# 第2章：序列密码

# 主要内容

1 序列密码的基本概念

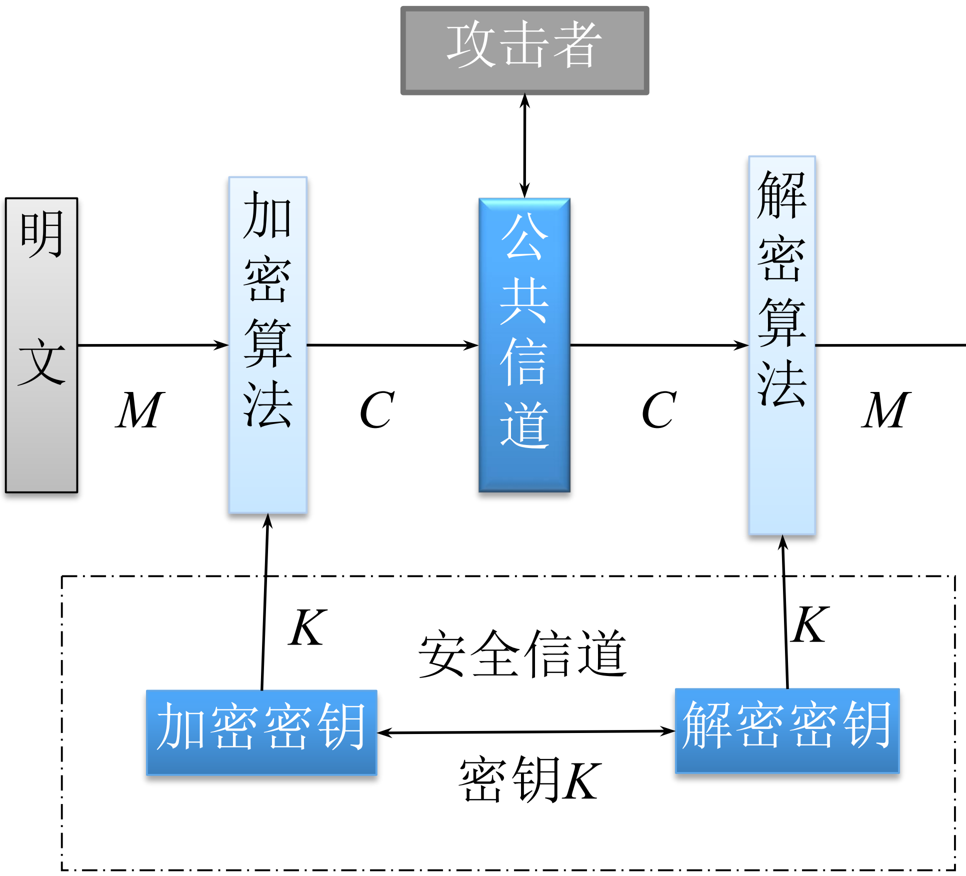
2 密钥流与密钥生成器

3 线性反馈移位寄存器序列

4 随机性概念与m序列的伪随机性

5 非线性组合子系统

6 流密码算法—A5


攻击者
明文
M
加密算法
C
公共信道
C
解密算法
M
明文
K
安全信道
K
加密密钥
解密密钥
密钥K

# 序列密码的起源

- 序列密码可追溯到20世纪20年代的**维尔南（Vernam）体制**
  - 当Vernam体制的密钥序列是**随机序列**时，就是“一次一密”密码体制
  - 将英文字母编成5比特的二元数字，随机选择二元序列作为密钥
  - 加密: 将明文和密钥的**按位**相加，即

$$c_i = m_i \oplus k_i \pmod{2} \quad \text{i=1,2,3,...}$$

- Shannon已经证明“一次一密”在理论上是不可破译的，但并不实用
  - 密钥的长度至少要等于明文长度

- 流密码（stream cipher，也称序列密码）是模仿一次一密系统的尝试
  - 理论比较成熟，而且实现简单、速度快、错误传播少
  - 军事和外交等领域的主要密码体制之一

# 序列密码的基本概念

- 序列密码的加密用一个随机序列(**密钥流**)与明文序列**按位叠加**产生密文，用同一随机序列与密文序列叠加来恢复明文
- 由种子密钥通过密钥流发生器得到的密钥流为：$K = k_1 k_2 \cdots k_n$，则加密变换为：

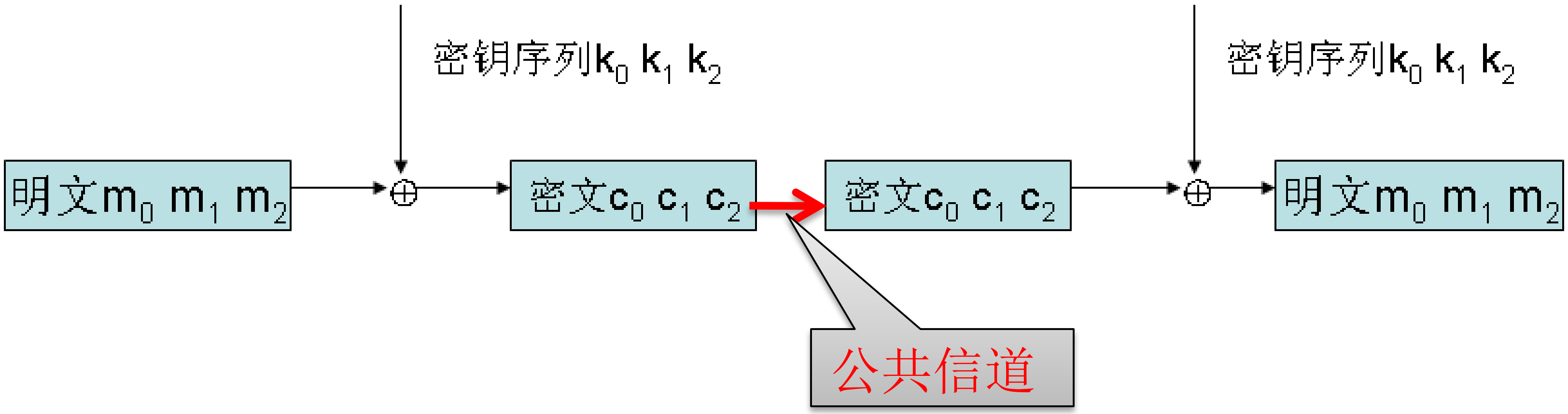
$$C = c_1 c_2 \cdots c_n$$

其中　$c_i = m_i \oplus k_i (i = 1, 2, \cdots, n)$

解密变换为：$m_i = c_i \oplus k_i (i = 1, 2, \cdots, n)$

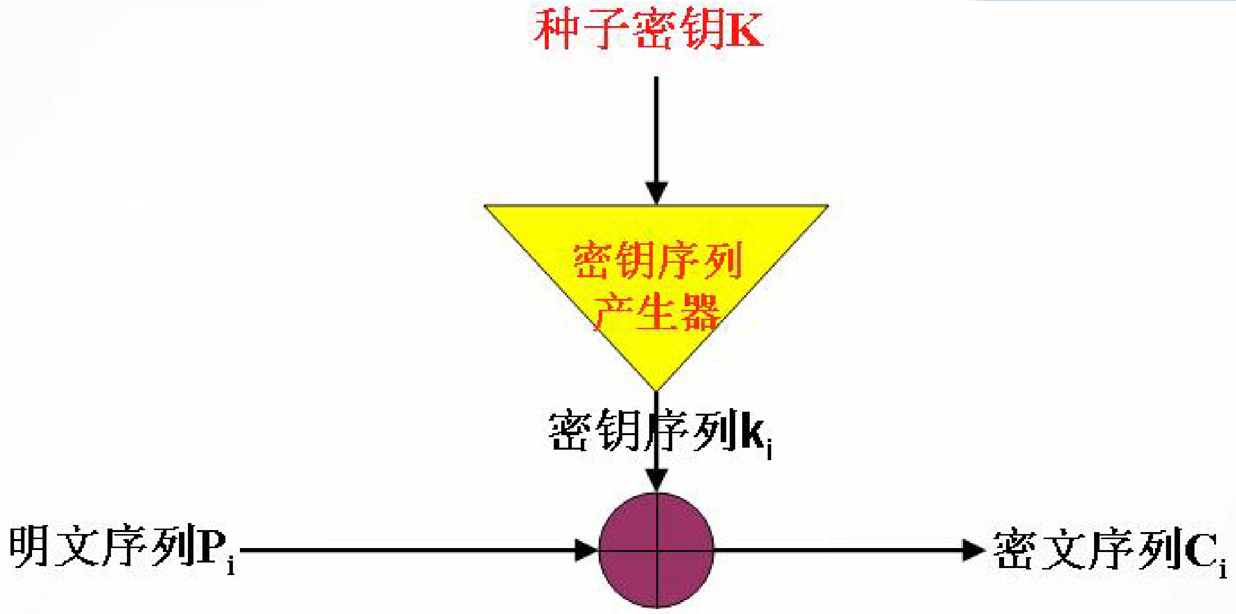
其中*m*，*k*，*c*是0、1序列，⊕表示模2加法（异或）

# 序列密码的加密和解密



密钥序列$k_0$ $k_1$ $k_2$
密钥序列$k_0$ $k_1$ $k_2$
明文$m_0$ $m_1$ $m_2$
密文$c_0$ $c_1$ $c_2$
密文$c_0$ $c_1$ $c_2$
明文$m_0$ $m_1$ $m_2$
公共信道

# 序列密码的原理




**特点:**

1. 加解密运算只是简单的模二加运算。
2. 密码安全强度主要依赖密钥流的安全性。

# 序列密码的分类

- 序列密码通常划分为两类：
  - 同步序列密码
  - 自同步序列密码

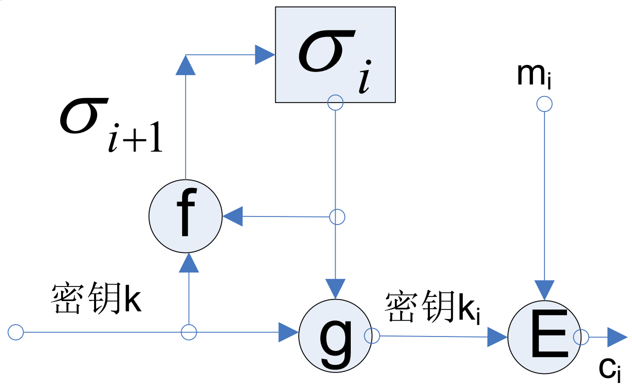
- **同步序列密码：**
  - **密钥序列的产生独立于明文消息和密文消息：**

$$\sigma_{i+1} = f(\sigma_i, k) \qquad (i = 0, 1, 2, \cdots)$$

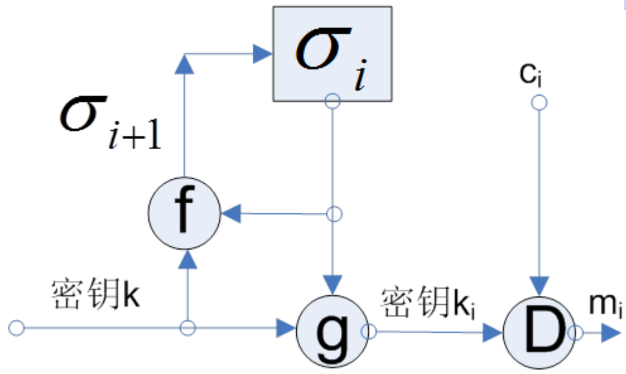
$$k_i = g(\sigma_i, k) \qquad (i = 1, 2, \cdots)$$

$$c_i = E_{k_i}(m_i) \qquad (i = 1, 2, \cdots)$$

# 同步序列密码的通用模型




加密




解密

- 特点：
  - **无错误传播：** 各符号之间真正独立。一个传播错误只影响一个符号，不会影响到后继的符号
  - **同步：** 发送方和接收方必须保持精确的、用同样的密钥并作用在同样的位置上，才能正确的解密

# 自同步序列密码

- **自同步序列密码：**
  - **密钥序列是密钥及固定大小的以往密文的函数：**

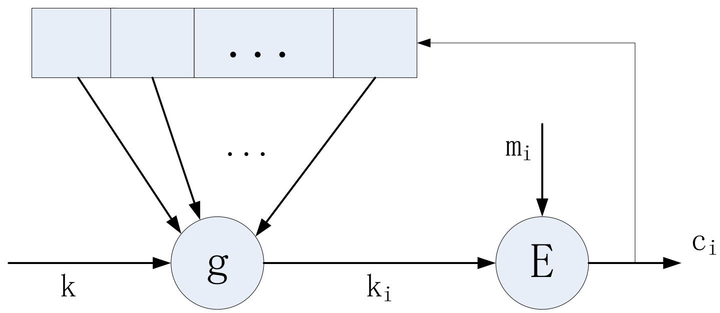
$$\sigma_i = (c_{i-t}, c_{i-t+1}, c_{i-t+2}, \ldots, c_{i-1}) \qquad (i = 0, 1, 2, \cdots)$$

$$k_i = g(\sigma_i, k)$$

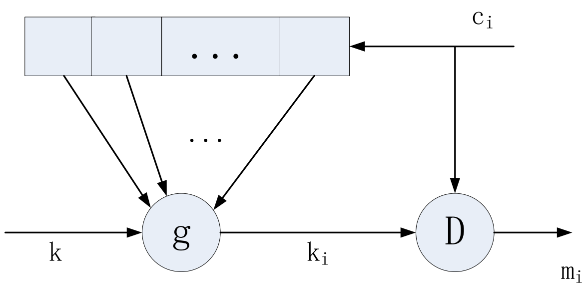
$$c_i = E_{k_i}(m_i)$$

其中$\sigma_0 = (c_{-t}, c_{-t+1}, c_{-t+2}, \ldots, c_{-1})$被称为初始状态

# 自同步序列密码的通用模型




加密




解密

❖特点：

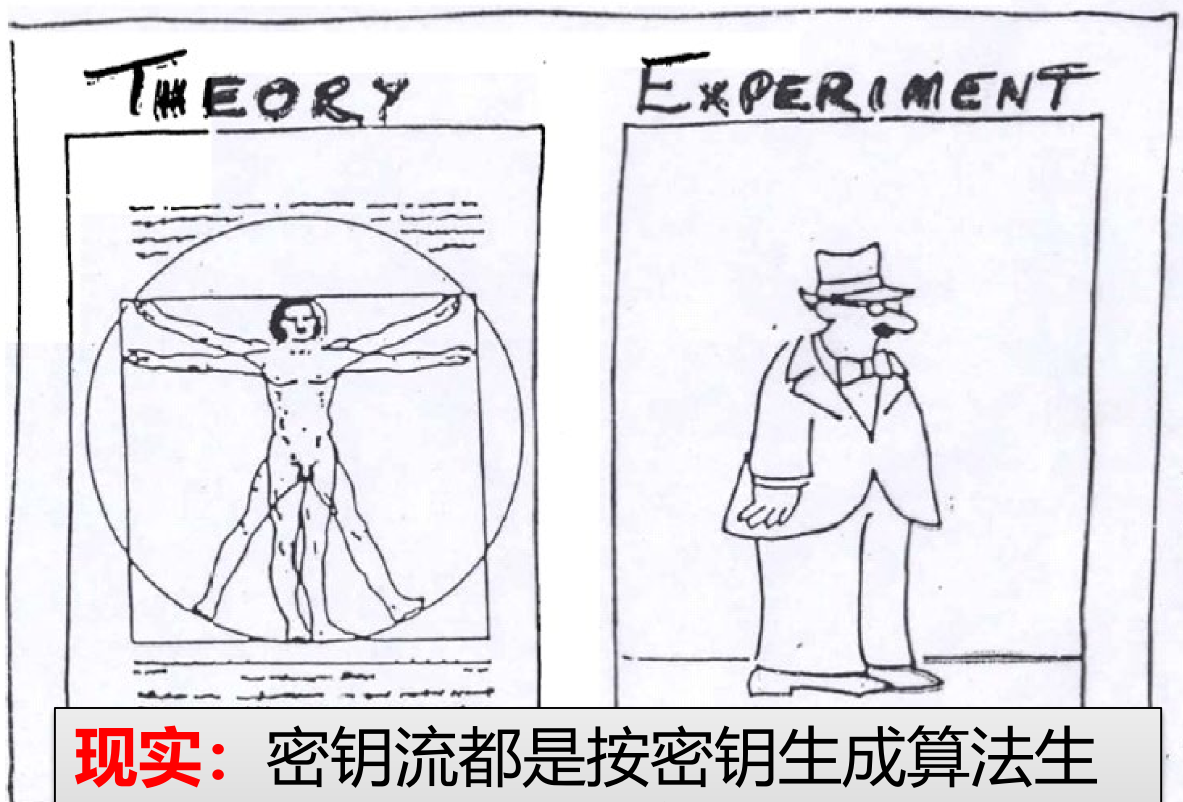
1.**有限错误传播：** 设密钥序列产生器具有n位存储，则一个符号的传输错误只影响到后面n符号的解密

