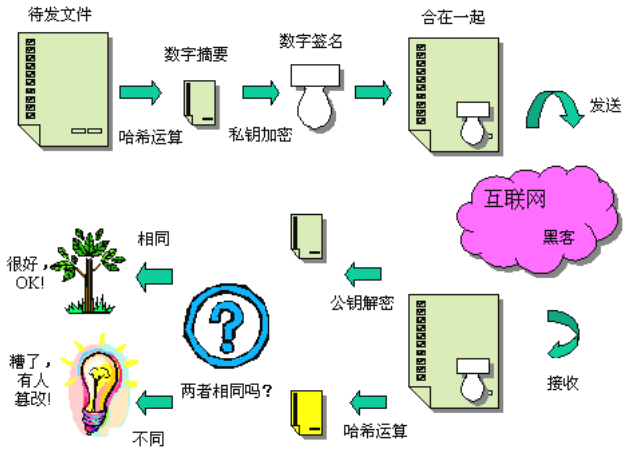
数字签名技术一般分为带仲裁和不带仲裁的两类。如果使用对称密钥进行数字签名，则必须使用仲裁者，非对称可以不带仲裁。

数字签名可以实现不可否认性和消息完整性认证（检验是否被篡改或伪造）

原理

(P,S,K,Sig,Ver),P:密钥生成算法，S:签名算法，K：验证算发，Sign，Verify

过程简图

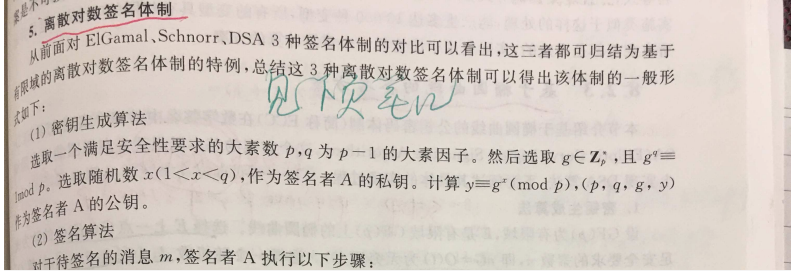


数字签名的实现方案

基于RSA的签名方案	密钥生成算法	1. 选取两个512位的大素数p和q，使得N(=pq)为1024位; 2. 可得Φ(N)=(p-1)(q-1); 3. 选取随机整数e(1<e<Φ(N))，满足gcd(e,Φ(N))=1; 4. 计算d: $d \equiv e^{-1} \pmod{\Phi(N)}$； 5. 公钥Pk=(n,e),私钥为{d,Φ(N),p,q}。 这里，先选择e再确定d的原因是：加密的重要性大于解密的重要性。
	签名算法	1. 利用一个安全的Hash函数h来产生消息摘要h(m); 2. $s \equiv h(m)^d \pmod n$,s就是消息m的签名； 3. 将(s,m)发送给B。
	验证算法	1. 利用上述Hash函数生成h(m); 2. 检验等式$h(m) \bmod n \equiv s^e \pmod n$是否成立，成立则签名有效，否则签名无效。
	正确	

