

密码学

第七讲 序列密码

张焕国

武汉大学计算机学院

空天信息安全与可信计算教育部重点实验室





内容简介

- 第一讲 信息安全概论
- 第二讲 密码学的基本概念
- 第三讲 数据加密标准 (DES)
- 第四讲 高级数据加密标准 (AES)
- 第五讲 中国商用密码 (SMS4)
- 第六讲 分组密码的应用技术
- 第七讲 序列密码**
- 第八讲 复习
- 第九讲 公钥密码 (1)





内容简介

第十讲 公钥密码 (2)

第十一讲 数字签名 (1)

第十二讲 数字签名 (2)

第十三讲 **HASH**函数

第十四讲 认证

第十五讲 密码协议

第十六讲 密钥管理 (1)

第十七讲 密钥管理 (2)

第十八讲 复习

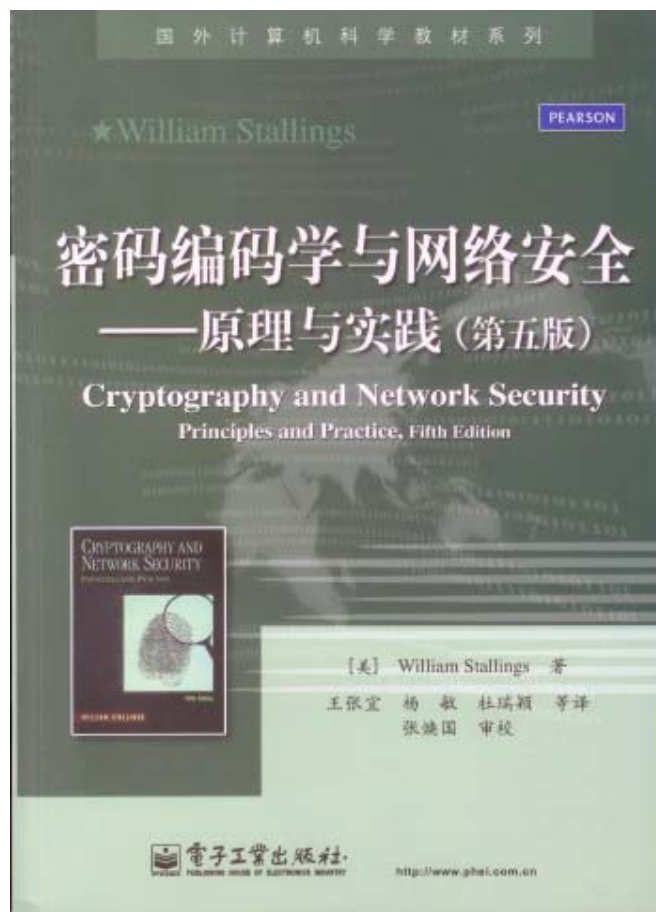


教材与主要参考书

教材



参考书



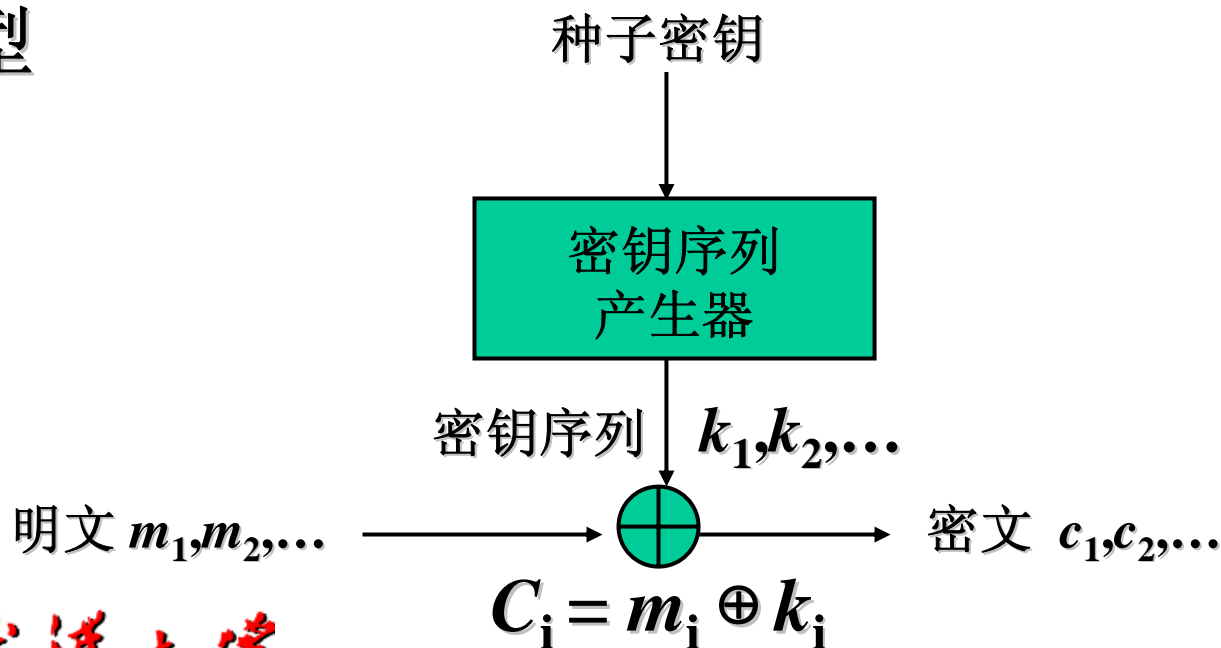
武汉大学

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





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。





一、序列密码的基本概念

1、序列密码的分类

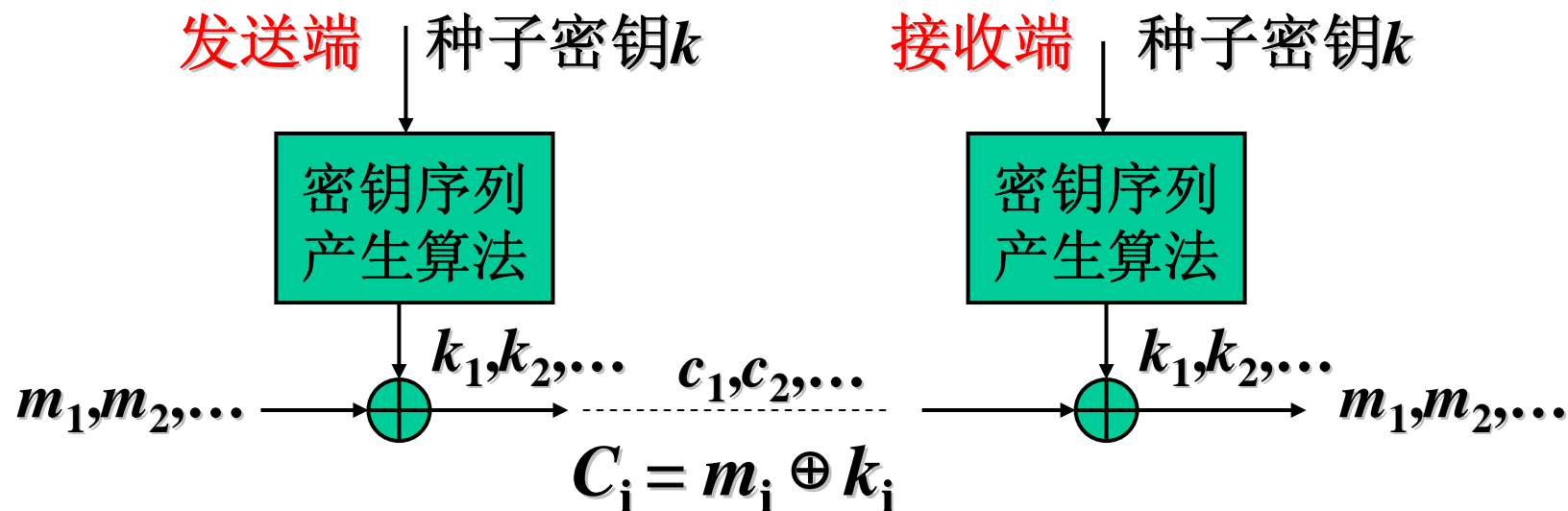
①同步序列密码 (Synchronous Stream Cipher)

