《密码学》习题&重点

判断题 10%　10 题

论述题 40%　 4 题

计算题 30%　 3 题

综合题 20%　 1 题

第 1 章 引言

1、安全威胁

(1)被动攻击

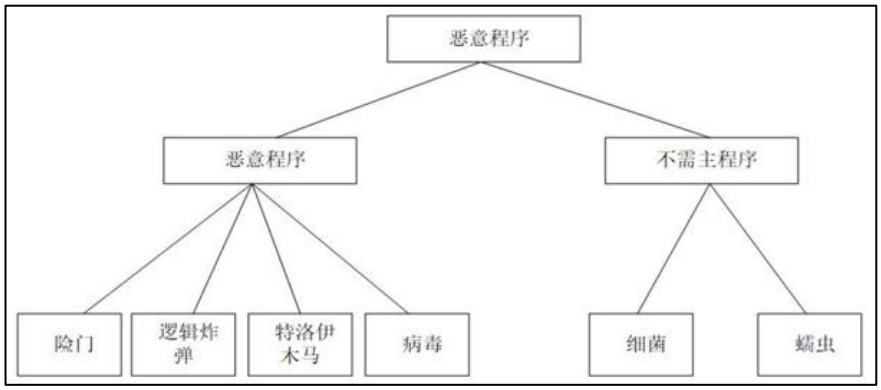
- 被动攻击即窃听：获取消息的内容、业务流分析
- 被动攻击不对消息做任何修改，难以检测的，抗击重点在于**预防而非检测**

(2)主动攻击

- 主动攻击即篡改数据流或产生假数据流：中断、篡改、伪造
- 抗击主动攻击的主要途径是**检测和恢复**

2、入侵者和病毒

恶意软件指病毒、蠕虫等恶意程序。



非法入侵的目的是非法窃取系统资源，对数据进行未授权的修改或破坏计算机系统。

3、安全业务：

(1)保密业务：保护数据以防被动攻击。

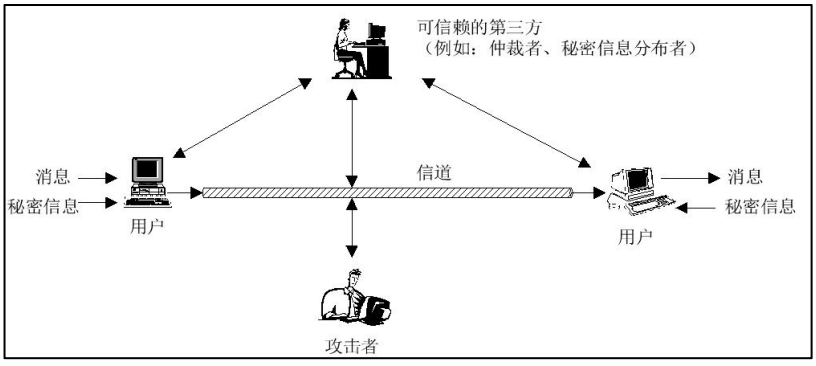
(2)认证业务：用于保证通信的真实性。

(3)完整性业务：保证所接收的消息未经复制、插入、篡改、重排或重放，与发出的消息完全一致。

(4)不可否认业务：用于防止通信双方中的某一方对所传输消息的否认。

(5)访问控制：防止对网络资源的非授权访问，控制的实现方式是认证。

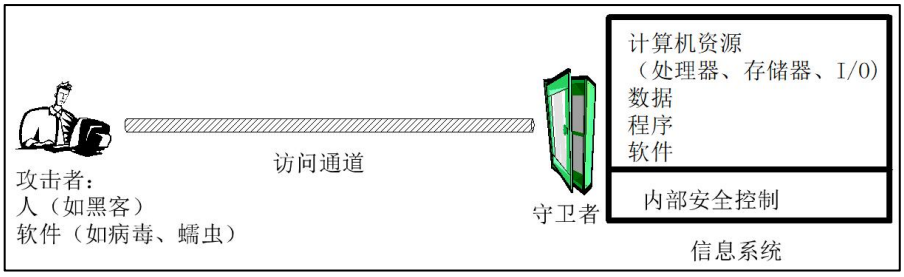
4、信息安全基本模型



通信双方欲传递某个消息，需建立一个逻辑上的信息通道——首先在网络中定义从发方到收方的一个路由，然后在该路由上共同执行通信协议。

(1)安全传输技术的成分：
• 消息的安全传输：包括对消息的加密和认证。
• 通信双方共享的某些秘密信息，如加密密钥。
(2)安全的网络通信必须考虑以下 4 个方面：
• 加密算法；
• 用于加密算法的秘密信息；
• 秘密信息的分布和共享；
• 使用加密算法和秘密信息以获得安全服务所需的协议。

5、信息系统的保护模型



对付未授权访问的安全机制可分为两道防线：
(1)守卫者，用于拒绝非授权用户的访问、检测和拒绝病毒。
(2)内部安全控制，用于管理系统内部的各项操作和分析所存有的信息，以检查是否有未授权的入侵者。
注：
信息安全又分为系统安全（包括操作系统的安全、数据库系统的安全等）、数据安全（包括数据的安全存储、安全传输）和内容安全（包括病毒的防护、不良内容的过滤等）三个层次。

6、保密通信系统
(1)密码系统主要包括以下几个基本要素：明文，密文，加密算法，解密算法和密钥。
(2)通信双方采用保密通信系统可以**隐蔽**和**保护**需要发送的消息，使未授权者不能提取信息。
(3)为了保护信息的保密性，抗击密码分析，保密系统应当满足下述 4 个要求：
• 系统即使达不到理论上是不可破的，也应当为实际上不可破的。从截获的密文或某些已知明文密文对，要决定密钥或任意明文在计算上是不可行的
• 系统的保密性不依赖于对加密体制或算法的保密，而依赖于密钥(Kerckhoff 原则)
• 加密和解密算法适用于密钥空间中的所有元素
• 系统便于实现和使用
(4)密钥体制分类：单钥体制，双钥体制
• 单钥体制：**加密密钥和解密密钥相同**，高保密性，强安全性，低费用芯片
(用途：数据加密、消息认证；方式：流密码、分组密码)
• 双钥体制(又称公钥体制)：将加密和解密能力分开
(用途：保密——多人加密一人解读，认证——一人加密多人解读)
注：系统的保密性取决于**密钥**的安全性，与算法的保密性无关，即由密文和加解密算法不可能得到明文。
(5)密钥攻击类型：
唯密文攻击(最困难、最易抵抗)、已知明文攻击、选择明文攻击 、选择密文攻击

(6)加密算法只要满足以下两条准则之一就称为是**计算上安全**的：

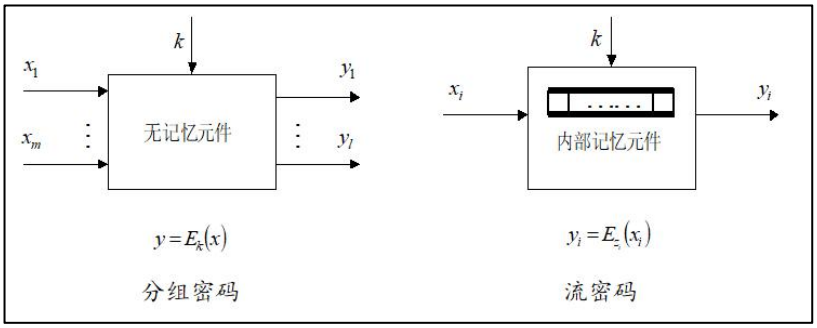
- 破译密文的代价超过被加密信息的价值。
- 破译密文所花的时间超过信息的有用期。

第 2 章 流密码

1、流密码的基本概念

• 基本思想：利用密钥和记忆元件的存储状态产生的密钥流对明文串加密。

• 分组密码与流密码的区别就在于有无记忆性。



(1)同步流密码

——加密器中的记忆元件的存储状态独立于明文字符的流密码(否则为自同步流密码)

因为此刻密文字符不依赖与此前明文字符，所以将同步流密码的加密器分成密钥流产生器和加密变换器(变换可逆)。

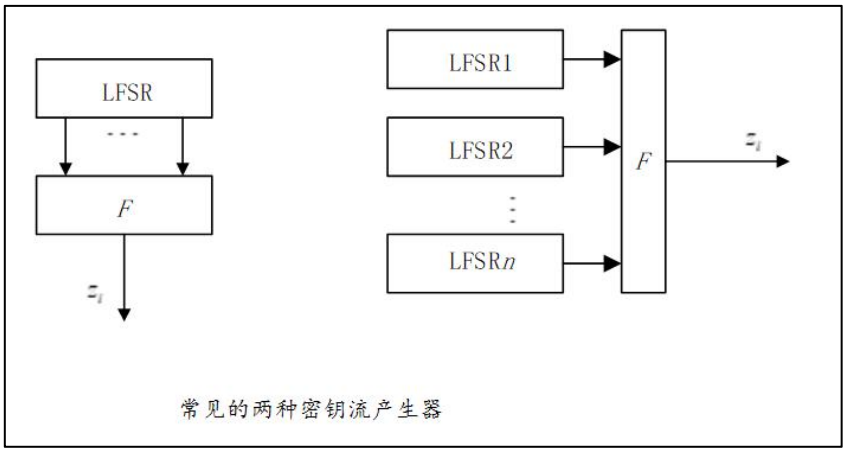
(2)密钥流产生器

关键：采用线性的状态转移函数和非线性的输出函数(非线性的状态转移函数的有限状态自动机理论不完善)

• 驱动部分控制生成器的状态转移，并为非线性组合部分提供统计性能好的序列

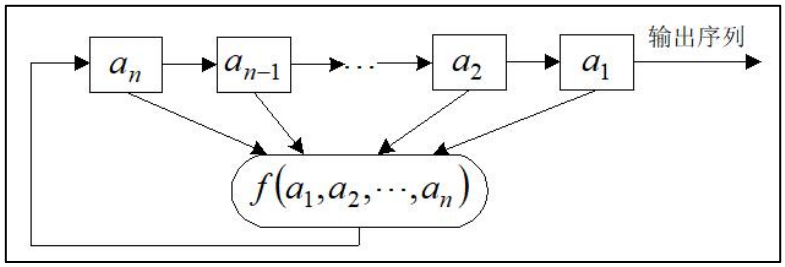
• 非线性组合部分利用序列组合出满足要求的密钥流序列

注：目前流行且实用的密钥流产生器的驱动部分是一个或多个线性反馈移位寄存器。



2、线性反馈移位寄存器

一个 n 级反馈移位寄存器由 n 个二元存储器与一个反馈函数组成。其状态周期等于输出序列周期，均不大于 2^n-1。周期能达到最大值的序列被称为 m 序列。



3、线性反馈移位寄存器的一元多项式表示

设n级线性移位寄存器的输出序列$\{a_i\}$满足递推关系

$$a_{n+k}=c_1a_{n+k-1}\oplus c_2a_{n+k-2}\oplus\cdots\oplus c_na_k \qquad (*)$$

对任何$k\geq 1$成立。这种递推关系可用一个一元高次多项式

$$p(x)=1+c_1x+\cdots+c_{n-1}x^{n-1}+c_nx^n$$

表示，称这个多项式为LFSR的特征多项式或特征多项式。

定义 2-2 设$p(x)$是$GF(2)$上的多项式，使$p(x)|(x^p-1)$的最小p称为$p(x)$的周期或阶。

定理 2-4 设$p(x)$是n次不可约多项式，周期为m，序列$\{a_i\}\in G(p(x))$，则$\{a_i\}$的周期为m。

定理 2-5 n级$LFSR$产生的序列有最大周期2^n-1的必要条件是其特征多项式为不可约的。

定义 2-4 若n次不可约多项式$p(x)$的阶为2^n-1，则成$p(x)$是n次本原多项式。

定义 2-6 设$\{a_i\}\in G(p(x))$，$\{a_i\}$为m序列的充要条件是$p(x)$为本原多项式。

4、非线性序列

(1)*J-K* 触发器

在 *J-K* 触发器的非线性序列生成器中，令驱动序列$\{a_k\}$和$\{b_k\}$分别为m级和n级m序列，则有

$$c_k=\overline{(a_k+b_k)}c_{k-1}+a_k=(a_k+b_k+1)c_{k-1}+a_k$$

即

$$c_k=\begin{cases}a_k, & c_{k-1}=0\\ \overline{b_k}, & c_{k-1}=1\end{cases}$$

当m和n互素并且$a_0+b_0=1$时，$\{c_k\}$序列的周期为$(2^m-1)(2^n-1)$。

(2)钟控序列生成器

假设 *LFSR1* 和 *LFSR2* 分别输出序列$\{a_k\}$和$\{b_k\}$，其周期分别为p_1和p_2。

• 当 *LFSR1* 输出 1 时，移位时钟脉冲通过与门使 LFSR2 进行一次移位，从而生成下一位。

• 当 LFSR1 输出 0 时，移位时钟脉冲无法通过与门影响 LFSR2。因此 LFSR2 重复输出前一位。

假设钟控序列$\{c_k\}$的周期为p，可得如下关系：

$$p=\frac{p_1p_2}{\gcd(w_1,p_2)}$$

其中，

$$w_1=\sum_{i=0}^{p_1-1}a_i$$

表示序列$\{a_k\}$一个周期中 1 出现的次数。

习题：

1、3 级线性反馈移位寄存器在 $c_3=1$ 时可有 4 种线性反馈函数，设其初始状态为 $(a_1,a_2,a_3)=(1,0,1)$，求各线性反馈函数的输出序列及周期。

