

第3章 大数分解

3.1 Pollard $p-1$ 因数分解法

RSA 公钥密码的出现,掀起了人们对大数分解进行研究的热潮,如果 $n=pq$ 被分解成功,则 RSA 系统便被攻破,所以必须选择好 p 和 q ,使之分解十分困难或不可能。

近些年来,数学家在大数分解上取得了引人注目的成就。1994 年 4 月,Lenstra 领导的跨国科研群体约 600 人集中了 1600 台计算机在网上联合作战,花费 38 个月的时间,分解成功了被称为 RSA129 的这个长达 129 位的十进制数。

1999 年 8 月,由 6 个国家的科学家组成的团队,将 RSA155 分解成功。

RSA155 = 1 094 173 864 157 052 742 180 970 732 204 035 761 200 373 294 544 920 599

013 842 131 476 349 984 288 934 784 717 997 257 891 267 332 497 625 752

899 781 833 797 076 537 244 027 146 743 531 593 354 333 897

p = 102 639 592 829 741 105 772 054 196 573 991 675 900 716 567 808 038 066 803 341

933 521 790 711 307 779

q = 106 603 488 380 168 454 820 927 220 360 012 878 679 207 958 575 989 291 522 270

608 237 193 062 808 643

对此数学家做了复杂性估计,每增加 10 位十进制数,分解的复杂性将增加 10 倍。按此估计,若 n 取 200 位十进制数将是很安全的。

现在回到 $p-1$ 因数分解法,或 Pollard $p-1$ 因数分解法。假定要对 n 进行因数分解, p 是 n 的某一个因数,若 $p-1$ 不存在大的素数因子,则下面的方法可供找出因数 p 。

S1 选一整数 k ,要求 k 为比界 B 小的所有数的倍数,比如 $k=B!$ 。

S2 在 $2 \sim n-2$ 间取一数 a ,可以是 2 或 3,也可以是随机的。

S3 计算 $a^k, (\text{mod } n)$ 。

S4 利用欧几里得算法计算 $d=(a^k, n)$ 。

S5 若 d 不是 n 的非平凡除数,则重新选择 a 和 k ,转 S1 重新开始。

算法的根据是:由于 k 被小于 B 和等于 B 的所有数除尽, p 是 n 的因子,若 $p-1$ 是小于 B 的素数的幂的乘积,则 k 是 $p-1$ 的倍数。根据 Fermat 定理 $a^k \equiv 1 (\text{mod } p)$,故

$$p \mid \gcd(a^k - 1, n)$$

失败的可能是由于 $a^k \equiv 1, (\text{mod } n)$ 。

例如 $n=540\,143, B=8, k=B!=840, a=2, 2^{840} (\text{mod } n) \equiv 53\,047,$

$$540\,143 = 10 \times 53\,046 + 9683$$

$$9683 = 540\,143 - 10 \times 53\,046$$

$$53\,046 = 5 \times 9683 + 4631$$

$$4631 = 53\,046 - 5 \times 9683$$

$$9683 = 2 \times 4631 + 421$$

$$421 = 9683 - 2 \times 4631$$

$4631 = 11 \times 421$, 所以 $\gcd(53\,046, 540\,143) = 421$, $540\,143 = 1283 \times 421$ 。

540 143 之所以被因数分解, 是由于 $421 - 1 = 420 = 2^2 \times 3 \times 7$ 。

即 540 143 有一因数 421, 而 421 的所有因数和幂都是小素数。与 $p-1$ 分解法类似的有 $p+1$ 分解法, 即 $p+1$ 可被分解为小素数乘积时, p 是数 n 的因数, n 可被因数分解,

即若 $n=r$, $p-1 = \prod_{i=1}^s p_i^{a_i}$ 或 $p+1 = \prod_{j=1}^s q_j^{B_j}$ 。

所有 $p_i \leq B, q_j \leq B$, 则 n 可被因子分解。

这里提供一信息, RSA 的 $n=pq$, 且要求对 $p-1$ 和 $p+1$ 的所有因数都不是小的数, q 也一样要求 $q-1, q+1$ 不存在小的因数。

* 3.2 连分数因数分解法

$$\begin{aligned} \text{定义 } a_0 + & \frac{1}{a_1 + \frac{1}{a_2 +}} \\ & \ddots \\ & + \frac{1}{a_m} \end{aligned}$$