- 密钥序列产生算法与明文（密文）无关，所产生的密钥序列也与明文（密文）无关。
- 在通信过程中，通信的双方必须保持精确的同步，收方才能正确解密，如果失步收方将不能正确解密。例如，如果通信中丢失或增加了一个密文字符，则收方的解密将一直错误。



一、序列密码的基本概念

①同步序列密码



设密文失步 $c = c_1, c_3, c_4, \dots, c_{n-1}, c_n$ (c_2 丢失)

$\oplus k = k_1, k_2, k_3, \dots, k_{n-2}, k_{n-1}$ (密钥正确)

$m = m_1, \times, \times, \dots, \times, \times$ (m_1 后明文全错)





一、序列密码的基本概念

①同步序列密码

- 对失步的敏感性，使我们能够容易检测插入、删除、重播等主动攻击。
- 另一个优点是没有错误传播，当通信中某些密文字符产生了错误（0错成1或1错成0，不是插入和删除），只影响相应字符的解密，不影响其它字符。
- 注意：错误与失步是不同的概念！

设密文错误 $c = c_1, c_2, c_3, \dots, c_{n-1}, c_n$ （ c_2 错）

$\oplus k = k_1, k_2, k_3, \dots, k_{n-1}, k_n$ （密钥正确）

$m = m_1, \times, m_3, \dots, m_{n-1}, m_n$ （仅 m_2 错）





一、序列密码的基本概念

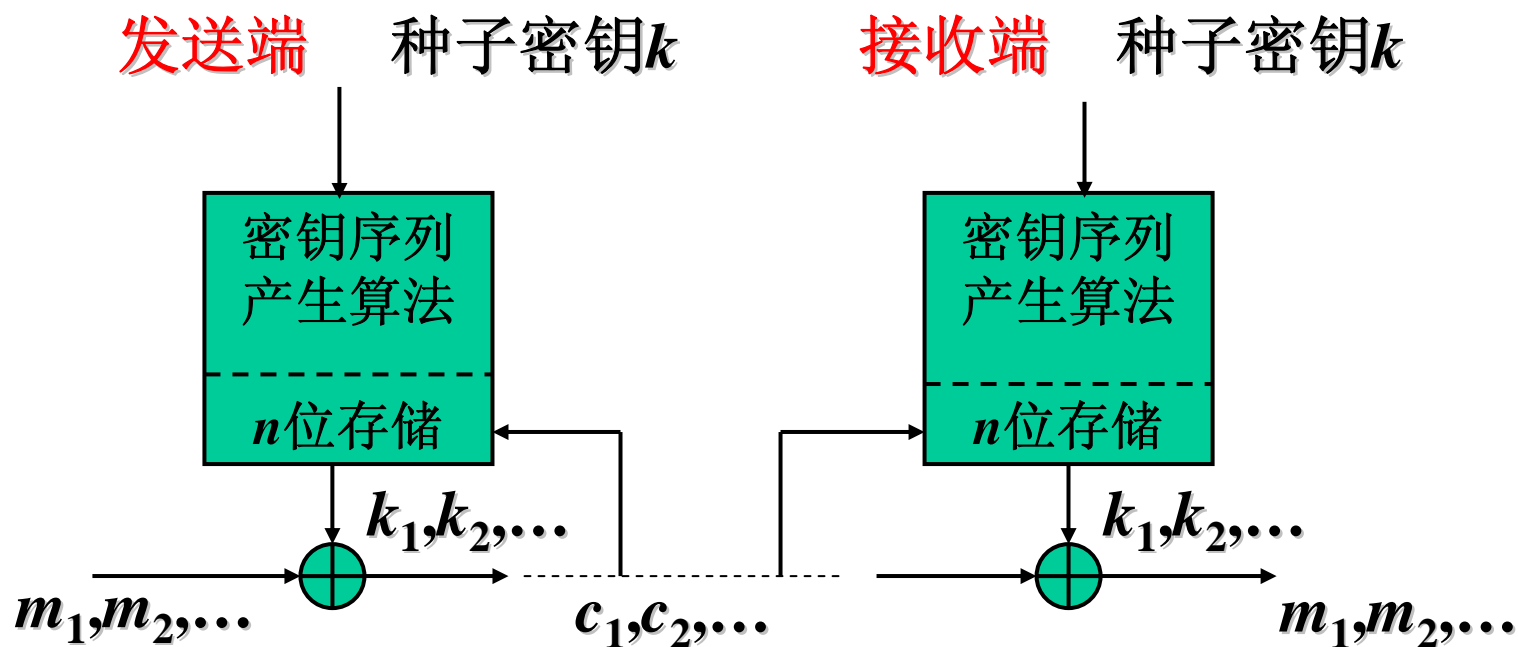
② 自同步序列密码 (Self- Synchronous Stream Cipher)