解：设 3 级线性反馈特征多项式为 $p(x)=1+c_1x+c_2x^2+c_3x^3$，若 $c_3=1$ 则 c_1,c_2 共有 $2^2=4$ 种可能，对应初态 $(a_1,a_2,a_3)=(1,0,1)$。4 种线性反馈函数分别记为：

$p_1(x)=1+x^3$ 输出序列 $a=101101101\cdots$，周期 $p=3$

$p_2(x)=1+x+x^3$ 输出序列 $a=1010011101\cdots$，周期 $p=7$

$p_3(x)=1+x^2+x^3$ 输出序列 $a=1011100101\cdots$，周期 $p=7$

$p_4(x)=1+x+x^2+x^3$ 输出序列 $a=101010\cdots$，得周期 $p=2$

2、设 $n=4$，$f(a_1,a_2,a_3,a_4)=a_1\oplus a_4\oplus 1\oplus a_2a_3$，初始状态为 $(a_1,a_2,a_3,a_4)=(1,1,0,1)$，求此非线性反馈移位寄存器的输出序列及周期。

解：由 $f(a_1,a_2,a_3,a_4)=a_1\oplus a_4\oplus 1\oplus a_2a_3$，初态为 $(a_1,a_2,a_3,a_4)=(1,1,0,1)$。线性递归可得：

$a_5=1\oplus 1\oplus 1\oplus 0=1$

$a_6=1\oplus 1\oplus 1\oplus 0=1$

$a_7=0\oplus 1\oplus 1\oplus 1=1$

$a_8=1\oplus 1\oplus 1\oplus 1=0$

$a_9=1\oplus 0\oplus 1\oplus 1=1$

$a_{10}=1\oplus 1\oplus 1\oplus 0=1$

可以得到输出序列为 $(1101111011\cdots)$，周期为 $p=5$。

3、设 J-K 触发器中$\{a_k\}$和$\{b_k\}$分别为 3 级和 4 级 m 序列，且$\{a_k\}=11101001110100\cdots$，$\{b_k\}=0010110110110000010110110110000\cdots$求输出序列$\{c_k\}$及周期。

解：由 J-K 触发器可知$c_k=(a_k+b_k+1)c_{k-1}+a_k$

$$c_k=\begin{cases}a_k, c_{k-1}=0\\ \overline{b_k}, c_{k-1}=1\end{cases}$$

此时$\{a_k\}$和$\{b_k\}$分别为 3 级和 4 级 m 序列，$(3,4)=1$，且$a_0+b_0=1$，则$\{c_k\}$的周期为$(2^3-1)(2^4-1)=7\times15=105$。

$\{c_k\}=11001001010100\cdots$。

4、设基本钟控序列生成器中$\{a_k\}$和$\{b_k\}$分别为 2 级和 3 级 m 序列，且$\{a_k\}=101101\cdots$，$\{b_k\}=10011011001101\cdots$求输出序列$\{c_k\}$及周期。

解：令基本钟控序列生成器中$\{a_k\}$的周期为p_1，$\{b_k\}$的周期为p_2，则输出序列$\{c_k\}$的周期为$p=\frac{p_1p_2}{\gcd(w_1,p_2)}$，$w_1=\sum_{i=0}^{p_1-1}a_i=2$，$p_1=2^2-1=3$，$p_2=2^3-1=7\Rightarrow p=\frac{3\times7}{\gcd(2,7)}=21$。

记 LFSR2 产生$\{b_k\}$，其状态向量为σ_k，可得σ_k的变化情况如下：

$\sigma_0\sigma_1\sigma_1\sigma_2\sigma_3\sigma_3\sigma_4\sigma_5\sigma_5\sigma_6\sigma_0\sigma_0\sigma_1\sigma_2\sigma_2\sigma_3\sigma_4\sigma_4\sigma_5\sigma_6\sigma_6\sigma_0\sigma_1\sigma_1\sigma_2$

输出序列$\{c_k\}=10001110011100011101\cdots$

5、设 *LFSR1* 为 3 级 m 序列生成器，其特征多项式为$f_1(x)=1+x+x^3$，设初态为$a_0=a_1=a_2=1$，于是$\{a_k\}$输出序列为{1,1,1,0,1,0,……}。又设 *LFSR2* 为 3 级 m 序列生成器，其特征多项式为$f_2(x)=1+x^2+x^3$，设初态为$b_0=b_1=b_2=1$，于是$\{b_k\}$输出序列为{1,1,1,0,0,1,0,……}。记其状态向量为σ_k，则σ_k的变化情况如下：

σ_0	σ_0	σ_1	σ_2	σ_3	σ_3	σ_4	σ_4	σ_4
σ_5	σ_6	σ_0	σ_0	σ_1	σ_1	σ_1		
σ_2	σ_3	σ_4	σ_4	σ_5	σ_5	σ_5		
σ_0	σ_1	σ_2	σ_2	σ_3	σ_3	σ_3		

所以 $\{c_k\}$ = 1,1,1,0,0,0,0,0,
1,0,1,1,1,1,1,
1,0,0,0,1,1,1,
0,1,1,1,1,1,1,
0,0,1,1,0,0,0,
1,1,1,1,0,0,0,
0,1,0,0,1,…..

第 3 章 分组密码体制

1、分组密码概述

(1)代换

为使加密运算可逆（使解密运算可行），明文的每一个分组都应产生惟一的一个密文分组，这样的变换是可逆的，称明文分组到密文分组的可逆变换为**代换**。不同可逆变换的个数有$2^n!$个(n为明文分组长)。

注：

• 加密映射和解密映射可由代换表来定义，这种定义法是分组密码最常用的形式，能用于定义明文和密文之间的任何可逆映射。

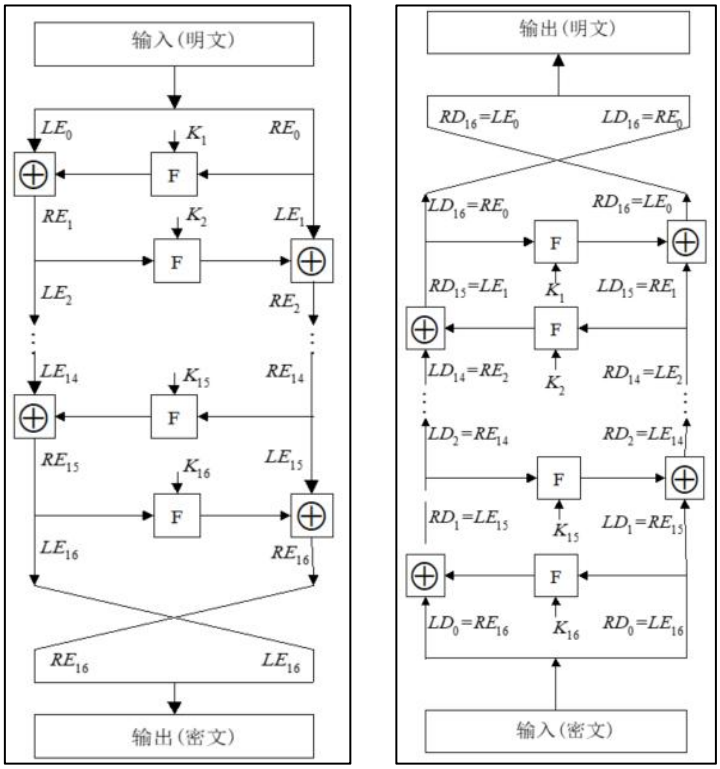
• 如果分组长度太小，如$n=4$，系统则等价于古典的代换密码，容易通过对明文的统计分析而攻破。如果分组长度n足够大，而且从明文到密文可有任意可逆的代换，那么明文的统计特性将被隐藏而使以上的攻击不能奏效。

(2)扩散和混淆——目的是抗击敌手对密码系统的统计分析

• 扩散，就是将明文的统计特性散布到密文中去，使明文和密文之间的统计关系变得尽可能复杂，以使敌手无法得到密钥。实现方式是使得密文中每一位由明文中多位产生。

• 混淆是使密文和密钥之间的统计关系变得尽可能复杂，以使敌手无法得到密钥。

(3)Feistel 密码结构



加密结构

• 加密算法的输入是分组长为$2w$的明文和一个密钥K_0。将每组明文分成左右两半LE_0和RE_0，在进行完n轮迭代后，左右两半再合并到一起以产生密文分组。其第i轮迭代的输入为前轮输出的函数：

$$LE_i = RE_{i-1}$$

$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, KE_i)$$

其中，K_i是第i轮用的子密钥，由加密密钥K得到。一般，各轮子密钥彼此不同而且与加

密密钥K不同。

• Feistel 网络中每轮结构都相同，每轮中右半数据被作用于轮函数F后，再与左半数据进行异或运算，这一过程就是**代换**。每轮轮函数的结构都相同，但以不同的子密钥K_i作为参数。代换过程完成后，再交换左、右两半数据，这一过程称为**置换**(代换-置换网络 SPN)。

• 分组越大、密钥越长、轮数越多，安全性越高，加密速度越慢；子密钥产生算法越复杂、轮函数复杂性越高，密码分析越困难。

解密结构

• 解密过程本质上和加密过程一样，算法使用密文作为输入，但使用子密钥K_i的次序与加密过程相反，即第一轮使用K_n，第二轮使用K_{n-1}，最后一轮使用K_1。保证了**解密和加密可采用同一算法**。

• 解密过程中有：

$$RE_{i-1} = LE_i$$

$$LE_{i-1} = RE_i \oplus F(RE_{i-1}, KE_i) = RE_i \oplus F(LE_i, KE_i)$$

2、分组密码算法 DES

(1)描述——明文分组长为 64 比特，密钥长为 56 比特

• 左边为明文处理：

• 初始置换 IP，用于重排明文分组的 64 比特数据

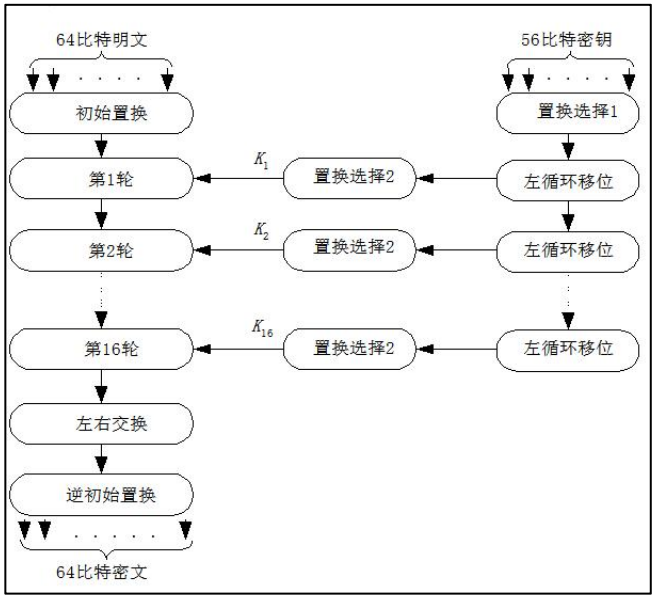
• 具有相同功能的 16 轮变换，每轮中都有置换和代换运算，第 16 轮变换的输出分为左右两半，并被交换次序

