第5章 密码

（计划学时 6）

本章主要内容

- 密码学的基本概念　　1学时
- 序列密码　　1学时
- 分组密码　　1学时
- 公钥密码　　2学时
- 数字签名　　1学时

教学目的与要求

1. 理解密码学的基本概念与主要分类。

2. 了解以伪随机序列为基础的序列密码有关知识。

3. 了解以DES为代表的分组密码基本概念。

4. 重点理解公钥密码的新理念。掌握**RSA**为典型的公开密码体系加、解密原理方法。

5. 了解数字签名与认证的基本原理方法。

参考文献

1.章照止：**现代密码学基础**

北京邮电大学出版社（20004年4月第一版）

2.王育民：**保密学---基础与应用**

西安电子科技大学出版社（1990年第一版）

3. 杨义先：**现代密码新理论**

北京：科学出版社（2002 第一版）

4. Bruce Schneier: **Applied Cryptography Second Edition**

(Protocols, algorithms, and source code in C)

第5章 密码

5.1 密码学的基本概念

（第18讲 2007.12.11.）

[温旧引新]

- 三大编码：

  信源编码、信道编码和加密编码

- 信源编码：

  为了压缩代码长度而进行的编码。

- 信道编码：

  为了减少差错而进行的编码。

本节的主要内容

- 密码学的内容和术语
- 密码体系分类
- 传统密码举例
- 对称密钥体系

外语关键词

密码学：Cryptology

保密： encipher; 泄密compromise

认证：authentication; 识别identification

密钥：key; (public key & private key)

攻击：attack; 抵赖repudiation

密码分析学： Cryptanalysis

5.1.1 密码学的内容和术语

- 密码是为解决信息安全而进行的编码。
- 安全指通信系统抗御外来攻击的能力。
- 外来攻击主要有两大类，
  - 被动攻击——来自接收端，以截获或窃听通信内容为目的，攻击者一般并不改变通信内容；
  - 主动攻击——来自发送端，冒充合法发信人，发布恶意信息，篡改或伪造信息，以达到骗取钱财、机密，甚至破坏通信系统。

防范攻击的方法

加密

- 针对被动攻击，密码可以使系统对所传输信息具有保密功能。
- 针对主动进攻，密码还应具备“认证”功能，对发信人身份、消息来源以及消息完整性等加以认证，阻止非法发信人，识别虚假、伪造消息。

认证

- 保密和认证是密码的两大基本功能。

密码学两大分支

- 密码学(Cryptology)
- 密码编码学(Cryptography)：研究如何编制密码码、设计密码体制的学问。
- 密码分析学(Cryptanalysis)：研究如何破译密码和攻击密码体系的学问。
- 二者矛盾斗争、相辅相成，共同发展。

基本术语

明文(plaintext)：发送方（sender）未经过加密处理的信息，其内容是容易理解的。

密文(ciphertext)：发送方经过加密处理后的信息，文字被改变，其内容是难以理解的。

密钥(key)：加密与解密处理时所用的秘密信息。

加密(encryption或encipher)：把明文加工成密文的变换。 $C = E_{k1}[M]$

解密(decryption或decipher)：接收方(receiver)把密文解译成明文的变换。 $M = D_{k2}[C]$

5.1.2 密码体系分类

- 从保密部位分
  - 信道加密
    - 物理防范
    - 专线传递
    - 流量余迹消除
    - 潜信道
  - 消息加密
    - 信息加密
      - 数据加密
        - 序列加密
        - 分组加密
          - 对称单钥保密体系
          - 公开密钥保密系统
      - 模拟消息加密
        - 时域置乱
        - 频域置乱
    - 信息伪装
      - 信息隐藏
      - 数字水印
      - 数字指纹
      - 叠像术

5.1.3 传统密码举例

（classical cryptography）

1、 逆序密码

- 将要加密的明文字符分段颠倒首尾次序。
- 设：明文*m*=during the last twenty years there has been an explosion of public academic research in cryptography (近20年对密码学的研究已急剧加速)；
- 加密算法：将原文每5字符一组，各组取逆序连接成密文：

  C=nirudlehtgwttsayytnetsraehereheebsaxenanisolpp fonocilbuedacaercimcraesrcnihgotpyyhpar
- 这里，*K*=5是密钥。

2、 凯撒（Caesar）密码

- 将要加密的明文字符按照字母表的顺序平移一个预先指定的序号，得到新的字符。如将**data security**平移**5**个序号得到：

明文	d	a	t	a	s	e	c	u	r	i	t	y
字母位置	4	1	20	1	19	5	3	21	18	9	20	25
平移5位	9	6	25	6	24	10	8	26	23	14	25	30 (4)
密文	I	F	Y	F	X	J	H	Z	W	N	Y	D

- 循环移位5位，C=(M+5) mod 26
- 循环移位k位数学表达：

  C＝(M+k) mod 26，这里k即密钥
- 为了更安全，如果使用两个密钥，

  $C＝(k_1M +k_2) \bmod 26$
- 特点：每个字符移相同位数，文章中同一字符加密结果相同。称为单表加密。
- 缺点：各字符相应出现的概率不变，可通过概率分析破译。

3、维吉利亚（Vigenere）密码

- 如取单词best为密钥，即给出：(2 ,5, 19, 20)
- 加密时，各字符相应的移位规律是2位、5位、19位、20位，以后按此规律循环重复。
- 特点：各字符按照密钥给出的规律移位，相同字符加密后结果未必相同。称为多表加密。
- 理想的密钥应越长越好，如给定一本书，从某页某行开始移位（密本加密）。

4、希尔（Hill）密码

将长为*L*的明文字符串通过线性变换，变为长为*L*的密文字符串：***C=KM mod 26***（字母表按**0—25**排序）

例如：***M*=Hill**，四个字母序号分别是**(7,8,11,11)**，

若变换矩阵：$K=\begin{pmatrix} 8 & 6 & 9 & 5 \\ 6 & 9 & 5 & 10 \\ 5 & 8 & 4 & 9 \\ 10 & 6 & 11 & 4 \end{pmatrix}$

则：$C = KM = \begin{pmatrix} 8 & 6 & 9 & 5 \\ 6 & 9 & 5 & 10 \\ 5 & 8 & 4 & 9 \\ 10 & 6 & 11 & 4 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 8 \\ 11 \\ 11 \end{pmatrix} = \begin{pmatrix} 24 \\ 19 \\ 8 \\ 23 \end{pmatrix} \bmod 26$

即：**C = YTIX**

5.1.4 对称密钥系统

- 传统密码体系属于“对称密钥系统”。
- 加密过程是对明文M作变换：$C=T_k(M)$
- 解密过程是对密文C作逆变换：$M=T_k^{-1}(C)$
- 密钥就是变换与逆变换中使用的参数k
- 特点：加密算法与解密算法可逆，加、解密的密钥相同。称为对称密钥（symmetric cipher）体系。
- 缺点之一：只重视保密功能，不关注认证功能。
- 缺点之二：密钥交换困难（见以下分析）

保密通信过程的流程

保密通信的流程示意图

密钥交换

- 协议(protocol)：为了使对称密钥（单钥制）体系实现信息的保密传输，通信双方必须事先约定通信双方的通信步骤和各个技术细节。

- 密钥交换(key exchange)：并且设法在不让其他任何人知晓的条件下，双方获取约定的密钥。

- 密钥交换难题：在公共信道中进行密钥交换是比较困难的任务（而专用的安全信道代价太大）。

本节要点

1. 密码学的基本概念：

(1) 功能：保密与认证

(2) 术语：明文、密文、密钥、加密算法和解密算法。

2. 传统密码举例：

逆序密码，恺撒密码，维吉利亚密码，希尔密码。

3. 对称密钥体系：

(1) 特点：加密算法与解密算法可逆，密钥相同。

(2) 密钥交换：设法在不让其他任何人知晓的条件下，双方获取约定的密钥。

- 思考：

  密码分析学（攻击）对密码编码学有何作用？

- 作业：

  P181页：第1题

第5章 密码

5.2 序列(流)密码

（第19讲 2007.12.11.）

[温旧引新]

- 密码的基本功能：

  保密与认证

- 编码术语：

  明文、密文、密钥、加密与解密算法。

- 编码方式：

  分组编码与序列编码。

本节的主要内容

- 序列密码体系
- 伪随机序列
- 反馈移位寄存器
- m序列的产生与特点
- 非线性复合序列

外语关键词

序列密码：stream cipher

伪随机序列：pseudo-random sequence

线性反馈移位寄存器：linear feedback shift register (LFSR)

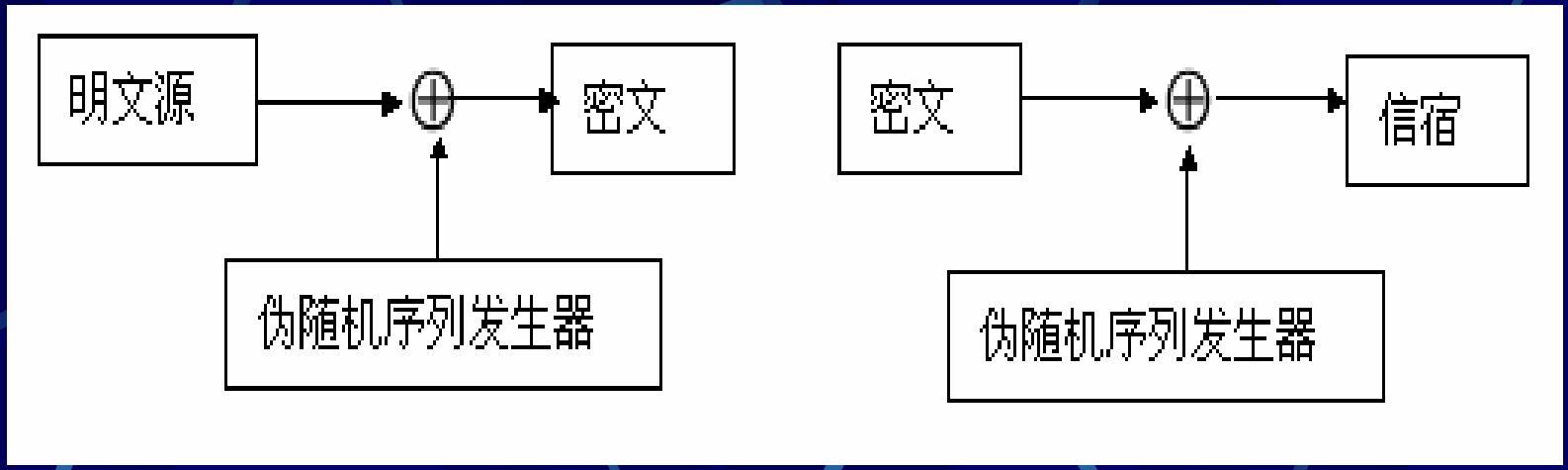
非线性复合序列：non-linear multiplexed sequence

序列复杂度：complexity

5.2.1 序列(流)密码体系

- 明文序列被源源不断地送入加密器，在密钥作用下进行加密运算，产生的密文序列源源不断地被输出来，实时地完成加密。
- 最简单的加密器是二进制明文序列$\{m_i\}$与二进制密钥序列$\{k_i\}$的模二加运算：

$$C_i = (m_i + k_i) \bmod 2 \quad i = 1，2，3，\ldots\ldots;$$



模二加的序列(流)加密、解密过程

- 加密和解密所用的密钥均由伪随机序列发生器产生，作为单钥制的对称保密系统，加、解密两端的伪随机序列发生器应完全相同，并且与密文同步。

5.2.2 伪随机序列

为何用伪随机序列?

- 香农理论告诉我们只有一字一密才是不可破译的，就是说密钥应当完全随机且无穷长。
- 无限长随机序列难以产生，无法重现。
- 实用的密钥序列总是用伪随机序列去代替。

一、伪随机序列的定义

- “伪随机序列”实际上是一个长周期序列，在一个周期内数据分布貌似无序，人们把它作为一个准随机序列看待。
- 只要它的周期足够长，数据分布足够乱，在实际中就可以把它当作随机序列使用。
- 它能够按相同规律再生，就解决了异地密钥交换问题。

二、密钥对伪随机序列的要求

1. 极大的周期：

现代密码机数据率高达$10^8 bit/s$，如果使用10年内不会重复的密钥流，密钥的周期应不少于3×10^{16}或2^{55}。

2. 良好的统计特性：

戈龙布（Glomb）对随机性提出以下判据：

（1）一个周期中，0和1的个数至多相差1；

（2）一个周期中，游程长度为1的游程占1/2；游 程长度为2的游程占1/4；……；并且0游程和1游程各占一半。

（3）序列的相关函数为单位冲击函数：

$$R(\tau)=\frac{1}{T}\sum_{k=0}^{T-1}(-1)^{a_k}(-1)^{a_{k+\tau}}=\begin{cases}1 & (\tau=0)\\ 0 & (\tau\neq 0)\end{cases}$$