用 $[a_0, a_1, a_2, \dots, a_m]$ 表示, 称为连分数, 其中 $a_0 \in \mathbb{Z}, a_i \in \mathbb{N}, 1 \leq i \leq m$ 。若 $m \rightarrow \infty$, 则表示为 $[a_0, a_1, \dots]$ 。

显然, 连分数是一有理数。

引理: 任一有理数都可以用有限的连分数表示。

证 证明的过程也就是求连分数的步骤, 即证明是构造性的。设 $\frac{a}{b}$ 是有理数, 令

$s_0 = a, s_1 = b, s_i = q_i s_{i+1} + s_{i+2}, 0 \leq s_{i+2} < s_{i+1}, i \geq 0$, 直到 $s_{n+2} = 0$ 为止, 这时 m 为奇整数,

$$s_m = q_m s_{m+1}$$

所以

$$\begin{aligned} \frac{a}{b} &= \frac{s_0}{s_1} = \frac{q_0 s_1 + s_2}{s_1} = q_0 + \frac{s_2}{s_1} = q_0 + \frac{s_2}{q_1 s_2 + s_3} \\ &= q_0 + \frac{1}{q_1 + \frac{s_3}{s_2}} = q_0 + \frac{1}{q_1 + \frac{1}{\frac{s_2}{s_3}}} = q_0 + \frac{1}{q_1 + \frac{1}{\frac{q_2 s_3 + s_4}{s_3}}} = q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{s_4}{s_3}}} = \dots \end{aligned}$$

$$\begin{array}{ccc}
 =q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{1}{\ddots + \frac{1}{q_{m-1} + \frac{s_{m+1}}{s_m}}}}} & = & q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{1}{\ddots + \frac{1}{q_{m-1} + \frac{1}{q_m}}}}}
 \end{array}$$

$\frac{a}{b}$ 可以用连分数 $[q_0, q_1, \dots, q_m]$ 来表示, $q_i > 0, i > 1$ 。

算法: 已知 $\frac{a}{b}$ 是有理数

S1 $s_0 \leftarrow a, s_1 \leftarrow b, i \leftarrow 0$ 。

S2 $s_i = q_i s_{i+1} + s_{i+2}, i \leftarrow i + 1$ 。

S3 若 $s_{i+2} = 0$, 则转 S4, 否则转 S2。

S4 输出 $[q_0, q_1, \dots, q_m]$, 其中 m 是 $s_{n+2} = 0$ 。

例 3-1 $[1, 2, 2, 2] = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}} = 1 + \frac{1}{2 + \frac{1}{\frac{5}{2}}} = 1 + \frac{1}{2 + \frac{2}{5}} = 1 + \frac{5}{12} = \frac{17}{12}$

例 3-2 $[-2, 2, 4, 2, 4] = -2 + \frac{1}{2 + \frac{1}{4 + \frac{1}{2 + \frac{1}{4}}}} = -2 + \frac{1}{2 + \frac{1}{4 + \frac{1}{\frac{9}{4}}}}$

$$\begin{aligned}
 &= -2 + \frac{1}{2 + \frac{1}{4 + \frac{4}{9}}} = -2 + \frac{1}{2 + \frac{9}{40}} \\
 &= -2 + \frac{1}{\frac{89}{40}} = -2 + \frac{40}{89} = -\frac{138}{89}
 \end{aligned}$$

例 3-3 $\frac{17}{12} = 1 + \frac{5}{12} = 1 + \frac{1}{\frac{12}{5}} = 1 + \frac{1}{2 + \frac{2}{5}} = 1 + \frac{1}{2 + \frac{1}{\frac{5}{2}}} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}} = [1, 2, 2, 2]$

例 3-4 $-\frac{138}{89} = -2 + \frac{40}{89} = -2 + \frac{1}{\frac{89}{40}} = -2 + \frac{1}{2 + \frac{9}{40}} = -2 + \frac{1}{2 + \frac{1}{\frac{40}{9}}}$

$$\begin{aligned}
 &= -2 + \frac{1}{2 + \frac{1}{4 + \frac{4}{9}}} = -2 + \frac{1}{2 + \frac{1}{4 + \frac{1}{\frac{9}{4}}}} = -2 + \frac{1}{2 + \frac{1}{4 + \frac{1}{2 + \frac{1}{4}}}} \\
 &= [-2, 2, 4, 2, 4]
 \end{aligned}$$