- 密钥序列产生算法与明文 (密文) 相关, 则所产生的密钥序列与明文 (密文) 相关。
- 设密钥序列产生器具有 n 位存储, 则加密时一位密文错误将影响后面连续 n 个密文错误。在此之后恢复正确。
- 解密时一位密文错误也将影响后面连续 n 个明文错。在此之后恢复正确。
- 加解密会造成错误传播。但错误传播有界, 在错误过去之后恢复正确。



一、序列密码的基本概念

② 自同步序列密码



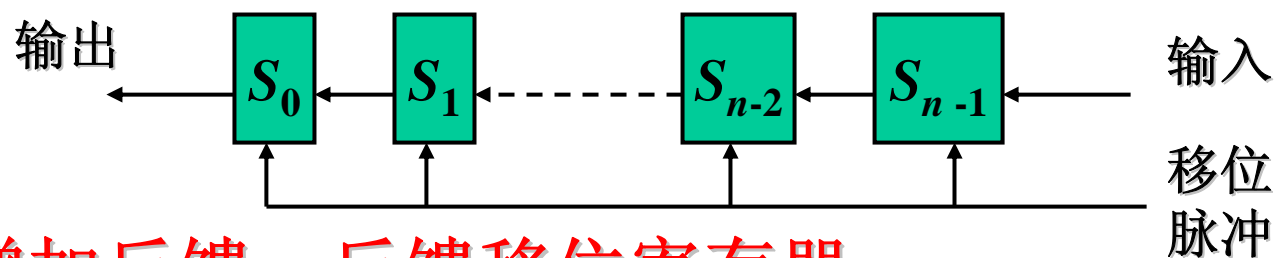
C_i 的错误将影响 n 位



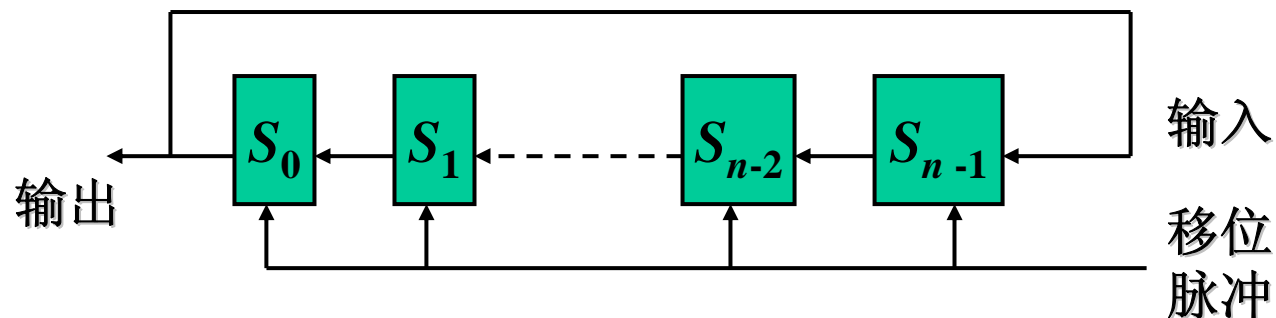
二、线性移位寄存器序列密码

1、线性移位寄存器 (Linear Shift Register)