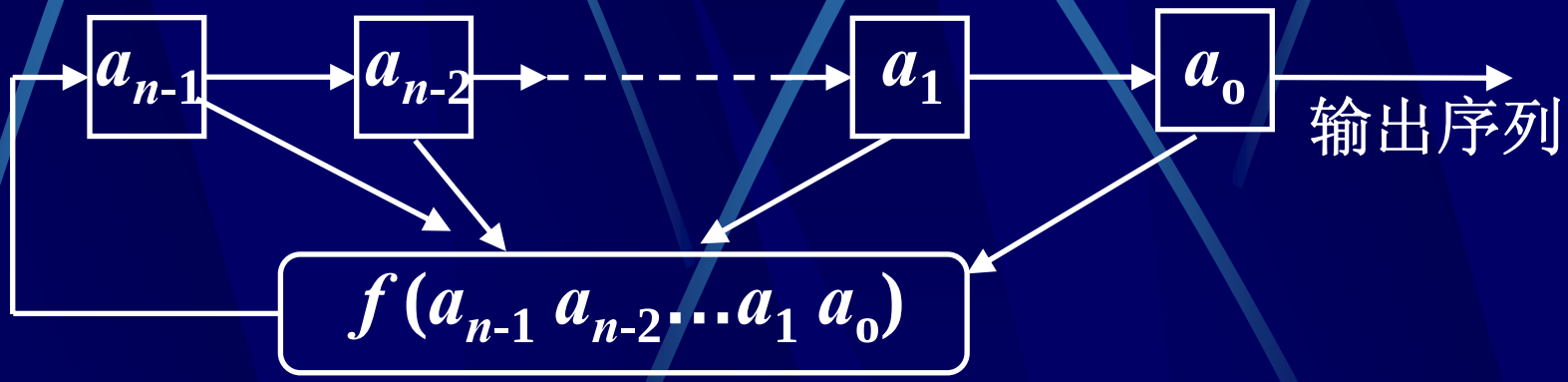
式中：T为周期，τ 为预先指定的一个位移值。

3. 足够的复杂度：

序列的复杂度指序列有足够繁多的变化花样。

5.2.3 反馈移位寄存器（Feedback Shift Register）

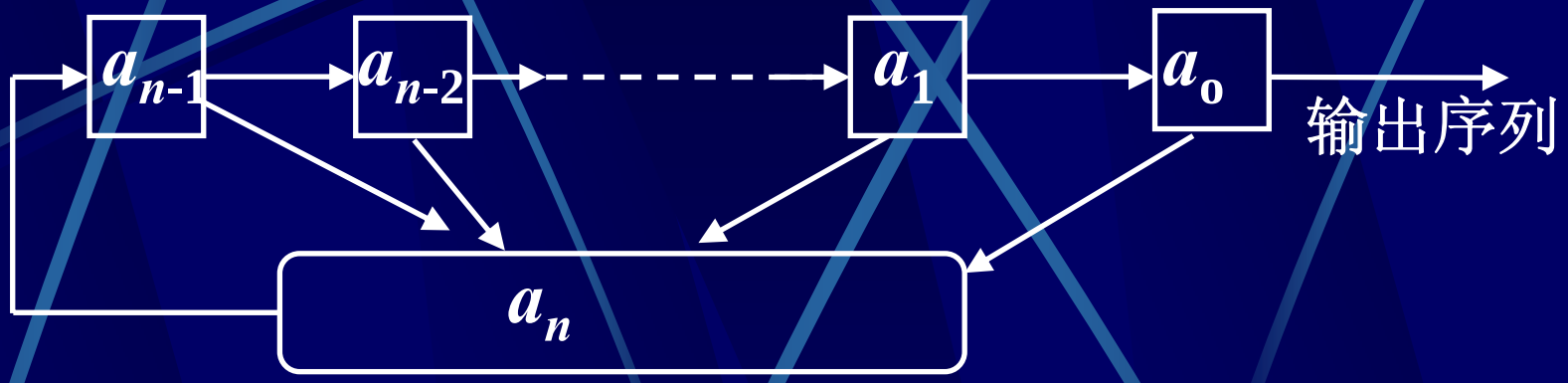
目前设计伪随机序列，多半采用移位寄存器生成，称为**SR**(Shift Register)序列。



$$f(a_0\,a_1\,a_2\ldots\ldots a_{n-1}) = c_0\,a_0 \oplus c_1\,a_1 \oplus \ldots \oplus c_{n-2}\,a_{n-2} \oplus c_{n-1}\,a_{n-1}$$

其中：
$$c_i = \begin{cases} 0 & \text{表示该位无反馈} \\ 1 & \text{表示该位有反馈} \end{cases}$$

视：$f(a_0 a_1 a_2 \cdots\cdots a_{n-1}) = a_n$



可写：$a_n = c_0 a_0 \oplus c_1 a_1 \oplus \cdots \oplus c_{n-2} a_{n-2} \oplus c_{n-1} a_{n-1}$

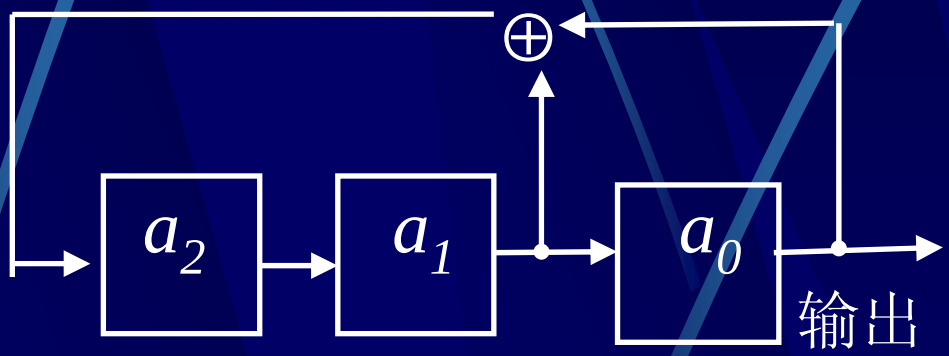
或：$a_n \oplus c_{n-1} a_{n-1} \oplus c_{n-2} a_{n-2} \oplus \cdots\cdots \oplus c_1 a_1 \oplus c_0 a_0 = 0$

由此引入特征多项式：

$$x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \cdots\cdots + c_1 x \oplus c_0$$

用它来描述反馈移位寄存器的结构。

- 如：n=3的三级移位寄存器
- 若初态(001)（不能是全零!），每个时钟节拍寄存器移1位，状态变化见表：



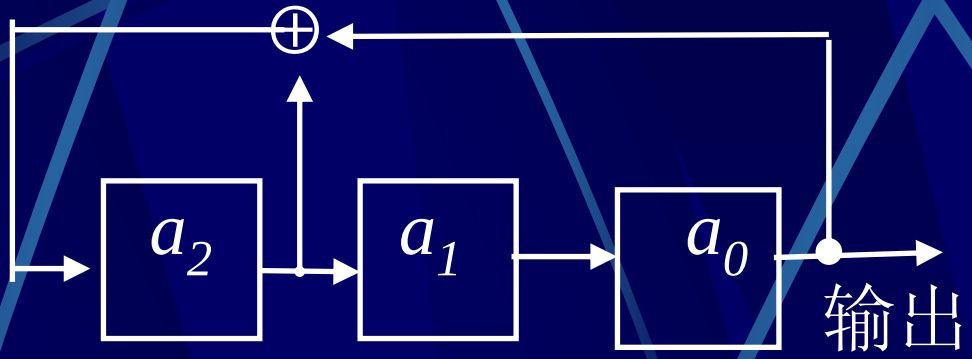
时序	状态
1	001
2	100
3	010
4	101
5	110
6	111
7	011

反馈函数为： $a_3 = a_1 \oplus a_0$

特征多项式为：x^3+x+1

- 它产生周期为$2^3-1=7$的序列：1001011的循环

- 调整抽头位置后：



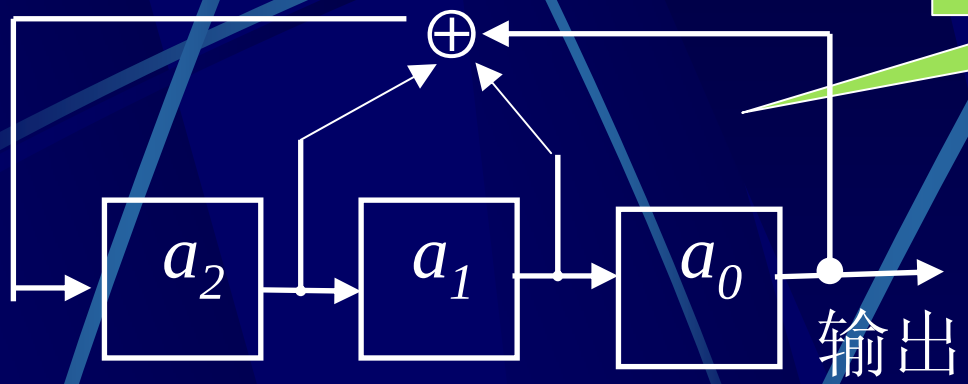
时序	输出
1	001
2	100
3	110
4	111
5	011
6	101
7	010

反馈函数为：$a_3 = a_2 \oplus a_0$

特征多项式为：x^3+x^2+1

- 也产生周期为$2^3-1=7$的序列：1001110的循环

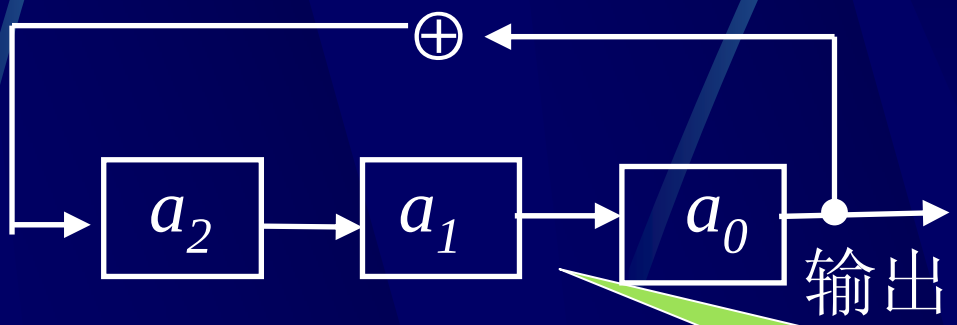
- 存器级数相同，反馈函数不同，也会产生周期不同的序列：



反馈函数为：$a_3 = a_2 \oplus a_1 \oplus a_0$

周期为4，输出1001的循环

时序	上图	下图
1	001	001
2	100	100
3	110	010
4	011	001
5	001	
6		
7		



反馈函数为：$a_3 = a_0$

周期为3，输出100的循环

5.2.4 m序列

一、定义

- 以上各例中，反馈函数均是线性函数。线性反馈移位寄存器称为LFSR;
- 同样级数的LFSR，抽头不同，产生伪随机序列周期就不同。给定级数的LFSR中输出序列周期最长的SR序列，称为*m序列*。
- 上例中，n=3的三级移位寄存器输出序列周期最长为m=7，所以前面两个周期为7的伪随机序列是m序列。

二、关于m序列的一些重要结论：

- *m*序列的周期为$m=2^n-1$（*n*为线性移位寄存器级数）；
- 线性反馈移位寄存器产生*m*序列的充要条件是特征多项式为x^m-1 的本原多项式。
- *n*级移位寄存器，可产生λ(n)种不同的m序列，

$$\lambda(n)=\varphi(m)/n \quad (\ m=2^n-1\ )$$

式中：φ(m)是欧拉数，可由下式求出：

$$\Phi(m)=\begin{cases} m-1 & \text{当}m\text{为素数} \\ m(1-\frac{1}{p_1})(1-\frac{1}{p_2})\cdots\cdots(1-\frac{1}{p_k}) & \text{当}m\text{为合数：}m=p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_k^{\alpha_k} \end{cases}$$

[例1] $n=3$，$m=2^3-1=7$，$\varphi(7)=6$，$\lambda(3)=6/3=2$

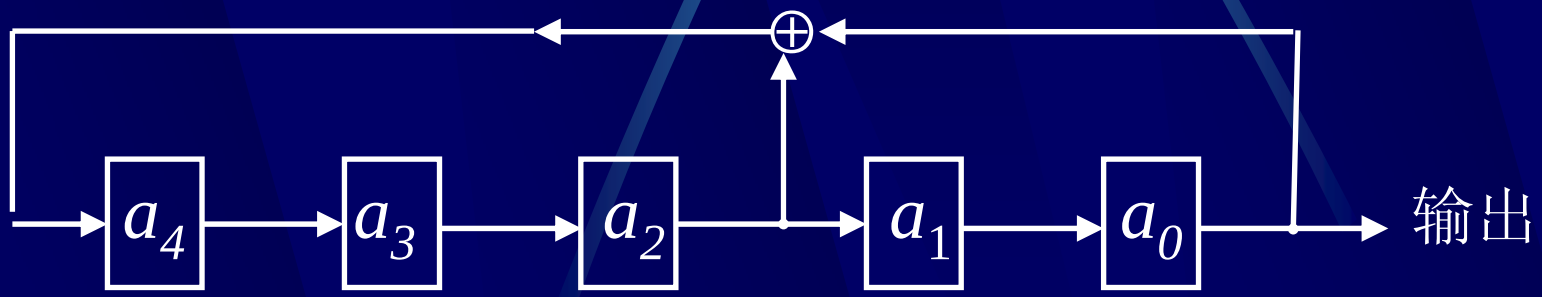
x^7-1的本原多项式是 x^3+x+1 与 x^3+x^2+1

[例2] $n=4$，$m=2^4-1=15$，$\varphi(15)=8$, $\lambda(4)=8/4=2$

$x^{15}-1$本原多项式是 x^4+x+1 与 x^4+x^3+1

[例3] $n=5$，$m=2^5-1=31$，$\varphi(31)=30$，$\lambda(5)=30/5=6$；

由$x^{31}-1$的一个本原多项式 x^5+x^2+1，得到电路：



$n=5$级移位寄存器构成的m序列发生器

三、m序列的统计性质

（1）在一个周期内，1和0出现的次数分别为2^{n-1}和$2^{n-1}-1$；

（2）总游程数共有2^{n-1}个，0、1游程各半。

其中：长为n的1游程和长为$n-1$的0游程各一个；长为$r \leqslant n-2$的1游程和0游程各2^{n-r-2}个；

（3）异相自相关函数为：$R(\tau)=\dfrac{-1}{2^n-1}$

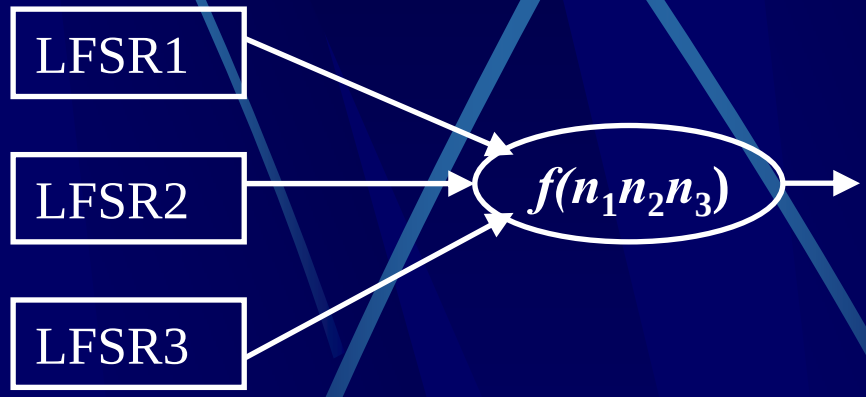
四、m序列的复杂度

- 不难在n=3的m序列1001110(循环) 中找到n<3位的全部码元组合(00, 01, 10, 11)；同理，n=4的m序列中可以找到 n<4位的全部码元组合；推而广之，任何有限位数的码元组合都可以在足够长的m序列中找到。
- 因此，任意给定的一个有限长度的随机序列，总可以找到一个恰能包含它的m序列，序列越复杂，找到的m序列的级数n就越大，因此就可以用这个n值作为该随机序列复杂度的定义。