注：DES 每轮变换与 Feistel 网络中每轮结构都相同，公式如下：

$$LE_i = RE_{i-1}$$

$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, KE_i)$$

• 最后再经过一个逆初始置换 IP 的逆，从而产生 64 比特的密文



• 右边为密钥产生：

• 输入算法的 56 比特密钥首先经过一个置换运算，然后将置换后的 56 比特分为各为 28 比特的左、右两半，分别记为C_0和D_0，在第i轮分别对C_{i-1}和D_{i-1}进行左循环移位。

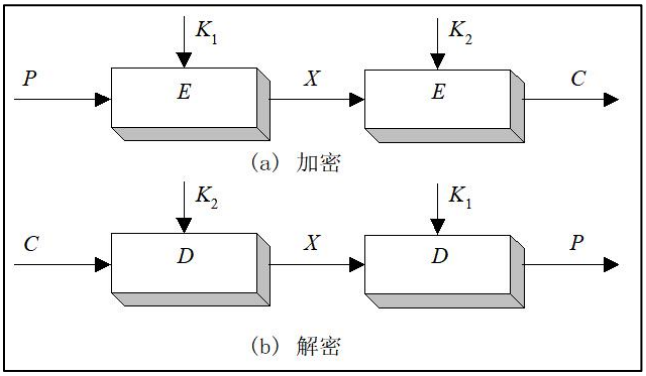
•移位后的结果作为求下一轮子密钥的输入和置换选择 2 的输入，通过置换选择 2 产生的 48 比特的本轮的子密钥K_i，作为函数$F(R_{i-1}, K_i)$的输入。

(2)二重 DES

明文为 P，密文为 C，则

$$C = E_{K_2}[E_{K_1}[P]]$$
$$P = D_{K_1}[D_{K_2}[P]]$$

注：加解密密钥使用顺序相反



(3)三重 DES

• 两个密钥

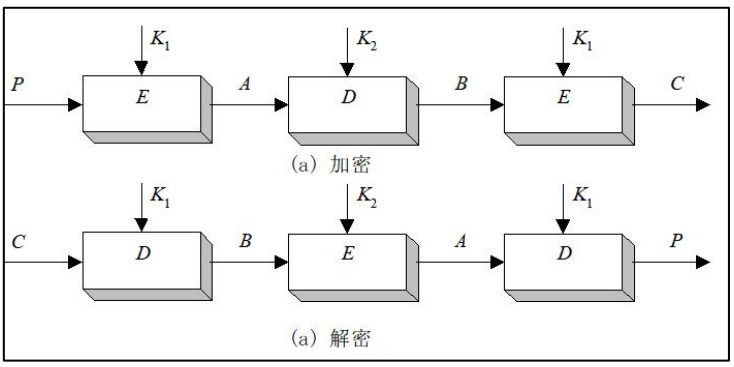
$$C = E_{K_1}[D_{K_2}[E_{K_1}[P]]]$$

注： 第 2 步解密的目的仅在于使得用户可对一重 DES 加密的数据解密

• 三个密钥、

$$C = E_{K_3}[D_{K_2}[E_{K_1}[P]]]$$

密钥长度为 168 比特



3、分组密码的运行模式

模式	描述	用途
电码本 ECB	每个明文组独立地以同一密钥加密	传送短数据(如一个加密密钥)
密码分组 CBC	加密算法的输入是当前明文组与前一密文组的异或	传送数据分组；认证
密码反馈 CFB	每次只处理输人的 j 比特,将上一次的密文用作加密算法的输入以产生伪随机输出，该输出再与当前明文异或以产生当前密文	传送数据流；认证
输出反馈 OFB	与 CFB 类似，不同之处是本次加密算法的输入为前一次加密算法的输出	有扰信道上(如卫星通信)传送数据流

(1)电码本模式 ECB

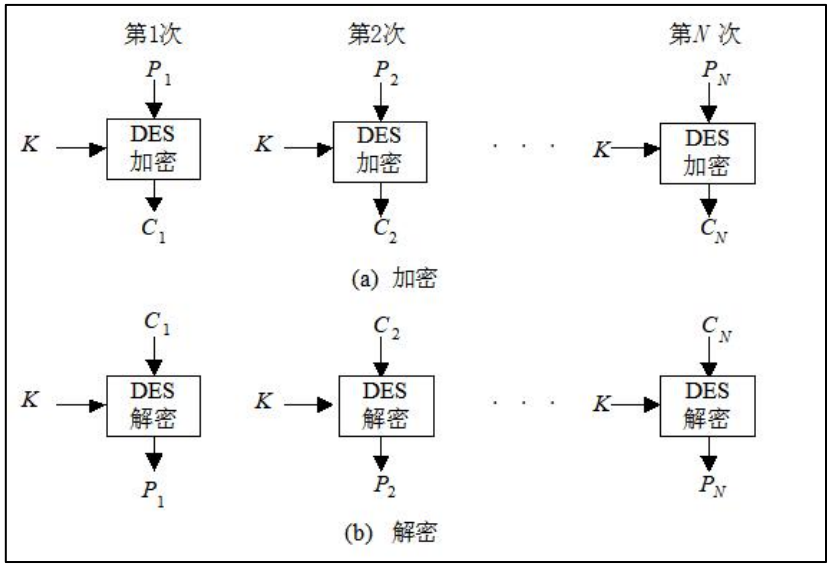
• 一次对一个 64 比特长的明文分组加密，而且每次的加密密钥都相同

• 当密钥取定时，对明文的每一个分组，都有一个惟一的密文与之对应

• ECB 在用于短数据（如加密密钥）时非常理想

• ECB 的最大特性是同一明文分组在消息中重复出现的话，产生的密文分组也相同(缺陷)

注：如果我们需要安全地传递 DES 密钥，ECB 是最合适的模式



(2)密码分组链接模式 CBC

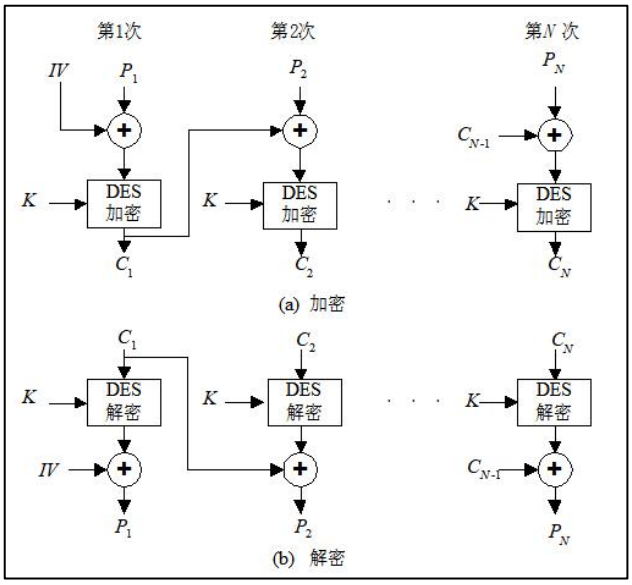
• 一次对一个明文分组加密，每次加密使用同一密钥,加密算法的输入是当前明文分组和前一次密文分组的异或，重复的明文分组不会在密文中暴露出重复关系

• 解密时，每一个密文分组被解密后，再与前一个密文分组异或，即：

$$D_K[C_n] \oplus C_{n-1} = D_K\left[E_K[C_{n-1} \oplus P_n]\right] \oplus C_{n-1}$$
$$= C_{n-1} \oplus P_n \oplus C_{n-1} = P_n \qquad (\text{设 } C_n = E_K[C_{n-1} \oplus P_n])$$

• 数据较长时，一般使用 CBC 模式，CBC 模式对于加密长于 64 比特的消息非常合适

• CBC 模式还能用于完整性认证，验证是否被篡改，但具体位置未知

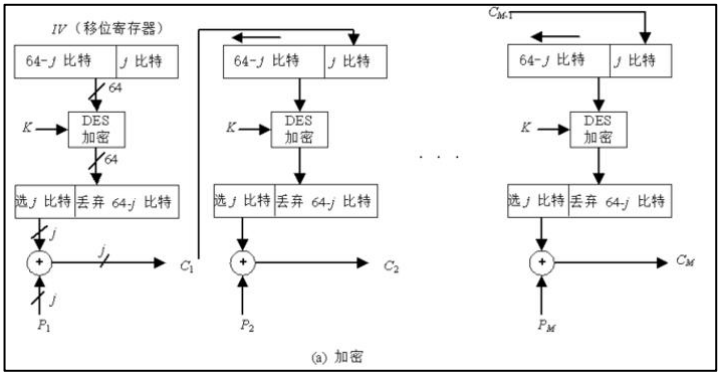


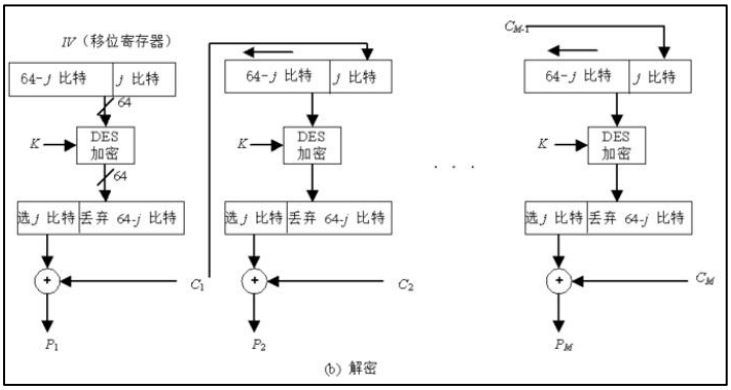
注：

• DES 是分组长为 64 比特的分组密码，但利用 CFB 或 OFB 模式可将 DES 转换为流密码，不需要对消息填充，而且运行是实时的

• 流密码具有密文和明文一样长的性质，如果需要发送的每个字符长为 8 比特，就应使用 8 比特密钥来加密每个字符；如果密钥长超过 8 比特，则造成浪费

(3)密码反馈模式 CFB





• 解密时仍用加密算法

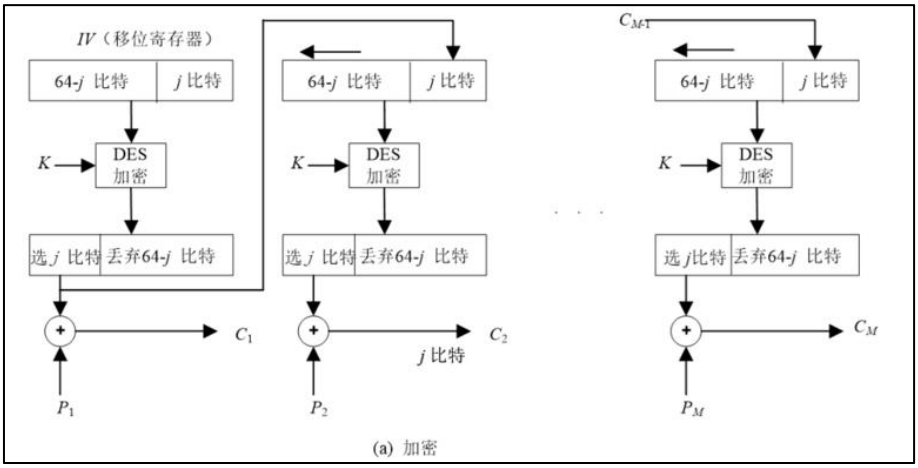
• CFB 模式除了能获得保密性外，还能用于认证

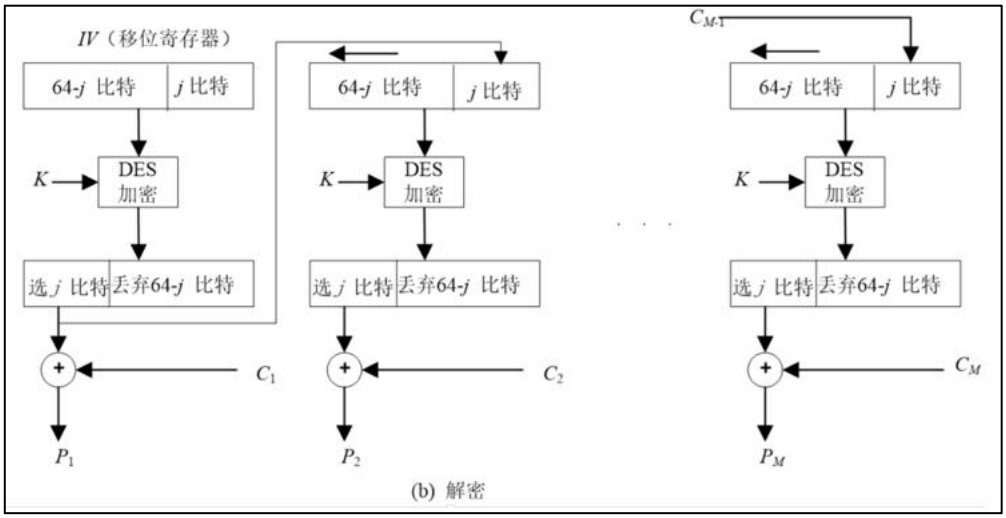
(4)输出反馈模式 OFB

• OFB 模式的结构类似于 CFB，不同之处如下 OFB 模式是将加密算法的输出反馈到移位寄存器，而 CFB 模式中是将密文单元反馈到移位寄存器

• 优点：传输过程中的比特错误不会被传播

• 缺点：比 CFB 模式更易受到对消息流的篡改攻击





4、AES 算法

(1)*Rijndael* 的数学基础和设计思想

***x* 乘法**

$$x \cdot a(x) = x \cdot (a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0)$$