● 例1 移位寄存器



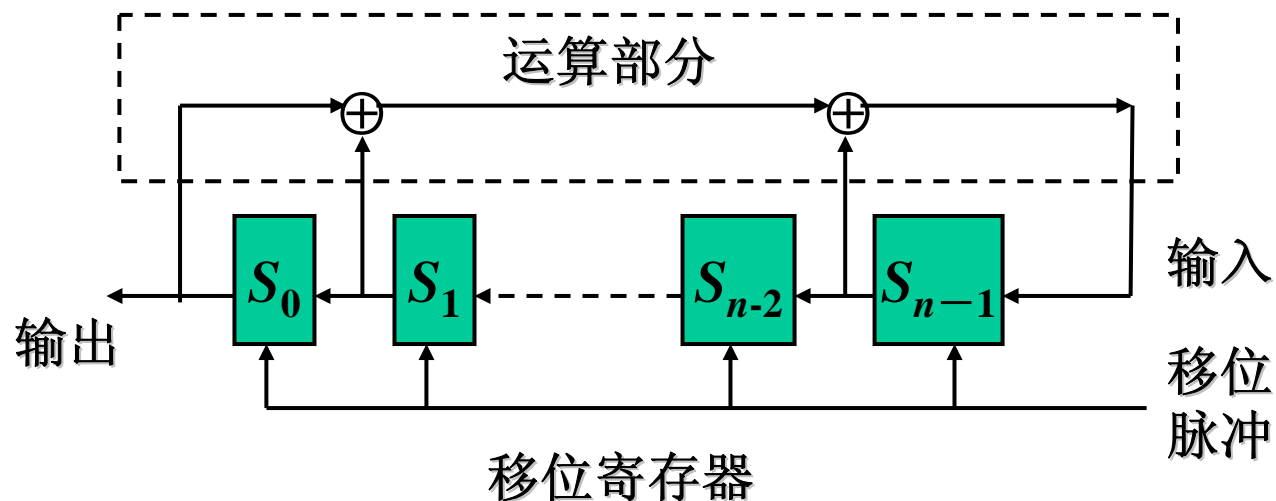
● 例2 增加反馈，反馈移位寄存器



二、线性移位寄存器序列密码

1、线性移位寄存器 (Linear Shift Register)

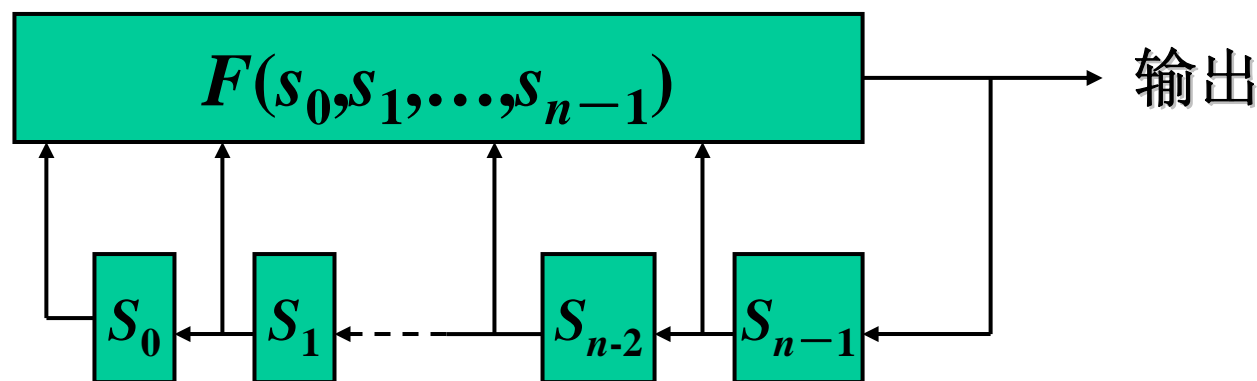
● 例3 增加运算，线性移位寄存器



二、线性移位寄存器序列密码

1、线性移位寄存器 (Linear Shift Register)

● 一般模型





二、线性移位寄存器序列密码

1、线性移位寄存器 (Linear Shift Register)

- 图中 s_0, s_1, \dots, 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 $F(s_0, s_1, \dots, s_{n-1})$ 计算产生。
- 称函数 $F(s_0, s_1, \dots, s_{n-1})$ 为反馈函数。
- 如果反馈函数 $F(s_0, s_1, \dots, s_{n-1})$ 是 s_0, s_1, \dots, 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。