- 根据这个定义， *m*序列的复杂度就是*n*。
- *m*序列的复杂度是很小的。如周期为**16777215**的*m*序列的复杂度才24；
- 作为密钥用的伪随机序列，复杂度过低有何问题呢？答案是安全性太低！理论上已经证明，只要知道*n*级*m*序列中相继**2*n***位，就能推测出它的特征多项式，进而得到整个序列。
- 结论：*m*序列是构造密钥序列的好素材。其周期长度可以做到很大，随机性也符合**Glomb**要求。缺点是复杂度太低，必须设法解决。

5.2.5 非线性复合序列

用多个**LFSR**产生的**m**序列加以非线性复合，能够得到周期很大、复杂度较高的序列。

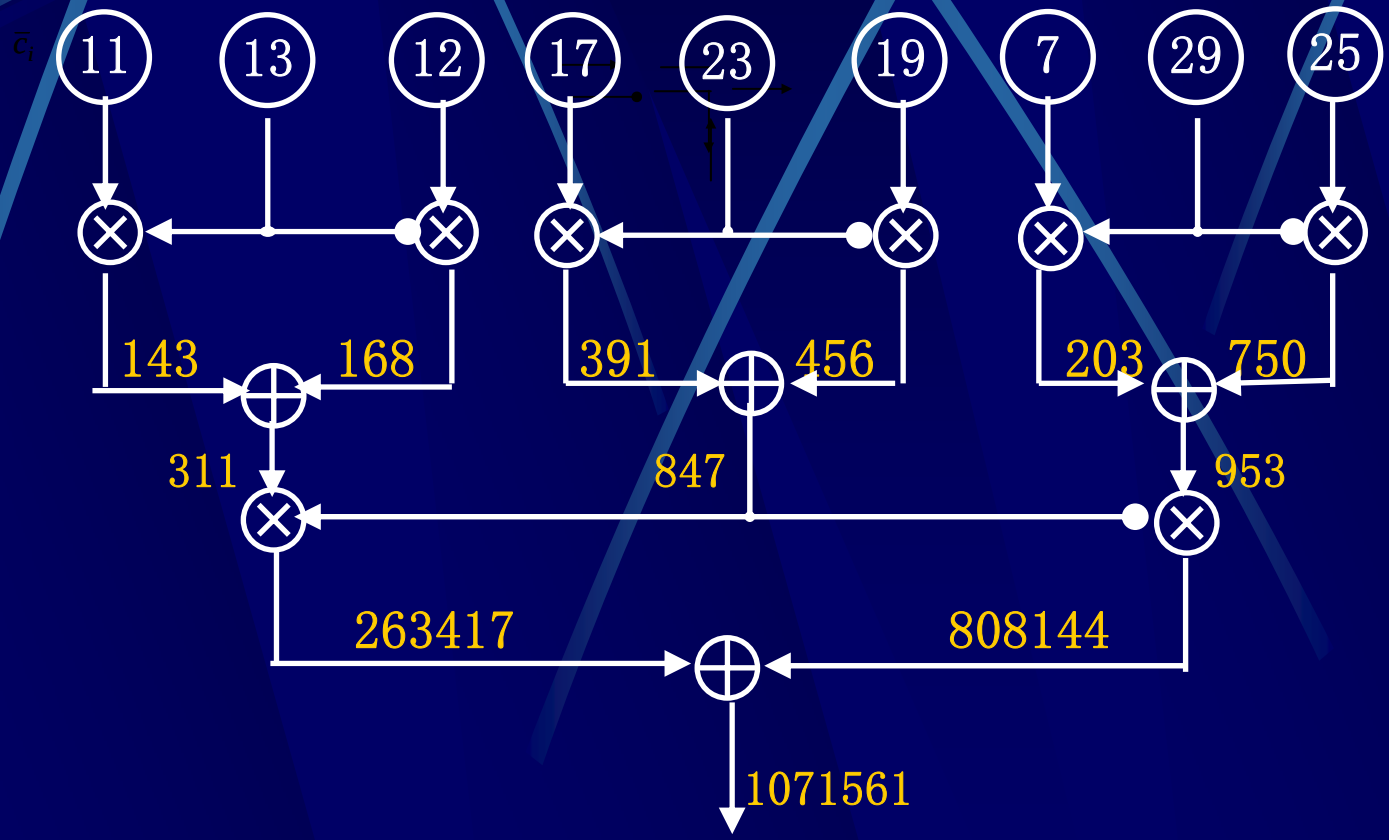


例如，三个**LFSR**的级数分别为**3**、**5**、**7**；非线性复合函数为：$f(x_1,x_2,x_3) = x_3+x_1x_2+x_1x_2x_3$；则组合序列的周期为：

$(2^3-1)(2^5-1)(2^7-1) = 7\times31\times127=27559$；

复杂度为：$f(n_1,n_2,n_3)=7+3\times5+3\times5\times7=127$；

1973年P.R.Geffe提出一种组合序列设计方案，可产生复杂度很高的的非线性复合序列。



本节要点（小结）

1. 序列密码的基本概念：

(1)序列加、解密方式。

(2)对系列密钥的要求。

2. 用反馈移位寄存器产生伪随机序列：

反馈函数，特征多项式。

3. m序列：

(1) 定义，产生（设计）方法。

(2)特点：周期为$m=2^n-1$，个数为$\lambda(n)=\varphi(m)/n$

统计性质：01分布，游程分布，自相关函数

复杂度为n。

- 思考：

  怎样产生周期长、统计性好、复杂度高的伪随机序列？

- 作业：

  P181页：2、3题

第5章 密码

5.3 分组密码

（第20讲 2007.12.13.）

上节回顾

1. 序列密码的基本概念：

(1) 序列加、解密方式。

(2) 对系列密钥的要求。

2. 用反馈移位寄存器产生伪随机序列：

反馈函数，特征多项式。

3. m序列的特性：

(1) 定义，产生（设计）方法。

(2) 特点：周期为$m=2^n-1$

统计性质：01分布，游程分布，自相关函数

复杂度为n。

本节的主要内容

- 分组加密方式
- DES分组加密原理
- DES解密算法

外语关键词

分组密码：block cipher

线性变换：linear transform

循环代换：cyclic substitution

仿射变换： affine transformation

扩展与压缩：Expand **and** compression

置换与定位：**displace and** location

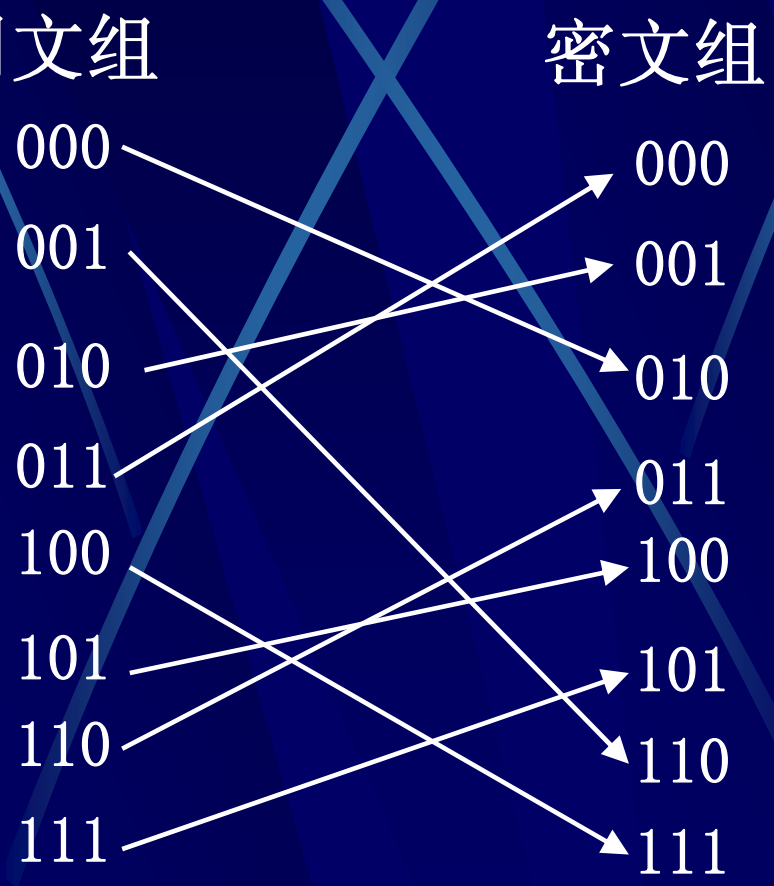
5.3.1 分组加密方式

- 明文M=($m_1\ m_2\ m_3\ m_4 m_5\ m_6\ m_7\ m_8 \ldots$)
- 分组M=($m_1\ m_2\ m_3\ m_4$)($m_5\ m_6\ m_7\ m_8$)……

  $= M_1\ M_2 \ldots\ldots$
- 分组加密： $C_1 = \mathrm{E_k}(M_1)$， $C_2 = \mathrm{E_k}(M_2) \ldots\ldots$
- 连接输出：$C = C_1\ C_2\ C_3\ C_4 \ldots\ldots$

如，分组长度为3，采用置换对应方式加密：

- 1对三
- 2对七
- 3对二
- 4对一
- 5对八
- 6对五
- 7对四
- 8对六



- 写为矩阵形式：

$$\begin{pmatrix} 010 \\ 110 \\ 001 \\ 000 \\ 111 \\ 100 \\ 011 \\ 101 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix}$$

即，加密：$C = K \cdot M$

解密：$M = K^{-1} \cdot C$

常用的加密方式有：

（1）左移循环代换：

$M=(m_1m_2\ldots\ldots m_n) \rightarrow C=(m_2m_3\ldots\ldots m_nm_1)$

（2）右移循环代换：

$M=(m_1m_2\ldots\ldots m_n) \rightarrow C=(m_nm_1m_2\ldots\ldots m_{n-1})$

（3）模二加一代换；

$M=(m_1m_2\ldots\ldots m_n) \rightarrow C=M+1 \bmod 2$

（4）线性变换：

$M=(m_1m_2\ldots\ldots m_n) \rightarrow C=A\cdot M$

（5）仿射变换：

$M=(m_1m_2\ldots\ldots m_n) \rightarrow C=A\cdot M+b$

5.3.2 DES分组加密算法

（DES(Data Encryption Standard）

- DES是IBM公司开发的数据加密标准
- 1976年作为联邦标准被美国国家安全局采用
- 属于公开算法的对称密钥体系
- 有分组（块）加密和序列（流）加密两种形式。
- 可以用硬件或软件实现（代码公开）
- DES保密完全依靠密钥。

数据处理方式

- 明文分组，M 64*bit*（8 *byte*）
- 密文分组，C 64*bit*
- 子密钥，48*bit* x16，来自56*bit*的原始密钥
- 16轮加密：$DES(M)=IP^{-1} \bullet T_{16}T_{15}\cdots T_2T_1 \bullet IP$
  - 注：IP，Initiate Process，初始置换
  - IP^{-1}，逆初始置换
  - T_i，第i轮（ round ），共16轮
- 解密函数 $DES^{-1}(C)= DES^{-1}(DES(M))$