2.**自同步：** 只要接收方连续收到n个正确的密文符号，密钥序列产生器便会自动地恢复同步

3.**消除明文统计特性**

# 密钥流与密钥生成器

- 密钥流算法应该能产生**随机性**和**不可预测性好**的密钥序列
- **保持同步**是序列密码在实际应用中的关键




**现实：**密钥流都是按密钥生成算法生成，且要求通信双方能产生相同的密钥序列，所以不可能是真随机的——**伪随机序列**

- 为尽可能的提高安全性，对密钥流的基本要求：

1. **极大的周期：**随机序列是非周期的，而按任何算法产生的序列都是周期的，因此应要求密钥流具有尽可能大的周期
2. **良好的统计特性：**随机序列有均匀的游程分布

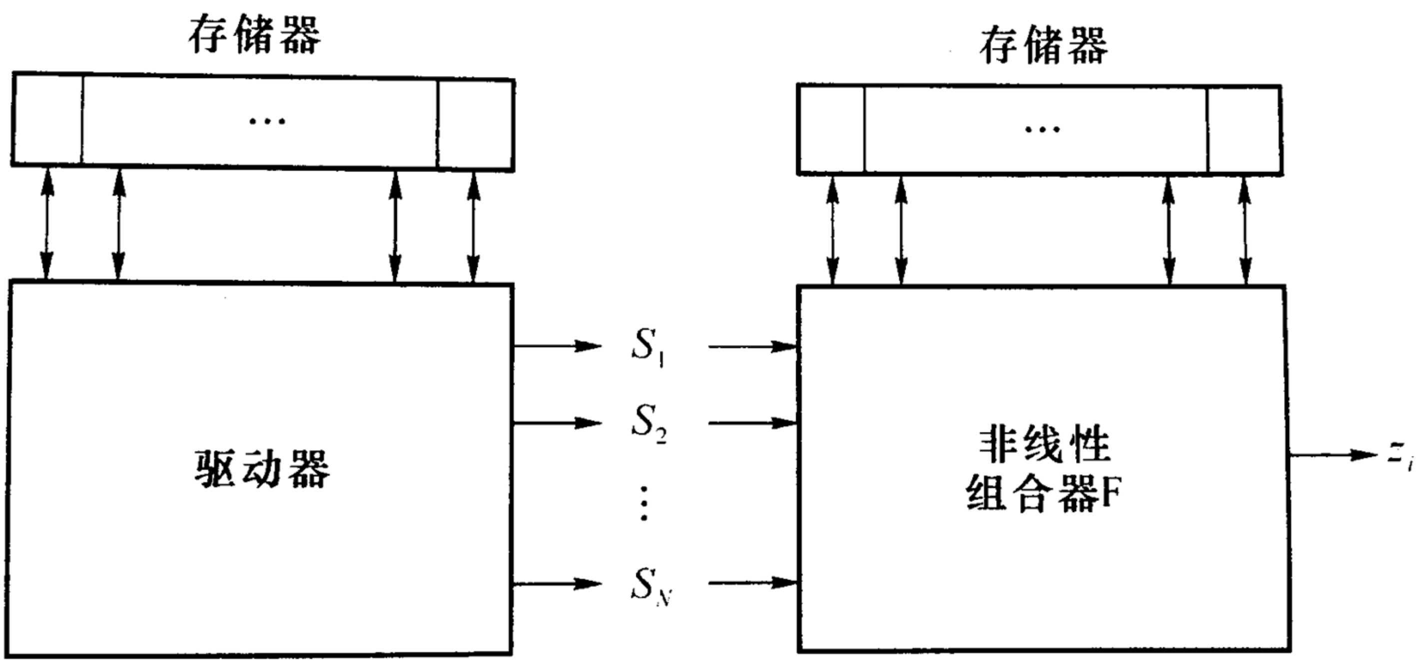
- **游程**指序列中相同符号的连续段，其前后均为异种符号
- ……0 111 0000 10……

有长为3的1游程、长为4的0游程、长为1的1游程。一般要求其在周期内满足：同样长度的0游程和1游程的个数相等，或近似相等

3. **很高的线性复杂度：**不能用级数较小的线性移位寄存器LFSR近似代替

4. 用统计方法由密钥序列$k_0k_1k_2\ldots k_i\ldots$提取密钥生成器结构或种子密钥在计算上不可行

# 密钥流生成器的组成

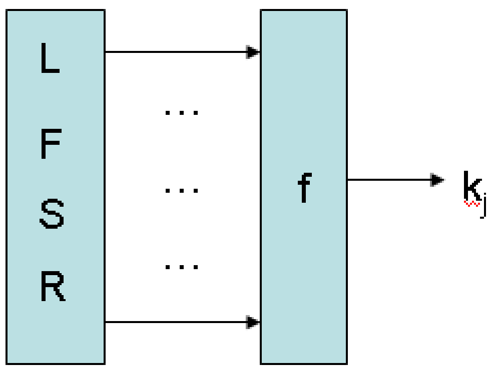



- 按Rainer Rueppel理论，密钥生成器可分解成：
  - **驱动部分**
  - **非线性组合部分**

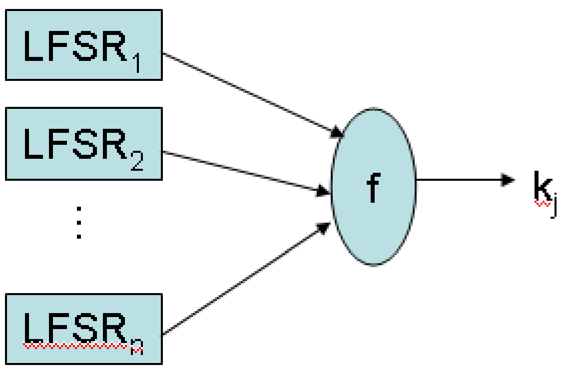
- **驱动部分：** 控制生成器的状态序列，为非线性组合部分提供统计性能良好的序列
  - 周期很大
  - 分布较随机
- **非线性部分：** 将驱动部分提供的序列组合成密码特性好的序列
  - 可隐蔽驱动序列与密钥k之间明显的依赖关系

- 目前密钥流生成器大都基于移位寄存器FSR
  - 基于移位寄存器的密钥流序列称为移位寄存器序列

- 通常由线性移位寄存器(LFSR)和一个非线性组合函数即布尔函数组合，构成一个密钥流生成器
  - （a）由一个线性移位寄存器和一个滤波器构成
  - （b）由多个线性移位寄存器和一个组合器构成




（a）

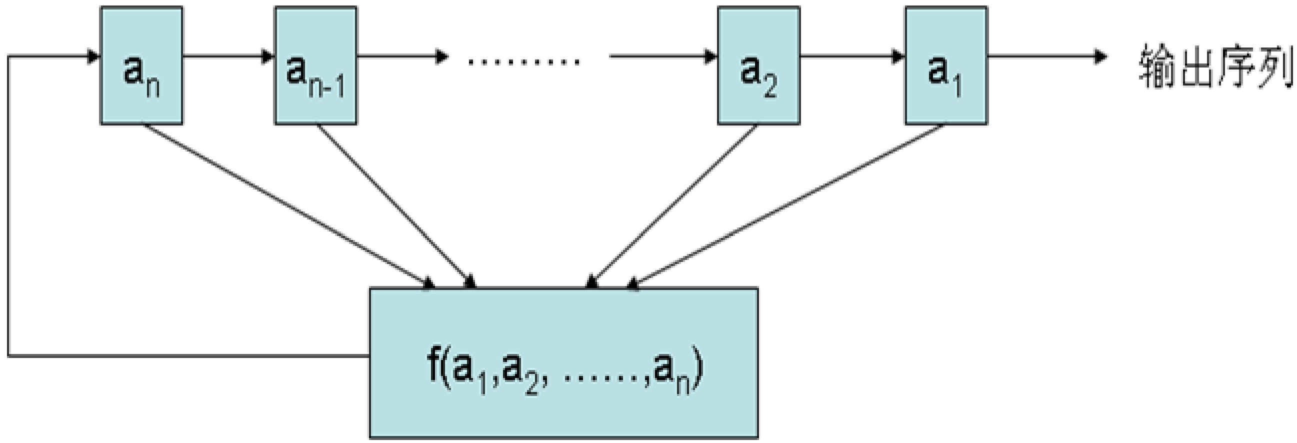



（b）

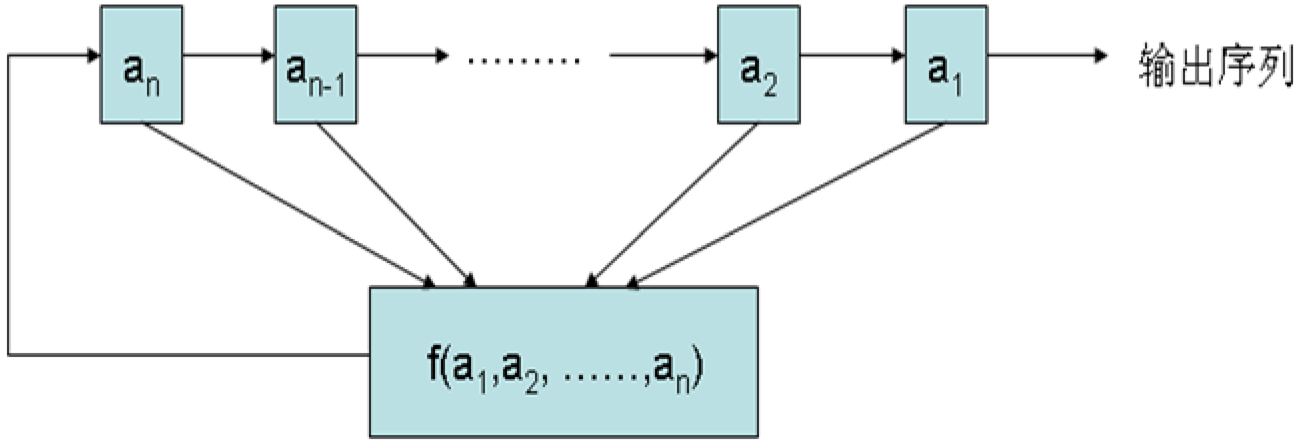
# 线性反馈移位寄存器序列

- 产生密钥流最重要的部件是**线性反馈移位寄存器（LFSR，Linear Feedback Shift Register）**，主要基于以下原因：
  1. 非常适合硬件实现
  2. 能产生大的周期序列
  3. 能产生统计特性好的序列
  4. 能够应用代数方法进行很好的分析

# 反馈移位寄存器

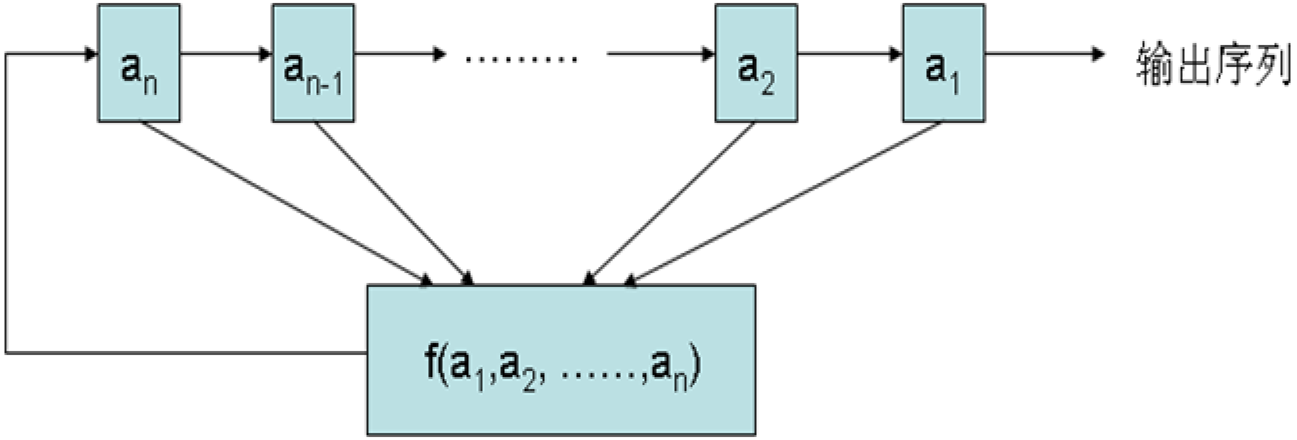



- GF(2)上一个n级反馈移位寄存器由n个二元存储器与一个反馈函数$f(a_1a_2 \ldots a_n)$组成
  - 每个存储器称为移位寄存器的一级
  - 在任一时刻，这些级的内容构成该FSR的状态;对应于一个GF(2)上的n维向量，共有$2^n$种可能的状态

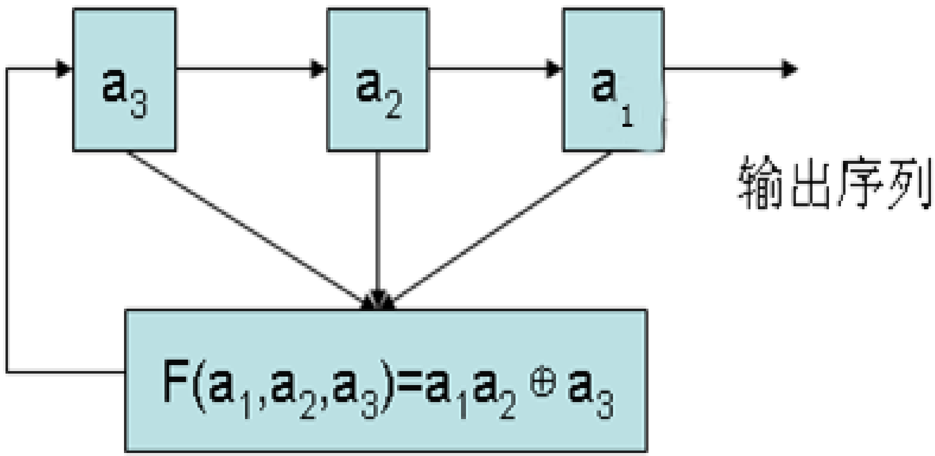



- 状态可用n长序列$a_1, a_2, a_3, \ldots, a_n$或n维行向量$(a_1, a_2, a_3, \ldots, a_n)$表示
- 每一级存储器$a_i$将其内容向下一级$a_{i-1}$传递，并根据存储器当前状态计算$f(a_1, a_2, a_3, \ldots, a_n)$作为$a_n$下一时间的内容

