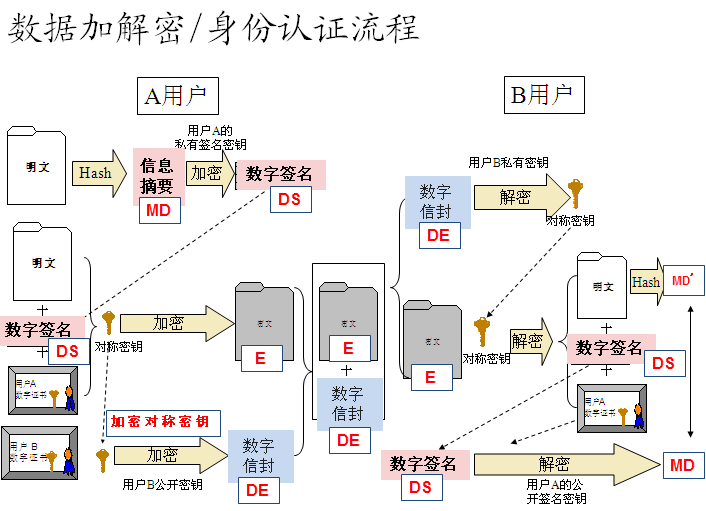
1. 数据验证流程：

****

1. 明文摘要数字签名
2. 明文、数字签名、发送方公钥密文
3. 对称密钥数字信封
4. 发送方将密文、数字信封发给接收方
5. 数字信封对称密钥
6. 密文明文、数字签名、发送方公钥
7. 数学签名摘要1
8. 明文摘要2
9. 验证“摘要1=?摘要2”来确定原文是否被修改
10. 公钥和私钥的说明：
11. 在非对称秘钥下出现
12. 公钥和私钥成对出现
13. 公钥私钥都能够加密/解密
14. 公开的密钥叫公钥，只有自己知道的叫私钥
15. 用公钥加密的数据只有对应的私钥可以解密
16. 用私钥加密的数据只有对应的公钥可以解密
17. 例子：
18. 景氏替换解密算法（对称密钥）：

明文：good good study, day day up.

密钥：google

景氏替换加密算法：

将明文中的所有的字母“d”替换成密钥

密文：

googoogle googoogle stugoogley, googleay googleay up.

密钥：google（加密解密为同一个秘钥）

景氏替换解密算法：

将密文中的所有与密钥相同的字符串换成“d”

明文：good good study, day day up.

1. WinRAR的解压密码（对称密钥）
2. 借条（非对称密钥）：

发送方用发送方私钥对明文加密得到密文，接收方用数字证书认证中心网站上下载的公钥将密文解密得到明文

* 防抵赖：这个能用发送方公钥解密的文件一定是由发送方私钥加密的，可用于证明数字签名的法律效应
* 防篡改：接收方虽然能用发送方公钥解密文件，但由于没有发送方私钥所以没法篡改明文加密得到新密文

1. 身份确认（非对称密钥）：

A在键盘上随机敲一段文字T

B用自己的私钥加密T变为S发还给A

A用B的公钥解密S，如果是T则能确认对方身份确为A

p.s.：如果T不是随机而是固定的，那么会被他人模仿

1. RSA算法（最常用的一种非对称加密算法）：

1977年，由三位数学家Rivest、Shamir 和 Adleman 设计

1. 欧拉函数：1~n中与n形成互质关系的数的个数

e.g.：（有且只有1,3,5,7与8互质）

设因式分解

则

【①若

pf：Step1：在下存在唯一解

证：∵

∴

【p.s.：这样的求法：辗转相除

，

，

从上向下迭代即可求出】

∴是一个解

若都是方程组的解，则

∴ ∴在下有唯一解

Step2：不同必定不同

证：∵

Step3：若与互质，则

分别与互质

②与不互质的元素只有】

1. 欧拉定理：设两个正整数a和n互质，则

* 证明：

设为1~n中所有与n互质的数，则由于a和n互质：。

又∵

∴

∴

由于中所有元素与n互质，

故等式两边可以消去：

* 应用：求的个位数

解：由于7和10互质，所以

所以所以的个位数为1

1. 费马小定理：，p：素数

【欧拉定理推论】

1. 模反元素：

设两个正整数a和n互质，则一定可以找到整数b，使得，这时，b叫做a的“模反元素”

【欧拉定理保证b存在：，

但不唯一，】

1. 算法：
2. 随机选择两个不相等的质数（质数越大越难破解）
3. 求，的二进制位数就是密钥长度，RSA密钥一般为1024位，重要场合2048位
4. 计算
5. 随机选择一个整数，实际应用中常常选择65537
6. 计算对于的模反元素，即