DES分组加密流程

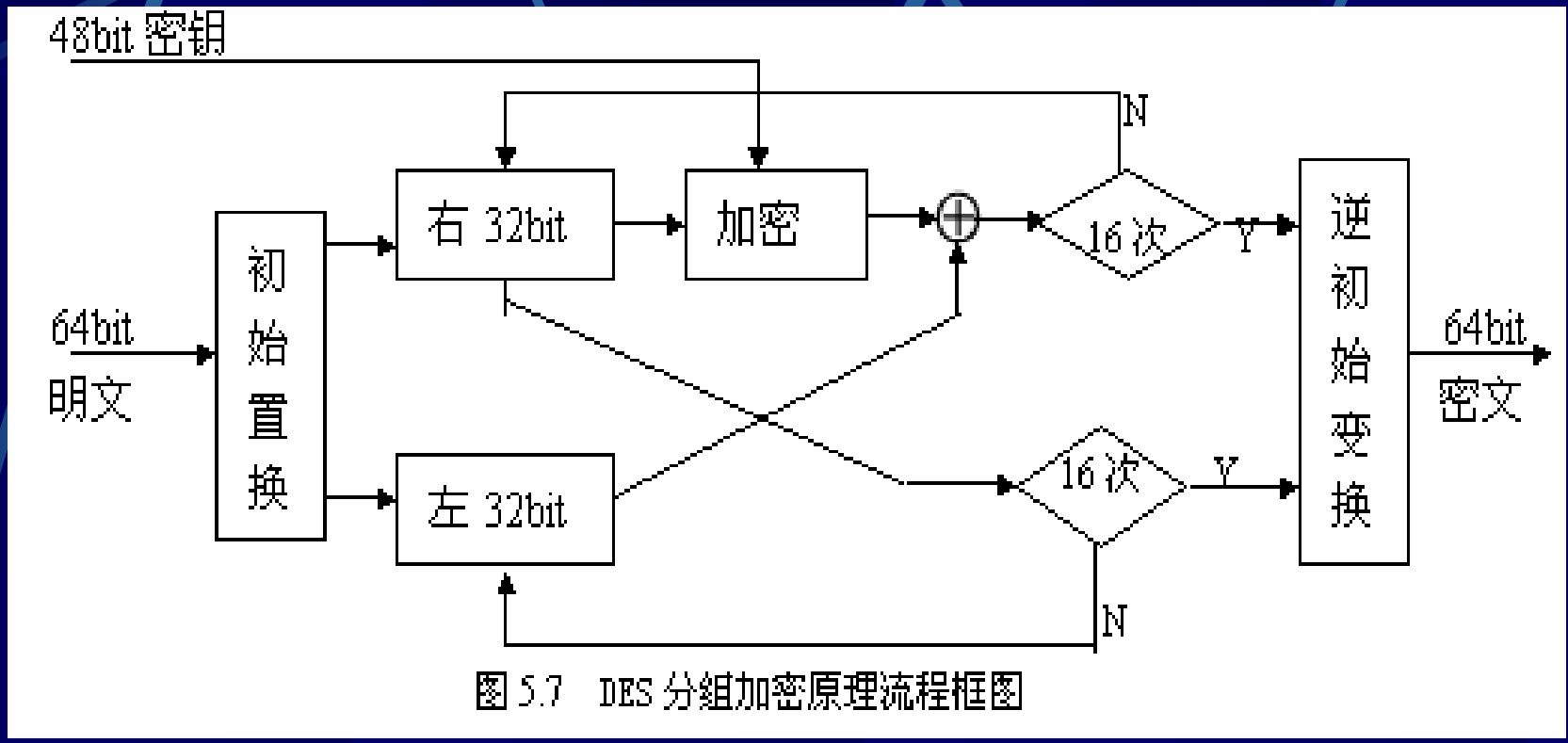


图 5.7　DES 分组加密原理流程框图

关键算法

1. 初始置换与逆置换
2. 换位加密（16轮迭代）
3. f 函数（扩展、加密、压缩、置换）
4. 子密钥生成

1、初始置换与逆初始置换

目的：置乱
增加复杂度

按列逆序写 →IP→

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	28	20	12	4
61	53	45	37	29	21	13	5
62	54	46	38	30	22	14	6
63	55	47	39	31	23	15	7
64	56	48	40	32	24	16	8

入：

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

按行调整次序，**1~8**行重排为**51627384**行

1、初始置换与逆初始置换

$\leftarrow IP^{-1} \leftarrow$

不难看到
恢复原位

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

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

按行读出，按列逆序写入，**1~8**行分别写入**24681357**列

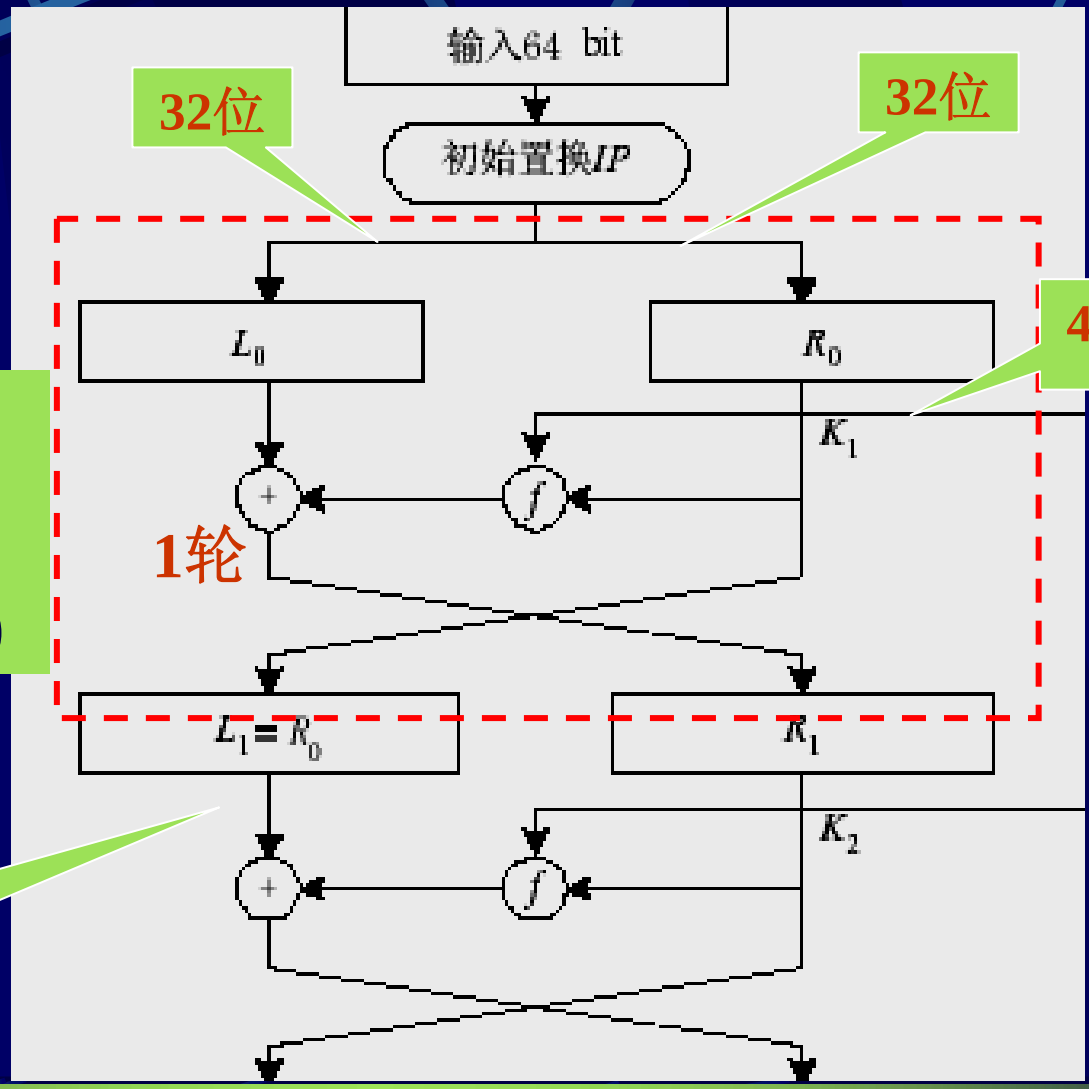
2、换位加密

每轮输出：

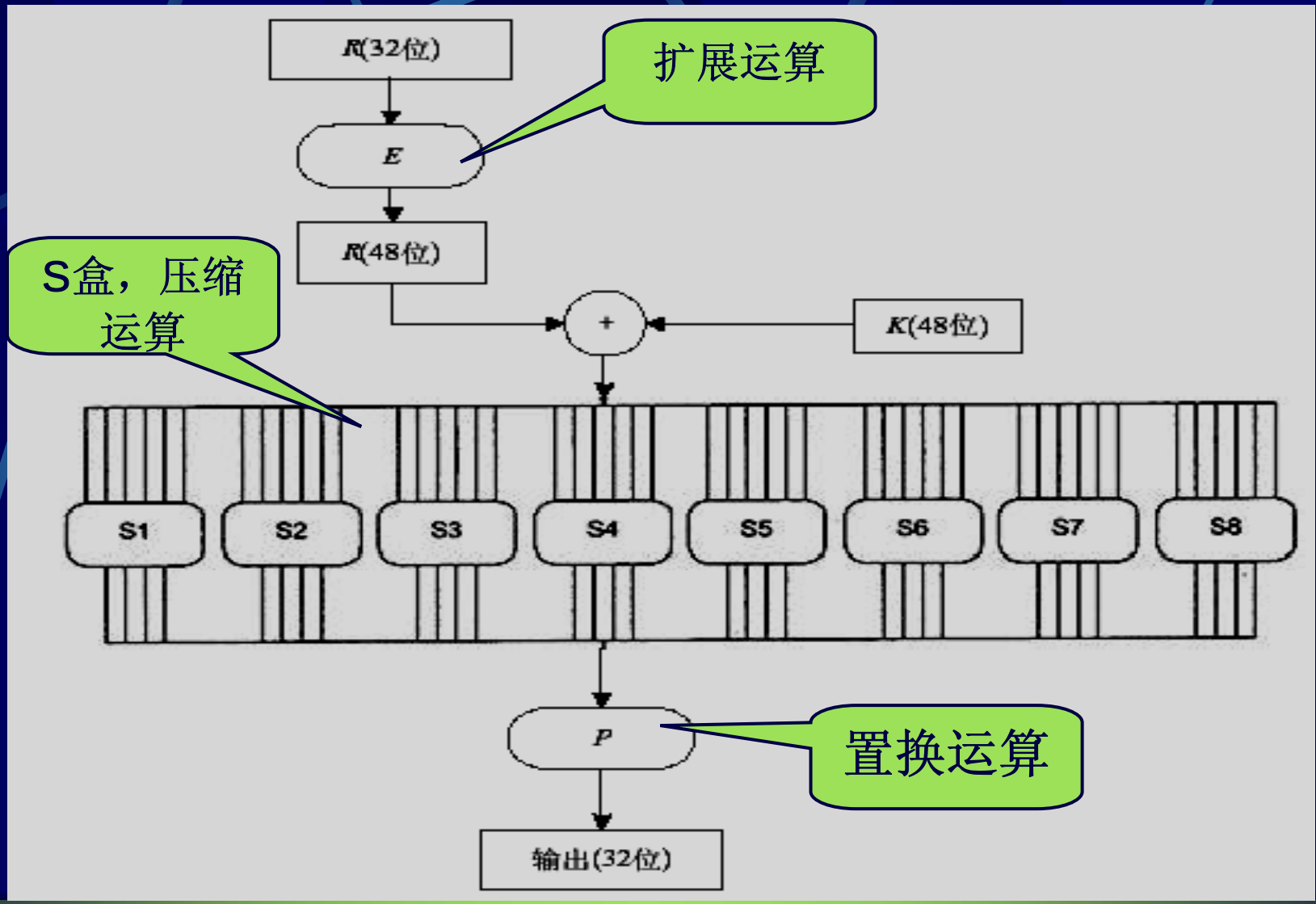
$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(k_i, R_{i-1})$

共16轮



3、f函数运算



(1) 扩展运算

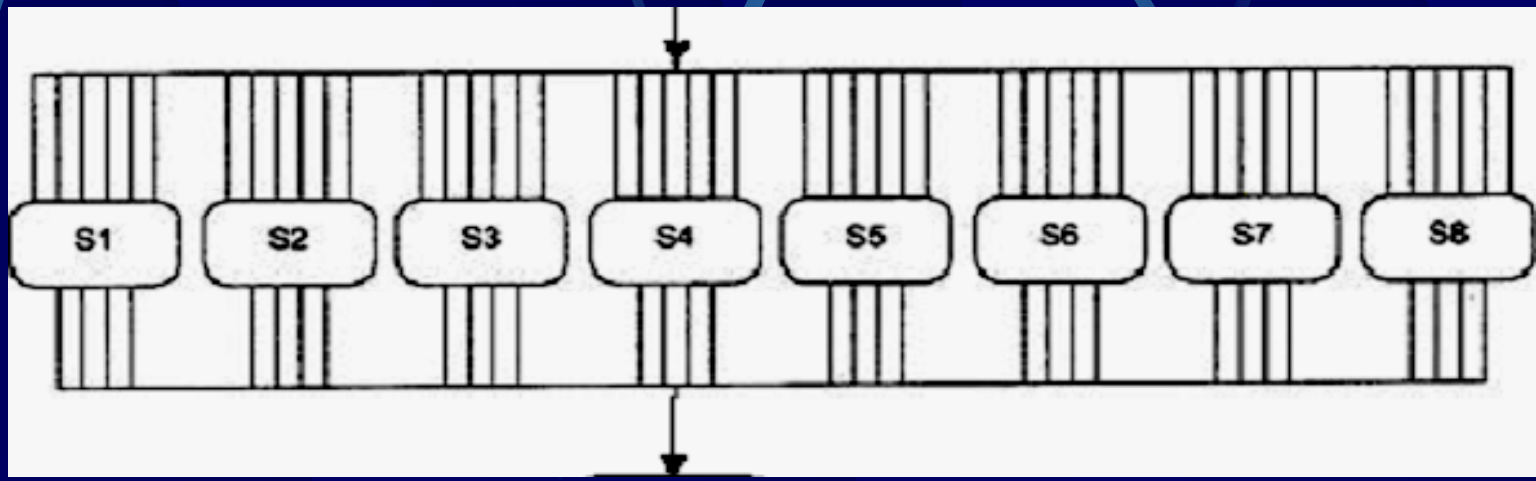
- 扩展的方法是将32bit明文中的一部分数据重复后，变为48bit输出，再与48bit密钥模二加。