- 函数$f(a_1, a_2, a_3, \ldots, a_n)$是n元布尔函数，称为反馈函数; 函数值为0或1



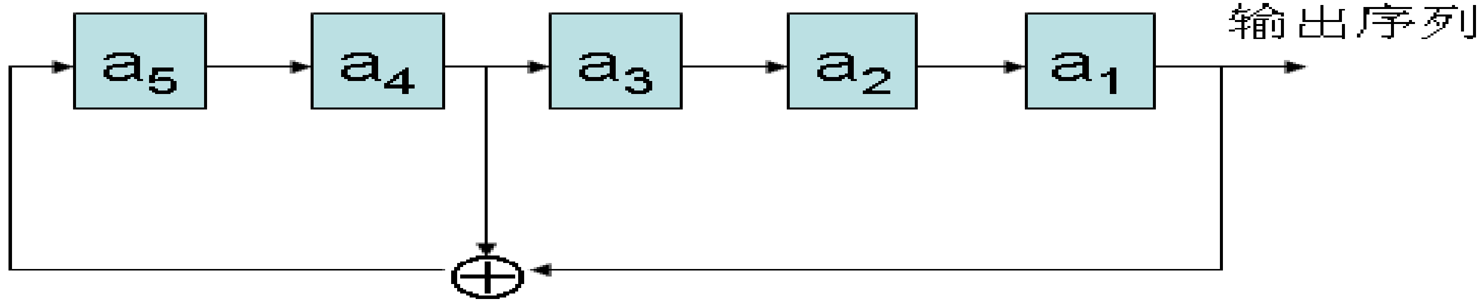
# LFSR举例




一个3级反馈移位寄存器

| 状态($a_3a_2a_1$) | 输出 |
|---|---|
| 1 0 1 | 1 |
| 1 1 0 | 0 |
| 1 1 1 | 1 |
| 0 1 1 | 1 |
| 1 0 1 | 1 |
| 1 1 0 | 0 |
| ………… | …. |

初始状态为($a_1$,$a_2$,$a_3$)＝(1,0,1)，输出可由上表求出，其输出序列为10111011101…，周期为4




- 设一个GF(2)上的5级反馈移位寄存器：

  初始状态： $s_0=(1, 0, 0, 1, 1).$

  反馈函数： $f(x_1, x_2, x_3, x_4, x_5)=x_1 \oplus x_4$

| 时刻 | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 状态 | 1, 0, 0, 1, 1 | 0, 0, 1, 1, 0 | 0, 1, 1, 0, 1 | 1, 1, 0, 1, 0 | 1, 0, 1, 0, 0 | 0, 1, 0, 0, 1 |
| 输出 | 1 | 0 | 0 | 1 | 1 | 0 |

- 反馈移位寄存器输出序列：

  1001101001000010101110110001111100110…

  $s_{31}$

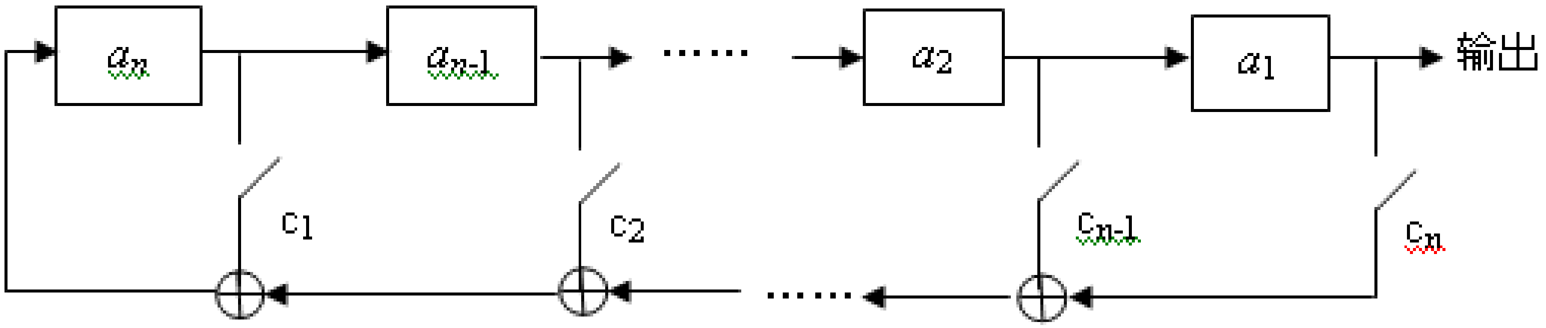
- 输出一个周期序列：周期为31=$2^5$−1

# 线性反馈移位寄存器（LFSR）

- 如果反馈函数f($a_1, a_2, \ldots, a_n$)是$a_1, a_2, \ldots, a_n$的线性函数，则称为线性反馈移位寄存器（LFSR）
- LFSR的反馈函数f可写为

$$f(a_1, a_2, \ldots, a_n) = c_n a_1 \oplus c_{n-1} a_2 \oplus \ldots\ldots \oplus c_2 a_{n-1} \oplus c_1 a_n$$

其中$c_i = 0$或1




结构图

- n级LFSR最多有$2^n$个不同的状态
  - 初始状态为零，则其状态恒为零
  - 若其初始状态非0，则其后继状态不会为0
- 因此n级LFSR的状态周期≤$2^n$-1
- 输出序列的周期与状态周期相等，所以**≤$2^n$-1**
- 选择合适反馈函数可使序列周期达到最大值$2^n$ -1，周期达到最大值的序列称为m序列

# LFSR的状态转移变换

- 设线性反馈移位寄存器的反馈函数为 $f(x_1,x_2,\cdots,x_n)=-\sum_{i=1}^{n}c_i x_{n-i+1}$　$\alpha^i=(x_1,x_2,\cdots,x_n)$ 为时刻$i$的状态，令

$$\overline{A}\alpha^i=\overline{A}(x_1,x_2,\cdots,x_n)=(x_2,x_3,\cdots,x_n,f(x_1,x_2,\cdots,x_n))$$

称 $\overline{A}$ 为该LFSR的状态转移变换.

# 状态转移变换和状态转移矩阵

例 对于例2.1中的线性反馈移位寄存器，其状态转移变换为：

$$\overline{A}:\ (x_1, x_2, x_3, x_4) \to (x_2, x_3, x_4, x_1 + x_2)$$

我们在 $F_2^4$ 中取基：

$$\varepsilon_1 = (1, 0, 0, 0),\ \varepsilon_2 = (0, 1, 0, 0),\ \ldots,\ \varepsilon_4 = (0, 0, 0, 1)$$

从而 $\overline{A}$ 在该组基下的矩阵为

$$A = \begin{pmatrix} \overline{A}\varepsilon_1 \\ \overline{A}\varepsilon_2 \\ \overline{A}\varepsilon_3 \\ \overline{A}\varepsilon_4 \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

若LFSR的反馈函数为

$$f(x_1,x_2,\cdots,x_n)=-(c_1x_n+c_2x_{n-1}+\cdots+c_{n-1}x_2+c_nx_1)$$

且在 $F_q^n$ 中取定如下的标准基：

$\varepsilon_1=(1,0,0,\cdots,0),\ \varepsilon_2=(0,1,0,\cdots,0),\ \varepsilon_3=(0,0,1,\cdots,0),\ldots\ldots,\varepsilon_n=(0,0,0,\cdots,1)$

则$n$级线性反馈移位寄存器的状态转移矩阵为：

$$A=\begin{pmatrix} 0 & 0 & \cdots & 0 & -c_n \\ 1 & 0 & \cdots & 0 & -c_{n-1} \\ 0 & 1 & \cdots & 0 & -c_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -c_1 \end{pmatrix}$$

# LFSR的特征多项式

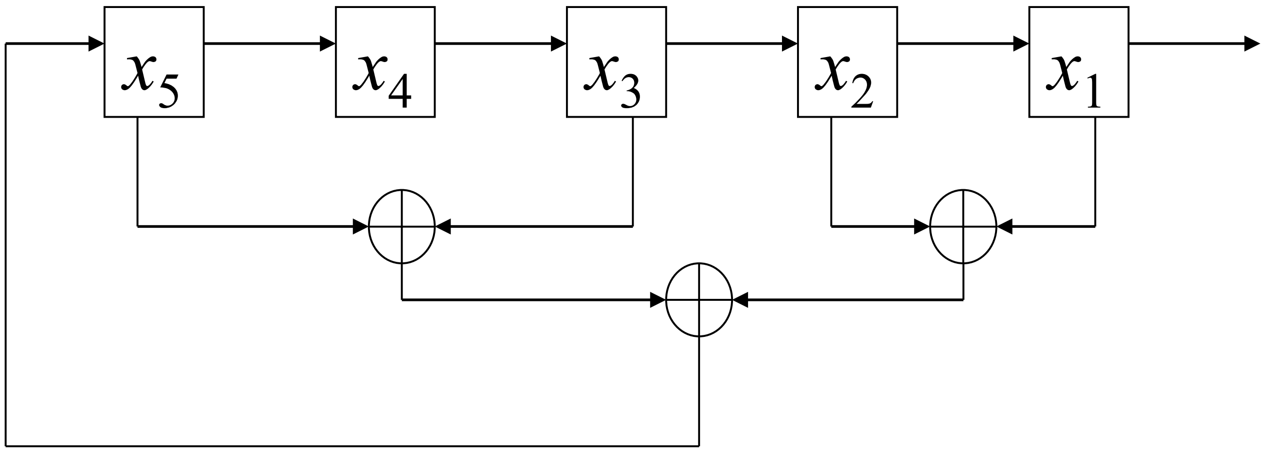
- 设线性反馈移位寄存器的状态转移矩阵为***A***，则称多项式 $f(x)=|xE-A|$ 为反馈移位寄存器的特征多项式，其互反多项式称为联接多项式.

于是

$$f(x)=|xE-A|=\begin{vmatrix} x & 0 & \cdots & 0 & c_n \\ -1 & x & \cdots & 0 & c_{n-1} \\ 0 & -1 & \cdots & 0 & c_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & x+c_1 \end{vmatrix}$$

$$=x^n+c_1x^{n-1}+c_2x^{n-2}+\cdots+c_{n-1}x+c_n$$

- 例求下图所示二元域上的LFSR的特征多项式与联接多项式.




特征多项式 $f(x)=x^5+x^4+x^2+x+1$

联接多项式 $\overline{f}(x)=x^5+x^4+x^3+x+1$

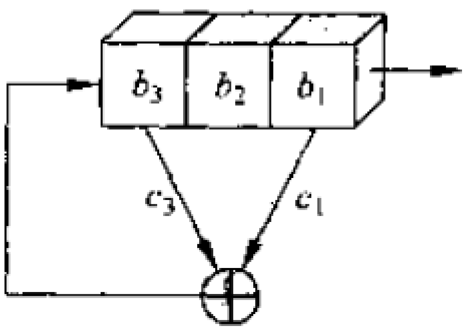
# 线性移位寄存器的一元多项式表示

- 设n级线性移位寄存器的输出序列$\{a_i\}$满足递推关系$a_{k+n} = c_1 a_{k+n-1} \oplus c_2 a_{k+n-2} \oplus \ldots \oplus c_n a_k$，$k \geq 1$。将这种递推关系用一元高次多项式

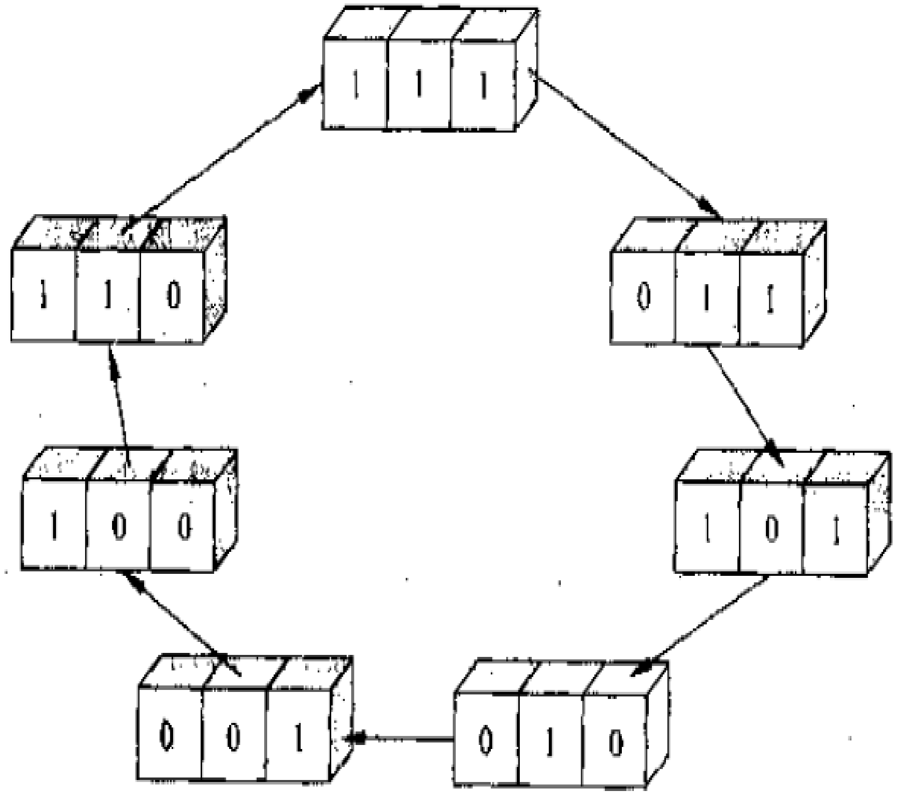
$$p(x) = 1 + c_1 x + \cdots + c_{n-1} x^{n-1} + c_n x^n$$