二、线性移位寄存器序列密码

1、线性移位寄存器

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

$$F(s_0, s_1, \dots, s_{n-1}) = g_0 s_0 + g_1 s_1 + \dots + g_{n-1} s_{n-1}$$

其中， g_0, g_1, \dots, g_{n-1} 为反馈系数。

- 在 $GF(2)$ 的情况下，式中的 $+$ 即为 \oplus ，反馈系数 $g_i \in GF(2)$ ，如果 $g_i=0$ ，则表示式中的 $g_i s_i$ 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。





二、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：

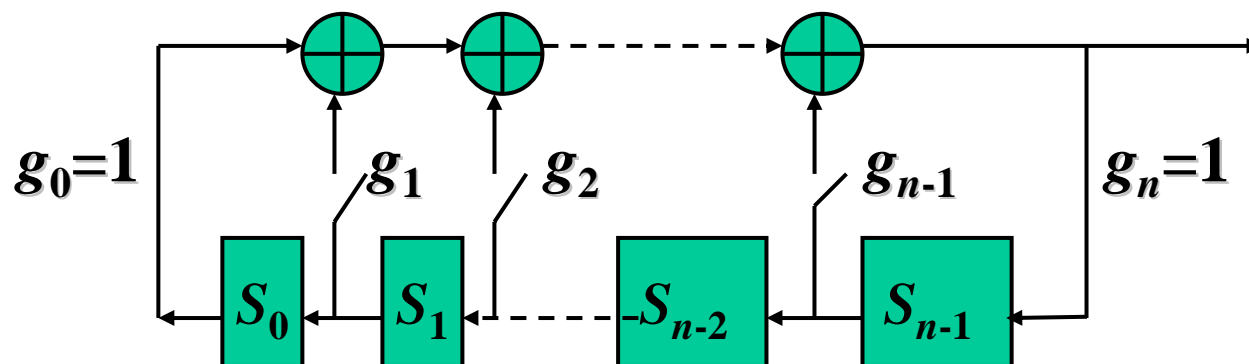
$$g(x) = g_n x^n + g_{n-1} x^{n-1} + \dots + g_1 x + g_0$$

- 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。



二、线性移位寄存器序列密码

1、线性移位寄存器





二、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 $\leq 2^n - 1$ ，其输出序列的周期 $\leq 2^n - 1$ 。
- 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。





二、线性移位寄存器序列密码

1、线性移位寄存器

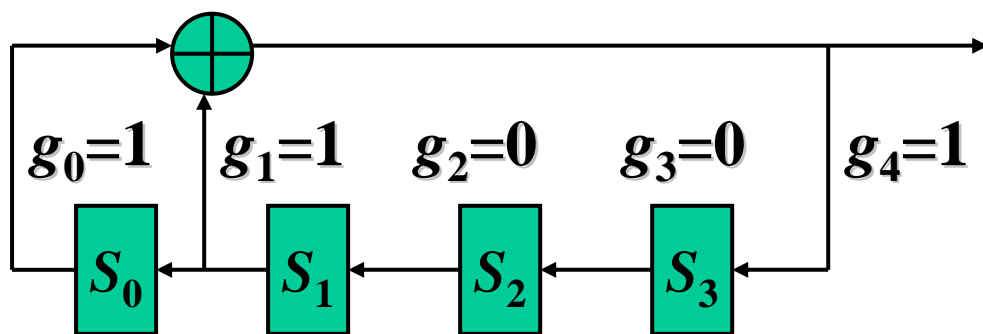
- 仅当连接多项式 $g(x)$ 为本原多项式时，其线性移位寄存器的输出序列为 m 序列。
- 设 $f(x)$ 为 $GF(2)$ 上的多项式，使 $f(x) \mid x^p - 1$ 的最小正整数 p 称为 $f(x)$ 的周期。如果 $f(x)$ 的次数为 n ，且其周期为 $2^n - 1$ ，则称 $f(x)$ 为本原多项式。
- 已经证明，对于任意的正整数 n ，至少存在一个 n 次本原多项式，而且存在有效的产生算法。