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

由32位扩展到48位的序列编排

(2) 压缩运算

- 将加密后的48bit数据再压缩成32bit。办法是：分成8组，每组6bit,分别从8个数据表（称为S盒）查表，各自得到4bit的压缩数据。



由48位压缩到32位的S盒数据对照表

行	列	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	10	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	11	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

- 第一分组数据$x=(x_0x_1x_2x_3x_4x_5)=(001101)$，应查$S_1$表；
- 由$(x_5x_0)=(10)$ 确定查第三行；
- 由$(x_4x_3x_2x_1)=(0110)=6$确定为6的这一列；
- 交叉处数据为2，即$(y_3y_2y_1y_0)=(0010)$，表明S盒的输出为$y=(y_0y_1y_2y_3)=(0100)$。→*由48位压缩到32位*

(3) 置换运算

- 压缩运算后的8组数据（每组4位，共32位），还要进行一次坐标变换，才送回去与左半32bit进行模二加。

	Y_0	Y_1	Y_2	Y_3
S_1	1	2	3	4
S_2	5	6	7	8
S_3	9	10	11	12
S_4	13	14	15	16
S_5	17	18	19	20
S_6	21	22	23	24
S_7	25	26	27	28
S_8	29	30	31	32

→

	Y_0	Y_1	Y_2	Y_3
S_1	16	7	20	21
S_2	29	12	28	17
S_3	1	15	23	26
S_4	5	18	31	10
S_5	2	8	24	14
S_6	32	27	3	9
S_7	19	13	30	6
S_8	2	11	4	25

4、子密钥生成

（1）去校验位：64bit密钥每8bit为一个校验段，末位是奇校验位。去掉第8，16，24，32，40，48，56，64位后，剩下的56bit为有效密钥；

（2）置换处理：将 56bit的有效密钥分成两半，左、右各28bit，并且排列次序也重排 。

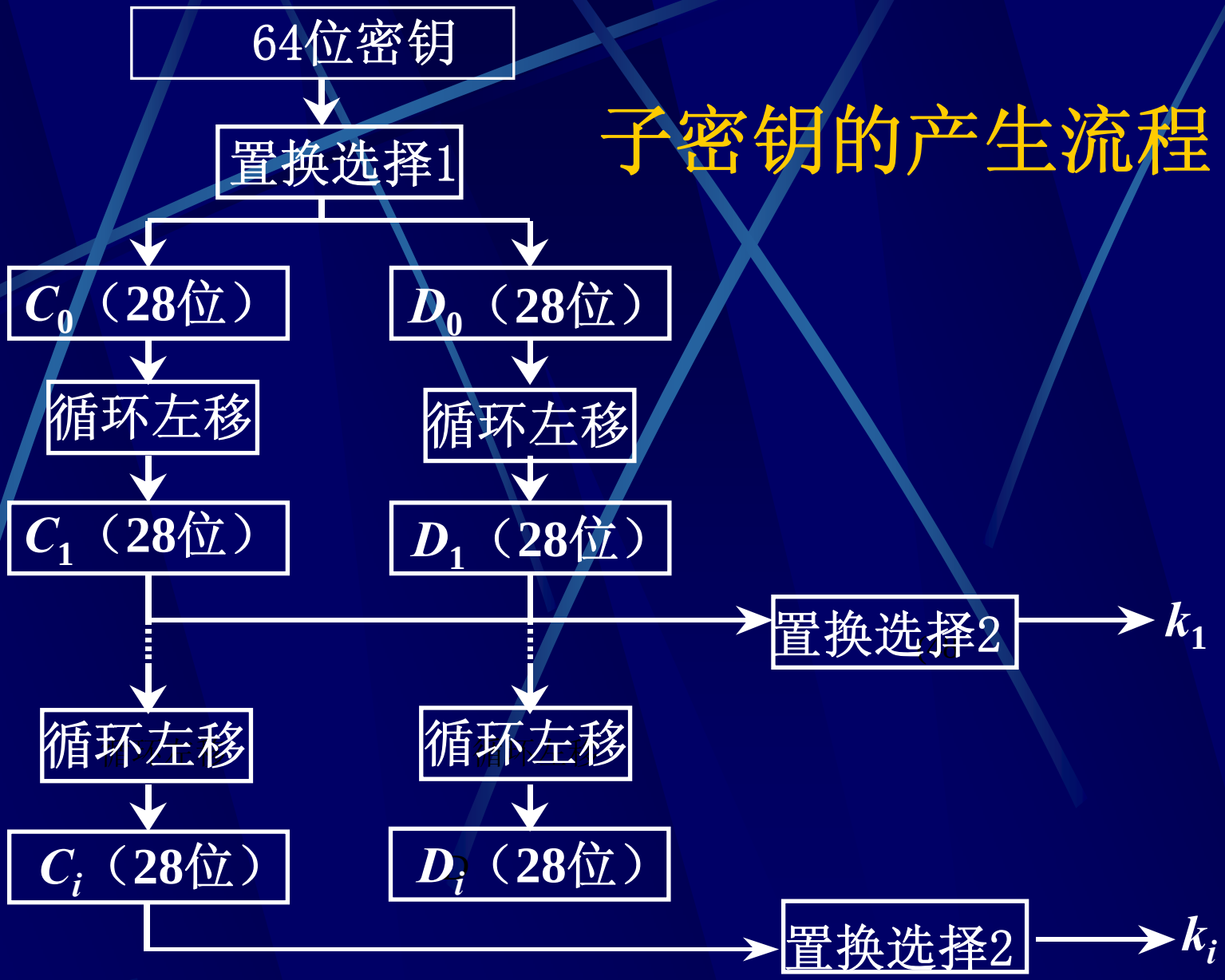
C_0（左28bit）							D_0（右28bi）t						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

(3) 循环移位：每轮迭代加密所取用的子密钥都不相同。首先将两表数据依次进行不同位数的循环左移：（第1、2、16轮移1位，其它轮移2位）

(4) 定位选择：然后从循环移位后的56bit的数据中去掉9，18，22，25，35，38，43，54 这 8位，留下的48bit作为加密用的子密钥。并且按下表对号入座，再次将排列次序进行置乱。

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

子密钥的产生流程


64位密钥
置换选择1
C_0（28位）
D_0（28位）
循环左移
循环左移
C_1（28位）
D_1（28位）
置换选择2
k_1
循环左移
循环左移
C_i（28位）
D_i（28位）
置换选择2
k_i

5.3.3、DES解密算法

- $DES^{-1}(C) = DES^{-1}(DES(M))$

$= (IP^{-1}\ T_1T_2\cdots T_{15}T_{16}\ IP) \cdot (IP^{-1}T_{16}T_{15}\cdots T_2T_1\ IP)$

解密　　加密

- 首先： $IP \cdot IP^{-1} = 1$
- 随后，两个T_{16}的作用相抵消：$T_{16}T_{16} = 1$
- 以后各对T_i 一一相消:$T_iT_i = 1$

$T_{16}T_{16}=1$的证明：

加密时的T16

$R_{16}= R_{15}$；

$L_{16}=L_{15}\oplus f(k_{16}, R_{15})$

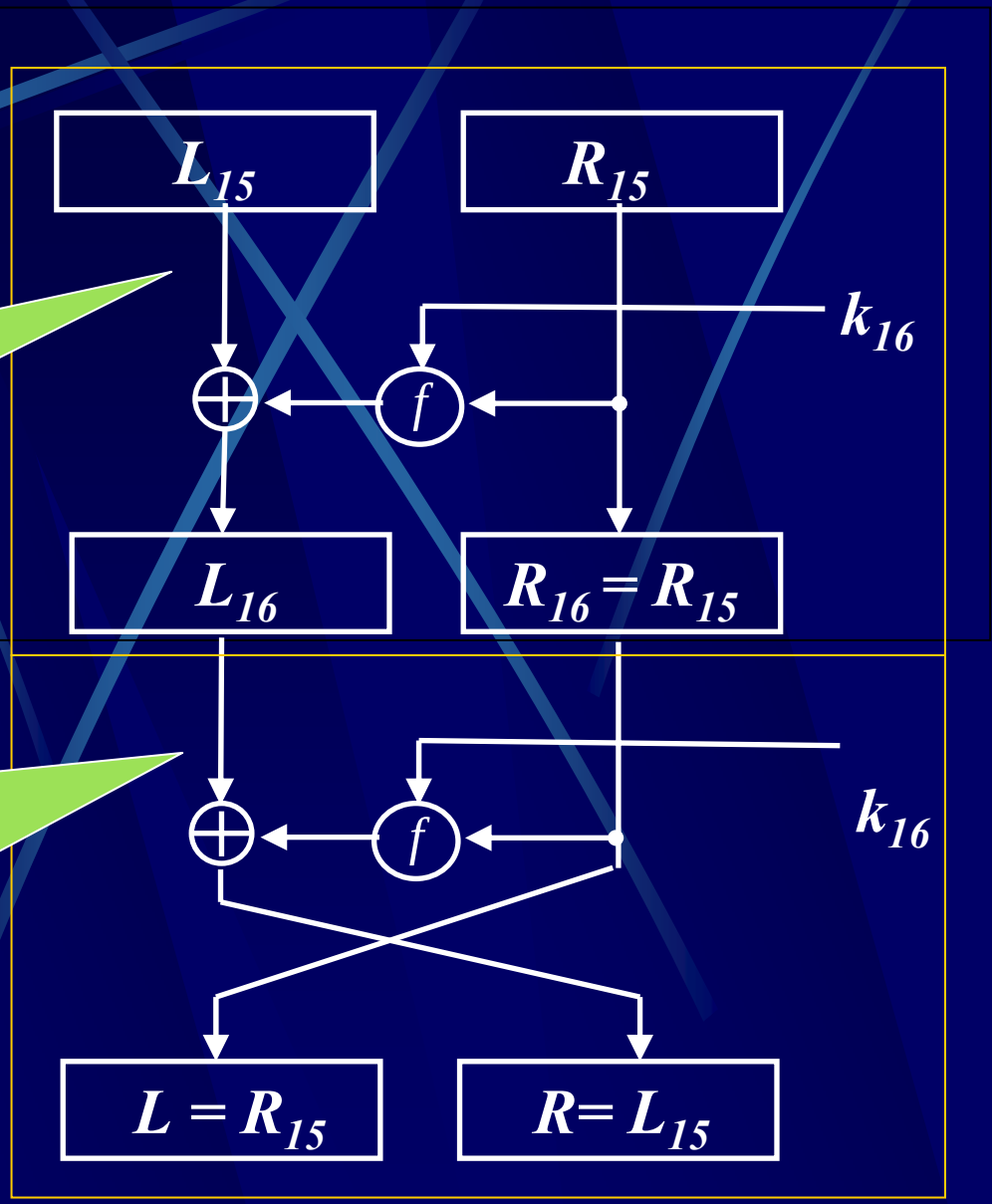
解密时的T16

$L=R_{16}= R_{15}$；

$R=L_{16}\oplus f(k_{16}, R_{16})$

$=[L_{15}\oplus f(k_{16}, R_{15})]$

$\oplus f(k_{16}, R_{15})= L_{15}$



本节要点（小结）

1. 分组密码的基本概念：

(1)分组加、解密方式。

(2)加密变换与解密变换。

2. DES分组加密流程：

（1）初始变换与逆变换；

（2）换位加密与16次迭代；

（3）f 函数（扩展、压缩与模2加）；

（4）子密钥的产生。

2. DES解密算法：

第5章 密码

5.4 公开密钥密码

（第21讲 2007.12.13.）

[温旧引新]

对称密钥系统(symmetric cipher)

- 解密是加密的逆运算：

$$D_k(\ ) = E_k^{-1}(\ );$$

- 解密与加密使用同样的密钥，
- 掌握了加密方法与密钥的人，就能解密。
- 这样的密码系统被成为**对称密钥体系**（也叫**单密钥制系统**）。

[今日学习]

5.5 公开密钥系统（asymmetric cipher)

- 参考文献：

1. 卢开澄：**计算机密码学** 清华大学出版社（**1998**年**7**月第二版）
2. 章照之：**现代密码学基础** 北京邮电大学出版社 （2004年4月第1版）

本节内容

- 公钥密码体系产生的背景
- 公钥密码体系的产生
- 公钥密码系统的特点
- 公开密钥体制的构建
- 现代密码学的新理念

外语关键词

公钥密钥体系： asymmetric cipher

计算复杂性：computational complexity

素数分解：factorization **of** prime number

离散对数：Discrete Logarithms

单向陷门函数：trap-door one-way function

公钥与私钥：Public Key and Private **Key**

5.5.1 公钥密码体系产生的背景：

1、对称密码体制面对的形势

20世纪末，电子商务与因特网迅速发展，一方面极大地方便了人们对信息的交换和共享，另一方面，也给安全通信提出了很多新的需求，传统的单钥制密码系统对此无能为力。如何更好地解决信息安全问题，已成为刻不容缓的任务。

- 一方面是越来越多的政务、业务、商务以及个人事务已经离不开网络。

- 另一方面是不容乐观的网络案件：计算机犯罪、病毒传播、黑客入侵呈逐年增加的趋势，造成严重的经济损失和社会危害。内容涉及金融欺诈、诽谤、非法入侵、知识产权、泄密等。

- 2000年《国家信息安全报告》揭示，当时80%的网站存在安全隐患，20%的网站有严重安全问题；每月都有一个省的信息港或证卷所被黑，银行被窃，主页被改。以9分为满分计算，中国的信息安全强度只有5.5分；

2、传统密码体制已不适应形势的发展

1. 功能上的缺憾

传统密码学不擅长认证功能。

2. 使用上的难题

密钥交换是对称密钥体制的难题。

3. 管理上的困难

密钥管理问题成了通信的瓶颈。

3、非对称密钥体系开始出现

在这种情况下，一种新的密码体制出现了。它就是公开密钥系统，也称**非对称密钥体系**（也叫**双密钥制系统**）。从理论到实践上逐步解决了对称密钥存在的问题，并从根本上提升了密码学的理念，标志着密码学进入了崭新的发展阶段。