表示，称该多项式为该LFSR的联接多项式

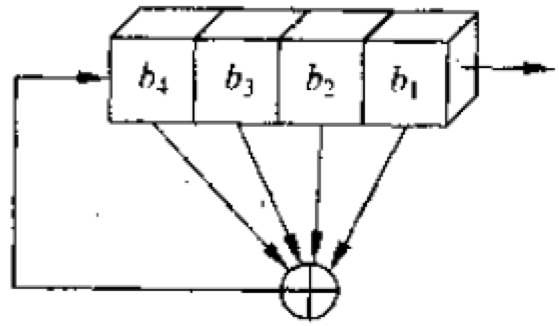
# LFSR周期分析

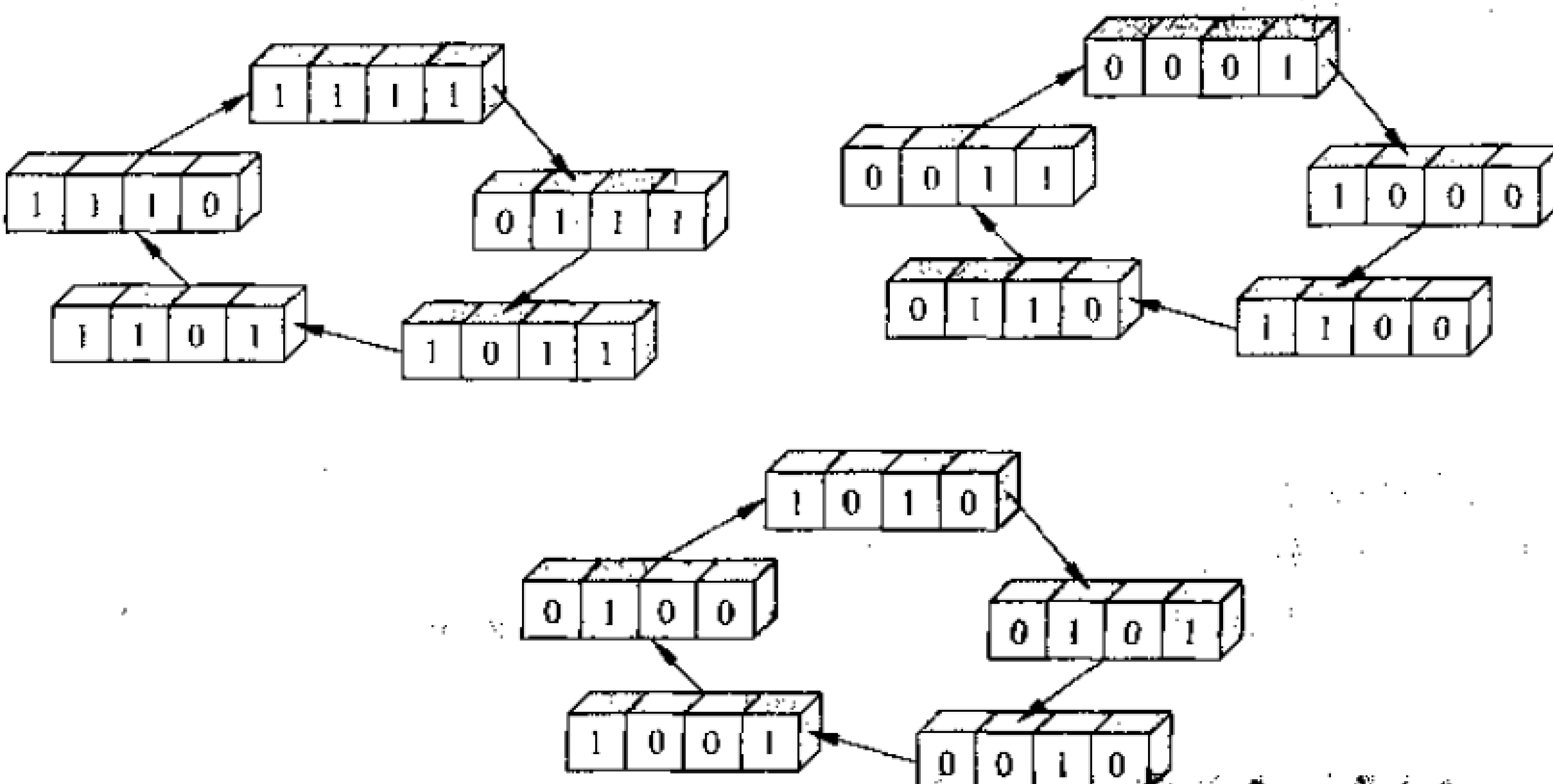



$$p(x) = x^3 + x + 1$$



$$p(x) = x^4 + x^3 + x^2 + x + 1$$


$b_4$ $b_3$ $b_2$ $b_1$


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

**定理：** 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：** 若n次不可约多项式p(x)的阶为$2^n$ - 1，则称p(x)是n次本原多项式

- 使得$p(x)|(x^p-1)$的最小p称为p(x)的**阶**

**定理：** 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

设$n$级线性移位寄存器的输出序列$\{a_i\}$满足递推关系

$$a_{n+k} = c_1 a_{n+k-1} \oplus c_2 a_{n+k-2} \oplus \cdots \oplus c_n a_k \qquad (*)$$

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

$$p(x) = 1 + c_1 x + \cdots + c_{n-1} x^{n-1} + c_n x^n$$

表示，称这个多项式为LFSR的联接多项式。

设$n$级线性移位寄存器对应于递推关系(*)，由于$a_i \in GF(2)$ ($i$=1,2,...,$n$)，所以共有$2^n$组初始状态，即有$2^n$个递推序列，其中非恒零的有$2^n$-1个，记$2^n$-1个非零序列的全体为$G(p(x))$。

## 定义2-1

给定序列$\{a_i\}$，幂级数$A(x)=\sum_{i=1}^{\infty} a_i x^{i-1}$ 称为该序列的生成函数。

## 定理2-1

设 $p(x)=1+c_1x+\cdots+c_{n-1}x^{n-1}+c_nx^n$ 是GF(2) 上的多项式，

任一序列$\{a_i\}$的生成函数$A(x)$ 满足：

$$A(x)=\frac{\phi(x)}{p(x)}$$

$$\phi(x)=\sum_{i=1}^{n}\left(c_{n-i}x^{n-i}\sum_{j=1}^{i}a_jx^{j-1}\right)$$

## 定理2-1 证明：

$$a_{n+1} = c_1 a_n \oplus c_2 a_{n-1} \oplus \cdots \oplus c_n a_1$$

$$a_{n+2} = c_1 a_{n+1} \oplus c_2 a_n \oplus \cdots \oplus c_n a_2$$

$$\cdots$$

$$A(x) = \sum_{i=1}^{\infty} a_i x^{i-1}$$

两边分别乘以$x^n$ , $x^{n+1}$,…，再求和，可得

$$A(x) - \left(a_1 + a_2 x + \cdots + a_n x^{n-1}\right) = c_1 x\left[A(x) - \left(a_1 + a_2 x + \cdots + a_{n-1} x^{n-2}\right)\right]$$

$$+ c_2 x^2\left[A(x) - \left(a_1 + a_2 x + \cdots + a_{n-2} x^{n-3}\right)\right] + \cdots + c_n x^n A(x)$$

移项整理得

$$\left(1 + c_1 x + \cdots + c_{n-1} x^{n-1} + c_n x^n\right) A(x)$$

$$= \left(a_1 + a_2 x + \cdots + a_n x^{n-1}\right) + c_1 x\left(a_1 + a_2 x + \cdots + a_{n-1} x^{n-2}\right) + \cdots + c_{n-1} x^{n-1} a_1$$

即 $$p(x)A(x) = \sum_{i=1}^{n}\left(c_{n-i} x^{n-i} \sum_{j=1}^{i} a_j x^{j-1}\right) = \phi(x)$$

注意在 GF(2)上有: $a + a = 0$

**定理2-2** $p(x)|q(x)$ 的充要条件是 $G(p(x))\subset G(q(x))$

**证明:**

若 $p(x)|q(x)$，可设 $q(x)=p(x)r(x)$，因此

$$A(x)=\frac{\phi(x)}{p(x)}=\frac{\phi(x)r(x)}{p(x)r(x)}=\frac{\phi(x)r(x)}{q(x)}$$

所以若$a_i \in G\,(p(x))$，则$a_i \in G\,(q(x))$，即

$$G(p(x))\subset G(q(x))$$

反之，若$G(p(x)) \subset G(q(x))$，则对于多项式 $\phi(x)$，存在序列 $a_i \in G(p(x))$ 以 $A(x)=\dfrac{\phi(x)}{p(x)}$为生成函数。特别的，对于多项式 $\phi(x)=1$存在序列$a_i \in G(p(x))$以 $\dfrac{1}{p(x)}$ 为生成函数。由于$G(p(x)) \subset G(q(x))$，序列$a_i \in G(q(x))$，所以存在函数$r(x)$，使得$\{a_i\}$的生成函数也等于 $\dfrac{r(x)}{q(x)}$，从而 $\dfrac{1}{p(x)}=\dfrac{r(x)}{q(x)}$，即$q(x)=p(x)r(x)$，所以$p(x) \mid q(x)$。

**上述定理说明:可用 $n$级LFSR产生的序列，也可用级数更多的LFSR来产生。**

## 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2-3

若序列$\{a_i\}$的特征多项式$p(x)$定义在 GF(2) 上，$p$是$p(x)$的周期，则 $\{a_i\}$的周期$r|p$ 。

证明：

由$p(x)$周期的定义得 $p(x)|(x^p-1)$ ，因此存在$q(x)$，使得$x^p-1=p(x)q(x)$，又由$p(x)A(x)=\phi(x)$可以得到式子 $(x^p-1)A(x)=\phi(x)q(x)$。因$p(x)$的次数不超过$n$ 。

由 $x^p-1=p(x)q(x)$ 知$q(x)$的次数不超过$p-n$ 。又知 $\phi(x)$ 的次数不超过$n-1$，所以 $(x^p-1)A(x)$的次数不超过式 $(p-n)+(n-1)=p-1$ 。将 $(x^p-1)A(x)$ 写成$x^pA(x)-A(x)$ ，可看出对于任意正整数$i$都有$a_{i+p}=a_i$ 。

设$p=kr+t$ $(0\le t<r)$，则 $a_{i+p}=a_{i+kr+t}=a_{i+t}=a_i$,

所以 $t=0$ ，即$r|p$ 。

- 序列周期
  - 输出序列的周期取决于寄存器的初始状态和反馈函数。
  - GF(2)上$n$级LFSR最多有$2^n$个不同的状态，输出序列的周期最多为$2^n$-1
- $m$序列
  - 周期达到最大值的序列称为$m$序列，是最长线性反馈移位寄存器序列(maximum length sequence)的简称。

- **$m$序列是一种典型的伪随机序列。在通信领域有着广泛的应用，如扩频通信、卫星通信的码分多址（CDMA），数字数据中的加密、加扰、同步、误码率测量等领域。**

**定义2-3**

仅能被非0常数或自身的常数倍除尽，但不能被其它多项式除尽的多项式称为**即约多项式**或**不可约多项式**。

**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$m$。

**注：** 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**。

证明：

设$\{a_i\}$的周期为$r$，由定理2-3有$r|m$，所以 $r\le m$。设$A(x)$为 $\{a_i\}$的生成函数$A(x)=\frac{\phi(x)}{p(x)}$，即 $p(x)A(x)=\phi(x)\ne 0$，$\phi(x)$的次数不超过$n$-1。而

$$A(x)=\sum_{i=1}^{\infty}a_ix^{i-1}=a_1+a_2x+\cdots+a_rx^{r-1}+x^r\left(a_1+a_2x+\cdots+a_rx^{r-1}\right)$$

$$+\left(x^r\right)^2\left(a_1+a_2x+\cdots+a_rx^{r-1}\right)+\cdots=\frac{a_1+a_2x+\cdots+a_rx^{r-1}}{1-x^r}$$

$$=\frac{a_1+a_2x+\cdots+a_rx^{r-1}}{x^r-1}$$

$$\because \quad A(x)=\frac{a_1+a_2x+\cdots+a_rx^{r-1}}{x^r-1}=\frac{\varphi(x)}{p(x)}$$

$$\therefore \quad p(x)\left(a_1+a_2x+\cdots+a_rx^{r-1}\right)=\varphi(x)\left(x^r-1\right)$$

**因 $p(x)$是不可约的且$\phi(x)$的次数不超过$n$-1，所以，gcd($p(x)$, $\phi(x)$)=1，因此 $p(x)|x^r$-1。综上$r=m$ 。**

**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。

**证明:**

设$n$级LFSR产生的序列周期达到最大$2^n$-1，除0序列外，每一序列的周期由特征多项式惟一决定，而与初始状态无关。设特征多项式为$p(x)$，若$p(x)$ 可约，可设为$p(x) = g(x)\ h(x)$，其中$g(x)$是不可约多项式，且次数$k<n$。由于 $G(g(x)) \subset G(p(x))$，而$G(g(x))$中序列的周期一方面不超过$2^k$-1，另一方面又等于$2^n$-1，这是矛盾的，所以$p(x)$是不可约多项式。

该定理的逆不成立。

**例2-4** $f(x)=x^4+x^3+x^2+x+1$为 GF(2) 上的不可约多项式，这是因为一次多项式 $x$和$x+1$ 都不能整除，因此任一三次多项式也不能整除$f(x)$。而二次多项式有$x^2$, $x^2+1$， $x^2+x$， $x^2+x+1$。由$x$和$x+1$都不能整除$f(x)$ 知 $x^2$, $x^2+1$， $x^2+x$ 都不能整除$f(x)$，二次不可约多项式$x^2+x+1$ 不能整除$f(x)$可直接验证。

以 $f(x)$为特征多项式的LFSR的输出序列可由

$$a_k = a_{k-1} \oplus a_{k-2} \oplus a_{k-3} \oplus a_{k-4} \quad (k \geq 4)$$

和给定的初始状态求出，设初始状态为0001，则输出序列为00011000110001…，周期为5，不是$m$序列。

**定义2-4**

若$n$次不可约多项式$p(x)$的阶为$2^n$-1，则称$p(x)$是$n$次**本原多项式**。

**定理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。

**证明:**

若$p(x)$是本原多项式，则其阶为$2^n$-1，由定理2-4得$\{a_i\}$的周期等于$2^n$-1，即$\{a_i\}$为$m$序列。

反之，若$\{a_i\}$为$m$序列，即其周期等于$2^n$-1，由定理2-5知 $p(x)$是不可约多项式。由定理2-3知$\{a_i\}$的周期$2^n$-1整除$p(x)$的阶，而$p(x)$的阶不超过$2^n$-1，所以$p(x)$的阶为$2^n$-1，即$p(x)$是本原多项式。

**例2-5** 设$p(x)=x^4+x+1$，由于$p(x)|(x^{15}-1)$，但不存在小于15的常数$l$，使得$p(x)|(x^l-1)$，所以$p(x)$的阶为15。类似于例2-4，$p(x)$的不可约性可由$x$，$x+1$，$x^2+x+1$都不能整除$p(x)$得到，所以$p(x)$是本原多项式。

若LFSR以$p(x)$为特征多项式，则输出序列的递推关系为

$$a_k = a_{k-1} \oplus a_{k-4} \quad (k \geq 4)$$

若初始状态为1001，则输出为

100100011110101100100011111010

周期为$2^4-1=15$，即输出序列为$m$序列。

## m序列$\{a_i\}$特性

**1. 均衡特性(平衡性)**

m序列每一周期中1的个数比0的个数多1个，0出现$2^{n-1}-1$次，1出现$2^{n-1}$次。

**2. 游程特性（游程分布的随机性）**

m序列中，状态“0”或“1”连续出现的段称为游程。游程中“0”或“1”的个数称为游程长度。

m序列的一个周期($2^n-1$)中，游程总数为$2^{n-1}$，“0”、“1”各占一半。对$1 \le i \le n-2$长为$i$的游程有$2^{n-i-1}$个，长为$n$-1的0游程一个，长为$n$的1游程一个。

**3. 移位可加性**

两个彼此移位等价的相异m序列，按模2相加所得的序列仍为m序列，并与原m序列等价。$\{a_i\}$的自相关函数为：

$$R(\tau)=\frac{1}{n}\sum_{l=1}^{n}(-1)^{k_l+k_{l+j}}=\begin{cases}1, & \tau=0\\ \dfrac{-1}{2^n-1}, & 0<\tau\le 2^n-2\end{cases}$$