		性	$s \equiv h(m)^d \pmod n, de \equiv 1 \pmod{\varphi(n)}, \varphi(n) = (p-1)(q-1)$ $s^e \bmod n = h(m)^{ed} \bmod n = h(m)^{k\varphi(n)+1} \bmod n = h(m)\ h(m)^{k\varphi(n)} \bmod n$ $= h(m)\ [h(m)^{\varphi(n)}]^k \bmod n = h(m) \bmod n$（其中 k 为整数）
		安全性	1. 签名时使用了Hash函数可以防止利用同态的伪造攻击，有很好的抗攻击性。 2. RSA签名方案存在签名可重用的问题，同一消息在不同时刻签名不应是相同的。
基于离散对数的签名方案	ElGamal签名体制	密钥生成算法	1. 选取一个1024位的大素数p； 2. 选择一个生成元g和随机数x，计算$y \equiv g^x \pmod p$ 签名者的公钥是(p,g,y)，私钥是x。
		签名算法	签名者选取随机数k，计算$r \equiv g^k \pmod p, s \equiv [h(m)-xr]K^{-1} \pmod{p-1}$,签名为(r,s),其中h为安全的Hash函数
		验证算法	计算h(m)后，验证$y^x r^s \equiv g^{h(m)} \pmod p$是否成立
		正确性	$r \equiv g^k \pmod p, s \equiv [h(m)-xr]k^{-1} \pmod{(p-1)}$ $ks \equiv h(m)-xr \pmod{(p-1)}$ $g^{ks} \equiv g^{h(m)-xr} \pmod p$ $g^{ks}\ g^{xr} \equiv g^{h(m)} \pmod p$ $y^r r^s \equiv g^{h(m)} \pmod p$
		安全性	1. 随机数k的选取和保管：首先k值不能泄露，否则签名者的私钥泄露；其次，k不能重复使用；最后，签名者多次签名的的多个k之间无关联。 2. 如果不使用Hash函数，则签名方案容易受到伪造攻击
	Schnorr签名体制	特点	签名速度较快，签名长度较短
		密钥生成算法	1. 选取两大素数p(1024位)和q，q是p-1的大素因子且长度比p小得多； 2. 选取一个生成元，且$g^q \equiv 1 \pmod o$,$g \neq 1$； 3. 选取随机数$1<x<q$，计算$y \equiv g^x \pmod p$，公钥为(p,q,g,y)，私钥为x。
		签名算法	签名者选择随机数k,$1 \leq k \leq q-1$，然后进行如下计算： $r \equiv g^k \pmod p$ $e = h(m,r)$ $s \equiv (xe+k) \pmod q$ 签名为(e,s),其中h为安全的Hash函数。
		验证算法	签名接收者在收到消息m和签名(e,s)后，首先计算$r_1 \equiv g^s y^{-e} \pmod p$ ，然后验证e=h(e,r1),如果等式成立则签名有效，否则无效。
		正确性和安全性	如果所有算法按步骤执行，则接收者输出签名有效。 因为 $r \equiv g^k \pmod p, e = h(m, r), s \equiv (xe + k) \pmod q$ 所以 $r_1 \equiv g^s y^{-e} \equiv g^s g^{-xe} \equiv g^{s-xe} \equiv g^{xe+k-xe} \equiv g^k \equiv r \pmod p$ 因此 $h(m, r_1) = h(m, r) = e$ 安全性和ElGaml类似
	DSA签名体制	密钥生成算法	1. 选取160bite的素数q，选择512~1024bite的素数p，使得p-1能被q整除； 2. 选择$g = h^{(p-1)/q} \pmod p$,h满足$1<h<p-1$,且$g>1$; 3. 选择1~q之间的随机数x作为私钥，计算$y \equiv g^x \pmod p$，用户的公钥是(p,q,g,y)

	签名算法	选取随机数k，$r=(g^k \bmod p) \bmod q$,$s\equiv[h(m)+xr]k^{-1} \pmod q$，其中h是SHA1的特定Hash函数
	验证算法	收到m和签名值(r,s)后，计算 $w\equiv s^{-1} \pmod q$ $u_1\equiv h(m)w \pmod q$ $u_2\equiv rw \pmod q$ $v=(g^{u_1}y^{u_2} \bmod p) \bmod q$ 比较v和r，相同则签名有效，否则无效。
	正确性	若所有算法按步骤执行，则接收者输出签名有效。 因为 $r\equiv g^k \bmod p \pmod q, s\equiv[h(m)+xr]k^{-1} \pmod q$ $w\equiv s^{-1} \pmod q, u_1\equiv h(m)w \pmod q, u_2\equiv rw \pmod q$ 所以 $v=(g^{u_1}y^{u_2} \bmod p) \bmod q=[(g^{h(m)w}y^{rw}) \bmod p] \bmod q=[(g^{h(m)w}g^{xrw}) \bmod p] \bmod q=[g^{(h(m)+xr)w} \bmod p] \bmod q=(g^{k\cdot w} \bmod p) \bmod q=(g^k \bmod p) \bmod q=r$
	安全性	DSA是ElGamal的变形，因此安全性论述在此也同样使用。还有一点是签名算法计算的s正好为0时，会产生1除以0的情况，必须放弃这个签名。
	三种签名体制的对比	如前所述，ElGamal 签名方案的提出是 3 种方案中最早的，它也是后 2 种方案的基础。随后的 Schnorr 签名方案可以看作是 ElGamal 签名方案的一种变型，它缩短了签名的长度。而 DSA 签名方案是 ElGamal 签名方案的另一种变型，它也吸收了 Schnorr 签名方案的一些设计思想。下面就具体介绍这 3 种签名方案的联系与区别。 从参数的初始化上可以看到，DSA 方案和 Schnorr 方案中通过引入素数 q 并选择 $\mathbf{Z}_p^*$ 的 q 阶子群的生成元，修改了 ElGamal 方案中直接选择 $\mathbf{Z}_p^*$ 本身的生成元的做法，这样就使得方案的安全性依赖于 2 个不同的但又相关的离散对数问题，即 $\mathbf{Z}_p^*$ 上的离散对数问题和 q 阶循环子群上的离散对数问题。同时，这 2 种方案的签名文件长度较 ElGamal 方案也有所降低。且在 DSA 中规定 q 的长度是 160 比特，p 的长度可以是 512 比特与 1 024 比特之间 64 的倍数。在 Schnorr 签名方案中没有限制 p 和 q 的长度，签名过程中，Schnorr 方案所用的 H 函数并不只是消息 m 的函数(而是 m 和 r)，这点与另外两种方案不同。此外，DSA 签名方案中专门规定使用算法 SHA1，而在其他两种签名方案中并没有这样的要求。由于采用的算法各不相同，因此 3 种方案的签名验证等式和过程也不尽相同。 根据计算量和签名长度，表 8-1 中定量对比和分析这 3 种方案的效率。其中模加
	离散对数签名体制	


