密 码 学

第六讲　序列密码基础

王后珍

武汉大学国家网络安全学院

空天信息安全与可信计算教育部重点实验室

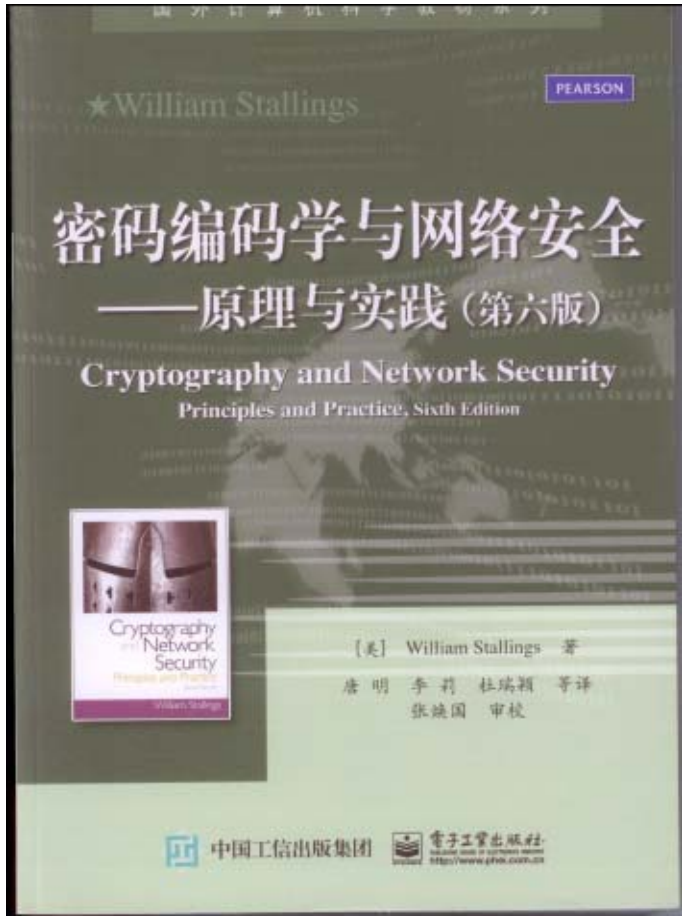
目 录

目 录

教材与主要参考书

教材

参考书



本讲内容

一、序列密码的基本概念

二、随机序列的安全性

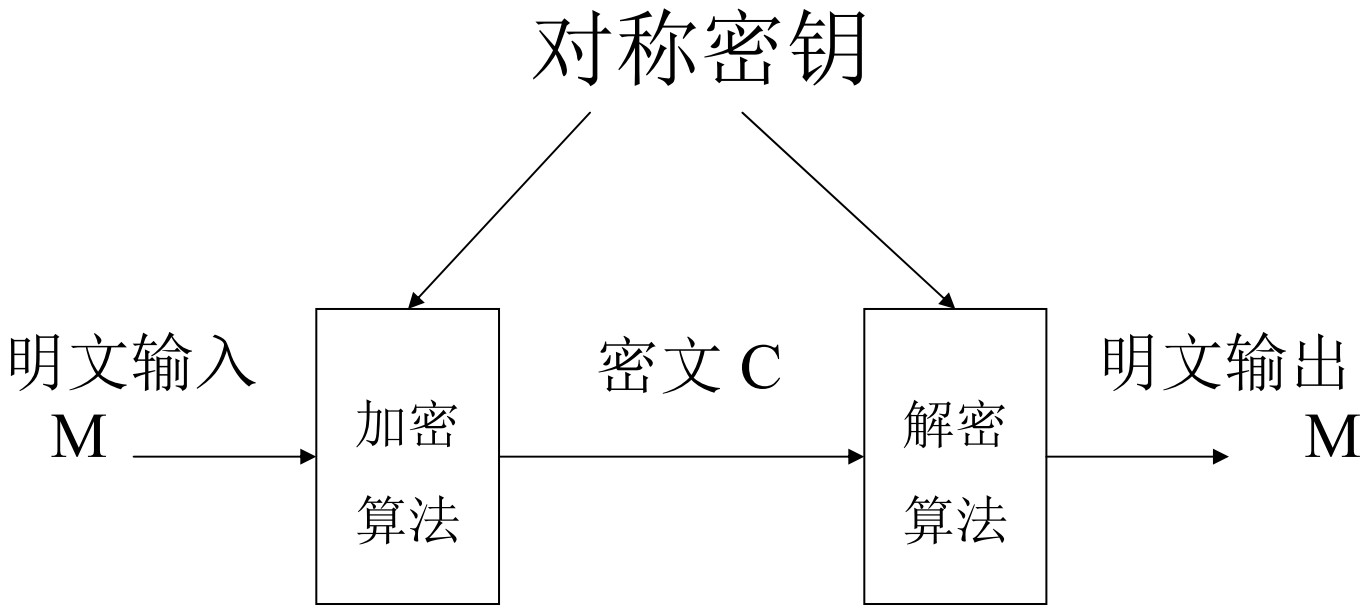
三、线性移位寄存器序列密码

四、非线性序列密码

回顾：密码的分类

- 按密码的历史发展阶段和应用技术分：手工密码、机械密码、电子机内乱密码和计算机密码
- 按密码转换的操作类型分：替代密码和移位密码

- 按保密程度划分，有理论上保密的密码、实际上保密的密码和不保密的密码
- 按密钥的类型分：对称密钥密码和非对称密钥密码
- 对称密码又称传统密码，按明文加密时的处理方法分：分组密码和序列密码



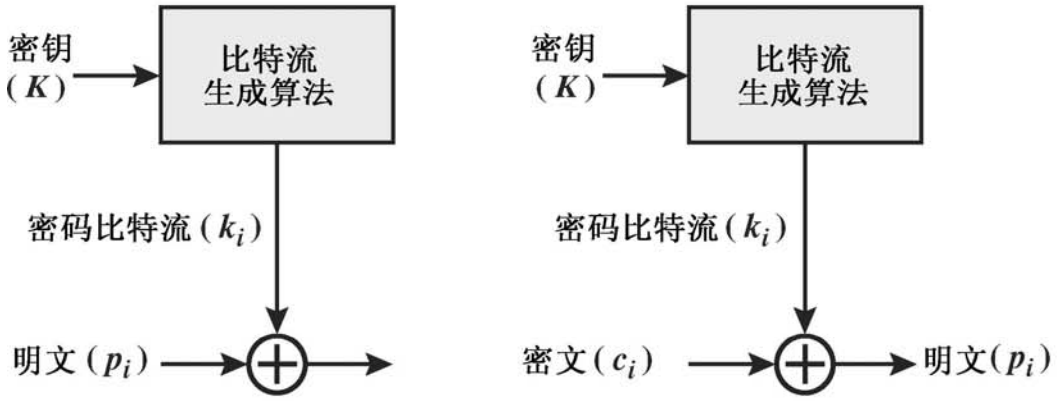
对称加密体制模型

序列(流)密码的概念

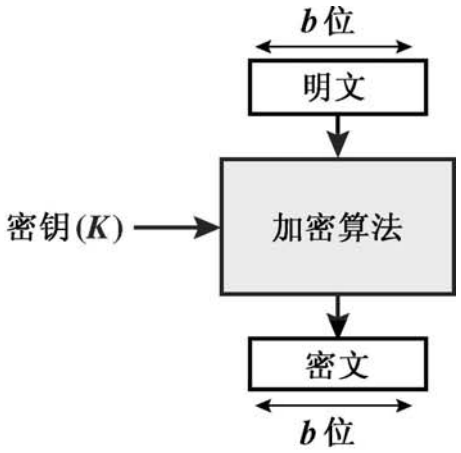
- 对称分组密码 与 序列密码的比较

	对称分组密码	序列密码
相同点	可以进行加密/解密	
	加密/解密时通常都使用相同的密钥	
	明文与密文一样长	
	速度快	
不同点	对数据采取分段处理	能处理无结构的数据流
	被处理数据之间存在着相关性	被处理数据之间不存在相关性
	能在软件环境中高速实现	较适合于硬件电路实现
	通常应用于商业领域	通常应用于军事领域
	通过在密钥的控制下对明文进行替代和换位来保密	通过将明文与密钥产生的密钥流叠加来保密