$a_7 = 0$

$$x.a(x) = a_6x^7 + a_5x^6 + a_4x^5 + a_3x^4 + a_2x^3 + a_1x^2 + a_0x$$

$$x \cdot a(x) = (a_6a_5a_4a_3a_2a_1a_0\,0)$$

$a_7 = 1$

$$a_7x^8 \bmod m(x) = x^8 \bmod m(x) = m(x) - x^8 = x^4 + x^3 + x + 1$$

$$x \cdot a(x) = (x^4 + x^3 + x + 1) + (a_6x^7 + a_5x^6 + a_4x^5 + a_3x^4 + a_2x^3 + a_1x^2 + a_0x)$$

$$x \cdot a(x) = (00011011) \oplus (a_6a_5a_4a_3a_2a_1a_0\,0)$$

(2)算法说明

Rijndael 的轮函数是由三个不同的可逆均匀变换组成的，由四个不同的计算部件组成，分别是：字节代换(ByteSub)、行移位(ShiftRow)、列混合(MixColumn)、密钥加(AddRoundKey)

例题 1

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

当（a_0 a_1 a_2 a_3）=（4d 90 4a d8）时，求b_0的值。

解：b_0=02 •4d⊕ 03 •90⊕ 01 •4a⊕ 01 •d8

02 •4d=00000010•01001101=10011010

03 •90=00000011•10010000=00000010•10010000⊕ 00000001•10010000

=00100000⊕ 00011011⊕ 00000001•10010000

=00111011⊕ 10010000

=10101011

b0=10011010⊕ 10101011⊕ 01001010⊕ 11011000=10100011=a3

例题 2

将 *Rijndael* 密码算法某一轮的后两个计算部件和下一轮的前两个计算部件组成组合部件，该组合部件的程序为：

MixColumn (State)；

AddRoundKey (State, Key(i));

ByteSub (State);

ShiftRow (State)

则该组合部件的逆变换程序为：

InvByteSub (State);

InvShiftRow (State);

InvMixColumn (State);

AddRoundKey (State, InvMixColumn (Key(i)))

注： *Rijndael* 密码的解密算法与加密算法的计算网络都相同，只是将各计算部件换为对应的逆部件

第 4 章 公钥密码

1、公钥密码体制

•在公钥密码体制以前的整个密码学史中，所有的密码算法都是基于**代换**和**置换**这两个工具

• 公钥密码体制则为密码学的发展提供了新的理论和技术基础：

• 公钥密码算法的基本工具不再是代换和置换，而是**数学函数**、

•公钥密码算法是以**非对称**的形式使用两个密钥，两个密钥的使用对保密性、密钥分配、认证等都有着深刻的意义

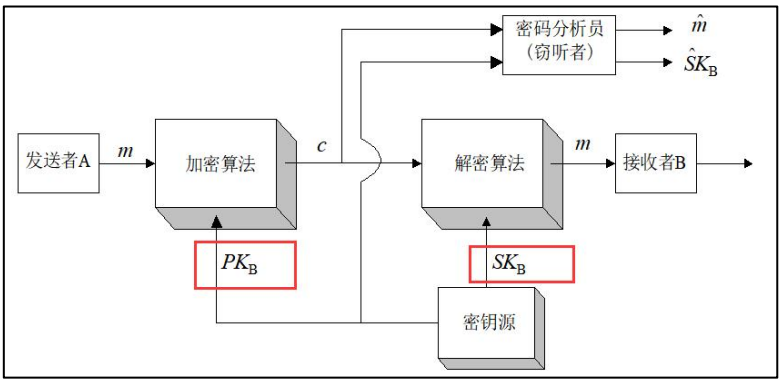
•公钥密码体制的概念是在解决单钥密码体制中最难解决的两个问题时提出的——**密钥分配**和**数字签字**

(1)原理

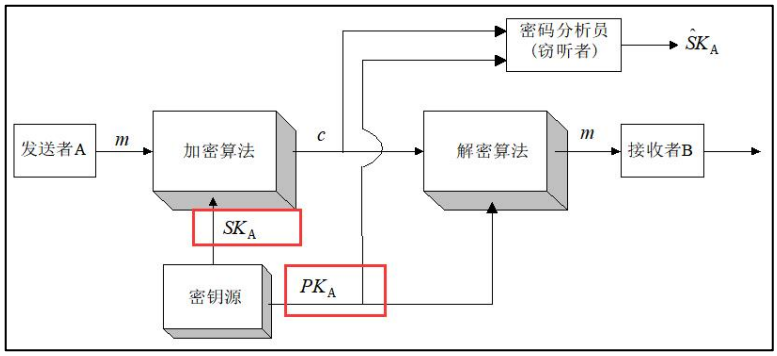
• 公钥密码算法的最大特点：采用两个相关密钥**将加密和解密能力分开**

• 算法有以下重要特性：已知密码算法和加密密钥，求解密密钥在计算上是不可行

• 加密框图：



• 认证框图：

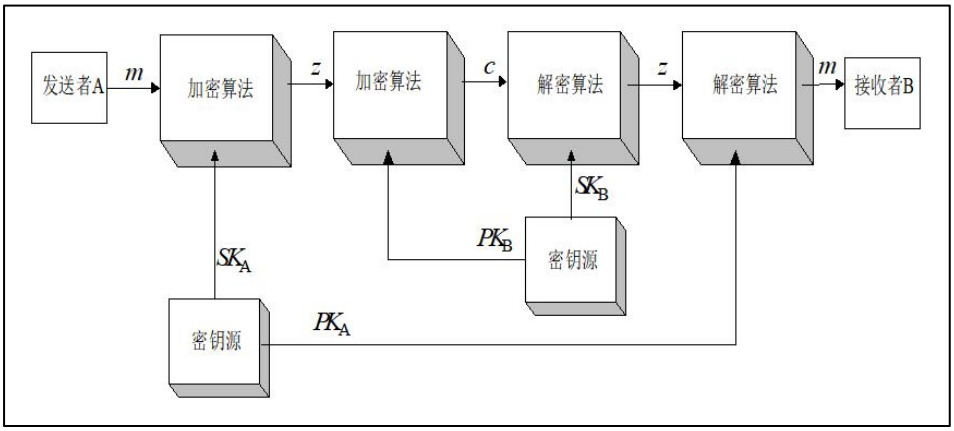


注：

•用户数目很多时，以上认证方法需要很大的存储空间，改进的方法是减小文件的数字签字的大小，即将文件经过一个函数压缩成长度较小的比特串，得到的比特串称为**认证符**

• 认证符具有这样一个性质：如果保持认证符的值不变而修改文件这在计算上是不可行的。用发送者的秘密钥对认证符加密，加密后的结果为原文件的数字签字

• 认证、保密框图：



(2)公钥密码算法应满足以下要求

(1) 接收方B产生密钥对（公开钥PK_B和秘密钥SK_B）是计算上容易的。

(2) 发方A用收方的公开钥对消息 加密以产生密文，即 $c=\mathrm{E}_{PK_B}[m]$ 在计算上是容易的。

(3) 收方B用自己的秘密钥对 c 解密，即 $m=\Delta_{SK_B}[c]$ 在计算上是容易的。

(4) 敌手由B的公开钥 PK_B 求秘密钥 SK_B 在计算上是不可行的。

(5) 敌手由密文 c 和B的公开钥 PK_B 恢复明文 m 在计算上是不可行的。

(6) 加、解密次序可换，即 $\mathrm{E}_{PK_B}[\Delta_{SK_B}(m)]=\Delta_{SK_B}[\mathrm{E}_{PK_B}(m)]$

其中最后一条虽然非常有用，但不是对所有的算法都作要求。

注： 公钥密码体制目前主要用于密钥管理和数字签字

(3)对公钥密码体制的攻击

穷搜索攻击、寻找从公开钥计算秘密钥的方法、可能字攻击

2、背包密码体制

(1)新背包——公开钥

$$b_i \equiv t \cdot a_i \bmod k, \quad i=1,2,\cdots,n$$

$$B \equiv t \cdot A \bmod k$$

(2)对明文分组 $x=(x_1x_2\cdots x_n)$ 加密

$$c=f(x)=B\cdot B_x \equiv t\cdot A\cdot B_x \bmod k$$

(3)解密

$$t^{-1}c \bmod k = AB_x$$

3、椭圆曲线密码体制

(1)椭圆曲线

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

加法法则：

$$P+O=P$$

$$P_1 = (x, y)\ ,\quad P_2 = -P_1 = (x, -y)$$

(2)有限域上的椭圆曲线

$E_p(a,b)$上的加法定义如下：

设$P, Q \in E_p(a,b)$，则

（1）$P + O = P$

（2）如果$P = (x, y)$，那么$(x, y) + (x, -y) = O$，即$(x, -y)$是P的加法逆元，表示为$-P$。

由$E_p(a,b)$的产生方式知，$-P$也是$E_p(a,b)$中的点，如上例，P=（13,7），$-P$=（13, -7），而$-7 \bmod 23 = 16$，所以$-P$=（13, 16），也在$E_{23}(1,1)$中。

（3）设$P = (x_1, y_1)$，$Q = (x_2, y_2)$，$P \neq -Q$，则$P + Q = (x_3, y_3)$由以下规则确定：

$$x_3 \equiv \lambda^2 - x_1 - x_2 (\bmod\ p)$$
$$y_3 \equiv \lambda(x_1 - x_3) - y_1 (\bmod\ p)$$

其中

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & P = Q \end{cases}$$

例题：

仍以$E_{23}(1,1)$为例，设P=(3,10)，Q=（9,7），则

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} \equiv 11 \bmod 23$$
$$x_3 = 11^2 - 3 - 9 = 109 \equiv 17 \bmod 23$$
$$y_3 = 11(3-17) - 10 = -164 \equiv 20 \bmod 23$$

所以P+ Q=（17,20），仍为$E_{23}(1,1)$中的点。

(3)椭圆曲线上的密码

• Diffie-Hellman 密钥交换

两用户 A 和 B 之间的密钥交换如下进行：

（1）A 选一小于 n 的整数 n_A，作为秘密钥，并由 $P_A = n_A G$ 产生 $E_p(a,b)$ 上的一点作为公开钥；

（2）B 类似地选取自己的秘密钥 n_B 和公开钥 P_B；

（3）A、B分别由 $K = n_A P_B$ 和 $K = n_B P_A$ 产生出双方共享的秘密钥。

这是因为 $K = n_A P_B = n_A(n_B G) = n_B(n_A G) = n_B P_A$。

• ElGamal 密码体制

首先选取一条椭圆曲线，并得 $E_p(a,b)$，将明文消息 m 嵌入到曲线上得点 P_m，再对点 P_m 做加密变换。

取 $E_p(a,b)$ 的一个生成元 G，$E_p(a,b)$ 和 G 作为公开参数。

用户 A 选 n_A 作为秘密钥，并以 $P_A = n_A G$ 作为公开钥。任一用户 B 若想向 A 发送消息，可选取一随机正整数 k，产生以下点对作为密文：

$$C_m = \{kG, P_m + kP_A\}$$

A 解密时，以密文点对中的第二个点减去用自己的秘密钥与第一个点的倍乘，即

$$P_m + kP_A - n_A kG = P_m + k(n_A G) - n_A kG = P_m$$

4、椭圆曲线公钥密码加密算法

(1)密钥产生

设接收方为 B，B 的秘密钥取为 $\{1,2,\cdots,n-1\}$ 中的一个随机数 d_B，记为 $d_B \leftarrow_R \{1,2,\cdots,n-1\}$，其中 n 是基点 G 的阶。