5.5.2 公钥密码体系的产生：

一种新理论新技术的产生，一方面是因为实际问题的需求，另一方面，也是科学发展的必然结果。**创新成果的产生，源于对前人成果的不断积累与改进，也离不开发明者开创性的思维。**

我们将通过三个典型案例的介绍，向同学们展示公钥密码体系的产生过程。

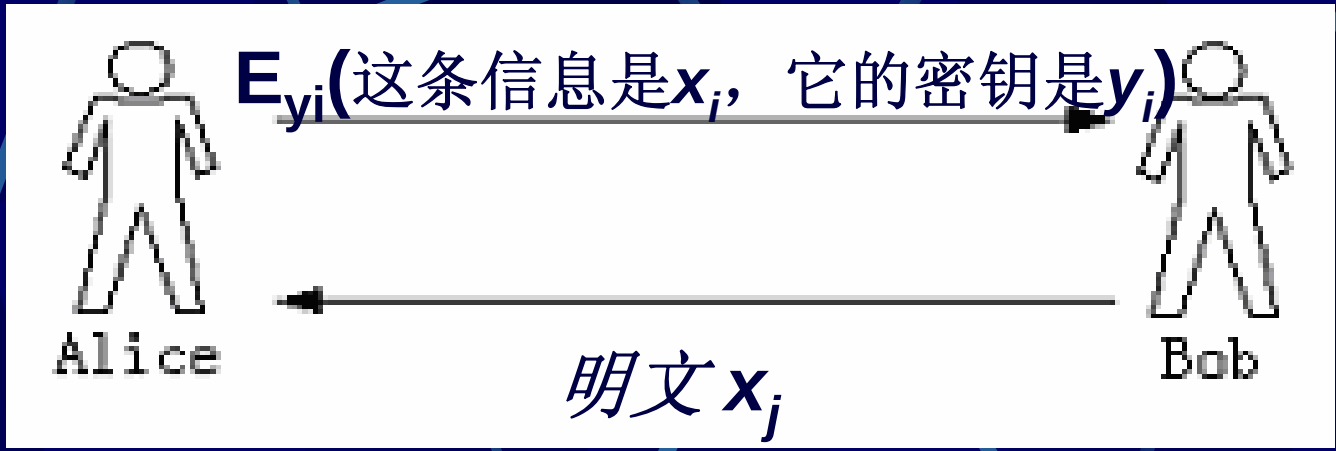
1.Merkle难题 （Puzzle）：

起初人们并没有想到去发明一个公钥密码，只不过是为了解决在普通信道中完成密钥交换问题而进行了一系列探索和努力。

1974年，**Merkle**为解决对称单钥制密钥体系的密钥交换问题，提出了一个设想。

Merkle密钥交换协议如下：

（1）Alice向Bob发送100万条消息；



- x_i是0 ~ 999999之间一个任意的但又各不相同的随机数；
- Y_i也是0 ~ 999999之间任意但又各不相同的随机数，分别作为每条消息的密钥。
- Alice秘密保存所有y_i与x_i的密钥对照表后，就把这100万条消息分别用所宣布的y_i加密，发给Bob。

(2) Bob收到100万条消息，从中任选一条，然后遍历100万个密钥进行尝试，总有一个y_j可将其解密，从而得知x_j的值。

(3) Bob以明文形式将x_j发给Alice

(4) Alice收到x_j，即知Bob所选的是哪个y_j，从而二人都有了同样的密钥y_j。

Merkle安全性探讨

- 非法窃听者**Eve**，即使收到了全部往来的明文和密文，为了找到x_j对应的y_j，他需要对约**100**万条消息一一作遍历**100**万个密钥的尝试。
- 若尝试**1**个密钥耗时**1**毫秒，尝试**100**万个密钥耗时1000秒≈**17**分钟，这才解译了一条消息，**100**万条消息约31年才能全部试完。
- 而合法用户乙最多只需要**17**分钟。

此设计方案显然是不现实的，但其思想是很先进的。他用的是公开的算法，通过普通（不安全）信道完成了密钥交换，其保密机制是基于计算复杂性，安全理念是基于破译的时效性，其设计理念已经进入了公钥密码的大门。

2 双钥制的提出

1976年11月，美国斯坦福大学电气工程系研究生W.Diffie和副教授Helman在IEEE上发表了题为“密码学新方向“的学术论文，利用离散对数复杂性实际地解决了单钥制的密钥交换问题。

(1) 离散对数问题

设p一个为大素数，已知x和b，不难求出：

$$y = b^x \mod p$$

例如，$y=3^6 \bmod 7=729 \bmod 7=1$;

然而若已知y和b，求逆运算却十分困难：

$$x = \log_b y \mod p$$

为了求上例的逆运算：$x=\log_3 1 \bmod 7=?$ 需要算出全部数据后，查表求反函数：

x	1	2	3	4	5	6
y	3	2	6	4	5	1

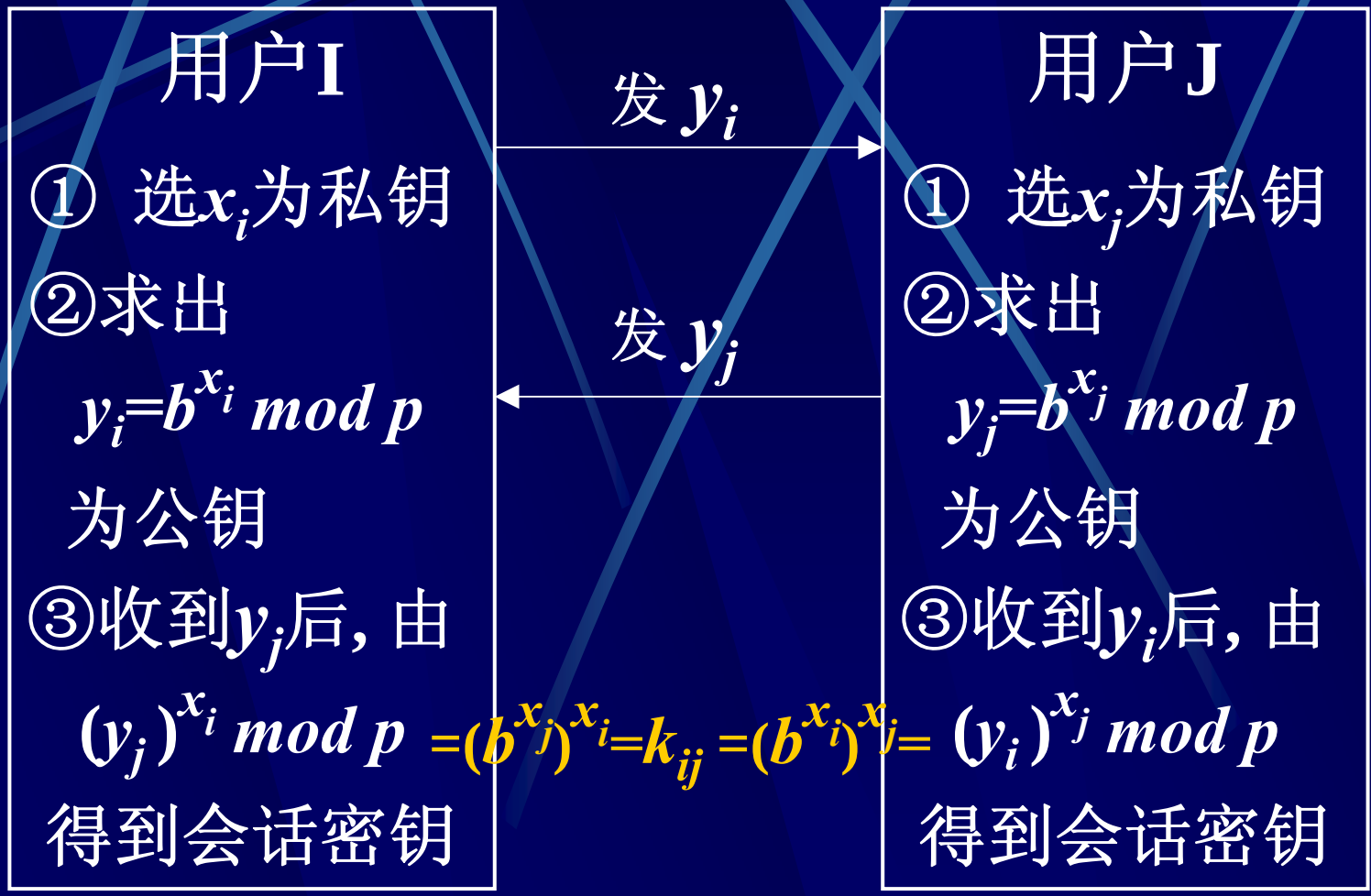
遍历法虽然是万能的方法，却是个笨方法。设想如果p是一个六位数，计算大约100万个x与y的对照表，估计也不会少于1毫秒吧，而如果p是一个一百零六位的数，按照同样的计算速度就需要10^{100}毫秒$=3.17\times10^{89}$年。

实际上，上面的估计还是太保守了。随着数位的增大，加减乘除都会变得更加繁杂，对于大数的乘方运算与求模运算，计算量已经远远增大，不能再沿用计算小数的速度来估计了。

（2）计算复杂性

- 所谓复杂性指计算所必须的基本运算步骤数量，它决定了计算所必须的机时和占用的计算机资源。
- 计算复杂性理论告诉我们，随着数位N的增大，计算量从小到大的次序是：常数→对数函数$logN$→线性函数aN→二次函数N^2→三次函数N^3→多项式函数N^d→亚指数函数2^{logN}→指数函数2^N或10^N。
- 多项式以下的运算都认为是有效算法，属于可解问题（*P*类），而比多项式发散更快的算法都被认为是难解问题（*NP*类，*NPC*类）。
- 数学上已经证明离散对数是非正常（*NP*）类复杂问题。

（3）Diffie和Helman的方案：



④ 尽管公布了y_i，y_j和p，基于离散对数的困难性，任何第三者难以求出x_i和x_j，因此无法获得k_{ij}。

Diffie和Helman虽然讨论的仍然是单钥制的密钥交换问题，但他们的观念是很超前的。他们采用的算法是公开的，他们的保密机制是基于计算复杂性，并且他们首次提出了双密钥的观点。因此这篇论文被认为是现代密码学的开篇之作。

3 RSA公开密钥体系

- **1978**年美国麻省理工大学的 **Ron.Rivest**、**Adi.Shamir** 和 **Len.Adleman** 提出**RSA**公钥密码体系。
- 它是第一个具有实用价值的公钥密码算法.
- 它是应用最广泛的公钥密码算法。
- 它的原理是根据数论的欧拉定理。
- 它的安全性是基于大数分解因数的复杂性。

1、RSA算法原理

设p和q为两个大素数，计算 $n = pq$

在小于n的 $n-1$个正整数中：

有$p-1$个数：$q, 2q, 3q, \ldots\ldots, (p-1)q$ 含因子q；

有$q-1$个正整数：$p, 2p, 3p, \ldots\ldots, (q-1)p$含因子$p$；

此外，其余的数都应当与n互素，其数目为：

$(n-1)-(p-1)-(q-1) = n-p-q+1 = pq-p-q+1 = (p-1)(q-1)$；

令： $\phi(n) = (p-1)(q-1)$

叫做欧拉数，代表与n互素的同余类的个数（除以n后余数相同的整数为一个同余类）。

- Euler(欧拉) 定理：若m与n为互素的正整数，则：　　$m^{\phi(n)} \equiv 1 \pmod{n}$
- 若k是任意整数,则 m^k也与n互素，于是有

  $m^{k\phi(n)} \equiv 1 \pmod{n} \Rightarrow m^{k\phi(n)+1} \equiv m \pmod{n}$,

  （对任意$0 \le m \le n$）
- 只要选择e,d, 满足$ed \equiv 1 \bmod \phi(n)$ ，

  即　$ed=k\phi(n)+1$,（d 叫做e的模$\phi(n)$ 逆元）
- 于是上式变为：

$$m^{ed} = m \pmod{n}$$

设m为明文，由：$m^{ed} = m \mod n$出发可以设计出一种加、解密方案：

- 如果用 $m^e \mod n$作为加密计算结果，就可用$(m^e)^d = m \mod n$来解密；
- 如果用 $m^d \mod n$作为加密计算结果，就可用 $(m^d)^e = m \mod n$来解密。

- 一般取 **(*n*, *e*)** 为公钥，对明文***m***进行加密得到密文：

$$c = m^e \bmod n$$

- d为私钥，解密算法是：

$$m = c^d \bmod n$$

- 当然也可以用私钥加密：

$$c' = m^d \bmod n$$

用公钥解密：$m = c'^{\,e} \bmod n$

[例1]取两个小素数p=11，q=3来演示RSA的加、解密过程。

解：先求出：$n=pq=33$, $\phi(n)=(p-1)(q-1)=20$；

在$1< e < \phi(n)$中,取e=7为公钥(与20互素即可)

不难看到：$7\times 3 \mod 20=1$

d= 3（为私钥）；

设明文$m=(010)_2=2$，则:

密文 $c = 2^7 \mod 33 = 128 \mod 33 = 29$；

解密：$m = 29^3 \mod 33 = 24389 \mod 33 = 2$；

2、RSA的安全性

因为e和d是在模$\Phi(n)$运算下的互为倒数，当e与d被确定后，应当把p、q和$\Phi(n)$都销毁，仅留下n，e和d是无法互相推导的。

除非能将n分解，求出p和q，而大数的分解因数是Np类难题。因此**RSA**的安全得到保证。

5.5.3 公钥密码系统的特点

1. 从形式上讲：由对称单钥制到非对称的双钥制

- 加密和解密所用的算法和密钥不同，而且二者不可互相推导。
- 加密过程：$C = E_{k1}[M]$
- 解密过程：$M = D_{k2}[C]$

2. 从使用上讲：由隐蔽密钥制到公开密钥制

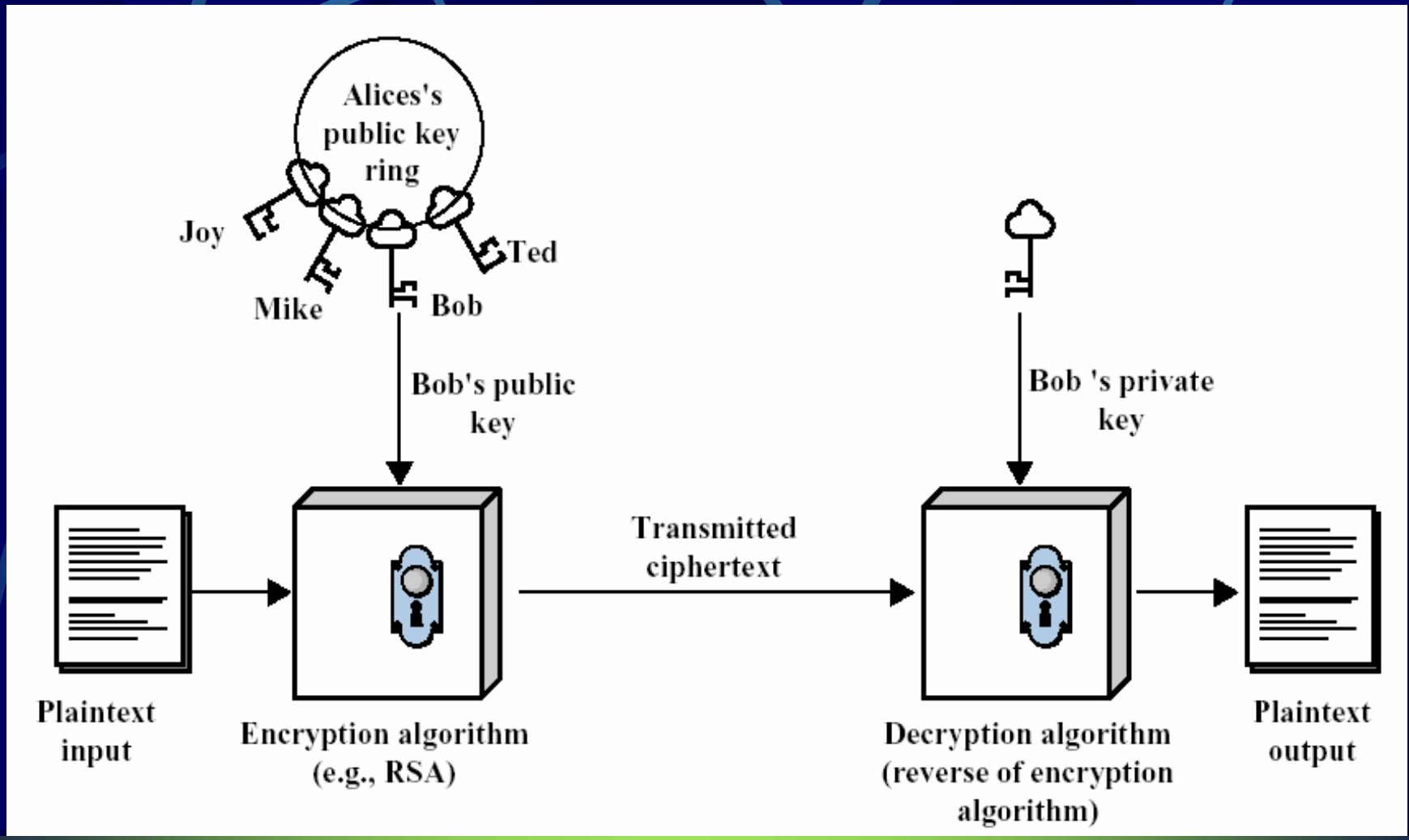
- 双钥制的出现引入了一个新观念：可以将一对不对称的密钥中，一个公开，叫做公钥，另一个保密，称为私钥，使用起来会更方便。
- ***N***个用户的系统，每个用户各自分得一对公钥与私钥，网络管理员只须保管（不必隐藏）***N***把公钥，私钥由用户个人保管，也解除了网管的负担。

3. 从功能上讲：由单纯保密功能到保密认证双重功能

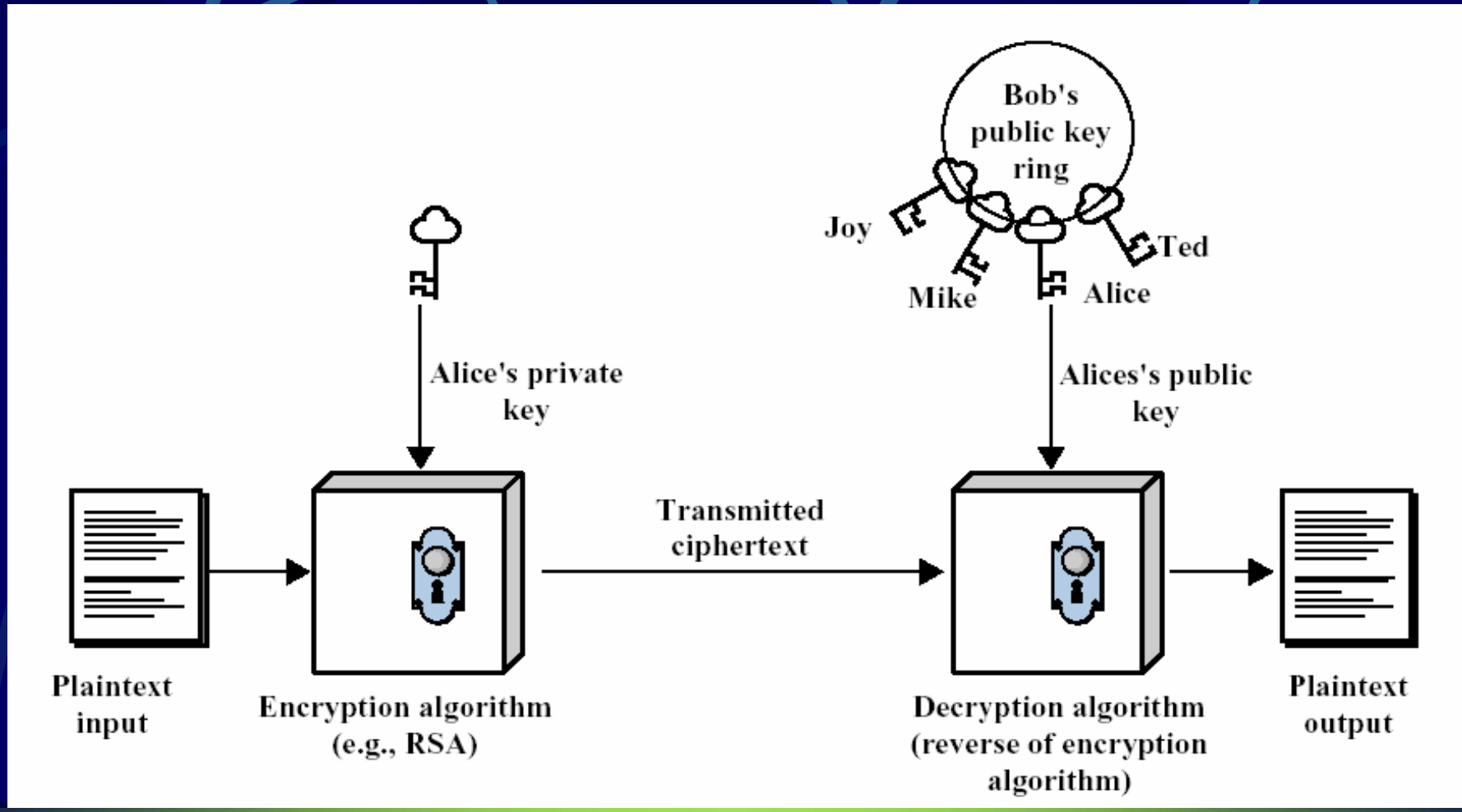
（1）当用于保密功能时：发信人用收信人的公钥加密，收信人用自己的私钥解密。其他人因没有密文所要求的私钥而不能解密。

（2）当用于认证功能时，发信人用自己的私钥加密，收信人用发信人的公钥解密，只要能解译密文，就表明这个密文是掌握私钥的那个人发出，别人做不出这样的密文。

基于公开密钥的加密过程



基于公开密钥的认证过程



5.5.4 公开密钥体制的构建

- 单钥制密码体系中，解密使用的密钥与加密密钥相同；能加密的人就能解密。
- 而公钥密码体系中，加密密钥公开了，解密密钥必须与加密密钥不同，并且从加密密钥无法推导出解密密钥；

- 单钥制密码体系中，解密用加密的逆运算实现，数学上两个算法互为反函数；
- 而公钥密码体系中，加密算法的反函数应当是不可行的。
- 可见构建公钥密码体系，需要找到除了反函数之外的另外一种逆运算方法。
- 这个方法的实现还需要使用不同于加密算法的另外一把密钥。

数学上是否存在这样的算法？

单向陷门函数
（one way trapdoor function）

- 单向陷门函数是满足下列条件的函数 **f(x)**：

(1)给定**x**，计算**y=f(x)**是容易的；**(**加密可行**)**

(2)给定**y**，计算$x=f^{-1}(y)$是不可行的; **(**安全有保障**)**

(3) 对给定的任何**y**，还存在另一种计算**x** 的方法 $x=f_k(y)$，在已知 **k** 的情况下完成计算是容易的. **k** 就是密钥。**(**合法用户得以解密**)**

RSA就给出了一种单向陷门函数

- 计算$c=m^e \pmod n$是容易的。
- 因为逆运算 $m=\sqrt[e]{\lambda \quad n+c}$ 十分复杂，且模商 λ 未知，引起多解性。因而它是单向函数，逆运算不可行。
- 对于已知密钥 d 的人，还存在另一种计算 m 的方法： $m=c^d \pmod n$
- RSA的安全性在于攻击者用逆运算求解不可行，由公钥e和n出发计算私钥d也是不可行的。

现在已找到多种公开密钥体制：

- **RSA**公钥密码体制
- 二次剩余公钥密码体制
- 背包公钥密码体制
- **Elgamal**公钥密码体制
- **MxEiece**公钥密码体制
- 椭圆曲线公钥密码体制

5.5.5 现代密码学的新理念：

1. 从基于算法的神秘性到基于算法的复杂性

- 传统的保密观念，寄托安全性于某种鲜为人知的奇妙算法上。其安全是暂时的，侥幸的，脆弱的。
- 现代密码学基于算法的复杂性，**不靠神奇靠麻烦**。其安全是牢固的，可信的，科学的。
- 从基于算法的神秘性到基于算法的复杂性是现代密码学设计理念上一次重大转变。不再关注从密文中提取信息有无可能性，转变为研究从密文中提取信息有无可行性。

2．算法可公开性

- 由于算法失去了保密价值，算法可以公开，让它在攻击中不断改进和完善，并以此显示其安全的坚固性。
- 算法公开了，合法用户与非法用户的区别在那里呢？合法用户拥有密钥，解译密文十分容易；非法用户没有密钥，破译密文谈何容易；**不藏算法藏密钥**是设计理念上一致的观点。

3 安全的相对性

- 当然，应充分估计破译者的计算能力和计算技术未来的发展，从这个意义讲，破译只是时间和金钱的问题。
- 如果破译工作所花的代价大于秘密本身的价值，或破译花费的时间大于秘密的有效期，则破译失去意义，而该保密系统就可以认为是安全的。这才是实事求是的科学的保密观和破译观。**不保绝对保相对**是密码设计理念上的又一个转变。

本节要点（小结）

1. 公开密钥体制的基理念：

(1) 系统安全基于计算复杂性，算法可以公开。

(2) 加密解密使用非对程的双密钥，一把可以公开。

(3) 科学的可靠的系统设计思想。

2. 公钥密码的功能：

(1) 用于保密：用接收方的公钥加密，私钥解密。

(2) 用于认证：用发送方的私钥加密，公钥验证。

3. RSA公钥密码的原理与方法：

(1) 系统搭建：

(2) 加、解密方法：

[讨论题]

- 如果要求系统所发的消息同时具有保密与认证两个功能，应当按照怎样设计加密程序？（究竟是先用自己私钥加密再用对方公钥加密好，还是先用对方公钥加密再用自己私钥加密好？为什么？）

[作业题]

P181页习题五　第4、5题

第5章 密码

5.6 数字签名

（第20讲 2007.12.18.）

上节回顾

1. 公开密钥体制的新理念：

(1)系统安全基于计算复杂性，算法可以公开。

(2)加密解密使用非对程的双密钥，一把可以公开。

(3)科学的可靠的系统设计思想。

2. 公钥密码的功能：

(1)用于保密：用接收方的公钥加密，私钥解密。

(2)用于认证：用发送方的私钥加密，公钥验证。

3. RSA公钥密码的原理与方法：

(1) 系统搭建：

(2)加、解密方法：

本节的主要内容

- 数字签名的基本概念
- Hash函数
- MD5算法
- 数字签名的标准DSS

外语关键词

电子商务：**Electronic Commerce**

数字签名：**digital signature**

身份识别：**Identify Certified**

用户认证：**User authentication**

完整性验证：**integrity verification**

不可否认性：**Non-repudiation**

5.6.1 数字签名的基本概念

- 多年来，手写签名因为其方便可靠，已成为签署文件、合同、甚至外交条约的必要手段。
- 传统的（手写）签名借助纸张将文件与签发人的笔迹（认可凭证）有效的结合在一起，使文件具有了可认证性。
- 由于电子文档的易拷贝性和可粘贴性，使机械地照搬手写签名到电子文档上的做法失效，必须引入功能相似，但方式不同的有效认证方式。