序列(流) 密码与分组密码的区别



(a) 使用算法比特流发生器的流密码



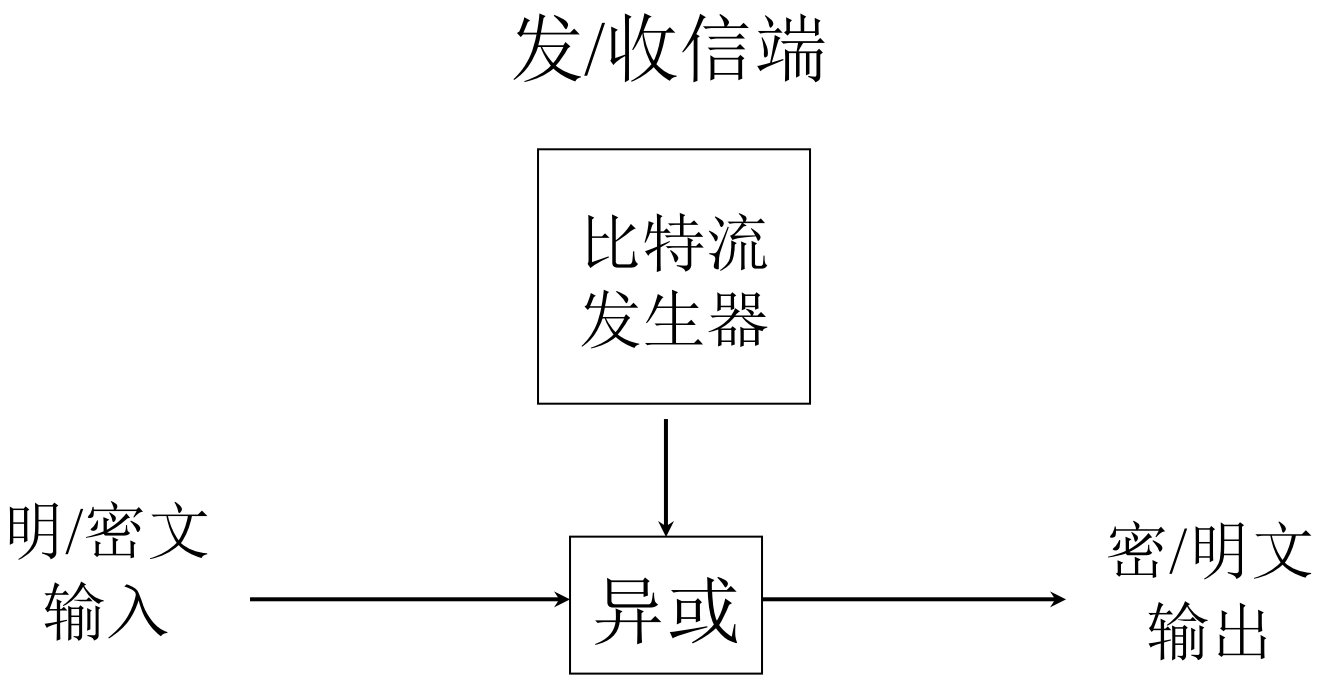
(b) 分组密码

- 序列密码
  - ✓ 序列密码加密过程是：把报文、语音、图像等原始信息转换为明文数据序列，再将其与密钥序列进行“异或”运算，生成密文序列发送给接收者。接收者用相同的密钥序列与密文序列再进行逐位解密(异或)，恢复明文序列。

- ✓ 序列密码加/解密的密钥，是采用一个比特流发生器随机产生二进制比特流而得到的。它与明文结合产生密文，与密文结合产生明文。
- ✓ 序列密码的安全性主要依赖于随机密钥序列。

Vernam密码与“一次一密”密码

- Vernam密码：
  - 加密：$c_i=m_i+k_i \bmod 2$
  - 解密：$m_i=c_i+k_i \bmod 2$
  - 例：11000⊕10010 ＝01010
- 一次一密：密钥为一个随机且不重复的字符序列。



序列密码

序列密码

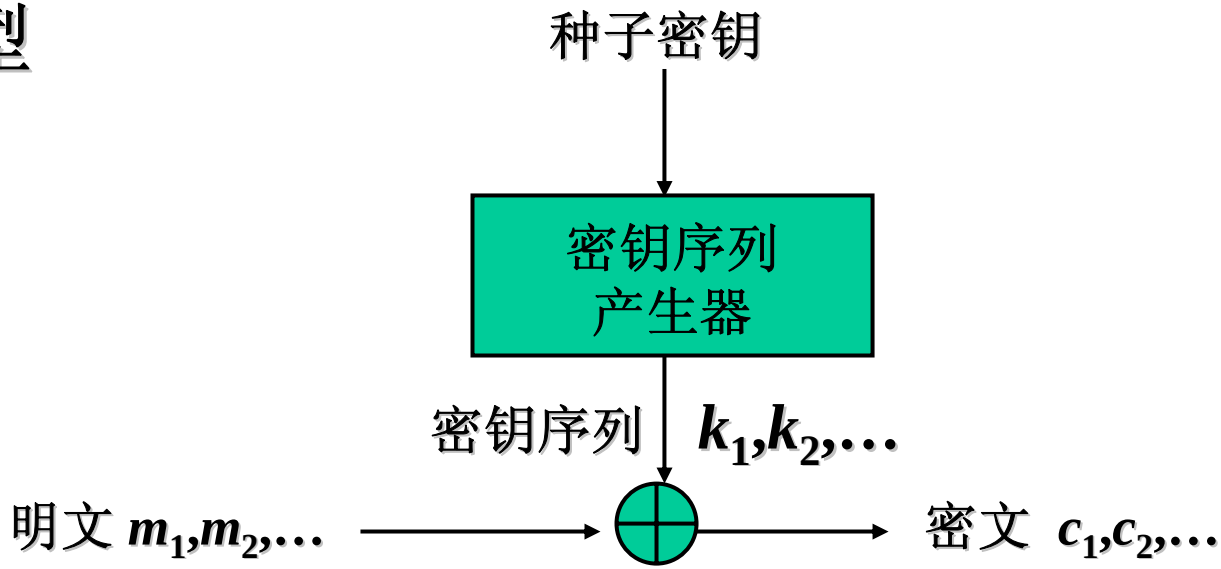
- 一次一密密码是绝对安全的密码，如果能以某种方式仿效一次一密密码，则将可以得到安全性很高的密码
- 人们试图以流密码方式仿效一次一密密码
- 流密码也称为序列密码，它是对明文以一位或者一个字节为单位进行操作
- 为了使加密算法更安全,一般选取尽可能长的密钥
- 但是长密钥的存储和分配都很困难
- 流密码采用一个短的种子密钥来控制密钥流发生器产生出长的密钥序列，供加解密使用