【用上述1)/①/Step1可求出一个解】

1. 公钥：；私钥：
2. 安全性分析：

已知公钥后只有知道的质因数分解才能算出从而算出从而结合算出，但对极大整数做因数分解非常困难，除了暴力破解没有什么有效方法

1. 公钥加密：

设加密信息为，计算，发送密文

（其中必须为小于整数，不是的用ASCII/UNICODE）

1. 私钥解密：

计算，得到原信息

pf：要证

∵

∴

∴只要证明即可

即证即可

∵

∴

∴只要证明即可

①

由欧拉定理：

∴

②

∵

∴ /

∴以为例，由费马小定理：

∴

由

∴

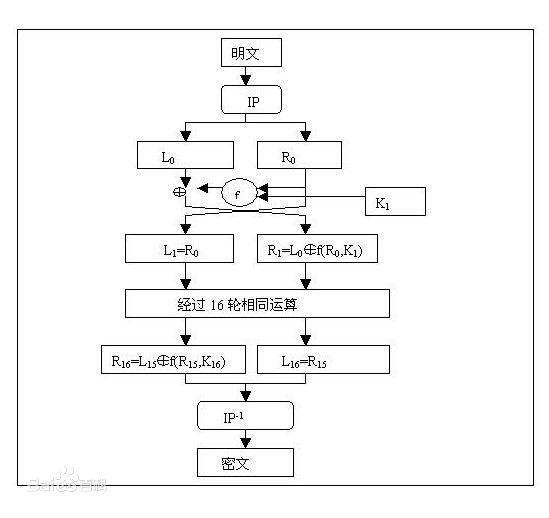
注意此时可以设：

∴（∵，）

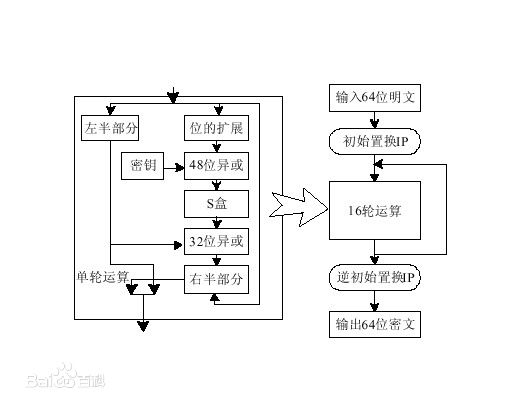
1. DES算法（最常用的一种对称加密算法）：

C++代码：<http://www.2cto.com/kf/201207/138943.html>

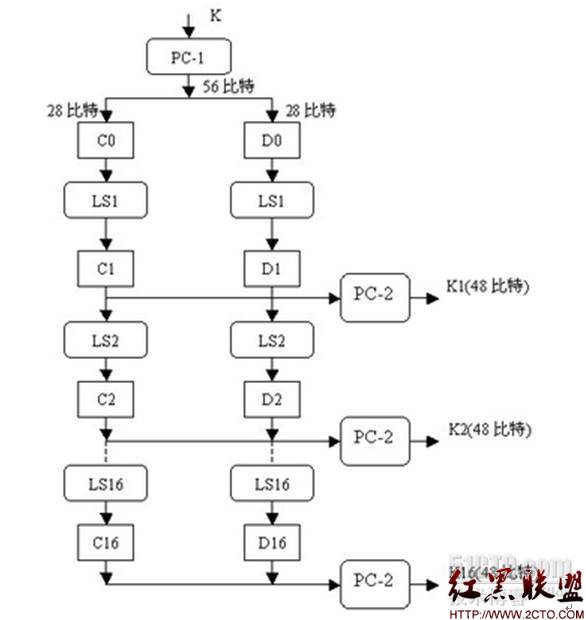
1. Data Encryption Standard数据加密标准
2. 算法流程图：