例 3-5

$$\begin{aligned}
\frac{294}{159} &= 1 + \frac{135}{159} = 1 + \frac{1}{\frac{159}{135}} = 1 + \frac{1}{1 + \frac{24}{135}} = 1 + \frac{1}{1 + \frac{1}{\frac{135}{24}}} \\
&= 1 + \frac{1}{1 + \frac{1}{5 + \frac{15}{24}}} = 1 + \frac{1}{1 + \frac{1}{5 + \frac{1}{\frac{24}{15}}}} = 1 + \frac{1}{1 + \frac{1}{5 + \frac{1}{1 + \frac{9}{15}}}}} \\
&= 1 + \frac{1}{1 + \frac{1}{5 + \frac{1}{1 + \frac{1}{\frac{15}{9}}}}} = 1 + \frac{1}{1 + \frac{1}{5 + \frac{1}{1 + \frac{1}{1 + \frac{6}{9}}}}} \\
&= 1 + \frac{1}{1 + \frac{1}{5 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}}}} = [1, 1, 5, 1, 1, 1, 2]
\end{aligned}$$

假定 n 是待因数分解整数, 以下运算都在 $\text{mod } n$ 下进行, $b_{-1} \leftarrow 0, b_0 \leftarrow a_0 = \lfloor \sqrt{n} \rfloor, x_0 \leftarrow \sqrt{n} - a_0, b_0^2, (\text{mod } n)$ 。

令 $i=1, 2, 3, \dots$, 进行以下运算, 并观察 $b_i^2 (\text{mod } n)$ 所得各数。

$$a_i \leftarrow \left\lfloor \frac{1}{x_{i-1}} \right\rfloor, \quad x_i \leftarrow \frac{1}{x_{i-1}} - a_i, \quad b_i \leftarrow a_i b_{i-1} + b_{i+1}, \quad (\text{mod } n), \quad b_i^2, \quad (\text{mod } n),$$

得一表将它们因数分解成小素数的积, 构成因数基 B , 对于每个 $b_i^2 (\text{mod } n)$, 含有属于 B 的因数表以对应向量 β 。

例 3-6 求 9073 的因数分解 (见表 3-1)。

$$i = 0$$

$$b_{-1} = 1, \quad b_0 = a_0 = \lfloor \sqrt{9073} \rfloor = 95, \quad x_0 = \sqrt{9073} - 95,$$

$$b_0^2 \equiv 9025, \quad (\text{mod } n) = -48, \quad (\text{mod } n)$$

$$i = 1$$

$$a_1 = \left\lfloor \frac{1}{\sqrt{9073} - 95} \right\rfloor = \left\lfloor \frac{\sqrt{9073} + 95}{48} \right\rfloor = 3,$$

$$x_1 = \frac{1}{\sqrt{9073} - 95} - 3 = \frac{\sqrt{9073} + 95}{48} - 3 = \frac{\sqrt{9073} - 49}{48},$$

$$b_1 = a_1 b_0 + b_{-1} = 3 \times 95 + 1 = 286$$

$$b_1^2 \equiv 81\,796 \equiv 139, \quad (\text{mod } n)$$

$i = 2$

$$a_2 = \left\lfloor \frac{48}{\sqrt{9073} - 49} \right\rfloor = \left\lfloor \frac{48(\sqrt{9073} + 49)}{6672} \right\rfloor = 1,$$

$$x_2 = \frac{48}{\sqrt{9073} - 49} - 1 = \frac{97 - \sqrt{9073}}{\sqrt{9073} - 49},$$

$$b_2 = a_2 b_1 + b_0 = 286 + 95 = 381,$$

$$b_2^2 \equiv 145\,161 \equiv -7, (\text{mod } n)$$

$i = 3$

$$a_3 = \left\lfloor \frac{\sqrt{9073} - 49}{97 - \sqrt{9073}} \right\rfloor = 26,$$

$$x_3 = \frac{\sqrt{9073} - 49}{97 - \sqrt{9073}} - 26,$$

$$b_3 = a_3 \cdot b_2 + b_1 = 26 \times 381 + 286 = 10\,192 \equiv 1119, (\text{mod } n),$$

$$b_3^2 (\text{mod } n) = 1\,252\,161, (\text{mod } n) \equiv 87$$

$i = 4$

$$a_4 = 2, \quad b_4 = 2619, b_4^2, (\text{mod } n) \equiv -27$$

表 3-1 例 3-6 表

p_i $b_i^2 (\text{mod } n)$	-1	2	3	7	87	139
$-48 \equiv -2^4 \times 3$	1	4	1			
139						1
$-7 \equiv -1 \times 7$	1			1		
87					1	
$-27 \equiv -3^3$	1		3			