一、序列密码的基本概念

①定义：明文、密文、密钥以位（或字符）为单位进行加解密。

- 为了安全，密钥必须有足够的长度、随机性，且经常更换。
- 为此，用一个短的种子密钥，通过算法产生好的密钥序列。

②模型



$$C_i = m_i \oplus k_i$$

一、序列密码的基本概念

③商农证明了“一次一密”是无条件安全的。于是，人们用序列密码模仿“一次一密”；

④加密运算最简单，而且是对合运算；

⑤安全取决于密钥序列产生算法；

⑥理论和技术都十分成熟；

⑦核心密码的主流密码。

二、随机序列的安全性

- 序列密码的安全，取决于密钥序列的安全性

1. 随机序列的安全性

- 在密码学中，对随机序列的基本要求是具有良好的随机性。
  - 长周期性、非线性、统计上的预期性、可伸缩性
  - 不可预测性等
- 一个有限长度的数字序列一定是可重复的，即有周期的。例如，长度为n的二元序列，其重复周期一定小于等于2^n。
- 我们希望密钥的重复周期能够等于2^n。当n足够大时，2^n是一个很大的数。因为序列的重复周期很长，从而使密钥空间很大。

二、随机序列的安全性

- 非线性是密码学的基本要求
  - 从计算复杂性角度来看，任何线性的数学问题都是容易计算的。
  - 如果序列的内在数学关系是线性的，则容易受到基于数学的攻击。
  - 目前数学上求解非线性问题仍是困难的。
  - 因此，要求序列的内在数学关系是非线性的，而且具有足够的复杂度。

二、随机序列的安全性

- 统计上的预期性
  - 序列的一些统计指标应达到预期值。
  - 例如，二元序列中0和1的频率应是相等的或接近相等的，0游程和1游程的频率的数量也应是相等的或接近相等的。
- 可伸缩性
  - 如果序列是随机的，那么任何从中抽取的子序列也应该是随机的。
  - 如果任何抽取的子序列不随机，那么这个序列也不是随机的。

二、随机序列的安全性

- 不可预测性
  - 一是指，人们不能人为有意识地重复产生该随机序列，
  - 二是指，不能由已知的序列数据求出未知的序列数据。
  - 第一个性质决定是“真”还是“伪”。
  - 第二个性质决定是“随机”还是“不随机”。
  - 其他性质决定随机性是“好”还是“不好”。
- 真随机性和伪随机性
  - 真随机序列具有上述两方面的不可预测性。
  - 伪随机序列，具有第二个性质，但不具有第一个性质。
  - 伪随机序列是可以人为有意识地重复的。
  - 序列密码只能使用好的伪随机序列！

二、随机序列的安全性

- 伪随机序列的不可预测性
  - 伪随机序列是可以人为有意识地重复的。
  - 其不可预测性主要强调不可由已知的序列数据求出未知的序列数据。
  - 为此，把不可预测性划分为前向不可预测性和后向不可预测性。
  - 所谓前向不可预测性是指，如果不知道产生该序列的种子，那么不管知道序列中前面的多少比特都无法预测序列中的下一比特。
  - 所谓后向不可预测性是指，从产生出的序列不能推断出产生它的种子值。

二、随机序列的安全性

- 伪随机序列的测试
  - 我们尚不能从理论上证明一个序列是随机序列。
  - 密码界采用的办法是对序列进行一系列的随机性指标测试，如果被测序列通过了这些测试，我们就认为它是随机的。
  - 2009年2月国家密码管理局颁布了《随机性测试规范》，规范给出了15项随机性检测。
  - 美国NIST也有类似规范。

4.3 Statistical analysis of pseudorandomness

The NIST STS (Statistical Test Suite) [22] is one of the most authoritative test tools for Random and Pseudorandom Numbers Generators for Cryptographic Applications. The STS is a statistical package consisting of 15 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based cryptographic random or pseudorandom number genera-

Table 3 Pseudorandom test results of MPH-160 example

Number	Statistical test	Testing times	$P-value_\tau$	Proportion
1	Rrequency	1	0.125200	0.9910
2	Block-frequency	1	0.259616	0.9910
3	Cumulative-sums	2	0.267573	0.9900
4	Runs	1	0.030197	0.9920
5	Longest-run	1	0.911413	0.9880
6	Rank	1	0.564639	0.9900
7	DFT	1	0. 036592	0.9920
8	Nonperiodic-templates	148	0.000991	0.9810
9	Overlapping-templates	1	0.733899	0.9890
10	Universal	1	0.542101	0.9880
11	Approximate entropy	1	0.990138	0.9870
12	Random-excursions	8	0.226559	0.9822
13	Random-excursions-variant	18	0.053823	0.9838
14	Serial	2	0.684890	0.9910
15	Linear-complexity	1	0.440975	0.9910

The proportion of each test falls inside of the interval $[0.9805607, 0.9994392]$ and the least $P-value_\tau \geqslant 0.0001$ are shown in Table 3. Therefore we can conclude that the data file sequence is random. Meanwhile,

二、随机序列的安全性

① 单比特频率检测

- 对于二元序列，统计被测序列中0和1出现的频率。如果0和1出现的频率相等或接近相等，则认为是合格的。

② 游程分布检测

- 对于二元序列，统计被测序列中0游程和1游程的数目。如果0游程和1游程的数目相等或接近相等，则认为是合格的。
- 定义：称二元序列中连续*i*个1组成的子序列为长度等于*i*的1游程，连续*i*个0组成的子序列为长度等于*i*的0游程。
- 举例：序列为100110101111000，则其中包含2个长度为1的0游程和1游程、包含1个长度为2的0游程和1游程、包含1个长度为4的1游程、包含1个长度为3的0游程，游程总数为8。

二、随机序列的安全性

③ 自相关检测

- 序列的自相关系数，反映该序列的自相关程度。
- 定义：设$\{k_i\}$是周期为p的序列，$k_0, k_1, \ldots, k_{p-1}$是其中一个周期子段，则$k_{0+\tau}, k_{1+\tau}, \ldots, k_{p-1+\tau}$也是一个周期子段。记这两个子段中相同的位数为$A$，不相同的位数为$D$，则自相关函数定义为：

$$R(\tau)=\frac{A-D}{P}$$

并且如果

$$R(\tau)=\begin{cases}1 & \tau=0\\ -1/p & 0<\tau\le p-1\end{cases}$$

则称其自相关系数达到最佳值。

二、随机序列的安全性

③ 自相关检测

- 自相关函数反映了一个周期序列在一个周期内平均每位的相同程度。
- 如果一个序列的自相关函数达到最佳值，则表明其具有良好的随机性。
- 举例，设一个周期 p =15的序列为**1 0 0 1 1 0 1 0 1 1 1 1 0 0 01 0 0 1 1 0 1 0 1 1 1 1 0 0 0……**。取出一个周期子段为**1 0 0 1 1 0 1 0 1 1 1 1 0 0 0**。令τ=0，则**R(τ) =0**。令τ=1，又得到一个周期子段**0 0 1 1 0 1 0 1 1 1 1 0 0 0 1**。简单计算可得**A=7，D=8，R(τ) = -1/15**。由此可知，这个序列的自相关函数达到最佳值，它具有良好的随机性。

二、随机序列的安全性

2、随机序列的产生

- 第一种方法是基于数学算法来产生随机序列。早期人们采用线性算法，后来发现不安全，现在都是采用非线性算法来产生随机序列。
  - 优点是可以得到统计随机特性很好的随机序列，且可以人为有意识地重复产生，适于序列密码应用。
  - 缺点是不是真随机的。
- 第二种方法是基于物理学来产生随机序列。早期最典型的是抛撒硬币或掷骰子。今天广泛采用基于电子噪声来产生随机序列。
  - 优点是真随机的。
  - 缺点是其序列的统计随机特性往往不够好。

随机数及其应用

- Entertainment 娱乐
  - Gambling 赌博
  - Lottery, lucky draw 抽奖
  - Games 游戏
- Cryptography 密码学
  - Key generation
- Computer simulation 模拟
- Software testing 软件测试
  - Generating testing data
- Randomized algorithms 随机化算法
  - Avoiding worst cases