B 的公开钥取为椭圆曲线上的点：

$$P_B = d_B G$$

其中 $G = G(x,y)$ 是基点。

(2)加密算法

设发送方是 A，A 要发送的消息表示成比特串 M，M 的长度为 $klen$。加密运算如下：

（1）选择随机数 $k \leftarrow_R \{1,2,\cdots,n-1\}$；

（2）计算椭圆曲线点 $C_1 = kG = (x_1, y_1)$，将 (x_1, y_1) 表示为比特串；

（3）计算椭圆曲线点 $S = hP_B$，若 S 是无穷远点，则报错并退出；

（4）计算椭圆曲线点 $kP_B = (x_2, y_2)$，将 (x_2, y_2) 表示为比特串；

（5）计算 $t = KDF(x_2 \| y_2, klen)$，若 t 为全 0 的比特串，则返回（1）；

（6）计算 $C_2 = M \oplus t$；

（7）计算 $C_3 = Hash(x_2 \| M \| y_2)$；

（8）输出密文 $C = (C_1, C_2, C_3)$。

(3)解密算法

（1）从 C 中取出比特串 C_1，将 C_1 表示为椭圆曲线上的点，验证 C_1 是否满足椭圆曲线方程，若不满足则报错并退出；

（2）计算椭圆曲线点 $S = hC_1$，若 S 是无穷远点，则报错并退出；

（3）计算 $d_B C_1 = (x_2, y_2)$，将坐标 x_2，y_2 表示为比特串；

（4）计算 $t = KDF(x_2 \| y_2, klen)$，若 t 为全0比特串，则报错并退出；

（5）从 C 中取出比特串 C_2，计算 $M' = C_2 \oplus t$；

（6）计算 $u = Hash(x_2 \| M' \| y_2)$，从 C 中取出 C_3，若 $u \neq C_3$，则报错并退出；

（7）输出明文 M'。

(4)证明

解密的正确性：

因为 $P_B = d_B G$，$C_1 = kG = (x_1, y_1)$，由解密算法的第（3）步可得

$$d_B C_1 = d_B kG = k(d_B G) = kP_B = (x_2, y_2)$$

所以解密算法第（4）步得到的 t 与加密算法第（5）步得到的 t 相等，由 $C_2 \oplus t$，便得到明文。

习题：

1、设通信双方使用 RSA 加密体制，接收方的公开钥是 $(e,n)=(5,35)$，接收到的密文是 $C=10$，求明文 M。

解：因为 $n=35=5\times 7\Rightarrow p=5,q=7$，则 $\varphi(n)=(p-1)(q-1)=4\times 6=24$，所以 $d\equiv e^{-1}\bmod\varphi(n)\equiv 5^{-1}\bmod 24\equiv 5\bmod 24$，即明文 $M\equiv C^d\bmod n\equiv 10^5\bmod 35\equiv 5$。

2、已知 $c^d\bmod n$ 的运行时间是 $O(\log^3 n)$，用中国剩余定理改进 RSA 的解密运算。如果不考虑中国剩余定理的计算代价，证明改进后的解密运算速度是原解密运算速度的 4 倍。

证明：RSA 的两个大素因子 p,q 的长度近似相等，约为模数 n 的比特长度 $\log n$ 的一半，即 $(\log n)/2$，而在中国剩余定理中，需要对模 p 和模 q 进行模指数运算，这与 $c^d\bmod n$ 的运行时间规律相似，所以每一个模指数运算的运行时间仍然是其模长的三次幂，即 $O[(\log n/2)^3]=O(\log^3 n)/8$。

在不考虑中国剩余定理计算代价的情况下，RSA 解密运算的总运行时间为两个模指数运算的运行时间之和，即 $O(\log^3 n)/8+O(\log^3 n)/8=O(\log^3 n)/4$，证得改进后的解密运算速度是原解密运算速度的 4 倍。

3、设 RSA 加密体制的公开钥是 $(e,n)=(77,221)$

(1) 用重复平方法加密明文 160，得中间结果为：

$$160^2\pmod{221}\equiv 185$$
$$160^4\pmod{221}\equiv 191$$
$$160^8\pmod{221}\equiv 16$$
$$160^{16}\pmod{221}\equiv 35$$
$$160^{32}\pmod{221}\equiv 120$$
$$160^{64}\pmod{221}\equiv 35$$
$$160^{72}\pmod{221}\equiv 118$$
$$160^{76}\pmod{221}\equiv 217$$
$$160^{77}\pmod{221}\equiv 23$$

若敌手得到以上中间结果就很容易分解 n，问敌手如何分解 n。

(2) 求解密密钥 d。

解：(1)由以上中间结果可得：

$160^{16}(\mathrm{mod}\,221)\equiv 35\equiv 160^{64}(\mathrm{mod}\,221)$

$\Rightarrow 160^{64}-160^{16}\equiv 0(\mathrm{mod}\,221)$

$\Rightarrow (160^{32}-160^{8})(160^{32}+160^{8})\equiv 0(\mathrm{mod}\,221)$ 。

$\Rightarrow (120-16)(120+16)\equiv 0(\mathrm{mod}\,221)$

$\Rightarrow 104\times 136\equiv 0(\mathrm{mod}\,221)$

由 $\gcd(104,221)=13,\gcd(136,221)=17$，可知分解为 $221=13\times 17$ 。

(2)解密密钥 $d=e^{-1}\bmod(\varphi(n))=77^{-1}\bmod(\varphi(13\times 17))=77^{-1}\bmod(12\times 16)$ ，由扩展的 Eucild 算法可得 $d=5$ 。

4、设背包密码系统的超递增序列为 $(3,4,9,17,35)$，乘数 $t=19$，模数 $k=73$，试对 good night 加密。

解：由 $A=(3,4,9,17,35)$，乘数 $t=19$，模数 $k=73$，可得

$B=t\times A\bmod k=(57,3,25,31,8)$ 。

明文“good night”的编码为“00111”，“01111”，“01111”，“00100”，“00000”，“01110”，“01001”，“00111”，“01000”，“10100”，则有：

$$
\begin{aligned}
&f(00111)=25+31+8=64,\\
&f(01111)=3+25+31+8=67,\\
&f(01111)=3+25+31+8=67,\\
&f(00100)=25,\\
&f(00000)=0,\\
&f(01110)=3+25+31=59,\\
&f(01001)=3+8=11,\\
&f(00111)=25+31+8=64,\\
&f(01000)=3,\\
&f(10100)=57+25=82=9\bmod 73.
\end{aligned}
$$

所以明文“good night”相应的密文为 $(64,67,67,25,0,59,11,64,3,9)$ 。

5、设背包密码系统的超递增序列为(3,4,8,17,33)，乘数 $t=17$，模数 $k=67$，试对 $25,2,72,92$ 解密。

解：因为 $t^{-1}\bmod k=17^{-1}\bmod 67=4\bmod 67$，则 $4\times(25,2,72,92)\bmod 67=(33,8,20,33)$ 。

所以其对应的明文分组为 $(00001,00100,10010,00001)$，由表可得明文为“ADRA”。

6、椭圆曲线 $E_{11}(1,6)$ 表示 $y^2 \equiv x^3 + x + 6 \bmod 11$，求其上的所有点。

解：模 11 的平方剩余有 $1,4,9,5,3$。

$x = 1,4,6$ 时，$y^2 \equiv 8 \bmod 11$，无解，曲线无与这一 x 相对应的点；

$x = 9$ 时，$y^2 \equiv 7 \bmod 11$，无解，曲线无与这一 x 相对应的点；

$x = 0$ 时，$y^2 \equiv 6 \bmod 11$，无解，曲线无与这一 x 相对应的点；

$x = 2$ 时，$y^2 \equiv 2 \bmod 11, y = 4$ 或 7；

$x = 3$ 时，$y^2 \equiv 3 \bmod 11, y = 5$ 或 6；

$x = 5,7,10$ 时，$y^2 \equiv 4 \bmod 11$，$y = 2$ 或 9；

$x = 8$ 时，$y^2 \equiv 9 \bmod 11, y = 3$ 或 8。

所以椭圆曲线 $E_{11}(1,6)$ 上的所有点为：

$\{(2,4),(2,7),(3,5),(3,6),(5,2),(5,9),(7,2),(7,9),(8,3),(8,8),(10,2),(10,9),O\}$。

7、已知点 $G = (2,7)$ 在椭圆曲线 $E_{11}(1,6)$ 上，求 $2G$ 和 $3G$。

解：(1)求 $2G$。

$$\lambda = \frac{3 \times 2^2 + 1}{2 \times 7} \bmod 11 = (13 \times 4) \bmod 11 = 8 \bmod 11,$$
$$x_{2G} = (8^2 - 2 - 2) \bmod 11 = 5 \bmod 11,$$
$$y_{2G} = [8 \times (2 - 5) - 7] \bmod 11 = 8 \bmod 11,$$

所以 $2G = (5,2)$。

(2)易知 $3G = 2G + G = (5,2) + (2,7)$。

$$\lambda = \frac{7 - 2}{5 - 2} \bmod 11 = (5 \times 7) \bmod 11 = 2 \bmod 11,$$
$$x_{3G} = (2^2 - 5 - 2) \bmod 11 = 8 \bmod 11,$$
$$y_{3G} = [2 \times (5 - 8) - 2] \bmod 11 = 3 \bmod 11,$$

所以 $3G = (8,3)$。

8、利用椭圆曲线实现 ElGamal 密码体制，设椭圆曲线是 $E_{11}(1,6)$，生成元 $G=(2,7)$，接收方 A 的秘密钥 $n_A=7$。

(1) 求 A 的公开钥 P_A。

(2) 发送方 B 欲发送消息 $P_m=(10,9)$，选择随机数 $k=3$，求密文 C_m。

(3) 显示接收方 A 从密文 C_m 恢复消息 P_m 的过程。

解：(1)易知公开钥 $P_A=7G=2\times 2G+3G$。

①求 $2\times 2G$。

$$\lambda=\frac{3\times 5^2+1}{2\times 2}\bmod 11=(10\times 3)\bmod 11=8\bmod 11,$$
$$x_{4G}=(8^2-5-5)\bmod 11=10\bmod 11,$$
$$y_{4G}=[8\times(5-10)-2]\bmod 11=2\bmod 11,$$

所以 $2\times 2G=(10,2)$。

②由题 19 可得 $3G=(8,3)$，即 $P_A=2\times 2G+3G=(10,2)+(8,3)$。

$$\lambda=\frac{3-2}{8-10}\bmod 11=(1\times 5)\bmod 11=5\bmod 11,$$
$$x_{7G}=(5^2-10-8)\bmod 11=7\bmod 11,$$
$$y_{7G}=[5\times(10-7)-2]\bmod 11=2\bmod 11,$$

所以 $P_A=(7,2)$。

(2)密文 $C_m=(kG,P_m+kP_A)$。

①求 kG：$kG=3G=(8,3)$。

②求 kP_A：$kP_A=2P_A+P_A=3G+7G=(2,7)+(7,2)$。

$$\lambda=\frac{2-7}{7-2}\bmod 11=-1\bmod 11,$$
$$x_{3P_A}=((-1)^2-2-7)\bmod 11=3\bmod 11,$$
$$y_{3P_A}=((-1)\times(2-3)-7)\bmod 11=5\bmod 11,$$

所以 $kP_A=(3,5)$。

③求 P_m+kP_A：$P_m+kP_A=(10,9)+(3,5)$。

$$\lambda = \frac{5-9}{3-10} \bmod 11 = -1 \bmod 11,$$
$$x_{P_m+kP_A} = ((-1)^2 - 10 - 3) \bmod 11 = 10 \bmod 11,$$
$$y_{P_m+kP_A} = ((-1) \times (10-10) - 9) \bmod 11 = 2 \bmod 11,$$