签名应有以下功能：

（1）签名是可以验证的，收到签字的人容易由签字确定来信人；

（2）签名是不可伪造的，除了签字者之外的任何人无法实现这个签字；

（3）签名是不可重用的，一个签字只对一个文件生效，无法用于其它文件；

（4）签名是不可改变的，一旦签字发出便不能再作修改；

（5）签名是不可抵赖的，存在某种方法充分证明该签字确为发信人所为；

电子文件实现签名的方式

- 数字签名是伴随电子文档的一段数字或代码串。
- 它应能体现发送方身份，它含有发送人所专有的、别人无法伪造的信息（如私人密钥）。
- 它必须与所发送的电子文档相关联，不是机械地放在一块，而是由该电子文档所产生，能体现该文档内容的数字或代码串。
- 比如，用发信人私钥加密所发信件得到的密文，就具有上述特征，它就是一种数字签名。

数字签名的作用：

- 鉴别信函、文件的来源：是否来自可靠的信源。
- 验证发信人身份：是否为合法的发信人。
- 鉴别信函、文件的真伪：是否被篡改或替代过。
- 确定信函文件的完好性：是否有传输差错。
- 解决争议：为判定是非提供依据（法律凭证）。
- 证明网络媒体知识产权：提供创作者的印记。

5.6.2 单向散列（Hash）函数

- 单向散列函数（**Hash**函数）也叫杂凑函数。它实际上是一种特殊的压缩算法，把任意长度的消息压缩成一定长度（如**128*bit***）的代码串（称之为消息摘要）。
- 消息摘要常作为原消息的“特征码”或“缩影”，被保存备案。
- 消息摘要更多用途是被私钥加密，形成短小精练的数字签名，附于原文档后，以备供认证。

单向散列函数的主要功能

- 单向散列函数虽然经常使用在数字签字中，但单向散列函数不等于数字签字。
- 单向散列函数最基本的功能是对文件作“完整性”检验。只有用私钥加密后才能构成数字签名。
- 当原文档*m*被传输或转存后，可再次计算其摘要值，比较是否发生变化（因为摘要值“放大”了文档的差别），用以判断原文档是否被有意或无意改动过。

对单向散列函数的要求：

(1) 函数h=H(m) 应能对任意长度的消息m作计算。

(2) 计算值h是容易的，而由h计算m是不可行的（单向性）。

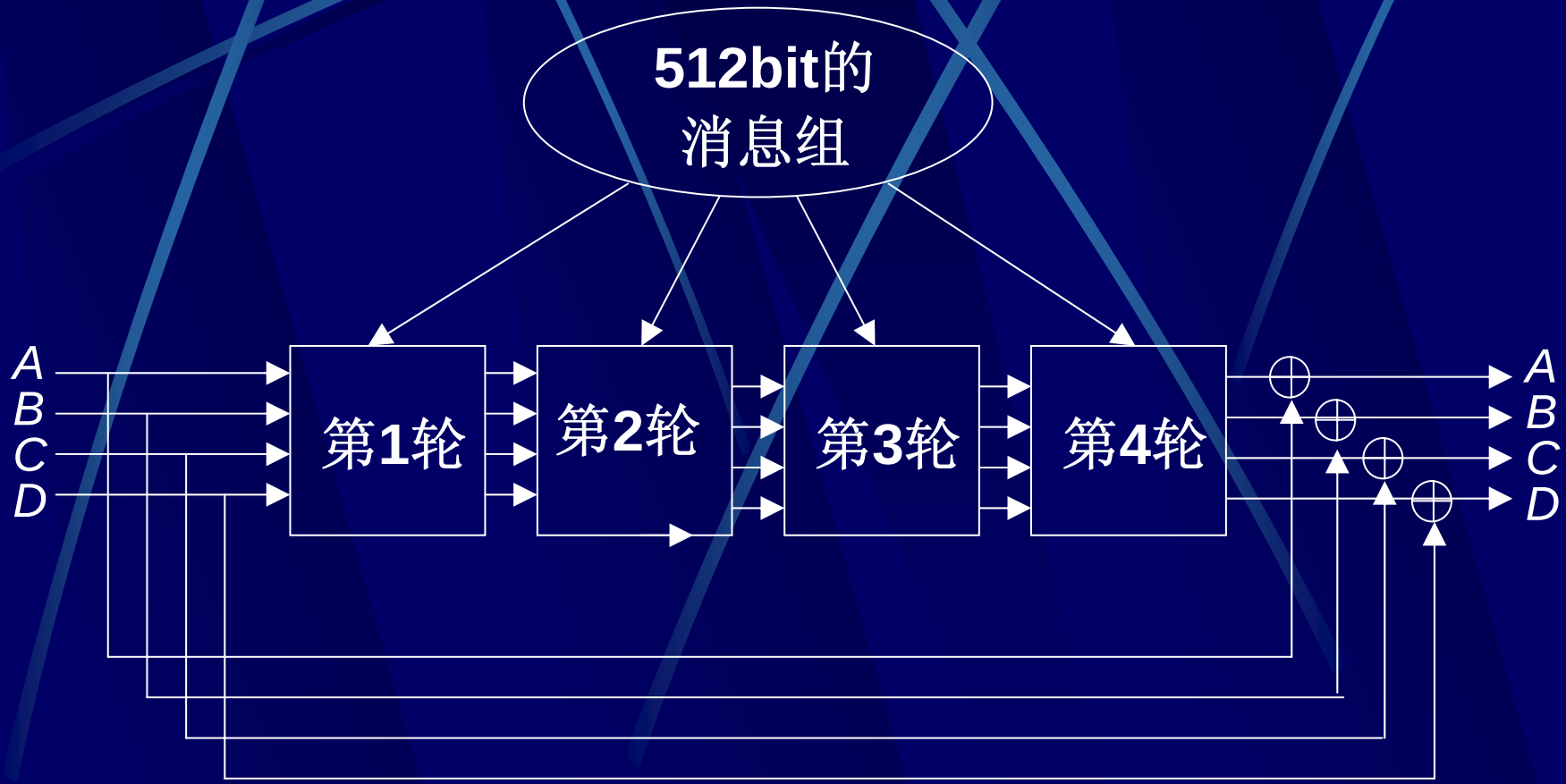
(3) 对于算法H(m)，要找出两个不同消息m_1和m_2有相同的摘要值：$H(m_1)=H(m_2)$ 也是非常困难的（或者说是不可行的）。

(4) 散列算法应当保证原消息的每一符号均与压缩结果相关联，以至于任意改变原消息的一个符号时将导致其信息摘要一半以上的bit变化。

5.6.3 MD5算法

- 能够实现散列函数的算法很多。MD5是一种用计算机软件实现的Hash函数。
- 首先将明文按512*bit*分段，依次处理各个段落。
- 把512bit的一个段落分成32*bit*的16小段，装入数组M(0),M(1),……M(15)，进行处理。
- 主循环4轮，每轮进行16次操作，每次处理1个小段。
- 经4轮处理，将512*bit*的分段压缩为4个32bit的变量，再与处理前的4个32bit变量模2加，所得结果作为初始值，进行下一个512*bit*分段的处理。
- 各轮结果都揉搓到一块，最后结果仍然只有128bit

MD5主循环4轮处理流程图



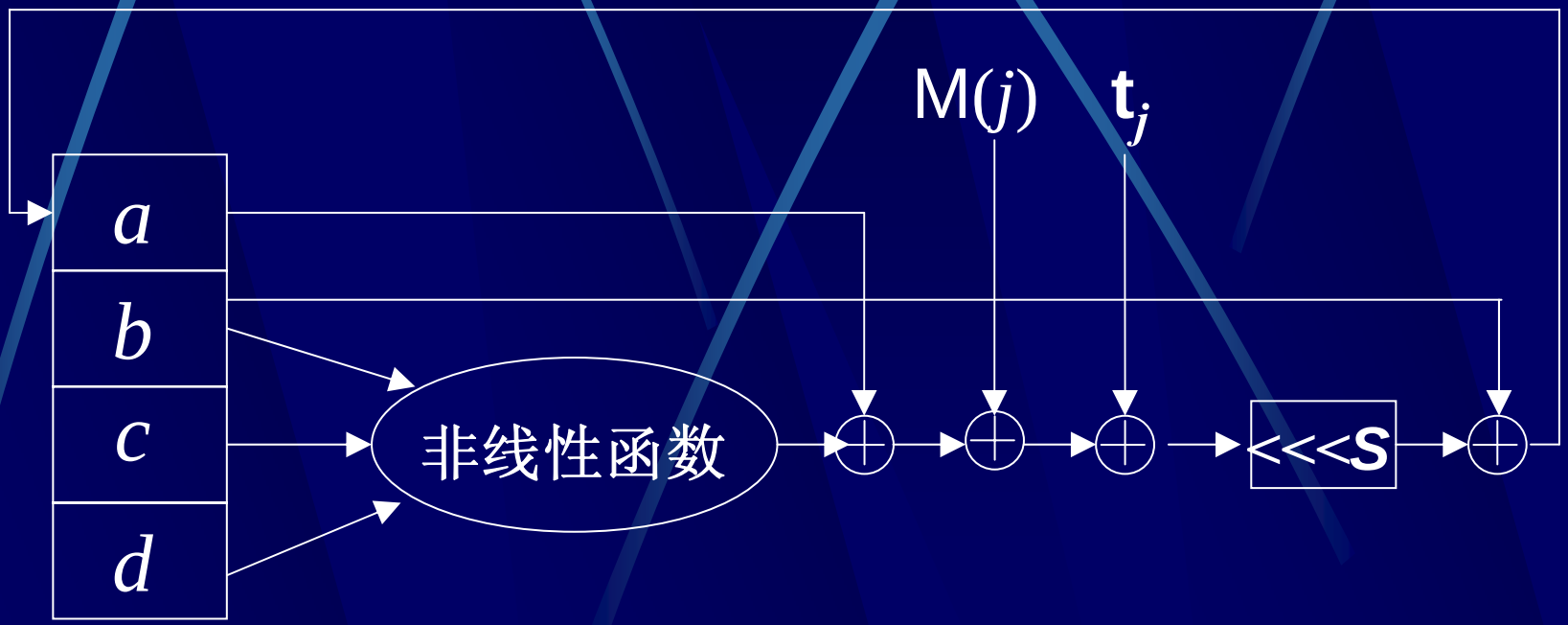
- 四个变量A,B,C,D 的初值为：A=67452301，B=*efcdab*89，　C=98*badcfe*，*D*=10325476；
- 4轮处理分别采用不同的非线性函数为：

$F(x,y,z)=(x\wedge y)\vee(\bar{x}\wedge z)$；　$G(x,y,z)=(x\wedge z)\vee(y\wedge\bar{z})$

$H(x,y,z)=x\oplus y\oplus z$；　$I(x,y,z)=y\oplus(x\vee\bar{z})$

- 各轮处理都要进行16次操作，每次把其中一个32bit的数组单元与4个变量进行非线性混杂。
- 每次操作流程都十分相似，不同仅在于所调用的内容：*a,b,c,d* 以及*M*(*j*), t_j 与*S*;

*MD5*每次处理的流程图：



5.6.4 数字签名的标准DSS

- 数字签名标准（DSS）是美国国家标准和技术研究所（NIST）于1991年8月公布的标准。它所采用的算法叫DSA.

- 系统参数：p是一个512位到1024位的大素数它满足离散对数难解问题; q是160位的素数, 且 $q \mid p-1$，$g=h^{(p-1)/q} \bmod p$, $h<p-1$且能使$g>1$

定义：$y=g^x \bmod p$

$h(\bullet)$为公开的*hash* 函数。

公开密钥：$k_1=(p\ .\ q\ .\ g\ .\ y)$，私有密钥：$k_2=(x)$；

- **签名算法：** 数字签名签名者拥有私钥 ***x***，对于随机数***k***和待签消息***m***，生成签名：

$$Sig_{k_2}(m,k)=(r,s)$$

$$r=(g^{k} \bmod p) \bmod q$$

$$s=[h(m)+xr]\,k^{-1} \bmod q$$

- **验证算法：** 验证者有公钥 $k_1=(p,g,y)$，收到的明文***m***和签字***(r, s)***，验证：

$$Ver_{k_1}(m,r,s)=\left(g^{e_1}y^{e_2} \bmod p\right) \bmod q \equiv r$$

$$e_1=h(m)s^{-1} \bmod q$$

$$e_2=rs^{-1} \bmod q$$

证明：

由：$k = \{h(m) + xr\}s^{-1} \bmod q$

$$g^{e_1} y^{e_2} = \left(g^{h(m)s^{-1}} y^{rs^{-1}} \bmod p\right) \bmod q$$

$$= \left(g^{h(m)s^{-1}} g^{xrs^{-1}} \bmod p\right) \bmod q$$

$$= \left(g^{[h(m)+xr]s^{-1}} \bmod p\right) \bmod q$$

$$= \left(g^{k} \bmod p\right) \bmod q = r$$

本节要点（小结）

1. 数字签名的

(1) 对数字签名的要求。

(2) 数字签名的实现方式。

(3) 数字签名的作用。

2. 单向散列函数：

(1) 单向散列函数的特点。

(2) 单向散列函数的主要功能。

3. MD5与DSS：

(1) MD5的算法：

(2) DSS的算法：