真随机序列

- 信息源
  - 电磁辐射，热噪声，人机交互等
- 不能用于加密，可以用于生成密钥

随机比特生成

- 基于硬件的生成器：

1.放射性衰变过程中粒子的消逝时间；

2.由半导体二极管或电阻器产生的热噪声；

3.自由运行的振荡器的频率不稳定性；

4.在一段固定的时间内对金属绝缘半导体电容进行充电；

5.密封的磁盘驱动器中的空气急流，它能在磁盘驱动器扇区寻道等待时间内产生随机波动；

6.来自麦克风的声音或照相机的视频输入。

随机比特生成

- 基于软件的生成器：

1.系统时钟；

2.敲击键盘或鼠标移动中的消逝时间；

3.输入、输出缓冲区的内容；

4.用户的输入；

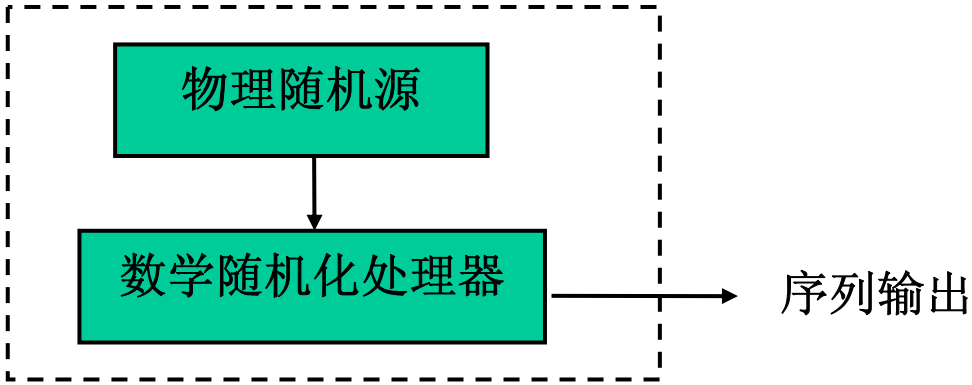
5.操作系统的参数值，例如系统加载量和网络统计量。

推荐使用密码学的Hash函数，如SHA-1对样本序列进行级联，以消除其输出中可能存在的偏差或者相关性。

武漢大学

二、随机序列的安全性

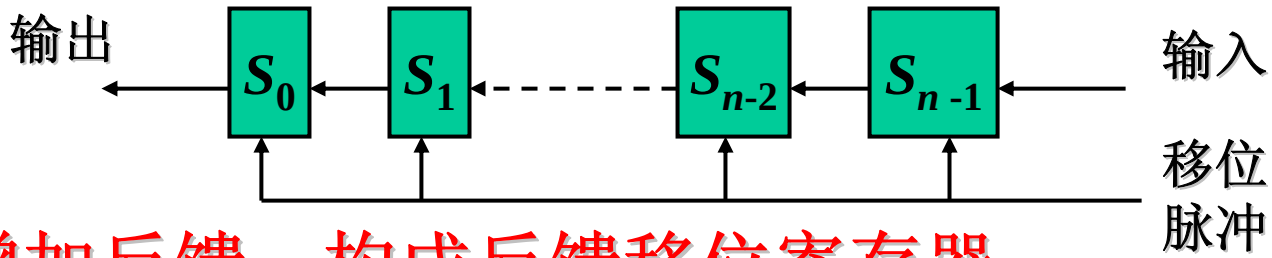
- 第三种方法是将前两种方法相结合。
  - 采用一个物理的随机源产生原始的随机序列，
  - 再接一个基于数学算法的随机化处理器。
  - 物理随机源的输出作为种子，输入给基于数学算法的随机化处理器作进一步的随机化处理，其输出可得到良好的随机序列。