所以 $P_m + kP_A = (10,2)$ 。

综上： $C_m = (kG, P_m + kP_A) = \{(8,3),(10,2)\}$ 。

(3)从密文 C_m 恢复消息 P_m 的过程如下：

$$P_m = (P_m + kP_A) - n_A(kG)$$
$$= (10,2) - 7(8,3)$$
$$= (10,2) - (3,5)$$
$$= (10,2) + (3,6)$$
$$= (10,9).$$

其中：

a)计算 $7(8,3)$

①先计算 $2(8,3)$ 。

$$\lambda = \frac{3 \times 8^2 + 1}{2 \times 3} \equiv 1 \bmod 11,$$
$$x_{2(8,3)} = 1^2 - 8 - 8 \equiv 7 \bmod 11,$$
$$y_{2(8,3)} = 1(8-7) - 3 \equiv 9 \bmod 11,$$

所以 $2(8,3) = (7,9)$ 。

②计算 $3(8,3) = 2(8,3) + (8,3)$ 。

$$\lambda = \frac{3-9}{8-7} \equiv 5 \bmod 11,$$
$$x_{3(8,3)} = 5^2 - 7 - 8 \equiv 10 \bmod 11,$$
$$y_{3(8,3)} = 5(7-10) - 9 \equiv 9 \bmod 11,$$

所以 $3(8,3) = (10,9)$ 。

③计算 $6(8,3) = 3(8,3) + 3(8,3)$ 。

$$\lambda = \frac{3\times 10^2+1}{2\times 9} = \frac{301}{18} \equiv 10 \bmod 11,$$
$$x_{6(8,3)} = 10^2 - 10 - 10 \equiv 3 \bmod 11,$$
$$y_{6(8,3)} = 10(10-3) - 9 \equiv 6 \bmod 11,$$

所以 $6(8,3) = (3,6)$。

④计算 $7(8,3) = 6(8,3) + (8,3)$。

$$\lambda = \frac{3-6}{8-3} = \frac{-3}{5} \equiv 6 \bmod 11,$$
$$x_{7(8,3)} = 6^2 - 3 - 8 \equiv 3 \bmod 11,$$
$$y_{7(8,3)} = 6(3-3) - 6 \equiv 5 \bmod 11,$$

所以 $7(8,3) = (3,5)$。

b)计算 $(10,2) - 7(8,3) = (10,2) - (3,5) = (10,2) + (3,-5) = (10,2) + (3,6)$。

$$\lambda = \frac{6-2}{3-10} = \frac{4}{-7} \equiv 1 \bmod 11,$$
$$x_{P_m} = 1^2 - 10 - 3 \equiv 10 \bmod 11,$$
$$y_{P_m} = 1(10-10) - 2 \equiv 9 \bmod 11,$$

所以 $(10,2) - 7(8,3) = (10,9)$。

第 5 章 密码分配与密钥管理

1、公钥的分配

(1)公开发布

• 用户将自己的公钥发给每一其它用户，或向某一团体广播，例如 PGP 中采用了 RSA 算法，很多用将自己的**公钥附加到消息上**，然后发送到公开区域

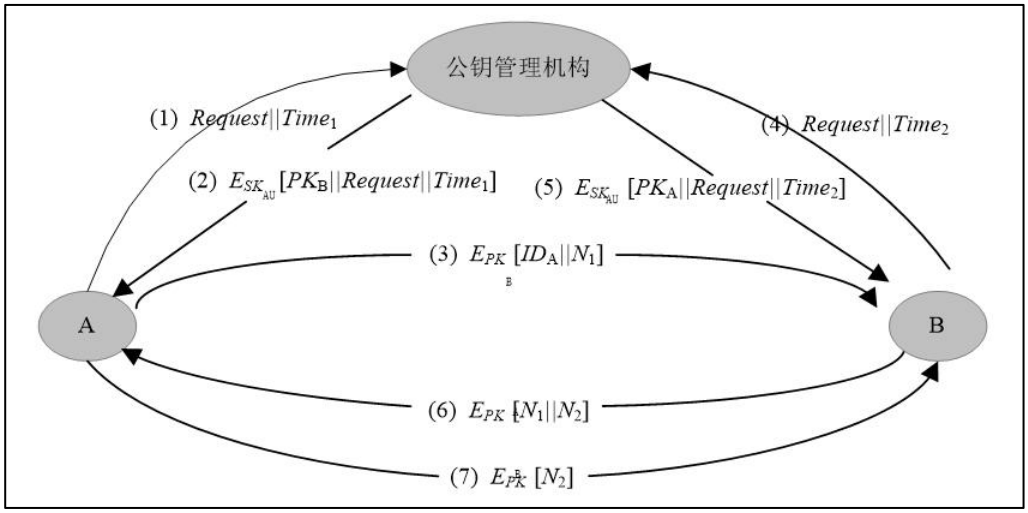
• 缺点：任何人都可伪造这种公开发布

(2)公用目录表

• 本方案的安全性虽然高于公开发布的安全性，但仍易受攻击，如果敌手成功地获取管理员的秘密钥，就可伪造一个公钥目录表

(3)公钥管理机构

每个用户都可靠地知道管理机构的公开钥，而只有管理机构自己知道相应的秘密钥

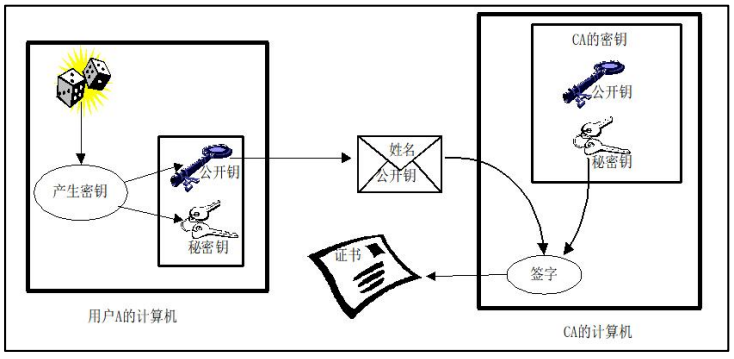


注： 必须定期地通过密钥管理中心获取通信对方的公开钥，以免对方的公开钥更新后无法保证当前的通信

(4)公钥证书

• 公钥证书由证书管理机构 CA 为用户建立，其中的数据项有与该用户的秘密钥相匹配的公开钥及用户的身份和时戳等，所有的数据项经 CA 用自己的秘密钥签字后就形成证书，即证书的形式为

$$C_A = E_{SK_{CA}}[T, ID_A, PK_A]$$

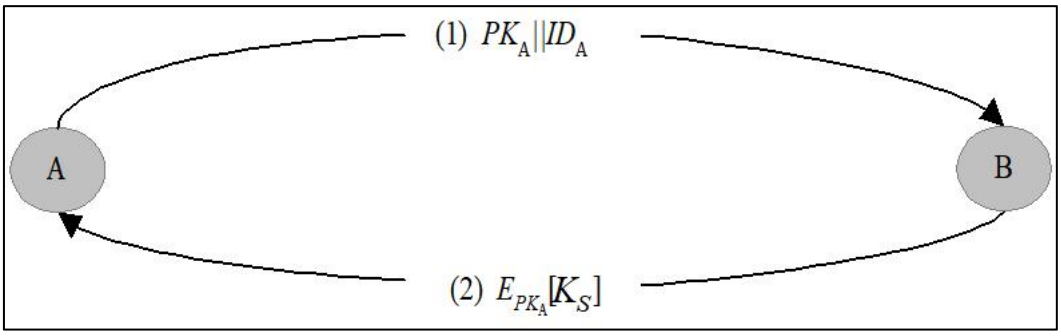


•用户可将自己的公开钥通过公钥证书发给另一用户，接收方可用 CA 的公钥对证书加以验证，即

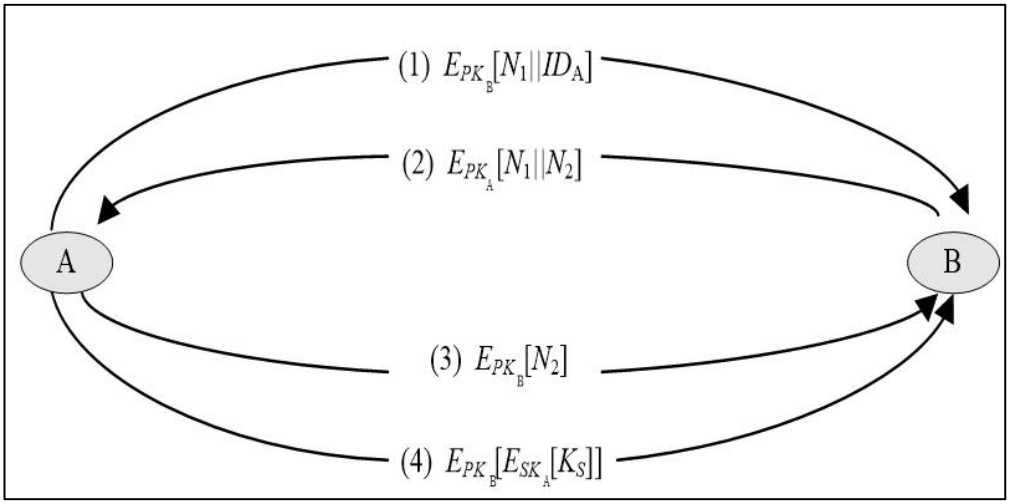
$$D_{PK_{CA}}[C_A] = D_{PK_{CA}}[E_{SK_{CA}}[T, ID_A, PK_A]] = (T, ID_A, PK_A)$$

2、用公钥加密分配单钥密码体制的密钥、

(1)简单分配



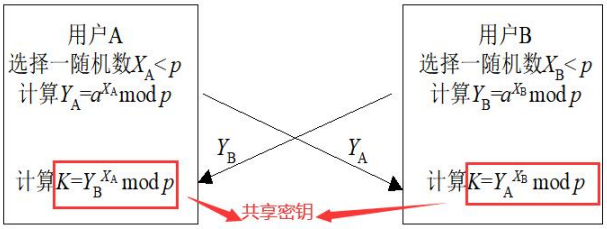
(2)具有保密性和认证性的密钥分配



上图所示的密钥分配过程具有保密性和认证性，因此既可防止被动攻击，又可防止主动攻击

3、Diffie-Hellman 密钥交换

算法的惟一目的：使得两个用户能够安全地交换密钥，得到一个共享的会话密钥，算法本身不能用于加、解密



$$Y_B^{X_A} \bmod p = (a^{X_B} \bmod p)^{X_A} \bmod p = (a^{X_B})^{X_A} \bmod p = a^{X_B X_A} \bmod p$$
$$= (a^{X_A})^{X_B} \bmod p = (a^{X_A} \bmod p)^{X_B} \bmod p = Y_A^{X_B} \bmod p$$

习题：

1、在公钥体制中，每一用户 U 都有自己的公开钥 PK_U 和秘密钥 SK_U。如果任意两个用户 A,B 按以下方式通信，A 发给 B 消息 $\left(E_{PK_B}(\mathrm{m}),A\right)$，B 收到后，自动向 A 返回消息 $\left(E_{PK_A}(\mathrm{m}),B\right)$，以使 A 知道 B 确实收到报文 m，

（1）问用户 C 怎样通过攻击手段获取报文 m?