从表 3-1 可见 $i=0,4$ 时两行分别为 $(1,4,1,0,0,0)$ 和 $(1,0,3,0,0,0)$ 。
这两行的和均为偶数。

$$(b_0^2)(b_4^2) \equiv (-48) \times (-27) = 4^2 \times 9^2, (\text{mod } 9073),$$

$$b_0 b_4 = 95 \times 2619 = 248\,805, \quad \equiv 3834, (\text{mod } 9073),$$

$$(3834)^2 \equiv (36)^2, (\text{mod } 9073),$$

$$\text{gcd}(3834 + 36, 9073) = \text{gcd}(3870, 9073) = 43,$$

$$9073 = 211 \times 43$$

例 3-7 分解 637 753(见表 3-2)。

$i = 0$

$$b_{-1} = 1, \quad b_0 = \lfloor \sqrt{637\,753} \rfloor = 798, \quad a_0 = 798,$$

$$b_0^2 = 636\,804 = 2^2 \times 3^2 \times 1789,$$

$$x_0 = \sqrt{637\,753} - 798$$

$$i = 1$$

$$a_1 = \left\lfloor \frac{1}{\sqrt{637\,753} - 798} \right\rfloor = 1,$$

$$x_1 = \frac{1}{\sqrt{637\,753} - 798} - 1 = \frac{799 - \sqrt{637\,753}}{\sqrt{637\,753} - 798},$$

$$b_1 = b_0 + b_{-1} = 798 + 1 = 799,$$

$$b_1^2 = 638\,401 = 648, (\text{mod } n)$$

$$648 = 2^3 \times 3^4$$

$$i = 2$$

$$a_2 = \left\lfloor \frac{\sqrt{637\,753} - 798}{799 - \sqrt{637\,753}} \right\rfloor = 1,$$

$$x_2 = \frac{\sqrt{637\,753} - 798}{799 - \sqrt{637\,753}} - 1 = \frac{2\sqrt{637\,753} - 1597}{799 - \sqrt{637\,753}},$$

$$b_2 = b_1 + b_0 = 799 + 798 = 1597,$$

$$b_2^2 = 2\,550\,409 = -603, (\text{mod } n),$$

$$-603 = (-1) \times (3) \times (201) = (-1) \times (3)^2 \times (67)$$

$$i = 3$$

$$a_3 = \left\lfloor \frac{799 - \sqrt{637\,753}}{2\sqrt{637\,753} - 1597} \right\rfloor = 2,$$

$$x_3 = \frac{799 - \sqrt{637\,753}}{2\sqrt{637\,753} - 1597} - 2 = \frac{3993 - 3\sqrt{637\,753}}{2\sqrt{637\,753} - 1597},$$

$$b_2 = 2 \times 1597 + 799 = 3993,$$

$$b_2^2 = 15\,944\,049 \equiv 224, (\text{mod } n), \quad 224 = 2^5 \times 7$$

以后的计算不一一给出,只叙述结果。

因数基 $B = \{-1, 2, 3, 7, 13, 59, 67, 73\}$

表 3-2 例 3-7 表

p_i	-1	2	3	7	13	59	67	73
$b_i^2 \pmod n$								
-13×73	1				1			1
$2^3 3^4$		1						
$-3^2 \times 67$	1						1	
$\rightarrow 2^5 7$		1		1				
-3×349								
3×149								
-907								
$\rightarrow 2^3 \times 3^2 \times 7$		1		1				

表 3-7 中左端 $\rightarrow$ 所指的两行对应元素之和构成偶数。

$$b_3 \equiv 3993, \pmod{n}, \quad b_7 \equiv 114\,199, \pmod{n}$$

$$3993 \times 114\,199 \equiv 3212, \pmod{n}$$

$$(3212)^2 \equiv 2^8 \times 3^2 \times 7^2 \pmod{n}, \quad \text{即 } (3212)^2 \equiv (2^4 \times 3 \times 7)^2, \pmod{n},$$

$$2^4 \times 3 \times 7 = 336,$$

$$\gcd(3212 + 336, 637\,753) = \gcd(3548, 637\,753) = 887,$$

$$\gcd(3212 - 336, 637\,753) = \gcd(2876, 637\,753) = 719,$$

$$887 \times 719 = 637\,753$$

3.3 Pollard ρ 法

Pollard ρ 法其基本思想为：选择出数 $f: Z_n \rightarrow Z_n, x_0 \in Z_n, x_i = f(x_{i-1}), i > 0$, 希望序列 $x_0, x_1, x_2, \dots$ 为 Z_n 的独立的随机数序列, 如果 p 是 n 的一个素数因子, 则 $\pmod{p}$ 将会发生碰撞, 即有两个整数 t 和 $l, l > 0$, 使 $x_t \equiv x_{t+l} \pmod{p}$ 。