（总流程）



（总流程中的f函数）



（子密钥生成方式）

1. 算法：

把64位明文用64位的密钥转化为64位密文

1. 将明文D1,D2,D3,......,D64位按照**IP**

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

的顺序变为**L0**=D58D50D42...D8,**R0**=D57D49D41...D7

p.s.：对应的逆置换应为**IP-1**

40,8,48,16,56,24,64,32,39,7,47,15,55,23,63,31,

38,6,46,14,54,22,62,30,37,5,45,13,53,21,61,29,

36,4,44,12,52,20,60,28,35,3,43,11,51,19,59,27,

34,2,42,10,50,18,58 26,33,1,41,9,49,17,57,25

验证：将1→58→1，16→4→16

1. 放大换位表：将R0的32位扩展为48位

32,1,2,3,4,5,4,5,6,7,8,9,8,9,10,11,

12,13,12,13,14,15,16,17,16,17,18,19,20,21,20,21,

22,23,24,25,24,25,26,27,28,29,28,29,30,31,32,1

1. 与48位子密钥**K1**做异或运算

子密钥K1生成方式：

1. PC1选位表：将64位密钥变为56位

57,49,41,33,25,17,9,1,58,50,42,34,26,18,

10,2,59,51,43,35,27,19,11,3,60,52,44,36,

63,55,47,39,31,23,15,7,62,54,46,38,30,22,

14,6,61,53,45,37,29,21,13,5,28,20,12,4

* 缺少的8,16,24,...,64的密钥作为奇偶校验位： 补1/0使每8位中1的个数都为奇数个

—— 奇校验位/偶数个——偶校验位

* A只做1次**K2**在K1基础上做

1. 将剩下56位分成两个28位半密钥，每个半密钥根据左移参数表“1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1”被左移1/2位（取决于这是第几次ii~v操作）

（左移“<<”截断超出部分，右端不足部分补0）

1. PC2选位表：将每个28位半密钥变为24位

14,17,11,24,1,5,3,28,15,6,21,10,

23,19,12,4,26,8,16,7,27,20,13,2,

41,52,31,37,47,55,30,40,51,45,33,48,

44,49,39,56,34,53,46,42,50,36,29,32

1. 将48位结果分为8组，每组6位，利用S盒将每组变为4位（即利用S盒将48位缩小回32位）

S1:

14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7,

0,15,7,4,14,2,13,1,10,6,12,11,9,5,3,8,

4,1,14,8,13,6,2,11,15,12,9,7,3,10,5,0,

15,12,8,2,4,9,1,7,5,11,3,14,10,0,6,13,

S2:

　　15,1,8,14,6,11,3,4,9,7,2,13,12,0,5,10,

　　3,13,4,7,15,2,8,14,12,0,1,10,6,9,11,5,

　　0,14,7,11,10,4,13,1,5,8,12,6,9,3,2,15,

　　13,8,10,1,3,15,4,2,11,6,7,12,0,5,14,9,

S3:

　　10,0,9,14,6,3,15,5,1,13,12,7,11,4,2,8,

　　13,7,0,9,3,4,6,10,2,8,5,14,12,11,15,1,

　　13,6,4,9,8,15,3,0,11,1,2,12,5,10,14,7,

　　1,10,13,0,6,9,8,7,4,15,14,3,11,5,2,12,

S4:

　　7,13,14,3,0,6,9,10,1,2,8,5,11,12,4,15,

　　13,8,11,5,6,15,0,3,4,7,2,12,1,10,14,9,

　　10,6,9,0,12,11,7,13,15,1,3,14,5,2,8,4,

　　3,15,0,6,10,1,13,8,9,4,5,11,12,7,2,14,

S5:

　　2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9,

　　14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6,

　　4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14,

　　11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3,

S6:

　　12,1,10,15,9,2,6,8,0,13,3,4,14,7,5,11,

　　10,15,4,2,7,12,9,5,6,1,13,14,0,11,3,8,

　　9,14,15,5,2,8,12,3,7,0,4,10,1,13,11,6,

　　4,3,2,12,9,5,15,10,11,14,1,7,6,0,8,13,

S7:

　　4,11,2,14,15,0,8,13,3,12,9,7,5,10,6,1,

　　13,0,11,7,4,9,1,10,14,3,5,12,2,15,8,6,

　　1,4,11,13,12,3,7,14,10,15,6,8,0,5,9,2,

　　6,11,13,8,1,4,10,7,9,5,0,15,14,2,3,12,

S8:

　　13,2,8,4,6,15,11,1,10,9,3,14,5,0,12,7,

　　1,15,13,8,10,3,7,4,12,5,6,11,0,14,9,2,

　　7,11,4,1,9,12,14,2,0,6,10,13,15,3,5,8,

　　2,1,14,7,4,10,8,13,15,12,9,0,3,5,6,11,

【具体方法】设6位数为b1b2b3b4b5b6，将b1b6换算成十进制数作为行标，b2b3b4b5换算成十进制数作为列标，（下标从0开始计数）在S表里找到对应一个0~15的数字，再转化为一个4位二进制数

【e.g.】以S2为例，输入为111010，则在S2中找第(10)2=2行，第(1101)2=13列的数，为3，转化回4位二进制数为0011

1. **L1**=R0,**R1**=L0与iv中结果做异或运算
2. 重复ii~v直到**L16**,**R16**，做逆置换IP-1得到密文