解：当 A 发给 B 消息 $\left(E_{PK_B}(\mathrm{m}),A\right)$ 时，A 的身份“A”并没有认证，而 B 在收到消息后也无法对发送者进行检验，且身份 A，B 均明文传输，因此用户 C 可通过如下手段获得报文 m：

当 A 发给 B 消息 $\left(E_{PK_B}(\mathrm{m}),A\right)$ 时，C 截取该消息并将身份 A 替换为自己的身份 C，将修改后的消息 $\left(E_{PK_B}(\mathrm{m}),C\right)$ 发给接收者 B；

B 提取消息后，根据身份“C”将返回消息 $\left(E_{PK_C}(\mathrm{m}),B\right)$；

C 再次劫取 B 返回的消息 $\left(E_{PK_C}(\mathrm{m}),B\right)$，用自己的私钥 SK_C 解密出消息 m，并用 A 的公钥对 m 加密后将消息 $\left(E_{PK_A}(\mathrm{m}),B\right)$ 发给 A。

这样，用户 C 获得了报文 m 而没有影响 A,B 之间的正常通信，实现了攻击。

（2）若通信格式变为：

A 发给 B 消息 $E_{PK_B}\left(E_{SK_A}(m),m,A\right)$

B 向 A 返回消息 $E_{PK_A}\left(E_{SK_B}(m),m,B\right)$

这时的安全性如何？分析这时 A、B 如何相互认证并传递消息 m。

解：根据消息格式，先对消息 m 进行了签名，然后再进行加密，传送的消息具有了保密性和认证性，敌手无法获得报文明文，安全性提高。

A，B 之间相互认证传递消息的过程如下：

B 收到消息 $E_{PK_B}\left(E_{SK_A}(m),m,A\right)$ 时，先用 B 自己的私钥解密得到消息 $\left(E_{SK_A}(m),m,A\right)$，然后根据提取的身份信息 A，用 A 的公钥对消息 m 的签名 $E_{SK_A}(m)$ 的正确性进行验证，如果验证通过，则说明消息确实来自 A。反之 A 用相同的方法可验证 $E_{PK_A}\left(E_{SK_B}(m),m,B\right)$ 确实来自 B，从而实现了相互认证。

2、Diffie-Hellman 密钥交换过程中，设大素数 p=11，a=2 是 p 的本原根，

（1）用户 A 的公开钥 Y_A=9，求其秘密钥 X_A。

解：X_A 满足 $Y_A = a^{X_A} \bmod p$ 即 $9=2^{X_A} \bmod 11$，所以有 X_A=6。

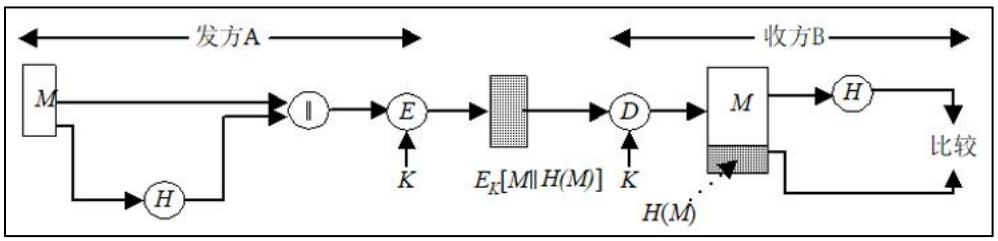
（2）设用户 B 的公开钥 $Y_B = 3$，求 A 和 B 的共享密钥 K。

解：由 Diffie-Hellman 协议可知 $K=Y_B^{X_A} \bmod p = 3^6 \bmod 11 = 3$。

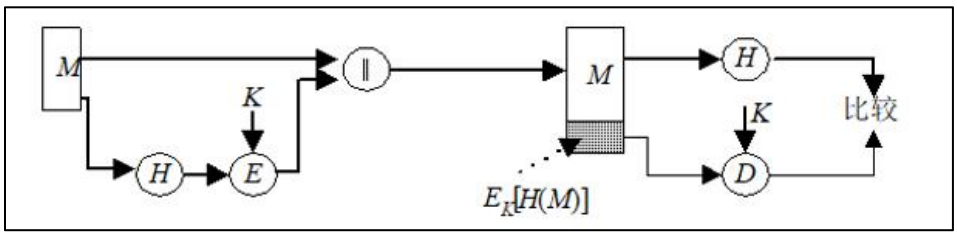
第 6 章 消息认证和哈希函数

1、哈希函数的使用方式

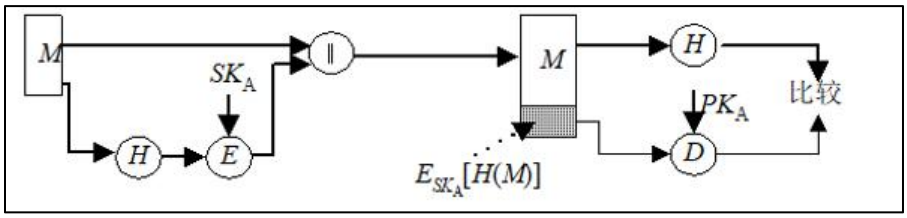
(1)消息与哈希码链接后用单钥加密算法加密，提供了**保密性**



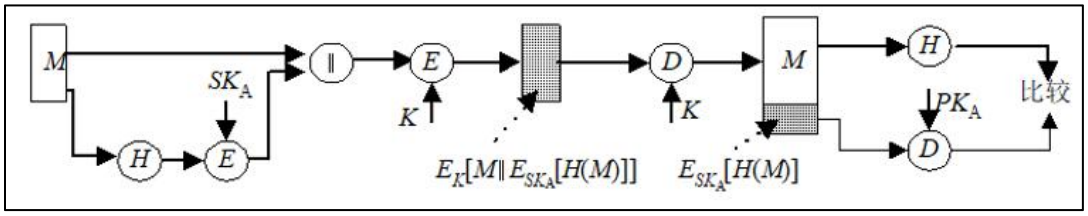
(2)用单钥加密算法仅对哈希码加密，用于**不要求保密性**的情况中可减少处理负担



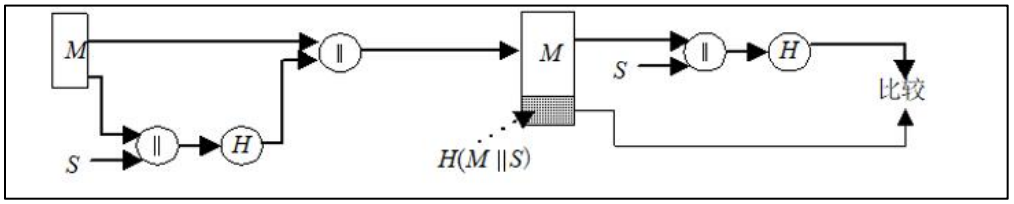
(3)用公钥加密算法和发方的秘密钥仅加密哈希码，对发方发送的消息提供了**数字签字**



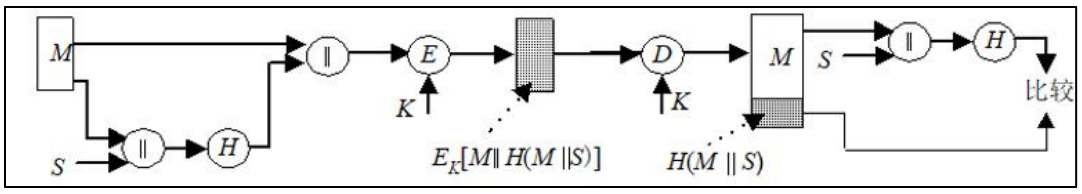
(4)消息的哈希值用公钥加密算法和发方的秘密钥加密后与消息链接，再对链接后的结果用单钥加密算法加密，这种方式提供了**保密性和数字签字**



(5)使用这种方式时要求通信双方共享一个秘密值，仅提供**认证**



(6)消息与哈希值链接以后再增加单钥加密运算，提供**保密性**



注： 由于加密算法的速度较慢，代价较高，而且很多加密算法还受到专利保护，因此在不要求保密性的情况下，方式(2)和(3)比其他方式更具有优势

2、哈希函数应满足的条件

(1)函数的输入可以是任意长

(2)函数的输出是固定长

(3)已知 x，求 $H(x)$较为容易，可用硬件或软件实现

(4)已知 h，求使得 $H(x)=h$ 的 x 在计算上是不可行的，这一性质称为函数的单向性，称 $H(x)$ 为单向哈希函数。

(5)已知 x，找出 $y(y\neq x)$使得 $H(y)=H(x)$在计算上是不可行的。如果单向哈希函数满足这一性质，则称其为弱单向哈希函数。

(6)找出任意两个不同的输入 x、y，使得 $H(y)=H(x)$在计算上是不可行的。

注： 如果单向哈希函数满足这一性质，则称其为强单向哈希函数。性质 5、6 给出了哈希函数**无碰撞性**的概念，如果哈希函数对不同的输入可产生相同的输出，则称该函数具有**碰撞性**。

第 7 章 数字签名和认证协议

1、数字签名应满足以下要求：

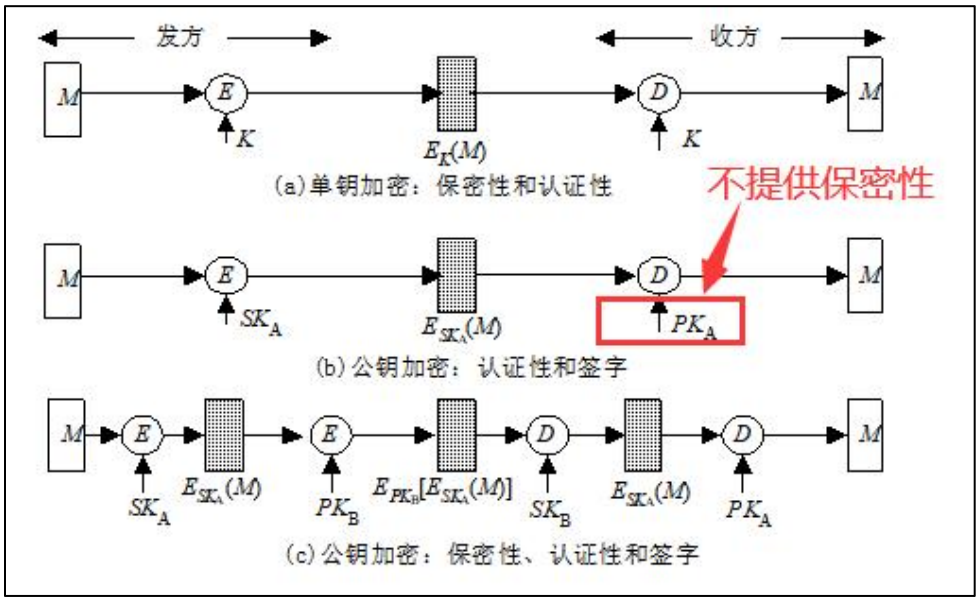
(1)签名的产生必须使用发方独有的一些信息以防伪造和否认

(2)签名的产生应较为容易

(3)签名的识别和验证应较为容易

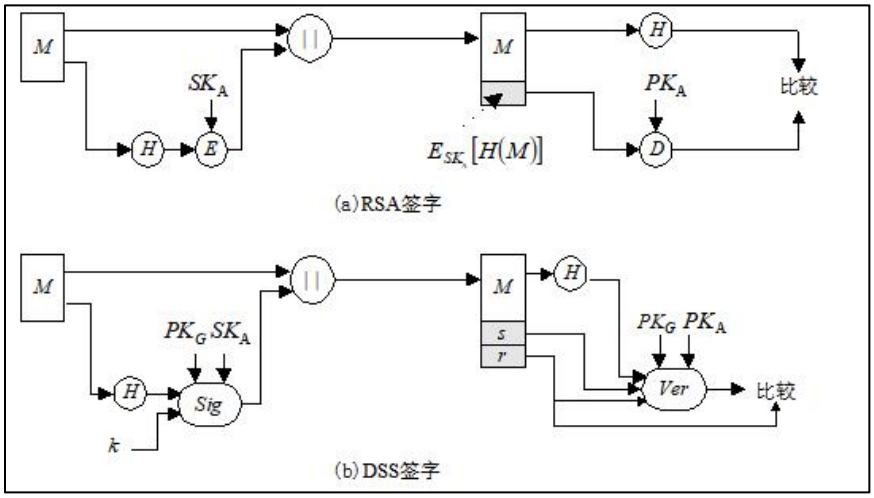
(4)对已知的数字签名构造一新的消息或对已知的消息构造一假冒的数字签名在计算上都是不可行的