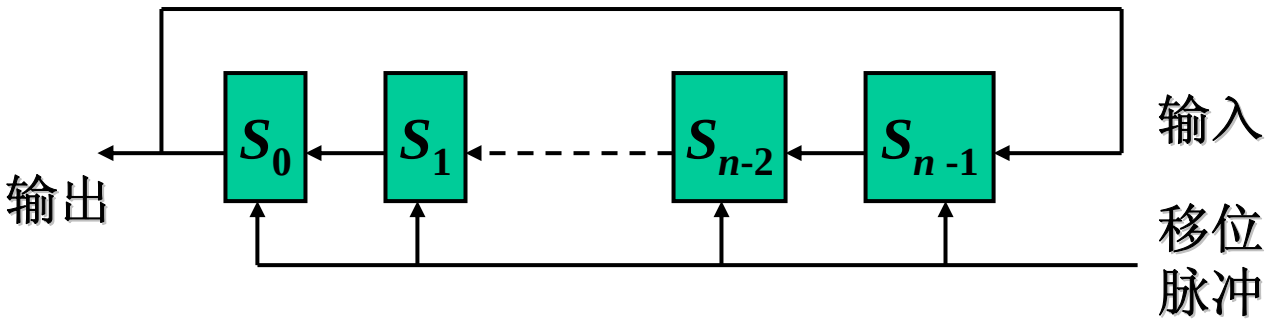
三、线性移位寄存器序列密码

1、线性移位寄存器（Linear Shift Registor）

- 例1 移位寄存器



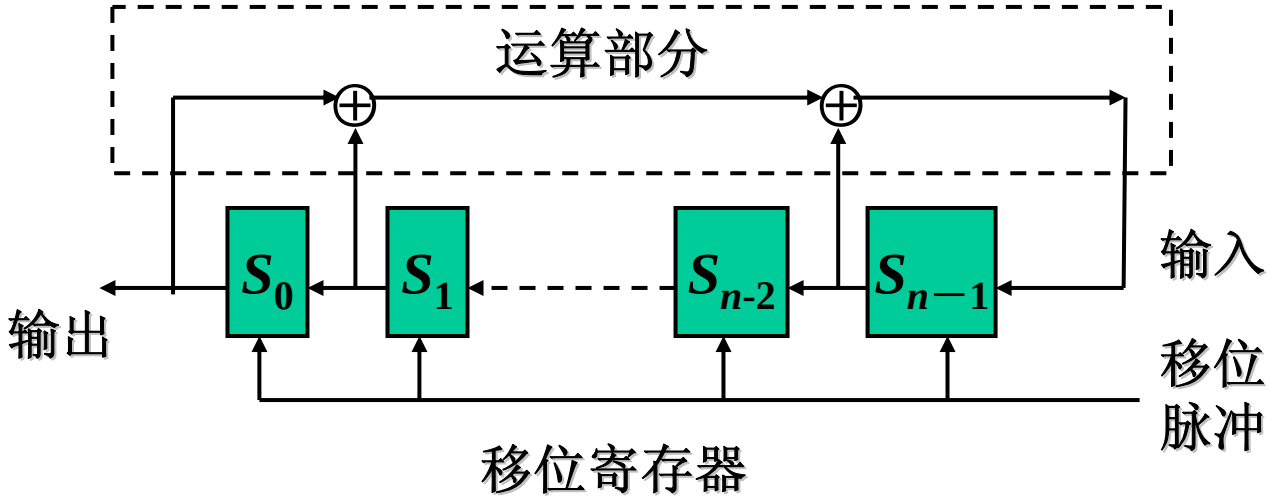
- 例2 增加反馈，构成反馈移位寄存器



三、线性移位寄存器序列密码

1、线性移位寄存器（Linear Shift Registor）

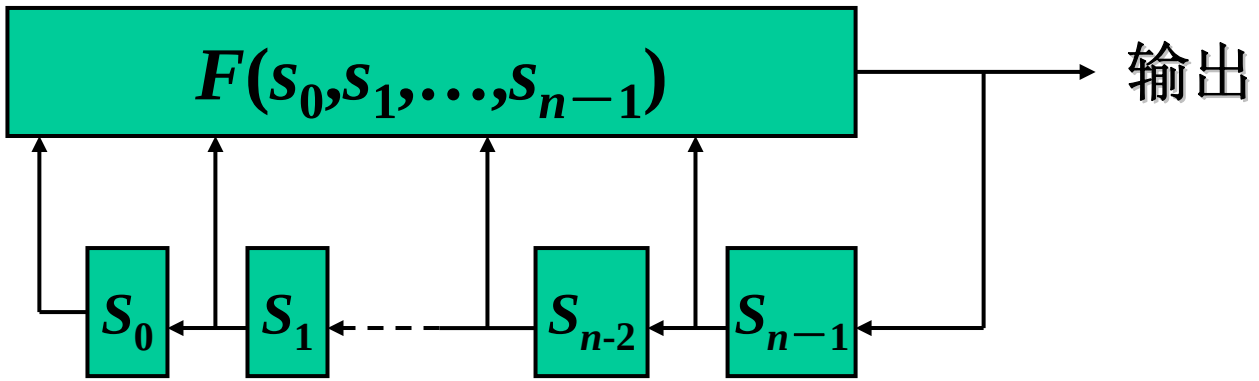
- 例3 增加运算，构成线性移位寄存器



三、线性移位寄存器序列密码

1、线性移位寄存器（Linear Shift Registor）

- 一般模型



三、线性移位寄存器序列密码

1、线性移位寄存器（Linear Shift Registor）

- 图中s_0，s_1，…，s_{n-1}组成左移移位寄存器，并称每一时刻移位寄存器的取值为一个状态。
- 移位寄存器的输出同时要送入s_{n-1}，其值要通过函数$F(s_0,s_1,\ldots,s_{n-1})$计算产生。
- 称函数$F(s_0,s_1,\ldots,s_{n-1})$为反馈函数。
- 如果反馈函数$F(s_0,s_1,\ldots,s_{n-1})$是$s_0,s_1,\ldots,s_{n-1}$的线性函数，则称为线性移位寄存器，否则称为非线性移位寄存器。

三、线性移位寄存器序列密码

1、线性移位寄存器

- 设$F(s_0,s_1,\ldots,s_{n-1})$为线性函数，则可写成

$$F(s_0,s_1,\ldots,s_{n-1}) = g_0s_0+g_1s_1+,\ldots,+g_{n-1}s_{n-1}$$

其中，g_0，g_1，…，g_{n-1}为反馈系数。

- 在$GF(2)$的情况下，式中的+即为⊕，反馈系数$g_i \in GF(2)$，如果$g_i=0$，则表示式中的g_is_i项不存在，因此表示s_i不连接。同理，$g_i=1$表示s_i连接。故g_i的作用相当于一个开关。

三、线性移位寄存器序列密码

1、线性移位寄存器

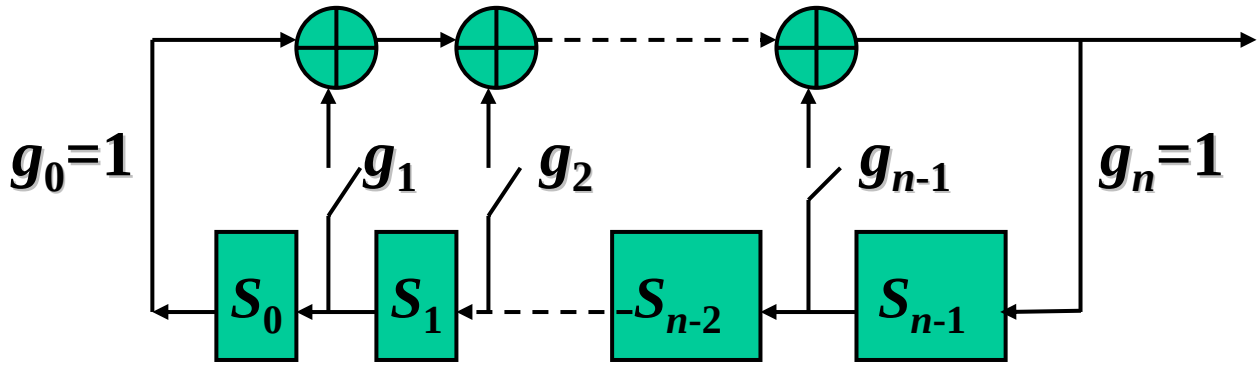
- 形式地，用x^i与s_i 相对应，则根据反馈函数可导出一个文字x的多项式：

$$g(x) = g_n x^n + g_{n-1} x^{n-1} + ,\ldots, + g_1 x + g_0$$

- 称$g(x)$ 为线性移位寄存器的连接多项式。
- 与图对照可知，g_n=g_0 =1。否则，若g_n=0则输出不反馈到s_{n-1}，若g_1=0则s_0不起作用，应将其去掉。

三、线性移位寄存器序列密码

1、线性移位寄存器



三、线性移位寄存器序列密码

1、线性移位寄存器

- *n*级线性移位寄存器最多有2^n个不同的状态。若其初始状态为零，则其后续状态恒为零。若其初始状态不为零，则其后续状态也不为零。因此，*n*级线性移位寄存器的状态周期≤2^n-1，

  其输出序列的周期≤2^n-1。

- 只要选择合适的连接多项式便可使线性移位寄存器的输出序列周期达到最大值2^n-1，并称此时的输出序列为最大长度线性移位寄存器输出序列，简称为*m*序列。

三、线性移位寄存器序列密码

1、线性移位寄存器

- 仅当连接多项式$g(x)$为本原多项式时，其线性移位寄存器的输出序列为m序列。

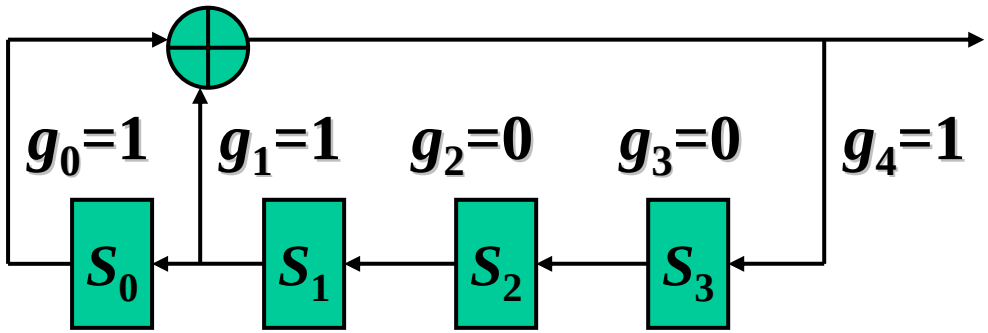
- 设$f(x)$为$GF(2)$上的多项式，使$f(x)|x^p-1$的最小正整数p 称为$f(x)$的周期。如果$f(x)$的次数为n，且其周期为2^n-1，则称$f(x)$为本原多项式。

- 已经证明，对于任意的正整数n，至少存在一个n次本原多项式，而且存在有效的产生算法。

三、线性移位寄存器序列密码

1、线性移位寄存器

- 举例：设$g(x)=x^4+x+1$，$g(x)$为本原多项式，以其为连接多项式的线性移位寄存器的输出序列为100110101111000…，它是周期为$2^4-1=15$的m序列。



0001	0101
0010	1011
0100	0111
1001	1111
0011	1110
0110	1100
1101	1000
1010	

三、线性移位寄存器序列密码

1、线性移位寄存器

- ***m*序列具有良好的随机性：**

①在一个周期内，0和 1出现的次数接近相等，即 0出现的次数为$2^{n-1}-1$，1出现的次数为2^{n-1} ；

②将序列的一个周期首尾相接，其游程总数$N=2^{n-1}$ 。

③其中1游程和0游程的数目各占一半。当$n>2$时，游程分布如下（$1\leqslant i\leqslant n-2$）：

- 长为i的1游程有$N/2^{i+1}$个；
- 长为i的0游程有$N/2^{i+1}$个；
- 长为$n-1$的0游程有1个；
- 长为n的1游程有1个.