二、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	





二、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：

游程：称序列中连续的 i 个1为长度等于 i 的1游程，同样，称序列中连续的 i 个0为长度等于 i 的0游程。

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} ；





二、线性移位寄存器序列密码

1、线性移位寄存器

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（ $1 \leq i \leq n-2$ ）：

■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。





二、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

定义： 设 $\{k_i\}$ 是周期为 p 的序列， k_0, k_1, \dots, k_{p-1} 是其中一个周期子段，则 $k_{0+\tau}, k_{1+\tau}, \dots, k_{p-1+\tau}$ 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：

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

- 自相关函数反映一个周期内平均每位的相同程度。





二、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

$$R(\tau) = \begin{cases} 1 & , \tau = 0 \\ -1/P & , 0 < \tau \leq P-1 \end{cases}$$

● *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

● $A=7$, $D=8$, $R(\tau) = -1/15$





二、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。





二、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

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

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

$$s'_0 = s_1$$

$$s'_1 = s_2$$

...

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

$$s'_{n-1} = g_0 s_0 + g_1 s_1 + \dots + g_{n-1} s_{n-1}$$





二、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: $S' = HS \pmod 2$

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

矩阵 H 称为连接多项式的伴侣矩阵。





二、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：

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

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

于是可求出一段长 $2n$ 位的密钥序列，

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

其中

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





二、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:

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

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

...

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

作矩阵

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

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





二、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 $S'=HS \bmod 2$, 有

$$S_2=HS_1$$

$$S_3=HS_2$$

...

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

于是,

$$Y=HX \bmod 2$$





二、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





二、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。





三、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：
 - ① 非线性移位寄存器序列
 - ② 对线性移位寄存器序列进行非线性组合
 - ③ 利用非线性分组码产生非线性序列





三、非线性序列密码

①非线性移位寄存器序列

令反馈函数 $f(s_0, s_1, \dots, s_{n-1})$ 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。**





三、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 $\text{GF}(2)$ 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。