Pollard ρ 算法：

S1 在 $\{0, 1, \dots, n-1\}$ 中随机取一数 $x_0, y_0 \leftarrow x_0, i \leftarrow 0$ 。

S2 $i \leftarrow i+1, x_i \leftarrow x_{i-1}^2 + 1 \pmod{n}, y_i \leftarrow (y_{i-1}^2 + 1)^2 + 1 \pmod{n}$ 。

S3 $g \leftarrow \gcd(x_i - y_i, n)$ 。

若 $1 < g < n$ 则作

始 输出 g , 结束终, 否则作

始 若 $g = n$ 则作

始 输出 $g = n$, 结束终, 否则

转 S2, 终。

例 3-8 $n = 82\,123, x_0 \leftarrow 631, y_0 \leftarrow 631,$

$$i = 1, x_1 = 631^2 + 1 = 398\,162 \equiv 69\,670, \pmod{82\,123},$$

$$y_1 = (69\,670)^2 + 1 = 4\,853\,908\,901 \equiv 28\,986, \pmod{n}$$

$$x_1 - y_1 = 40\,684,$$

$$82\,123 = 2 \times 40\,684 + 755,$$

$$d = (82\,123, 40\,684) = (40\,684, 755),$$

$$40\,684 = 53 \times 755 + 669,$$

$$d = (40\,684, 755) = (755, 669),$$

$$755 = 669 + 86, d = (669, 86),$$

$$669 = 7 \times 86 + 67, d = (86, 67) = (67, 19) = (19, 10) = (3, 1) = 1。$$

$$i = 2, x_2 = 28\,986, y_2 = 13\,166, (28\,986 - 13\,166, 82\,123) = 1。$$

$$i = 3, x_3 = 69\,907, y_3 = 40\,816, (x_3 - y_3, n) = 1。$$

$$i = 4, x_4 = y_2 = 13\,166, y_4 = 20\,459, (x_4 - y_4, n) = 1。$$

$$i = 5, x_5 = 64\,027, y_5 = 6685, (x_5 - y_5, n) = 1。$$

$$i = 6, x_6 = y_3 = 40\,816, y_6 = 75\,835, (x_6 - y_6, n) = 1。$$

$$i = 7, x_7 = 80\,802, y_7 = 17\,539, x_7 - y_7 = 63\,263。$$

$$\begin{aligned}
 d &= (82\,123, 63\,263) = (63\,263, 18\,860) = (18\,860, 6683) = (6683, 5494) \\
 &= (5494, 1189) = (1189, 738) = (738, 451) = (451, 287) = (287, 164) \\
 &= (164, 123) = (123, 41) = 41,
 \end{aligned}$$

$$n = 82\,123 = 41 \times 2003$$

类似地, $\text{mod } 41$ 计算如下, $631 \equiv 16, (\text{mod } 41)$,

$$i = 0, \quad x_0 \leftarrow 16$$

$$i = 1, \quad 16^2 + 1 = 257 \equiv 11, \quad (\text{mod } 41), \quad x_1 \leftarrow 11$$

$$i = 2, \quad 11^2 + 1 = 122 \equiv 40, \quad (\text{mod } 41), \quad x_2 \leftarrow 40$$

$$i = 3, \quad 40^2 + 1 = 1601 \equiv 2, \quad (\text{mod } 41), \quad x_3 \leftarrow 2$$

$$i = 4, \quad x_4 \leftarrow 5$$

$$i = 5, \quad x_5 \leftarrow 26$$

$$i = 6, \quad 26^2 + 1 = 677 \equiv 21, \quad (\text{mod } 41), \quad x_6 \leftarrow 21$$

$$i = 7, \quad 21^2 + 1 = 442 \equiv 32, \quad (\text{mod } 41), \quad x_7 \leftarrow 32$$