三、线性移位寄存器序列密码

1、线性移位寄存器

④自相关函数达到最佳值。

● 举例：*m*序列的自相关函数达到最佳

例： 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0

0 0 1 1 0 1 0 1 1 1 1 0 0 0 1

$A=7$，$D=8$，$R(\tau)=-1/15$

三、线性移位寄存器序列密码

1、线性移位寄存器

- $GF(q)$上的本原多项式和线性移位寄存器

- 如果线性移位寄存器中的 s_0，s_1，…，s_{n-1}为域$GF(q)$上的元素，并且反馈函数$f(s_0,s_1,\ldots,s_{n-1})$ 为域$GF(q)$上的线性函数，则称其为域$GF(q)$上的线性移位寄存器。
- 设$f(x)$为域$GF(q)$上的多项式，使$f(x)|\ x^p-1$的最小正整数p称为$f(x)$的周期。如果$f(x)$的次数为n，且其周期为q^n-1，则称$f(x)$为$GF(q)$上的本原多项式。
- 如果一个$GF(q)$域上的线性移位寄存器的连接多项式是域$GF(q)$上的本原多项式，则其输出序列为域$GF(q)$上m序列。域$GF(q)$上m序列同样具有良好的随机性。

三、线性移位寄存器序列密码

1、线性移位寄存器

- $GF(q)$上的本原多项式和线性移位寄存器
- 举例：$2^{31}-1=2147483647$是素数，所以域$GF(2^{31}-1)$是素域。多项式$p(x)=x^{16}-2^{15}x^{15}-2^{17}x^{13}-2^{21}x^{10}-2^{20}x^4-(2^8+1)$
  - $GF(2^{31}-1)$上的16次本原多项式，周期为$(2^{31}-1)^{16}-1\approx 2^{496}$
  - 以$p(x)$为连接多项式的线性移位寄存器的输出序列为域$GF(2^{31}-1)$上的m序列。
  - 4G密码标准祖冲之密码所使用的本原多项式。

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

- ***m*序列具有良好的随机性；**
- **50年代开始用作密钥序列，并用于军用。**
- **60年代发现其是不安全的。**

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

设m序列线性移位寄存器的状态为

$$S=(s_0,s_1,\ldots,s_{n-1})^T,$$

下一状态为$S'=(s'_0,s'_1,\ldots,s'_{n-1})^T$，其中

$$s'_0 = s_1$$

$$s'_1 = s_2$$

$$\ldots$$

$$s'_{n-2} = s_{n-1}$$

$$s'_{n-1} = g_0s_0+g_1s_1+,\ldots,+g_{n-1}s_{n-1}$$

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

写成矩阵形式：$S'=HS \mod 2$

$$H=\begin{pmatrix} 0\,1\,0\,\ldots 0 \\ 0\,0\,1\,\ldots 0 \\ 0\,0\,0\,\ldots 0 \\ \ldots \\ 0\,0\,0\,\ldots 1 \\ g_0\,g_1\cdots g_{n-1} \end{pmatrix} \quad S'=\begin{pmatrix} s'_0 \\ s'_1 \\ \vdots \\ s'_{n-1} \end{pmatrix} \quad S=\begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{pmatrix}$$

矩阵H称为连接多项式的伴侣矩阵。

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

进一步假设攻击者知道了一段长$2n$位的明密文对，即已知：

$$M = m_1, m_2, \ldots, m_{2n}$$

$$C = c_1, c_2, \ldots, c_{2n}$$

于是可求出一段长$2n$位的密钥序列，

$$K = k_1, k_2, \ldots, k_{2n}$$

其中

$$k_i = m_i \oplus c_i = m_i \oplus (m_i \oplus k_i)$$

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

由此可以推出线性移位寄存器连续 $n+1$个状态：

$$S_1 = (k_1, k_2, \ldots, k_n)^T$$

$$S_2 = (k_2, k_3, \ldots, k_{n+1})^T$$

$$\cdots$$

$$S_{n+1} = (k_{n+1}, k_{n+2}, \ldots, k_{2n})^T$$

作矩阵

$$X = (S_1, S_2, \ldots, S_n)^T$$

$$Y = (S_2, S_3, \ldots, S_{n+1})^T$$

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

根据$S'=HS \mod 2$，有

$$S_2=HS_1$$

$$S_3=HS_2$$

$$\cdots$$

$$S_{n+1}=HS_n$$

于是，

$$Y=HX \mod 2$$

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

因为*m*序列的线性移位寄存器连续 *n*个状态向量彼此线性无关，因此*X*矩阵为满秩矩阵，故存在逆矩阵X^{-1}，于是

$$H=YX^{-1} \bmod 2$$

求出*H*矩阵，便确定出连接多项式*g*(x)，从而完全确定线性移位寄存器的结构。

例：*m*序列 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0

连续 4个状态 1001，0011，0110，1101线性无关

三、线性移位寄存器序列密码

2、线性移位寄存器序列密码

求逆矩阵X^{-1}的计算复杂度为$O(n^3)$。一般，对于$n=1000$的线性移位寄存器序列密码，用每秒100万次的计算机，一天之内便可破译。

四、非线性序列密码

- 线性移位寄存器序列密码在已知明文攻击下是可破译的，可破译的根本原因在于线性移位寄存器序列是线性的，这一事实促使人们向非线性领域探索。
- 目前研究得比较充分的方法：
  - 非线性移位寄存器序列
  - 对线性移位寄存器序列进行非线性组合
  - 利用非线性分组码产生非线性序列

四、非线性序列密码

①非线性移位寄存器序列

令反馈函数$f(s_0,s_1,\ldots,s_{n-1})$为非线性函数便构成非线性移位寄存器，其输出序列为非线性序列。

- 称输出序列的周期达到最大值2^n的非线性移位寄存器序列为*M*序列。
- *M*序列的0，1分布和游程分布是均匀的，而且周期最大。

四、非线性序列密码

①非线性移位寄存器序列

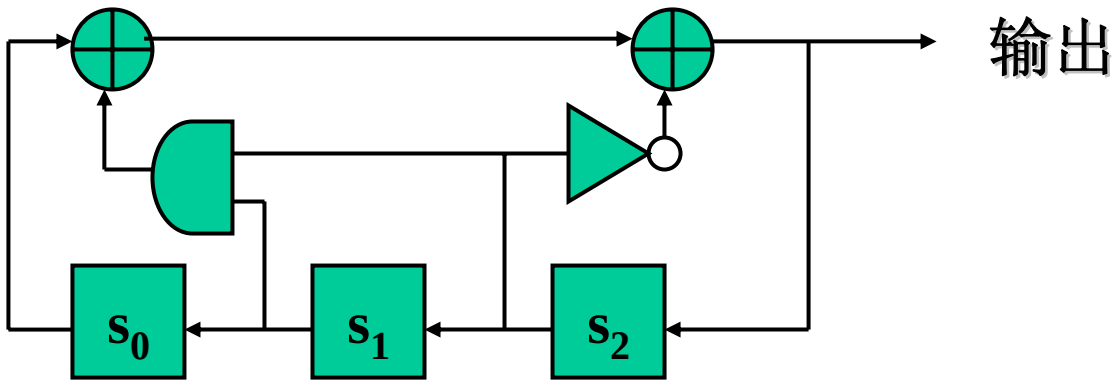
●非线性移位寄存器反馈函数的数量极大 ***GF*(2)**上的***n***级移位寄存器共有 2^n个状态，因此共有2^{2^n}种不同的反馈函数，其中 线性反馈函数只有 2^n-1种，其余均为非线性。

显然，非线性移位寄存器的空间极大！

四、非线性序列密码

①非线性移位寄存器序列

例：令$n=3$，$f(s_0,s_1,s_2)=s_0 \oplus s_2 \oplus 1 \oplus s_1 \bullet s_2$，由于与运算$\bullet$为非线性运算，故反馈函数为非线性反馈函数，其输出序列为10110100...，为M序列。