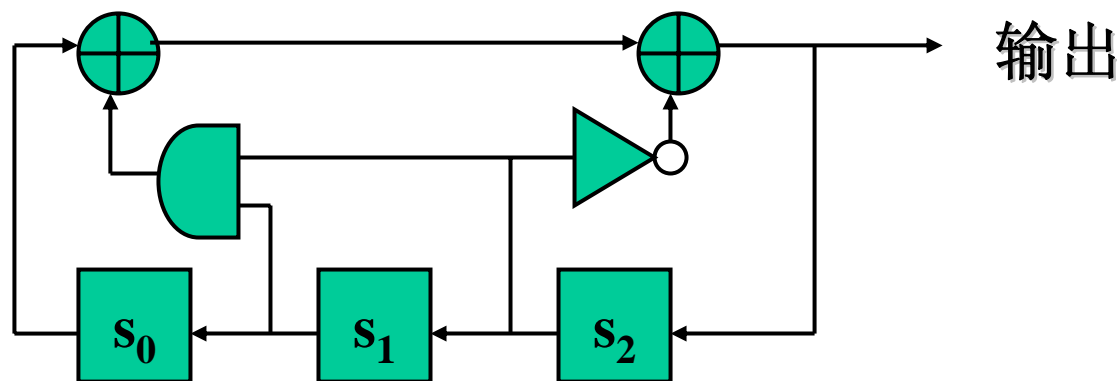
显然，非线性移位寄存器的空间极大！



三、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。





三、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。





三、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。

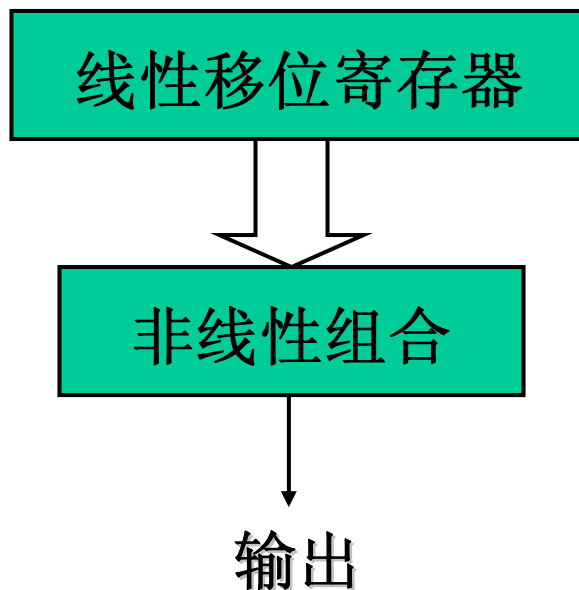




三、非线性序列密码

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

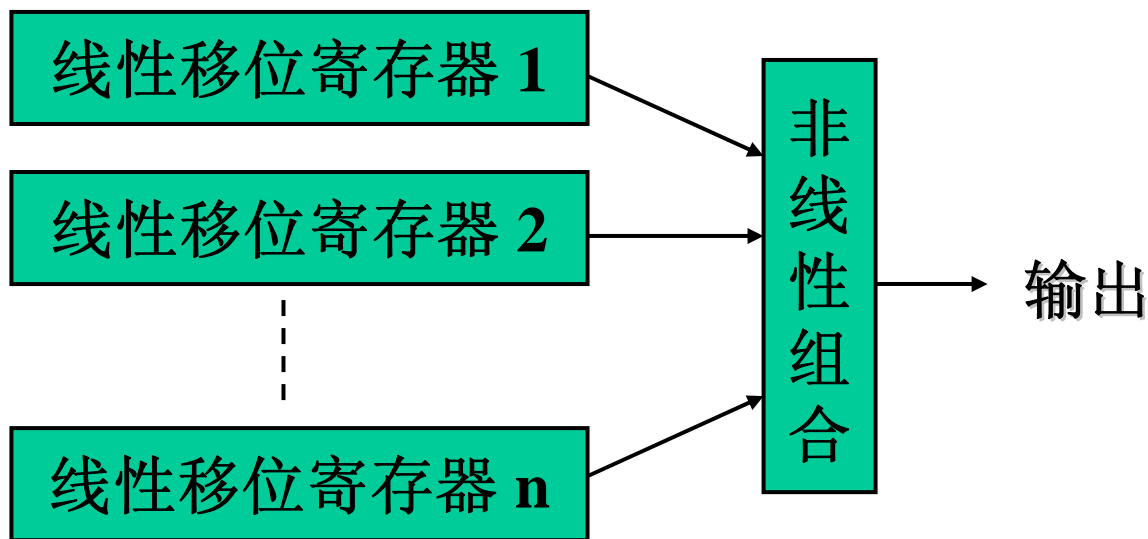





三、非线性序列密码

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





四、RC4序列密码

- RC4序列密码是美国RSA数据安全公司设计的一种序列密码。RSA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公司于1997年公布了RC4密码算法。
- 密钥40位的RC4密码，通过Internet 32小时可攻破。





四、RC4序列密码

- RC4密码与基于移位寄存器的序列密码不同。
- 它是一种基于非线性数据表变换的序列密码。
- 它以一个足够大的数据表为基础，对表进行非线性变换，产生非线性的密钥序列。






四、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, \dots, S_{255} 是 $0, 1, \dots, 255$ 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：

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

- 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 。





四、RC4序列密码

●RC4的下一状态函数定义如下：

- (1) $I=0, J=0$;
- (2) $I=I+1 \bmod 256$;
- (3) $J=J+S[I] \bmod 256$;
- (4) 交换 $S[I]$ 和 $S[J]$ 。

●RC4的输出函数定义如下：

- (1) $h=S[I]+S[J] \bmod 256$;
- (2) $k=S[h]$ 。





四、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

$S[0]=0, S[1]=1, S[2]=2, \dots, S[255]=255;$

(2) 用密钥填充另一个**256**字节的**R**表 **$R[0], R[1], \dots, R[255]$** ，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。

(3) $J=0;$

(4) 对于 **$I=0$** 到**255**重复以下操作：

① $J = (J + S[I] + R[I]) \bmod 256;$

② 交换 **$S[I]$** 和 **$S[J]$** 。





四、RC4序列密码

- **注意：**对S表初始化的过程是对S表进行随机化处理的过程，只有当这一过程完成后，才能计算产生密钥字符，才能进行加解密，否则将是不安全的。
- **RC4**算法的优点是算法简单，高效，特别适合软件实现。
- **RC4**是目前应用最广的商用序列密码。





作业题

1、p129第2题,第3题。

自选实践题

1、p129第10题。





谢 谢！



武汉大学