由以上*m*序列的性质可以看出：*n*阶*m*序列满足Golomb随机性假设。而且当*n*并不大时，通信伙伴生成*n*阶*m*序列的复杂度很小，得到的最小周期$2^n$-1却极大。如此看来，*m*序列似乎非常适合用作密钥流。

其实不然，如果*n*阶线性反馈移位寄存器序列用作密钥流，攻击者Eve截获了密文段$c_1c_2c_3\ldots c_{2n}$，并知道了对应的明文段$m_1m_2m_3\ldots m_{2n}$，由此可计算出了对应的废弃密钥段$k_1k_2k_3\ldots k_{2n}$。

实际上，当Eve获得了$n$阶线性反馈移位寄存器序列的任何一段的连续$2n$个比特$k_{j+1}k_{j+2}k_{j+3}\ldots k_{j+2n}$，他就获得了关于抽头系数$\{c_1, c_2, \ldots, c_n\}$的以下方程组：

$$k_l = c_1 k_{l-1} \oplus c_2 k_{l-2} \oplus \ldots \oplus c_n k_{l-n},$$

其中$l=j+n+1$，$j+n+2$，…，$j+2n$。

$$\begin{bmatrix} k_{j+n} & k_{j+n-1} & \cdots & k_{j+1} \\ k_{j+n+1} & k_{j+n} & \cdots & k_{j+2} \\ \vdots & \vdots & \ddots & \vdots \\ k_{j+2n-1} & k_{j+2n-2} & \cdots & k_{j+n} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} k_{j+n+1} \\ k_{j+n+2} \\ \vdots \\ k_{j+2n} \end{bmatrix}$$

- 攻击的实例

- 设一个流密码算法使用了一个GF(2)上的3级线性反馈移位寄存器作为密钥流生成器，已知明文0100010001的密文为1010110110，试破译该密码算法。
- 步骤：

(1)明文为0100010001，密文为1010110110，可以得出密钥序列

$$a_1=1,a_2=1,a_3=1,a_4=0,a_5=1,a_6=0$$

(2)求反馈函数

$$0100010001 \oplus 1010110110 = 1110100111$$

## 根据反馈函数的性质

$$\begin{cases} a_4 \equiv (c_3a_1 + c_2a_2 + c_1a_3) \bmod 2 \\ a_5 \equiv (c_3a_2 + c_2a_3 + c_1a_4) \bmod 2 \\ a_6 \equiv (c_3a_3 + c_2a_4 + c_1a_5) \bmod 2 \end{cases} \Rightarrow \begin{bmatrix} a_4 & a_5 & a_6 \end{bmatrix} = \begin{bmatrix} c_3 & c_2 & c_1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_3 & a_4 \\ a_3 & a_4 & a_5 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} c_3 & c_2 & c_1 \end{bmatrix} = \begin{bmatrix} a_4 & a_5 & a_6 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_3 & a_4 \\ a_3 & a_4 & a_5 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \Rightarrow$$

$$\begin{bmatrix} c_3 & c_2 & c_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \Rightarrow f(a_1, a_2, a_3) \equiv (a_1 + a_3) \bmod 2$$

以上事实说明，当Eve获得了$n$阶线性反馈移位寄存器序列的任意连续$2n$个比特，Eve就获得了整个密钥流。

实际上，对线性反馈移位寄存器序列还有更为有效的攻击方法。当Eve不知道阶数$n$时，他还可以进行测试。这种测试攻击方法被称为序列的综合。

# 线性反馈移位寄存器序列的综合

定理 如果一个比特流是一个周期序列，则它一定是线性反馈移位寄存器序列。

证明 设比特流$k$的最小周期是$N$。则

$$l>N\text{后，}k_l=k_{l-N}\text{。}$$

因此比特流$k$为$N$阶线性反馈移位寄存器序列，抽头系数为$\{c_1, c_2, \ldots, c_N\}=\{0，0，\ldots，0，1\}$（即极小多项式$f(x)=1+x^N$），初始状态为$k_1k_2k_3\ldots k_N$。

**定义：　一个周期序列作为一个线性反馈移位寄存器序列，它的最小阶数称为它的线性复杂度。对应于这个阶的特征多项式称为该序列的极小多项式.**

**注意：**　一个周期序列作为一个线性反馈移位寄存器序列，可以有很多不同的阶数，其中它的最小周期就是它的一个阶数。因此，周期序列的线性复杂度一定不超过它的最小周期。

所谓序列的综合，就是寻找周期序列的线性复杂度$n$，并且求出极小多项式$f(x)$。序列的综合的两种最著名的算法是Berlekamp-Massey算法和Games-Chan算法。

# 线性反馈移位寄存器序列的小结

线性反馈移位寄存器序列能够实现:

- 小的计算量（$n$阶线性递归生成， 通常$n$不大）；
- 极大的最小周期（对于$m$序列，最小周期为$2^n$-1）；
- 良好的伪随机性（对于$m$序列， Golomb随机性假设成立）。

然而小的计算量得到小的线性复杂度（对于$m$序列，线性复杂度为$n$），很容易由短的一段（对于$m$序列，由长度为$2n$的一段）推断出整个序列（用B-M算法）。因此，线性反馈移位寄存器序列不能作为密钥流。

能否让线性复杂度很大?

(不能)

兼顾小的计算量和大的线性复杂度，需要使用非线性的生成方式。

# 非线性组合序列

域GF(2)上的$n$维函数（$n$维布尔函数）

$n$维布尔函数是这样的函数

$$y=g(x_1, x_2, \ldots, x_n):$$

- $n$个自变量$x_1, x_2, \ldots, x_n$取值均为0和1;
- 因变量$y$取值为0和1。

$n$维布尔函数的代数正规型：

$y=g(x_1, x_2, \ldots, x_n)$

$=a(0)$

$+a(1)x_1+a(2)x_2+ \ldots+a(n)x_n$

$+a(1, 2)x_1x_2+\ldots+a(n\text{-}1, n)x_{n\text{-}1}x_n$

$+a(1, 2, 3)x_1x_2x_3+\ldots+a(n\text{-}2, n\text{-}1, n)x_{n\text{-}2}x_{n\text{-}1}x_n$

$+\ldots$

$+a(1, \ldots, n)x_1\ldots x_n$

其中常数$\{a(0)$，$a(1)\sim a(n)$，$a(1, 2)\sim a(n-1, n)$，$a(1, 2, 3)\sim a(n-2, n-1, n)$，…，$a(1, \ldots, n)\}$称为系数，它们取0或1为值。

使得系数不为0的项的最高次数称为$n$维布尔函数的次数。

关于$n$维布尔函数的注解：$x_1^2 \equiv x_1$，因此只有混合高次项；又因此最高次数不超过$n$；系数组$\{a(0)\sim a(1, \ldots, n)\}$中一共有$2^n$个系数；函数$g(x_1, x_2, \ldots, x_n)$与系数组相互唯一。

**当$g(x_1, x_2, \ldots, x_n)$只含有一次项时，称$g$为线性函数；**

**当$g(x_1, x_2, \ldots, x_n)$只含有0次项和一次项时，称$g$为仿射函数；**

**当$g(x_1, x_2, \ldots, x_n)$含有高次项时，称$g$为非线性函数。**

# 非线性前馈序列

若比特流$k$由如下的方式生成：

（1） 选择$n$阶$m$序列$s=s_1s_2s_3\ldots$，其极小多项式为$f(x)$，其初始状态为$s_1s_2s_3\ldots s_n$；

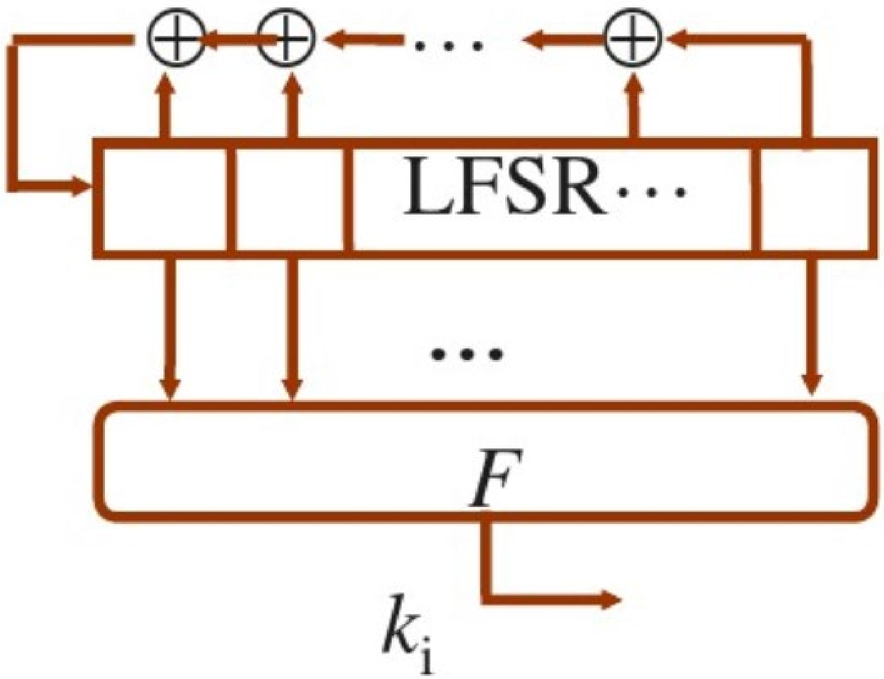
（2）对每个$l>0$，$k_l=g(s_l, s_{l+1}, \ldots, s_{l+n-1})$。

其中$g(x_1, x_2, \ldots, x_n)$为非线性的$n$维布尔函数。

**则**

- **称比特流$k$为非线性前馈序列。**
- **称$n$维布尔函数$y=g(x_1, x_2, \ldots, x_n)$为前馈函数，又称其为滤波函数。**
- **称$m$序列$s$为驱动序列。**

# 非线性前馈序列



当非线性前馈序列用作密钥流时，通常有三个部分可能作为通信伙伴的原始密钥：初始状态，极小多项式，非线性布尔函数。有以下三种不同的用法：

（1）（早期常用）原始密钥是初始状态，而将极小多项式和非线性布尔函数公开。此时原始密钥最短，但需要精心设计非线性布尔函数。

（2）（不常用）原始密钥是初始状态和极小多项式，而将非线性布尔函数公开。此时原始密钥长一些，但对非线性布尔函数的要求低一些。

（3）（很少用）原始密钥是初始状态、极小多项式、非线性布尔函数。此时原始密钥最长，但对非线性布尔函数的要求最低。

# 为什么$g$必须是非线性函数?

如果$g$是线性函数，则前馈序列$k$还是$n$阶$m$序列，并且与驱动序列$s$有相同的极小多项式（不给出证明）。

如果$g$是仿射函数，则前馈序列$k$是$n$阶$m$序列的补序列。

**希望：非线性前馈序列的线性复杂度极大，应该与$2^n$具有相同的数量级。**

# 前馈序列$k$的伪随机性如何？

$2^n$-1是前馈序列$k$的一个周期。换句话说，前馈序列$k$的最小周期必然是$2^n$-1的因子。

**希望：前馈序列$k$的最小周期就是$2^n$-1 (而不是$2^n$-1的真因子)。**

**希望：前馈序列$k$是0-1基本均衡的，即在一个最小周期内0和1的数量近似相等。**

**定义** 如果$n$维布尔函数$y=g(x_1, x_2, \ldots, x_n)$ 满足：当自变量$(x_1, x_2, \ldots, x_n)$跑遍 $2^n$个不同的值时，$y$取值为0的机会和$y$取值为1的机会相等（各为 $2^{n-1}$次），则称$g$为均衡函数。

**定理** **如果前馈函数是均衡函数，则**

**（1）前馈序列$k$的最小周期就是$2^n$-1（而不是$2^n$-1的真因子）。**

**（2）前馈序列$k$是0-1基本均衡的，即在一个最小周期内0和1的数量近似相等。**

**前馈序列最难设计的是游程分布的伪随机性和自相关的伪随机性。无论如何，需要精心地设计非线性布尔函数$g$。**

# 非线性组合序列

若比特流$k$由如下的方式生成：

（1） $M$个$n$阶$m$序列$s^{(j)}=s^{(j)}{}_1s^{(j)}{}_2s^{(j)}{}_3\ldots$； $j=1\sim M$;