四、非线性序列密码

②对线性移位寄存器序列进行非线性组合

- 非线性移位寄存器序列的研究比较困难
- 但人们对线性移位寄存器序列的研究却比较充分和深入。
- 于是人们想到，利用线性移位寄存器序列设计容易、随机性好等优点，对一个或多个线性移位寄存器序列进行非线性组合可以获得良好的非线性序列。

四、非线性序列密码

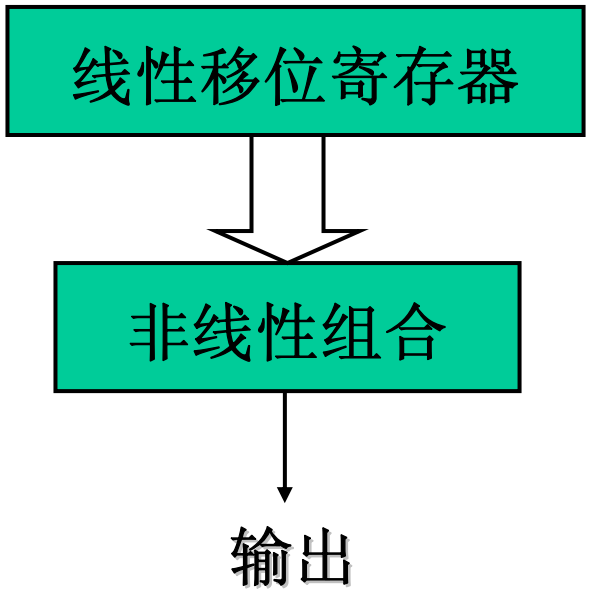
②对线性移位寄存器序列进行非线性组合

- 对一个**LSR**进行非线性组合
- 在这里用线性移位寄存器作为驱动源，来驱动非线性电路产生非线性序列。其中用线性移位寄存器序列来确保所产生序列的长周期和均匀性，用非线性电路来确保输出序列的非线性和其它密码性质。通常称这里的非线性电路为前馈电路，称这种输出序列为前馈序列。

四、非线性序列密码

②对线性移位寄存器序列进行非线性组合

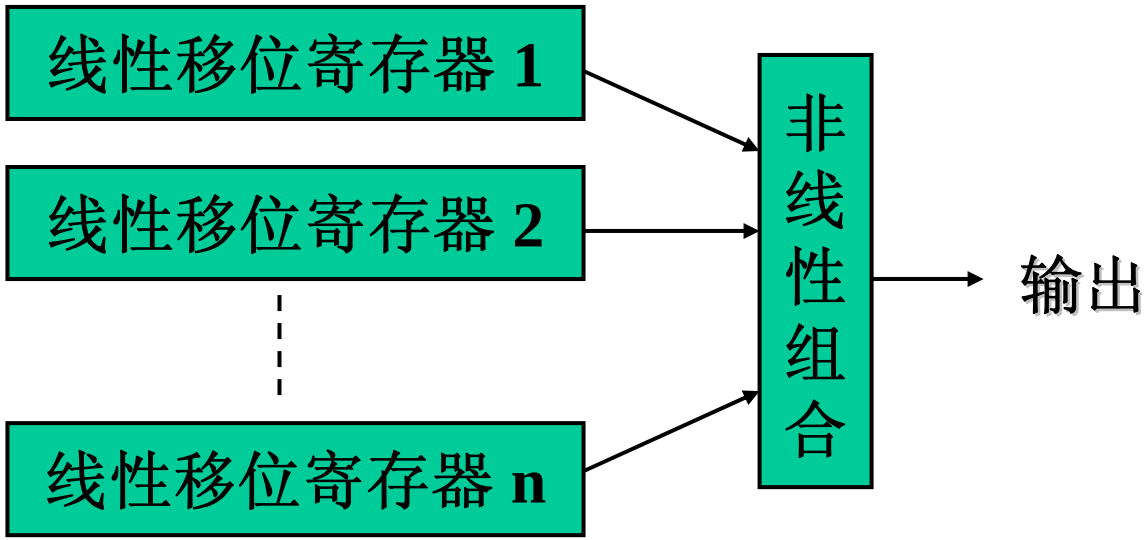
- 对一个**LSR**进行非线性组合



四、非线性序列密码

②对线性移位寄存器序列进行非线性组合

- 对多个LSR进行非线性组合



4.6 RC4序列密码

RC4是Ron Rivest在1987年为美国RSA数据安全公司设计的一种同步流密码。

RSA数据安全公司将其收集在加密工具软件BSAFE中。最初并没有公布RC4的算法。人们通过对软件进行逆向分析得到了算法。

在这种情况下RSA数据安全公司于1997年公布了RC4密码算法。

密钥40位的RC4，通过Internet 32小时攻破。

武漢大学

4.6 RC4序列密码

- RC4不是基于移位寄存器的流密码，而是一种基于非线性数据表变换的流密码
- 它以一个足够大的数据表为基础，对表进行非线性变换，产生非线性密钥流序列
- RC4 是一个可变密钥长度,面向字节操作的流密码
- 该字节的大小n可以根据用户需要来定义，一般应用中n取8位

武漢大学

4.6 RC4序列密码

- RC4算法描述：
  - 用从1到256个字节（8-2048比特）的可变长度密钥初始化一个256个字节的状态向量S，S的元素记为s[0], s[1], …, s[255],从始至终置换后的S包含从0~255的所有的8比特数。对于加密和解密，字节K由S中255个元素按一定方式选出一个元素而生成。每生成一个K的值，S中的元素个体就被重新置换一次。

4.6 RC4序列密码

- RC4使用256个字节的*S*表和两个指针（I和J）。
- S表的值$S_0, S_1, \ldots, S_{255}$是0, 1, …, 255的一个排列。
- I和J的初值为0。
- 我们把RC4算法看成一个有限状态自动机。把S表和I、J指针的具体取值称为RC4的一个状态：

$$T=\langle S_0, S_1, \ldots, S_{255}, I, J\rangle$$

- 对状态T进行非线性变换，产生出新的状态，并输出密钥序列中一个字节k 。

武漢大学

4.6 RC4序列密码

- RC4的下一状态函数定义如下：
  - (1) I＝0，J＝0;
  - (2) I= I+1 mod 256;
  - (3) J=J+S[I] mod 256;
  - (4) 交换S[I]和S[J]。
- RC4的输出函数定义如下：
  - (1) h= S[I] + S[J] mod 256;
  - (2) k = S[h] 。

武漢大学

4.6 RC4序列密码

- 在用RC4加解密之前，应当首先对S表初始化。初始化的过程如下：

(1) 对S表进行线性填充，即令

S[0]＝0，S[1]＝1，S[2]＝2，...，S[255]＝255；

(2) 用密钥填充另一个256字节的R表R[0],R[1],...，R[255]，如果密钥的长度小于R表的长度，则依次重复填充，直至将R表填满。

武漢大学

4.6 RC4序列密码

(3) J＝0;