2、消息加密产生数字前面的基本方式

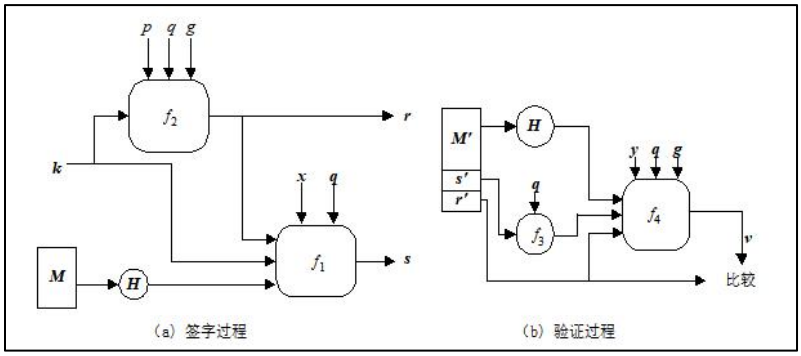


3、DSS 的基本方式

- RSA 算法既能用于加密和签名，又能用于密钥交换
- DSS 使用的算法只能提供数字签名功能



4、数字签名算法 DSA



注： 由于离散对数的困难性，敌手从 r 恢复 k 或从 s 恢复 x 都是不可行的

5、SM2 椭圆曲线公钥密码签名算法

(1)签名算法

设待签名的消息为M，A做以下运算：

①取 $\bar{M}=Z_A\|M$；

②计算 $e=H_v(\bar{M})$，将 e 转换为整数，H_v 是输出为 v 比特长的哈希函数；

③用随机数发生器产生随机数 $k\leftarrow_R\{1,2,\cdots,n-1\}$；

④计算椭圆曲线点 $C_1=kG=(x_1,y_1)$；

⑤计算 $r=(e+x_1)\bmod n$，若 $r=0$ 或 $r+k=n$ 则返回③；

⑥计算 $s=((1+d_A)^{-1}\cdot(k-r\cdot d_A))\bmod n$，若 $s=0$ 则返回③；

⑦消息M的签名为 (r,s)。

(2)验证算法

B收到消息 M' 及其签名 (r',s') 后，执行以下验证运算：

①检验 $r'\in[1,n-1]$ 是否成立，若不成立则验证不通过；

②检验 $s'\in[1,n-1]$ 是否成立，若不成立则验证不通过；

③置 $\bar{M}'=Z_A\|M'$；

④计算 $e'=H_v(\bar{M}')$，将 e' 转换为整数；

⑤计算 $t=(r'+s')\bmod n$，若 $t=0$，则验证不通过；

⑥计算椭圆曲线点 $(x_1',y_1')=s'G+tP_A$；

⑦计算 $R=(e'+x_1')\bmod n$，检验 $R=r'$ 是否成立，若成立则验证通过；否则验证不通过。

(3)正确性

正确性：如果 $\bar{M}=\bar{M}',(r',s')=(r,s)$，则 $e'=e$，要证 $R=r'=r$，只需证 $x_1'=x_1$。

$$x_1'=s'x_G+tx_A=sx_G+(r+s)x_A=sx_G+(r+s)d_Ax_G$$
$$=(s+rd_A+sd_A)x_G=(s(1+d_A)+rd_A)x_G$$
$$=(k-rd_A+rd_A)x_G=kx_G=x_1$$

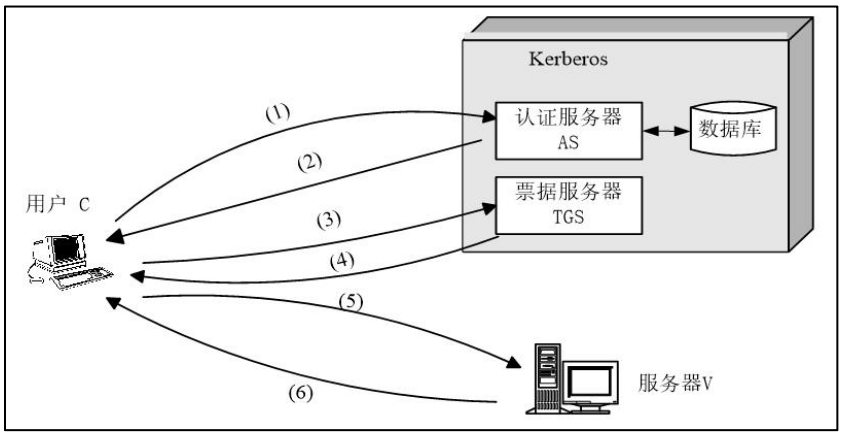
习题：

1、在 DSA 签名算法中，参数 k 泄露会产生什么后果？

解：若攻击者得到了一个有效签名 (r,s)，并且知道了 DSA 签名算法中的参数 k，那么在签名方程 $s=[k^{-1}(H(M)+xr)]\bmod q$ 中只存在一个未知数，即用户的秘密钥 x，所以攻击者可以求得秘密钥 $x=[(ks-H(M))r^{-1}]\bmod q$ 。因此，参数 k 泄漏将导致签名秘密钥的泄漏，攻击者可以伪造任意消息的签名。

第 10 章 网络加密与认证

1、Kerberos4



TS_i：第i个时戳，　　$lifetime_i$：第i个有效期限，
K_C：由用户口令导出的用户和AS的共享密钥，
$K_{c,tgs}$：C与TGS的共享密钥，K_V：TGS与V的共享密钥，
K_{tgs}：AS与TGS的共享密钥，$K_{c,v}$：C与V的共享密钥。

协议如下：

第I阶段（认证服务交换）用户从AS获取票据许可票据：

（1）C→AS：$ID_C \| ID_{tgs} \| TS_1$。

（2）AS→C：$E_{K_c}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| lifetime_t \| Ticket_{tgs}]$。

其中 $Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| lifetime_t]$

第II阶段（票据许可服务交换）用户从TGS获取服务许可票据：

（3）C→TGS：$ID_V \| Ticket_{tgs} \| Authenticator_c$。

（4）TGS→C：$E_{K_{c,tgs}}[K_{c,v} \| ID_V \| TS_4 \| Ticket_v]$

其中 $Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| lifetime_2]$

$Ticket_v = E_{K_v}[K_{c,v} \| ID_C \| AD_C \| ID_V \| TS_4 \| lifetime_4]$

$Authenticator_c = E_{K_{c,tgs}}[ID_C \| AD_C \| TS_3]$

第III阶段（客户机——服务器的认证交换）用户从服务器获取服务：

（5）C→V：$Ticket_v \| Authentica\ tor_v$

（6）V→C：$E_{K_{c,v}}[TS_5+1]$

其中 $Ticket_v = E_{K_v}[K_{c,v} \| ID_C \| AD_C \| ID_V \| TS_4 \| lifetime_4]$

$Authenticator_c = E_{K_{c,tgs}}[ID_C \| AD_C \| TS_3]$

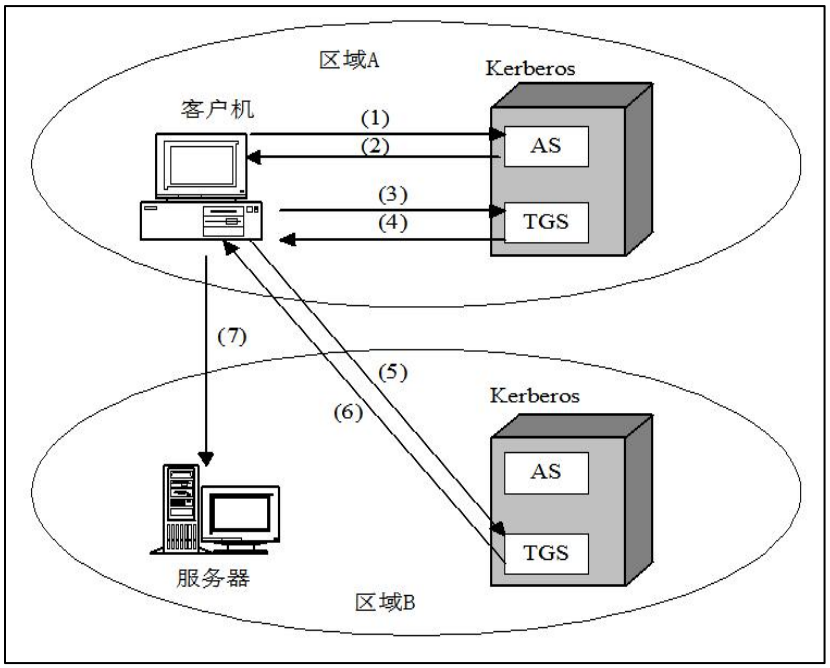
2、Kerberos 区域与多区域的 Kerberos

(1)Kerberos 的一个完整服务范围由一个 Kerberos 服务器、多个客户机和多个服务器构成，并且满足以下两个要求

• Kerberos 服务器必须在它的数据库中存有所有用户的 ID 和口令的哈希值，所有用户都已向 Kerberos 服务器注册

• Kerberos 服务器必须与每一服务器有共享的密钥，所有服务器都已向 Kerberos 服务器注册

(2)两个区域的 Kerberos



（1）客户向本地 AS 申请访问本区域 TGS 的票据：

C→AS：$ID_C \| ID_{tgs} \| TS_1$

（2）AS 向客户发放访问本区域 TGS 的票据：

AS→C：$E_{K_c}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| lifetime_1 \| Ticket_{tgs}]$

（3）客户向本地 TGS 申请访问远程 TGS 的票据许可票据：

C→TGS：$ID_V \| Ticket_{tgs} \| Authentica\ tor_c$

（4）TGS 向客户发放访问远程 TGS 的票据许可票：

TGS→C：$E_{K_{c,tgs}}[K_{c,v} \| ID_V \| TS_4 \| Ticket_v]$

（5）客户向远程TGS申请获得服务器服务的服务许可票据：

C→TGS_{rem}：$ID_{vrem} \| Ticket_{tgsrem} \| Authentica\ tor_c$

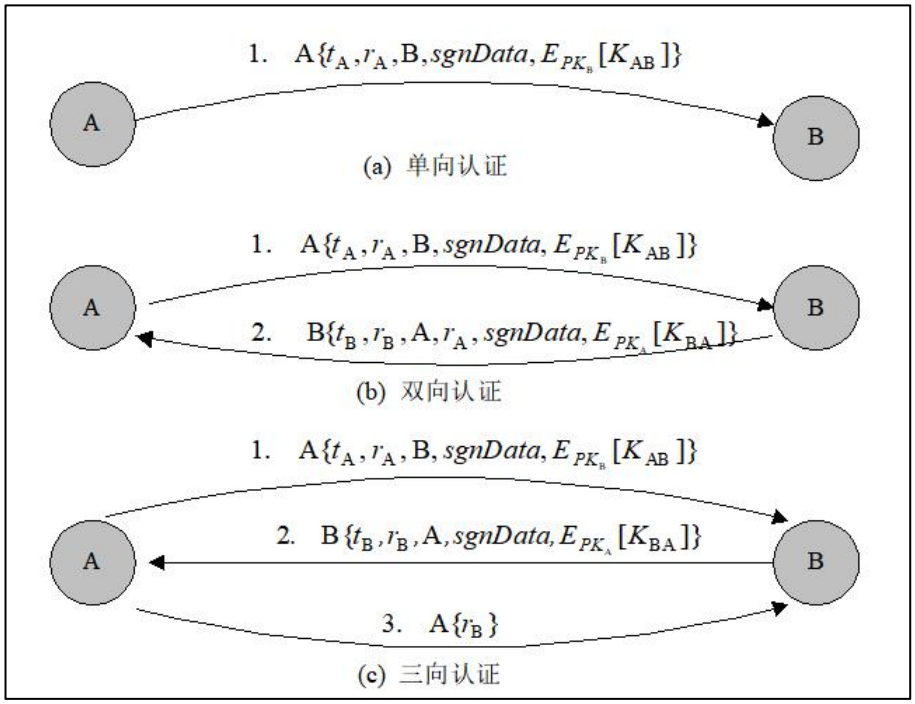
（6） 远程 TGS 向客户发放服务许可票据：

TGS→C：$E_{K_{c,tgsrem}}[K_{c,vrem} \| ID_{vrem} \| TS_6 \| Ticket_{vrem}]$

（7）客户申请远程服务器的服务：

C→V_{rem}：$Ticket_{vrem} \| Authentica\ tor_c$

3、认证过程



(1)单向认证

单向认证指用户 A 将消息发往 B，以向 B 证明：A 的身份、消息是由 A 产生的，消息的意欲接收者是 B，消息的**完整性**和**新鲜性**

(2)双向认证

双向认证是在上述单向认证的基础上，B 再向 A 作出应答，以证明：B 的身份、应答消息是由 B 产生的，应答的意欲接收者是 A，应答消息是完整的和新鲜的

(3)三向认证

在上述双向认证完成后，A 再对从 B 发来的一次性随机数签名后发往 B，即构成第三向认证。

三向认证的目的是双方将收到的对方发来的一次性随机数又都返回给对方，因此双方不需检查时戳只需检查对方的一次性随机数即可检查出是否有重放攻击。在通信双方无法建立时钟同步时，就需使用这种方法。