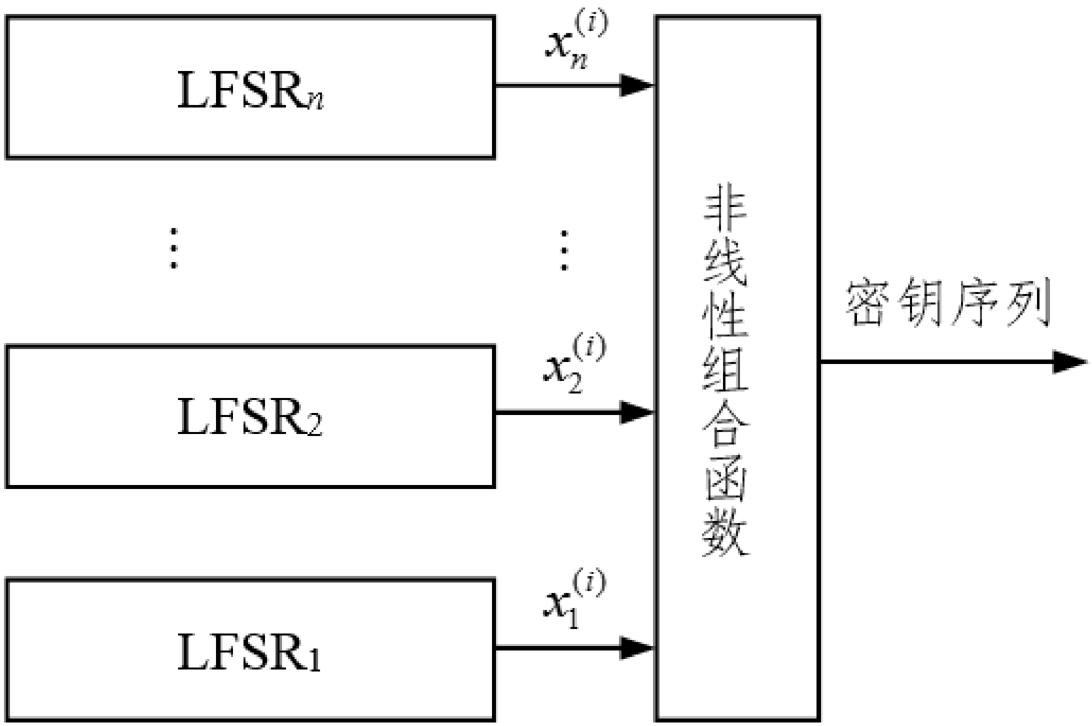
（2）对每个$l>0$,

$$k_l=g(s^{(1)}{}_l,\ s^{(2)}{}_l,\ \ldots,\ s^{(M)}{}_l)。$$

其中$g(x_1, x_2, \ldots, x_M)$为非线性的$M$维布尔函数。

则

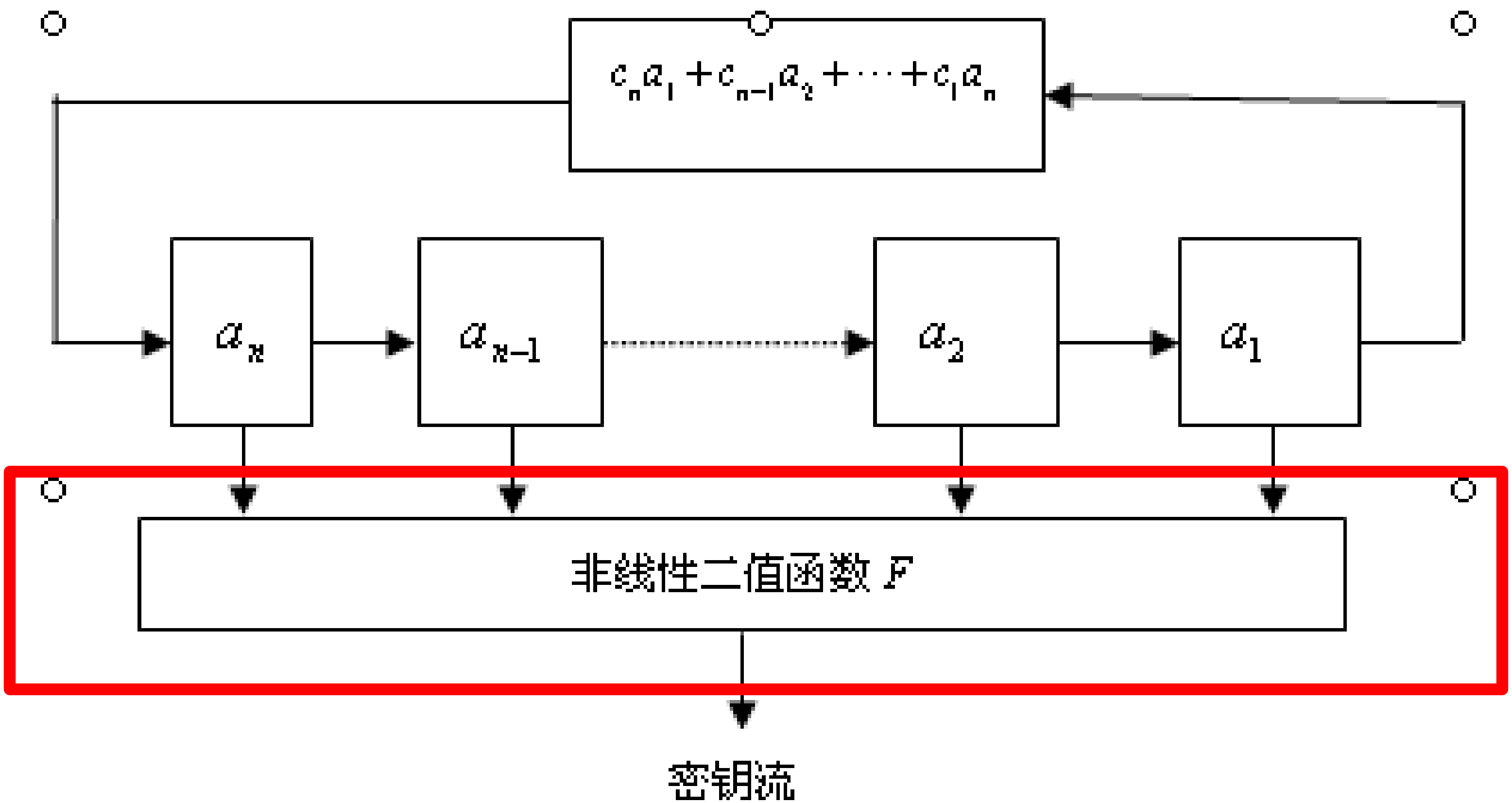
- 称比特流$k$为非线性组合序列。
- 称$M$维布尔函数$g(x_1, x_2, \ldots, x_M)$为组合函数。
- 称$M$个$n$阶$m$序列$s^{(j)}$为驱动序列。



LFSR$_n$
LFSR$_2$
LFSR$_1$
$x_n^{(i)}$
$x_2^{(i)}$
$x_1^{(i)}$
非线性组合函数
密钥序列

# 非线性组合部分

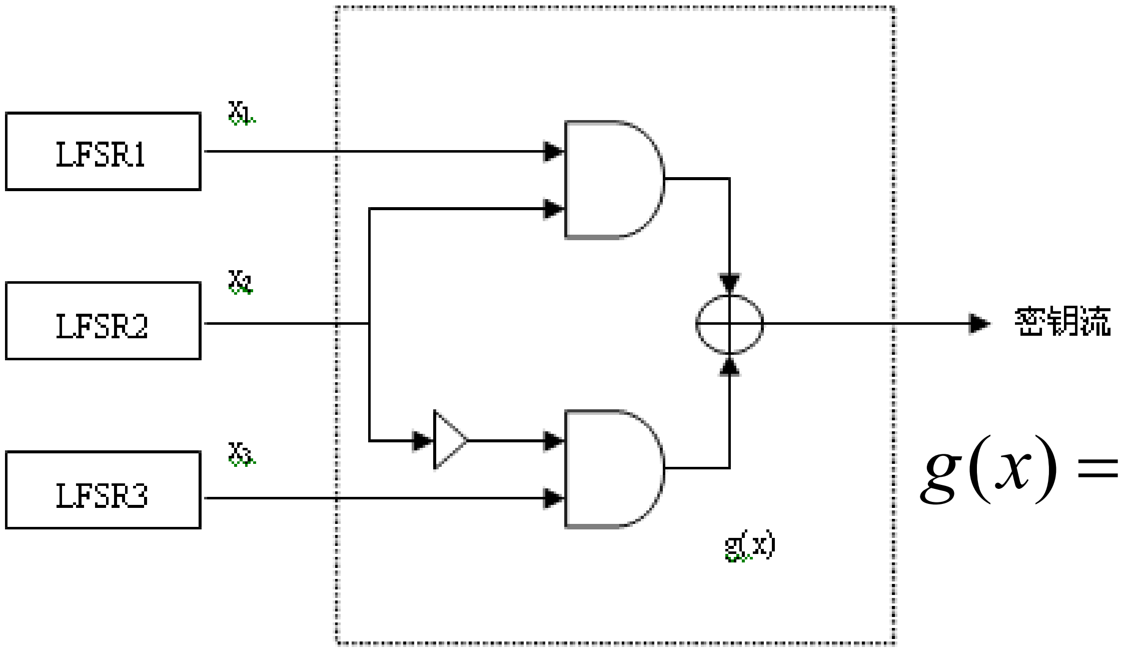
- 基于LFSR的序列密码（对一个LFSR进行非线性组合）：



# 常见的基于LFSR的密钥序列发生器

## 1. Geffe发生器

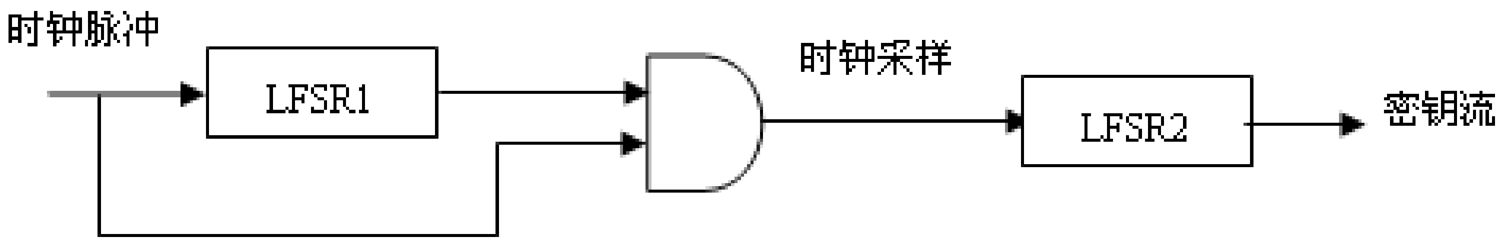
- 由三个线性反馈移位寄存器[LFSR1、LFSR2和LFSR3]以及一个非线性函数g(*x*)组成




$$g(x) = x_1 x_2 + \overline{x_2} x_3$$

## 2. 钟控发生器

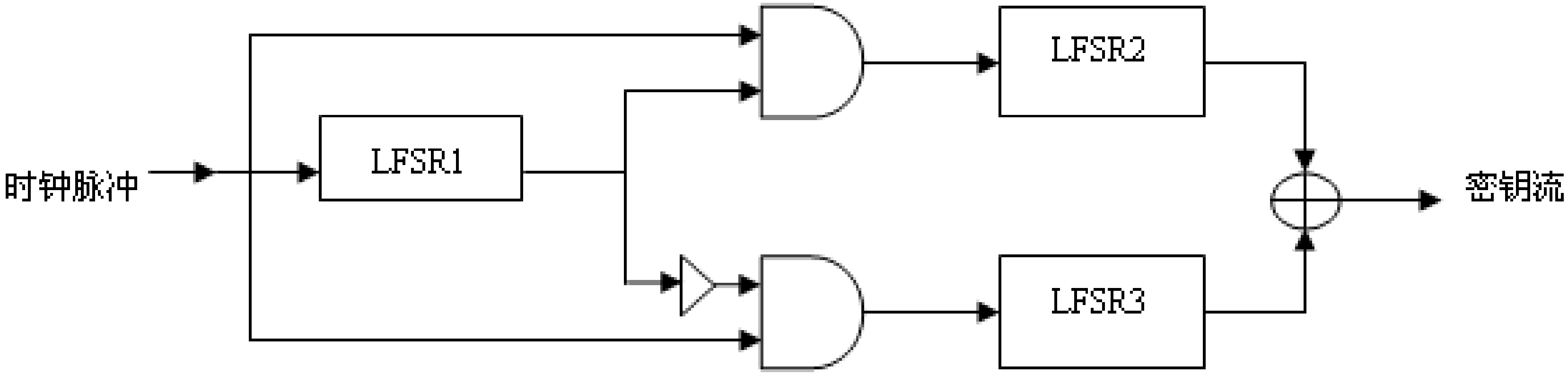
- 由控制序列（由一个或多个LFSR来控制生成）的当前值来决定采样序列寄存器移位次数
- 最基本的钟控发生器用一个LFSR控制另外一个LFSR的移位时钟脉冲，根据时钟脉冲的高低控制输出



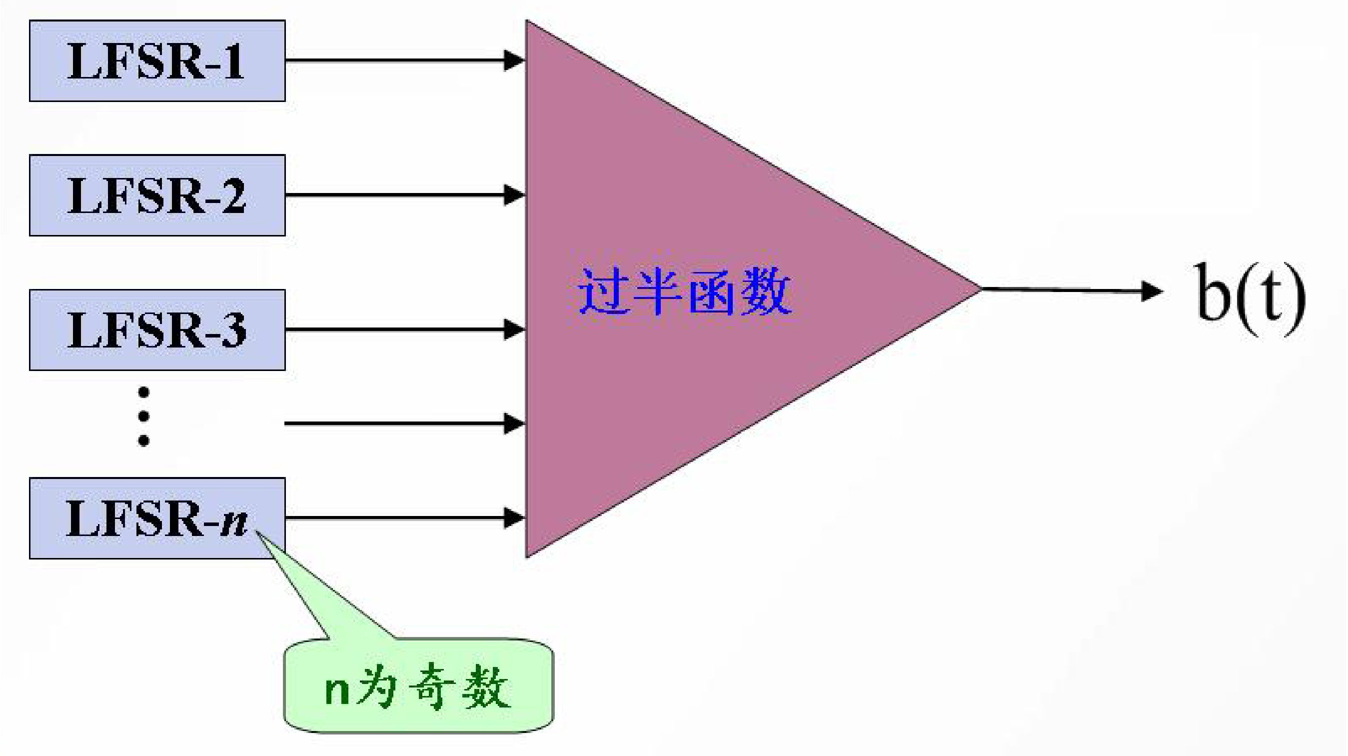

- 当LFSR1输出1时，移位脉冲通过与门使LFSR2进行一次移位；
- 当LFSR1输出0时，移位脉冲无法通过与门影响LFSR2，因此LFSR2重复输出前一位

## 3. 交错停走式发生器

也是一种钟控发生器，使用了三个不同级数的LFSR



# 4. 门限发生器



# 综合设计与使用

当前国际主流的流密码标准通常有以下结构：

- 非线性反馈+线性前馈；

  （例如Trivium）

- （非线性反馈，线性反馈）并行+非线性前馈；

  （例如Grain）

- 线性反馈+钟控+非线性前馈；
- 线性反馈+带记忆的非线性前馈；等等。

- ~~线性反馈+非线性前馈~~

当前，流密码的使用：

1. Alice和Bob协商了一个密钥种子；
2. Alice（Bob）向Bob（Alice）公开发送一个初始化向量；然后以（密钥种子，初始化向量）作为原始状态，共同将密钥流生成器递归1000步；然后以此时的状态作为初始状态，Alice（Bob）加密第一段消息，Bob（Alice）解密第一段消息；
3. ……

# 注解

- 密钥种子是秘密的，在多次加密/解密中保持不变；
- 初始化向量是公开发送的，每次加密/解密都要临时选择；
- 这样使用的目的是：

  （1）避免差错的扩散；

  （2）避免敌人截获长的密钥段。

# 常见流密码算法简介—RC4

- **RC4**：由MIT的Ron Rivest于1987年设计的、可变密钥长度、面向字节操作的、使用最为广泛的的序列密码之一；分析显示该密码的周期大于$10^{100}$

- RC4是一个典型的**基于非线性数组变换**的序列密码。它以一个足够大的数组为基础，对其进行非线性变换，产生非线性的密钥流序列

- **优点：容易用软件实现，加解密速度快**（大约比DES快10倍）

# Ron Rivest

•MIT计 安全公司的共同创始人
•耶鲁大
•斯坦福
•在密码 面做出了突出贡献
•最重要 Adleman（MIT的教授）一 Shamir、Adlema CM)颁发的图灵奖
•美国国 士，美国计算机协会院 艺术与科学院院士