(4) 对于I=0 到255重复以下操作，

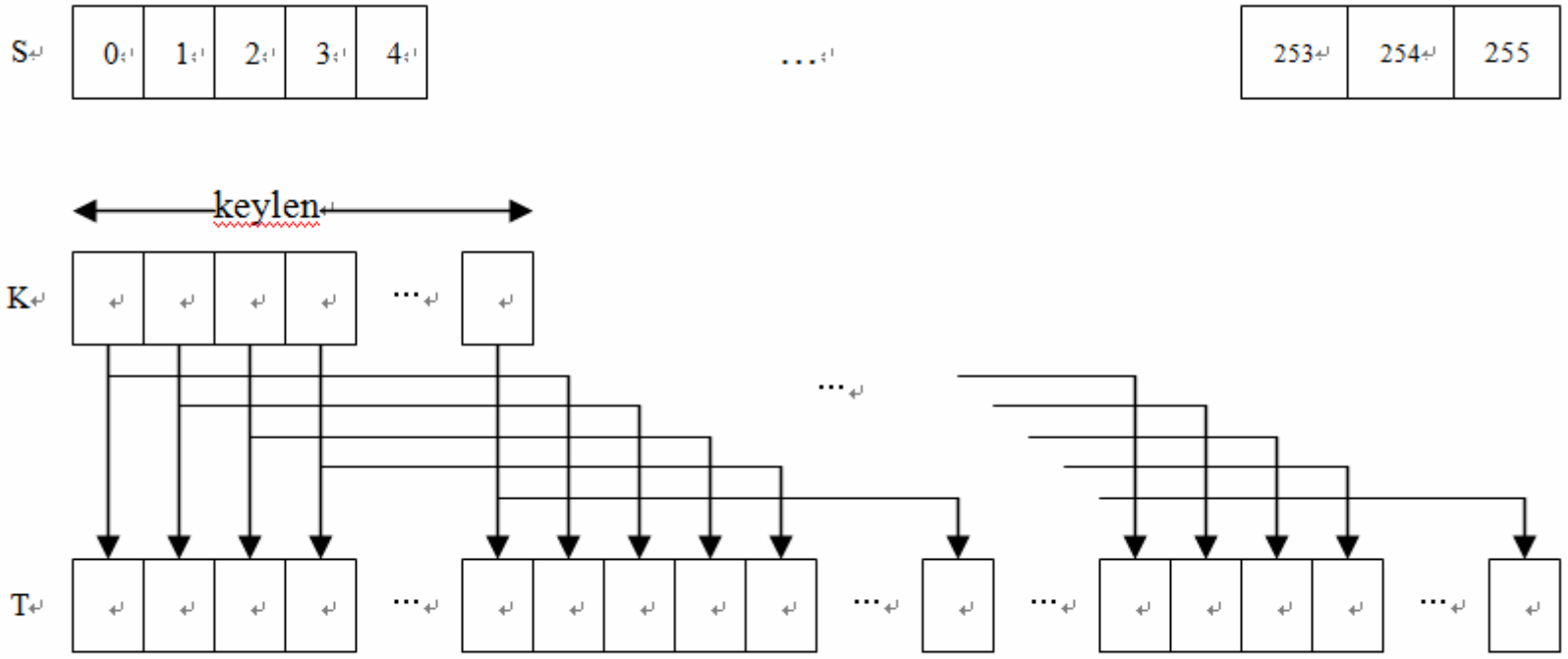
① J＝（J+S[I]+R[I]） mod 256;

② 交换S[I]和S[J]。

- RC4算法的优点是算法简单，高效，特别适合软件实现。

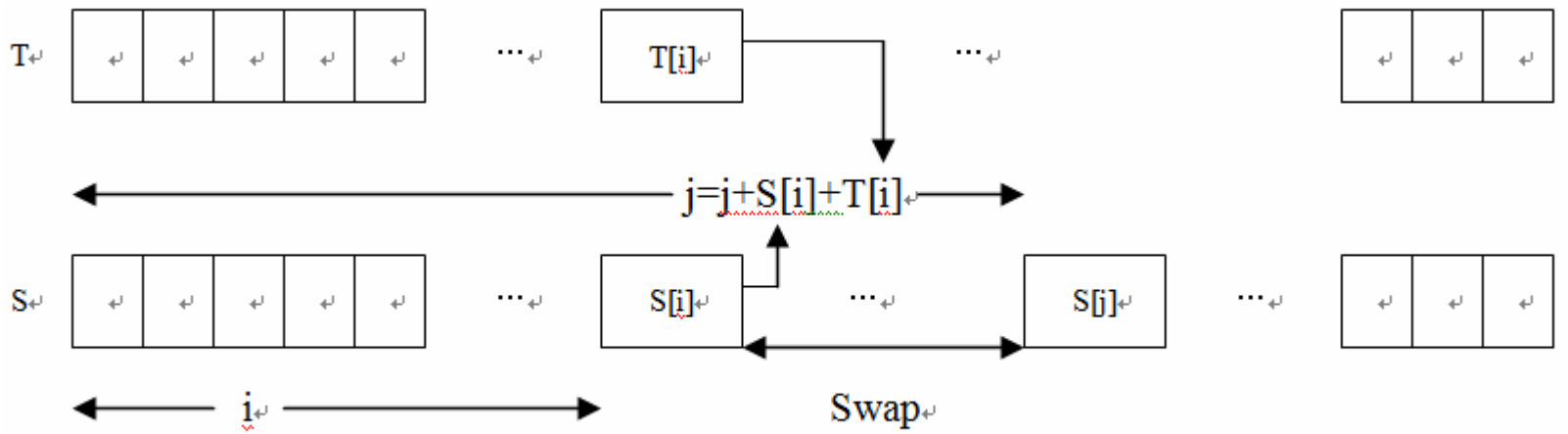
武漢大学

- /* 初始化 (注:示例图中T表即教材中的R表) */
- for i = 0 to 255 do
- S [i] = i ;
- T [i] = K [i mod keylen] ;



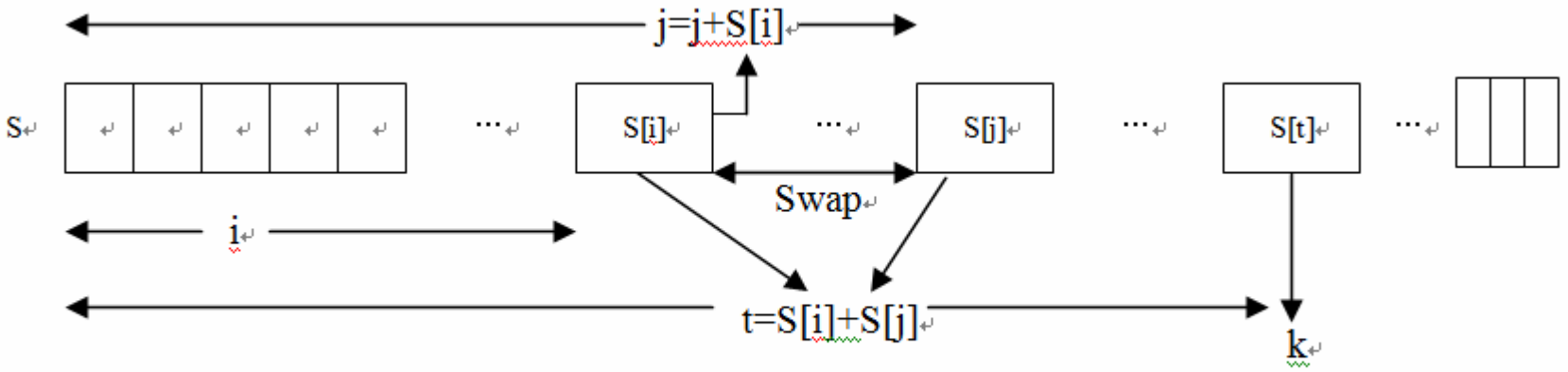
（a）S 和 T 的初始状态

- /* S的初始置换(注:示例图中T表即教材中的R表) */
- j = 0 ;
- for i = 0 to 255 do
- j = (j + S [i] + T [i]) mod 256 ;
- Swap (S [i] , S [j]) ;



（b）S 的初始置换

- /* 密钥流的生成 */
- i , j = 0 ;
- while (true)
- i = (i + 1) mod 256 ;
- j = (j + S [i]) mod 256 ;
- Swap (S [i] , S [j]) ;
- t = (S [i] + S [j]) mod 256 ;
- k = S [t] ;



（c）密钥流的生成

谢　谢！