5. 离散对数签名体制

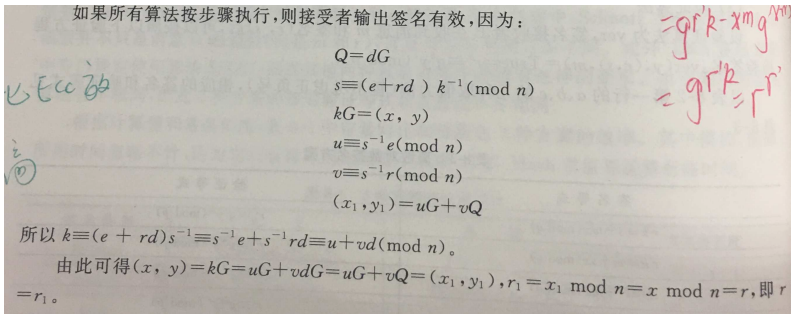
从前面对 ElGamal、Schnorr、DSA 3 种签名体制的对比可以看出，这三者都可归结为基于有限域的离散对数签名体制的特例，总结这 3 种离散对数签名体制可以得出该体制的一般形式如下：

(1) 密钥生成算法

选取一个满足安全性要求的大素数 p,q 为 $p-1$ 的大素因子。然后选取 $g\in Z_p^*$，且 $g^q\equiv 1 \bmod p$。选取随机数 $x(1<x<q)$，作为签名者 A 的私钥。计算 $y\equiv g^x \pmod p$，(p,q,g,y) 作为签名者 A 的公钥。

(2) 签名算法

对于待签名的消息 m，签名者 A 执行以下步骤：

基于椭圆曲线的签名方案	密钥生成算法	选择E上一点G∈E，G的阶为满足安全要求的素数n，即nG=O。 选取一个随机数d∈[1,n-1]，计算Q使得Q=dG，那么公钥为(n,Q)，私钥为d。
	签名算法	1. 用户随机选取整数k∈[1,n-1]，计算kG=(x,y),r≡x(mod n)； 2. 计算e = h(m)； 3. 计算s≡(e+rd)k^{-1}(mod n)，如果r=0或s=0，则另选随机数k，重新执行上面的过程，消息m的签名为(r,s)。
	验证算法	1. 计算e=h(m)； 2. 计算u≡s^{-1}e(mod n),v≡s^{-1}r(mod n)，(x_1,y_1)=uG+vQ,$r_1\equiv x_1$(mod n)； 3. 判断r和r1的关系，相等则签名有效，否则无效。
	正确性	 如果所有算法按步骤执行，则接受者输出签名有效，因为： $Q=dG$ $s\equiv(e+rd)\,k^{-1}\pmod n$ $kG=(x,y)$ $u\equiv s^{-1}e\pmod n$ $v\equiv s^{-1}r\pmod n$ $(x_1,y_1)=uG+vQ$ 所以 $k\equiv(e+rd)s^{-1}\equiv s^{-1}e+s^{-1}rd\equiv u+vd\pmod n$。 由此可得$(x,y)=kG=uG+vdG=uG+vQ=(x_1,y_1)$，$r_1=x_1 \bmod n=x \bmod n=r$，即 $r=r_1$。
	安全性	ECDSA安全性依赖于基于椭圆曲线的有限群上的离散对数难题，与RSA和有限域离散对数的数字签名相比，在相同的安全强度条件下，有如下特点 签名长度短、密钥存储空间小，特别是用于存储空间有限、带宽受限、要求高速实现的场合。

特殊数字签名

- 代理签名
- 盲签名
- 群签名
- 不可否认签名
- 门限数字签名
- 多重数字签名
- … ..

密钥管理

密钥管理概述

密钥管理的原则

- 区分密钥管理的策咯和机制
- 完全安全原则
- 最小权力原则
- 责任分离原则
- 密钥分级原则
- 密钥更换原则
- 密钥应该有足够的长度
- 密钥体制不同，密钥管理也不相同

密钥管理的层次结构

- 会话密钥
- 密钥加密密钥
- 主密钥

越低层的密钥更换越快。仅主密钥需要人工装入，其他各级密钥均可以由密钥管理系统按照某些协议来进行自动地分配、更换、撤销等。

密钥生命周期

分类: 网络信息安全

WittPeng
关注 - 0
粉丝 - 5
+加关注

« 上一篇： 各种内排序算法的实现及性能的比较
» 下一篇： 计算机通信与网络

posted @ 2018-06-10 22:17 WittPeng 阅读(5512) 评论(0) 编辑 收藏

· DXO预告iPhone 12 Pro评测：接连艾特库克 未获回应
· 比特币价格三年来首次突破16000美元 总市值超过工行
» 更多新闻...