# RC4算法描述

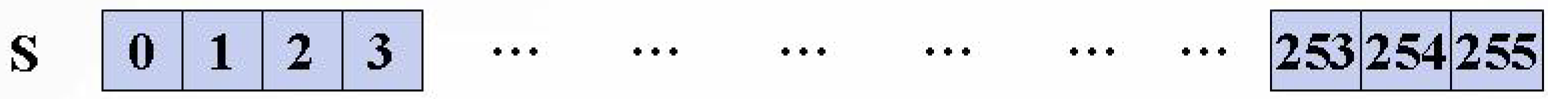
- RC4算法的大小根据参数n的值而变化，通常n=8，这样RC4可生成256（$2^8$）个元素的数据表S：$S_0$，$S_1$，$S_2$，…，$S_{255}$
- 种子密钥长度为1~256个字节（8~2048比特）的可变长度，用于初始化256个字节的初始向量S
- RC4有两个主要算法：
  - **密钥调度算法（KSA）**
  - **伪随机数生成算法（PRGA）**

# RC4的基本思想

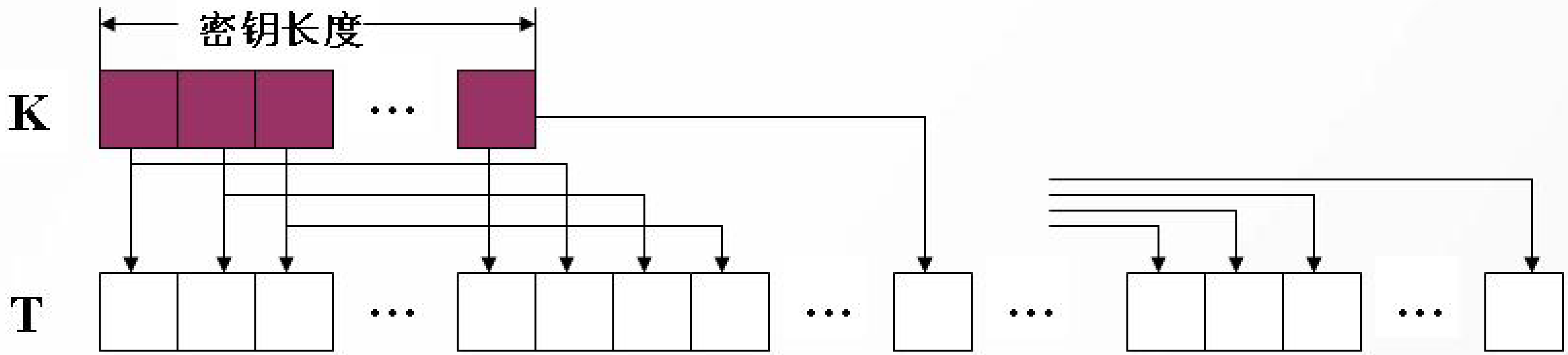
- 根据种子密钥，利用密钥调度算法对数据表S进行重新排列
- 利用伪随机数生成算法，从重新排列的数据表S中取出一个字节
- 每取出一个字节，数据表S将发生变化

# 数据表S的初始状态

初始化S，即S(i):=i(一个字节),0≤i≤255。



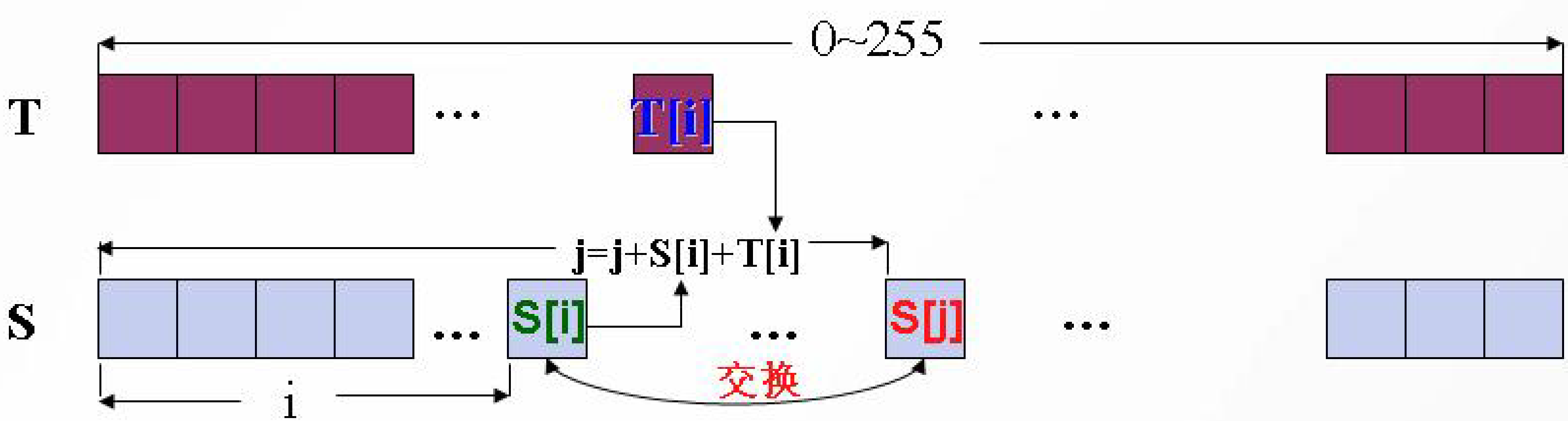

用主(种子)密钥K按字节填充另一个T表，
即T(i):=K(i mod keylen)，0≤i≤255。



# 数据表S的初始变换

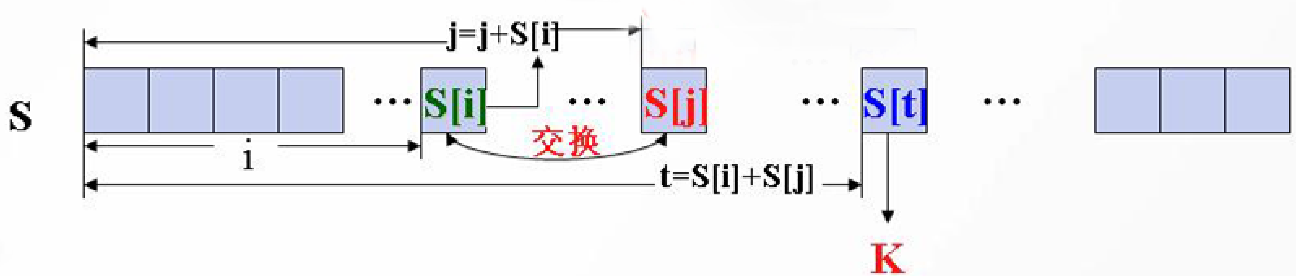
```
j:=0;
for i := 0 to 255 do
    begin
        j := (j + S[i] + T[i]) (mod 256);
        swap(S[i], S[j]);    // 交换S(i)和S(j)的内容;
    end
```

S[j]



# 密钥流的生成

```
i,j:=0
while (true)
  begin
     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];
 end
```



# RC4算法说明

- 加密时，将$k$的值与明文字节异或；解密时，将$k$的值与密文字节异或
- 为保证安全强度，目前的RC4至少使用128位密钥
- RC4算法可看成一个有限状态自动机，由$S$表和$i,j$索引组成RC4的一个状态：$T=(S_0,S_1,\ldots,S_{255},i,j)$。对状态$T$进行非线性变换，产生新的状态，并输出密钥序列中的一个字节$k$。大约有$2^{1700}$（$256!*2562$）种可能状态
- 用更大的数据表$S$和字长来实现这个思想是可能的，即定义16位RC4

# 举例说明

假如使用3位（从0到7）的RC4，其操作是对8取模（而不是对256取模）。数据表S只有8个元素，初始化为：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

选取一个密钥，该密钥是由0到7的数以任意顺序组成的。例如选取5、6和7作为密钥。该密钥如下填入密钥数据表中：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| K | 5 | 6 | 7 | 5 | 6 | 7 | 5 | 6 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

# 密钥调度算法KSA（举例）

然后利用如下循环构建实际的S数据表：

```
j:=0;
for i=0 to 7 do
    j:=(j+s(i)+k(i)) mod 8;
    swap(S(i),S(j));
```

该循环以j=0和i=0开始。使用更新公式后j为：

$$j=(0+S(0)+K(0)) \bmod 8=5$$

因此，S数据表的第一个操作是将S(0)与S(5)互换。

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S | 5 | 1 | 2 | 3 | 4 | 0 | 6 | 7 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

索引i加1后，j的下一个值为：

j=(5+S(1)+K(1)) mod 8=(5+1=6) mod 8=4

即将S数据表的S(1)和S(4)互换：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S | 5 | 4 | 2 | 3 | 1 | 0 | 6 | 7 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

当该循环执行完后，数据表S就被随机化：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S | 5 | 4 | 0 | 7 | 1 | 6 | 3 | 2 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

# 伪随机数生成算法PRGA（举例）

这样数据表S就可以用来生成随机的密钥流序列。从j=0和i=0开始，RC4如下计算第一个密钥字：

i=(i+1) mod 8=(0+1) mod 8 =1

j=(j+s(i)) mod 8=(0+s(1)) mod 8=(0+4) mod 8=4

swap S(1)和S(4)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S | 5 | 1 | 0 | 7 | 4 | 6 | 3 | 2 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

然后如下计算t和k:

t=(S(j)+S(i)) mod 8=(S(4)+S(1)) mod 8=(1+4) mod 8=5

k=S(t)=S(5)=6

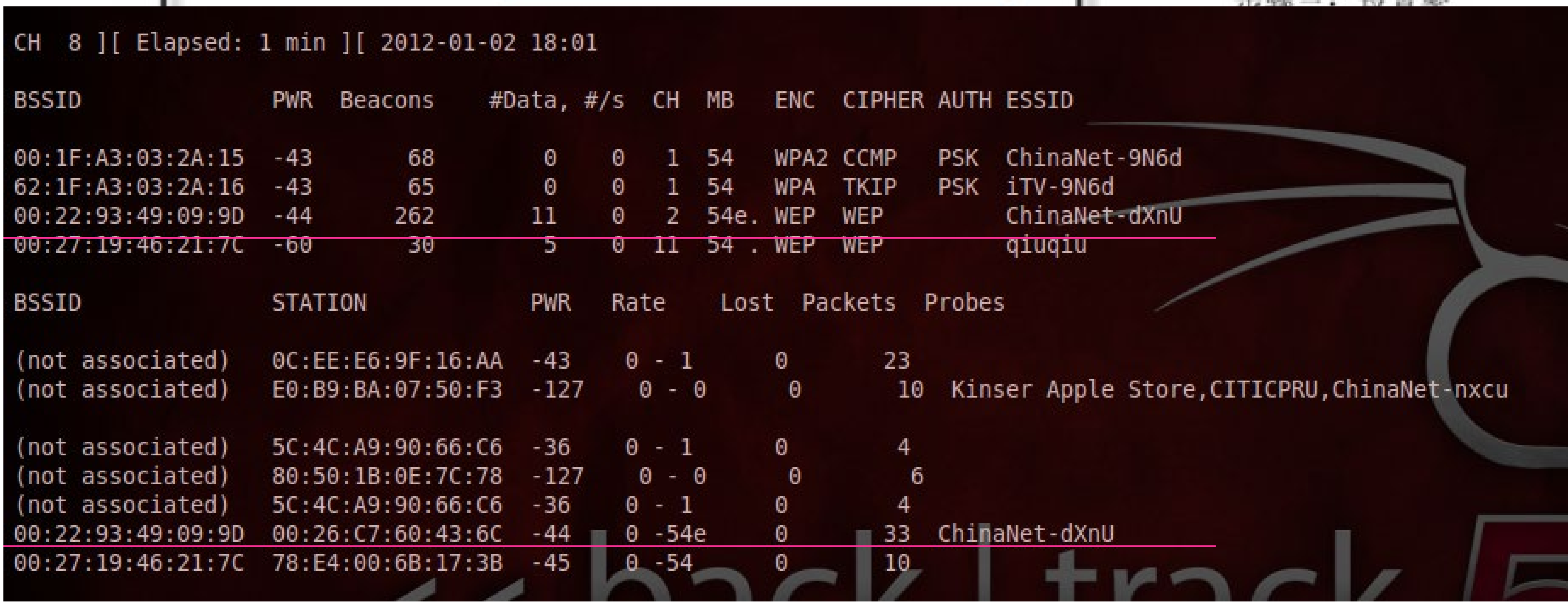
第一个密钥字为6，其二进制表示为110。反复进行该过程，直到生成的二进制的数量等于明文位的数量。

# RC4应用举例

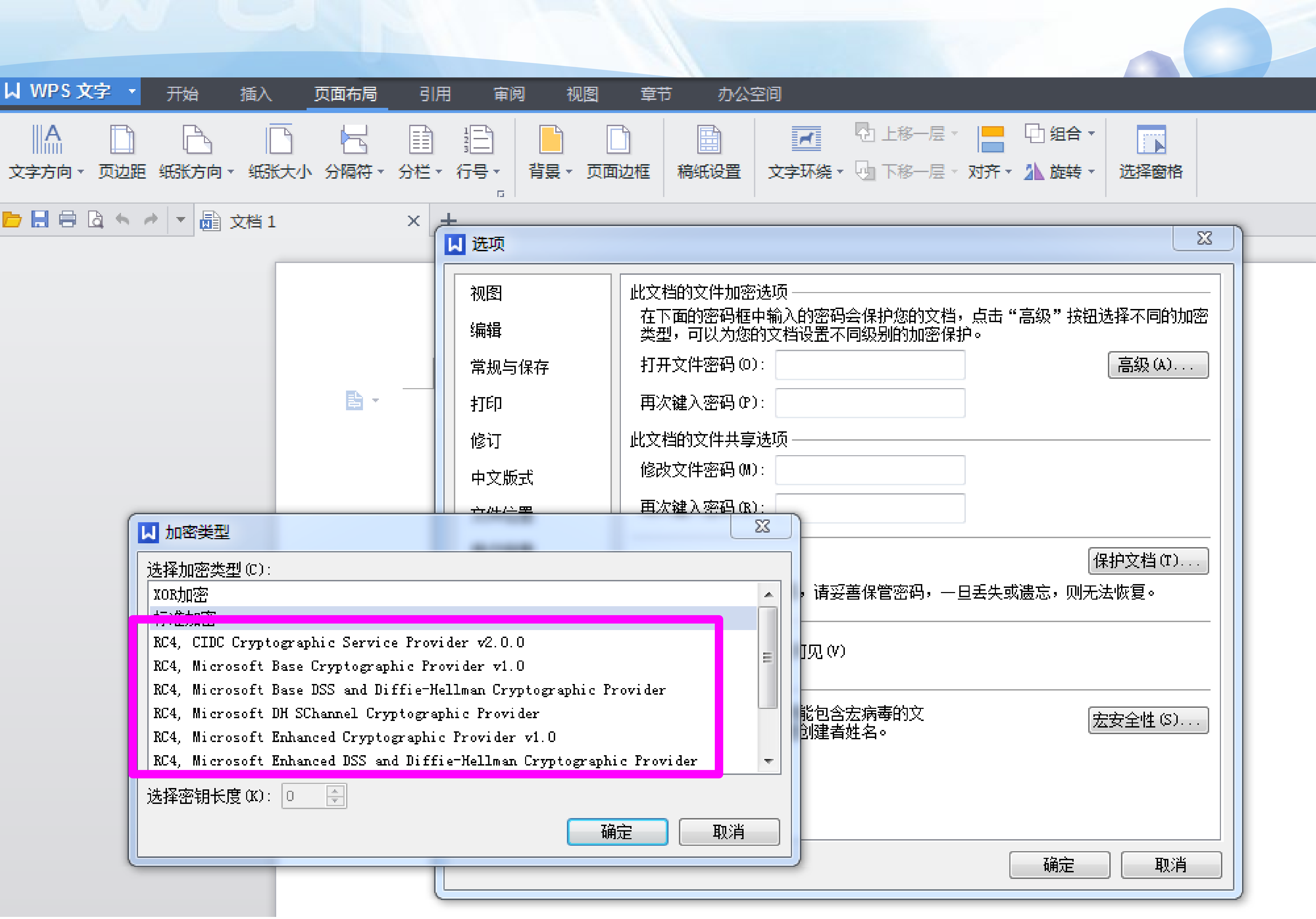
- **RC4目前使用中**
  - **SSL**（安全套接字）中广泛使用
  - **WEP(Wired Equivalent Privacy:有线对等保密) IEEE 802.11**
  - **Microsoft Windows**
  - **WPS Office**
  - **Lotus Notes**（世界领先的企业级通讯、协同工作及Internet/Intranet平台；IBM公司）

# RC4应用举例


中国第一大电子邮件服务商
163 网易免费邮
mail.163.com
126 网易免费邮
www.126.com
yeah.net
网易免费邮
网易手机号码邮箱
SHOUJI.163.com
登录163免费邮箱
cheungcumt
@163.com
密码
登 录
记住帐号
SSL安全登录 ?
忘记密码?
邮箱版本: 默认版本
注册网易免费邮

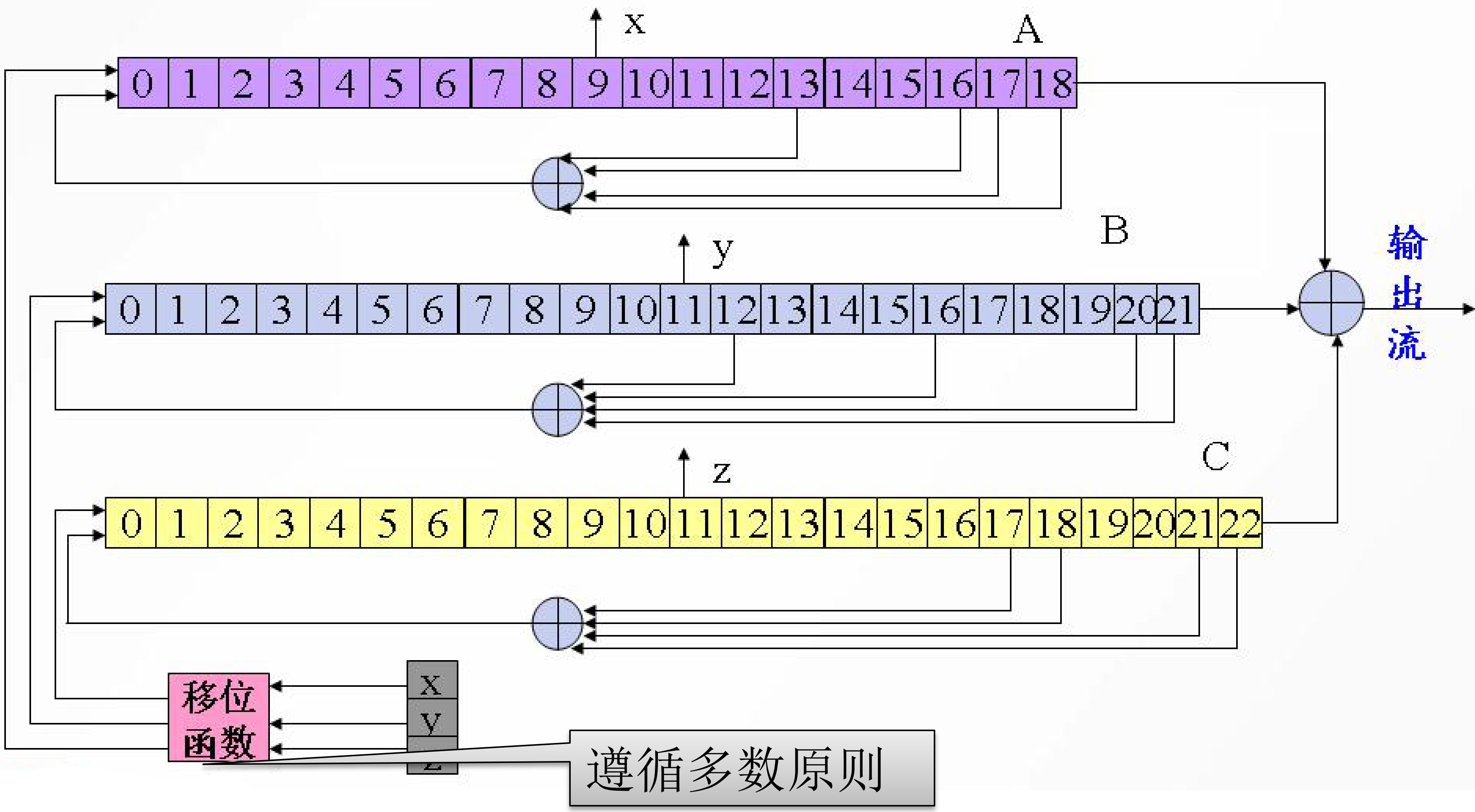
```
CH  8 ][ Elapsed: 1 min ][ 2012-01-02 18:01

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1F:A3:03:2A:15  -43       68        0    0   1  54   WPA2 CCMP   PSK  ChinaNet-9N6d
62:1F:A3:03:2A:16  -43       65        0    0   1  54   WPA  TKIP   PSK  iTV-9N6d
00:22:93:49:09:9D  -44      262       11    0   2  54e. WEP  WEP         ChinaNet-dXnU
00:27:19:46:21:7C  -60       30        5    0  11  54 . WEP  WEP         qiuqiu

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

(not associated)   0C:EE:E6:9F:16:AA  -43    0 - 1      0      23
(not associated)   E0:B9:BA:07:50:F3  -127    0 - 0      0      10  Kinser Apple Store,CITICPRU,ChinaNet-nxcu

(not associated)   5C:4C:A9:90:66:C6  -36    0 - 1      0       4
(not associated)   80:50:1B:0E:7C:78  -127    0 - 0      0       6
(not associated)   5C:4C:A9:90:66:C6  -36    0 - 1      0       4
00:22:93:49:09:9D  00:26:C7:60:43:6C  -44    0 -54e     0      33  ChinaNet-dXnU
00:27:19:46:21:7C  78:E4:00:6B:17:3B  -45    0 -54      0      10
```


WPS 文字
开始
插入
页面布局
引用
审阅
视图
章节
办公空间
文字方向
页边距
纸张方向
纸张大小
分隔符
分栏
行号
背景
页面边框
稿纸设置
文字环绕
上移一层
下移一层
对齐
组合
旋转
选择窗格
文档 1
选项
视图
编辑
常规与保存
打印
修订
中文版式
此文档的文件加密选项
在下面的密码框中输入的密码会保护您的文档，点击“高级”按钮选择不同的加密类型，可以为您的文档设置不同级别的加密保护。
打开文件密码 (O):
高级 (A)...
再次键入密码 (P):
此文档的文件共享选项
修改文件密码 (M):
保护文档 (T)...
，请妥善保管密码，一旦丢失或遗忘，则无法恢复。
宏安全性 (S)...
确定
取消
加密类型
选择加密类型 (C):
XOR加密
RC4, CIDC Cryptographic Service Provider v2.0.0
RC4, Microsoft Base Cryptographic Provider v1.0
RC4, Microsoft Base DSS and Diffie-Hellman Cryptographic Provider
RC4, Microsoft DH SChannel Cryptographic Provider
RC4, Microsoft Enhanced Cryptographic Provider v1.0
RC4, Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider
选择密钥长度 (K):
确定
取消

# 常见流密码算法简介—A5

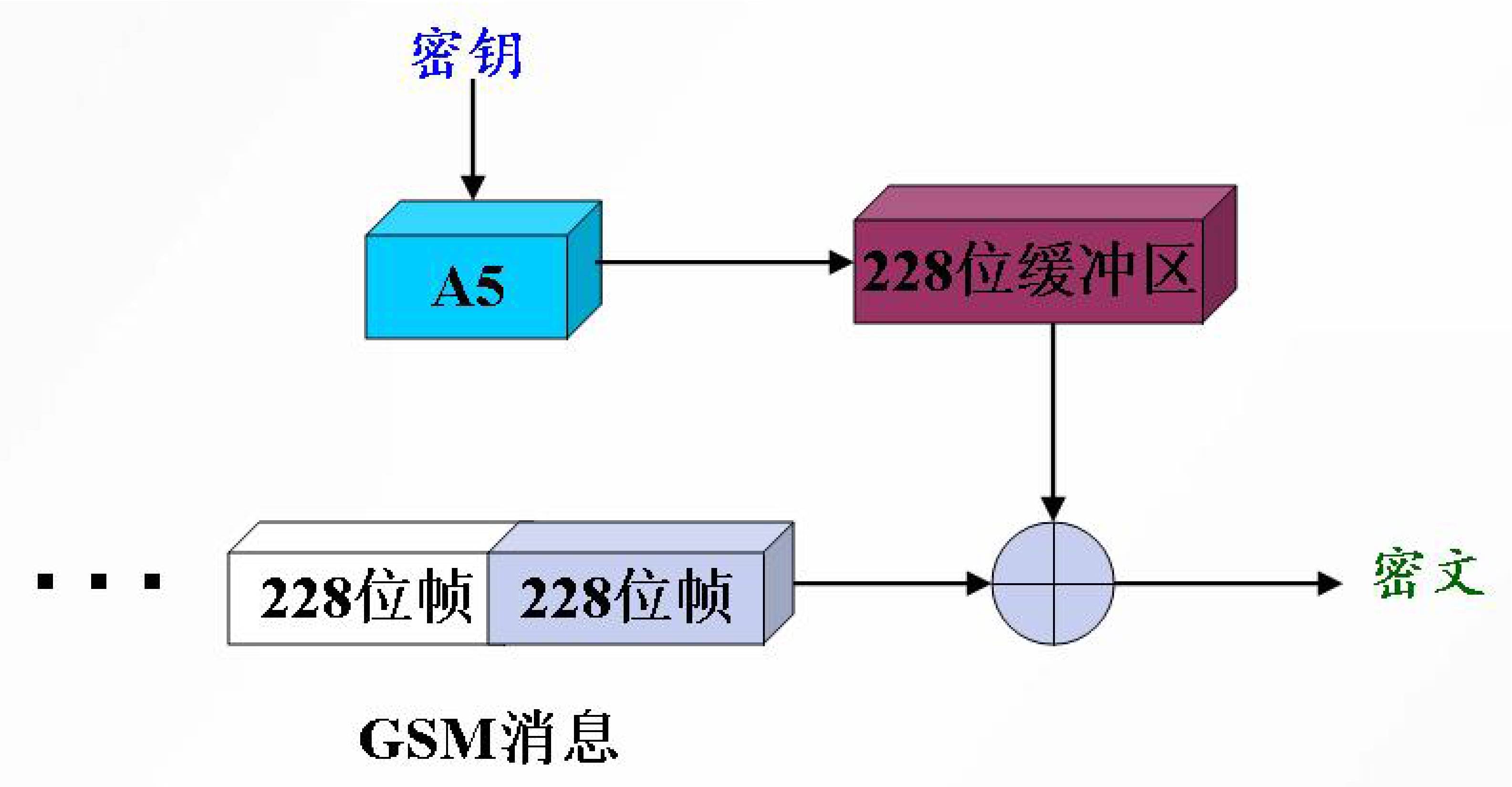
- A5算法已被应用于GSM通信系统中，用于加密从手机到基站的连接，以保护语音通信。一个GSM语言消息被转换成一系列的帧，每帧长228位，每帧用A5进行加密
- A5算法主要由三个长度不同的线性移位寄存器组成，即A, B, C。其中A有19位，B有22位，C有23位
- 移位由时钟控制的，且遵循“择多”的原则。即从每个寄存器中取出一个中间位，三个数中占多数的寄存器参加移位，其余的不移位。
  - 比如取出的三个中间位中有两个为“1”，则为“1”的寄存器进行一次移位，为“0”的不移。反过来，若三个中间位中有两个为“0”，则为“0”的寄存器进行一次移位，而为“1”的不移

# A5算法示意图



x
A
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
y
B
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
z
C
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
输出流
移位函数
x
y
z
遵循多数原则

# GSM中使用A5流密码算法



# A5的安全性

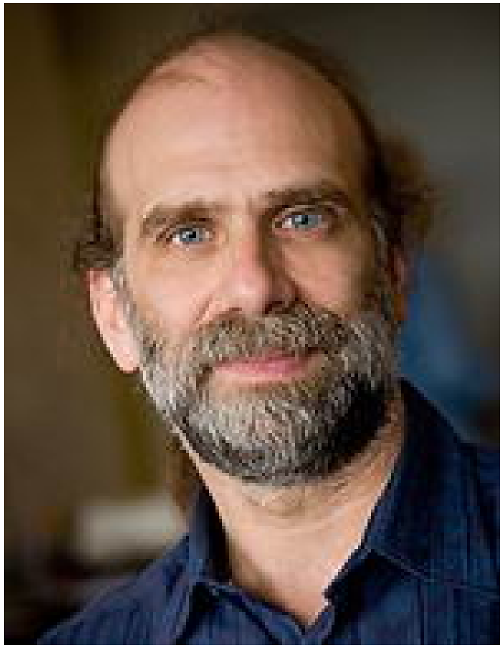
- 2000年：Alex Biryukov、Adi Shamir和David Wagner展示了一种攻击，在几分钟内就从小的已知明文中找到密钥，但它还需要一个$2^{48}$步的预处理过程

- 2003年：Ekdahl和Johannson可以运用2到5分钟的明文记录在几分钟内破解A5

# A5算法的创建者

Ross Anderson

http://www.cl.cam.ac.uk/~rja14/



Bruce Schneier

http://www.schneier.com/

# 其他序列密码

- 最优软件加密算法（SEAL，Software Encryption Algorithm）
  - 1993年由IBM公司的Rogaw和Coppersmith设计
- SNOW
  - 2000年由Partik Ekdahl和Thomas Johansson设计
- PKZIP算法
  - 广泛用于计算机文档数据压缩程序
- WAKE算法
  - 字自动密钥加密算法，由David Wheeler设计

# 序列密码相对于分组密码的优点

- 在硬件实施上，序列密码的速度一般要比分组密码快，而且不需要有很复杂的硬件电路。
- 在某些情况下（例如某些电信上的应用），当缓冲不足或必须对收到字符进行逐一处理时，序列密码就显得更加必要和恰当。
- 序列密码有较理想的数学分析。
- 序列密码能较好地隐藏明文的统计特性。

# 本章小结

- 序列密码算法的基本概念
- 密钥流生气器的基本设计方法
- LFSR的基本结构和性质
- 非线性组合部分
- 常见序列密码算法介绍
  - RC4—基于非线性数组变换
  - A5—基于LFSR