$$i = 8, \quad 32^2 + 1 = 1025 \equiv 0, \quad (\text{mod } 41), \quad x_8 \leftarrow 0$$

$$i = 9, \quad x_9 \leftarrow 1$$

$$i = 10, \quad x_{10} = 2$$

具体见图 3-1。

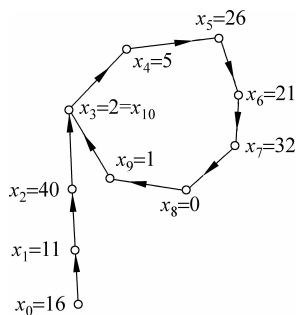


图 3-1 Pollard 迭代举例

3.4 Dixon 随机平方因数分解法

本节将举例说明该算法及其思想。

若

$$n = s^2 - t^2 = (s - t)(s + t)$$

$$n = ab = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2$$

这说明 n 的因数分解和将 n 表示成一对数的平方差相对应。令 $t = |\sqrt{n}|, |\sqrt{n}| + 1, \dots$, 测试 $t^2 - n$ 是否是一个完全的平方。假如找到 $t^2 - n = s^2$, 则 $n = t^2 - s^2$, 可利用其来对 n 进行因数分解。若 $n = ab$, 而 $a - b$ 的值比较小。

例如, $n=2\,027\,651\,281, \sqrt{n} \approx 45\,029$,

$$n = 45\,041^2 - 1020^2 = (45\,041 + 1020) \times (45\,041 - 1020) = 46\,061 \times 44\,021.$$

例 3-9 $n=2183$, 设能找到以下同余式,

$$453^2 \equiv 7, \pmod{2183}, \quad 1014^2 \equiv 3, \pmod{2183}, \quad 209^2 \equiv 21, \pmod{2183}$$

则

$$(453 \times 1014 \times 209)^2 \equiv (21)^2, \pmod{2183}$$

$$453 \times 1014 \times 209 = 96\,002\,478 \equiv 687, \pmod{2183}$$

所以

$$(687)^2 \equiv (21)^2, \pmod{2183}$$

$$(687)^2 - (21)^2 = (687 + 21) \times (687 - 21) \equiv 0, \pmod{2183}$$

$$687 + 21 = 708, \quad 687 - 21 = 666$$

令 $d_1=(708,2183), d_2=(666,2183)$, 但 $d_1=(708,59)=59$ 。

$$d_2 = (666,185) = (185,111) = (111,74) = (74,37) = 37,$$

$$2183 = 37 \times 59$$

关于大数的因数分解更进一步的讨论, 还需更上一层楼。

习 题

试用不同方法分解 $200\,819$ 。

第 4 章 线性反馈移位寄存器

4.1 流 码

若能以一种方式产生一随机序列,且由密钥来控制,则可由此序列进行加密,如图 4-1 所示。

这样的密码称为序列密码,也叫流码,其中 $\oplus$ 表示 mod 2 的加法。密钥流产生器是给定一种算法产生的密钥流,通常是 0, 1 数据流。人们希望它尽可能长,且有随机性,但严格地说它不可能做到完全随机,只能是伪随机。下面给出 Golomb 的随机公设,密钥 0,1 序列,例如 00110111,00 便称为 0 的 2 游程,11 是 1 的 2 游程,接着是 0 的 1 游程,最后是 1 的 3 游程。

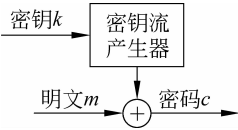


图 4-1 流码定义

假定 0,1 序列 $\{s_i\}$,存在 r 对所有的正整数 i 都满足 $s_i = s_{r+i}, i \geq 0$,使这等式成立的最小正整数 r 称为序列 $\{s_i\}$ 的周期。

若有两个子序列 $s_1, s_2, \dots, s_r; s_{1+\tau}, s_{2+\tau}, \dots, s_{r+\tau}$ 。

若 s_i 和 $s_{i+\tau}$ 相同的数目为 n_τ ,不相同的数目为 $d_\tau = r - n_\tau$,令 $R(\tau) = \frac{d_\tau - n_\tau}{r}, \tau = 0$,则 $n_\tau = r, d_\tau = 0, R(0) = 1$ 。 $\tau \neq 0, R(\tau)$ 为异相自相关函数。

Golomb 提出 0,1 序列的随机性公设:

- (1) 若 Y 是奇数,则 0,1 序列在一个周期内 0 的个数比 1 的个数多一个或少一个。
- (2) 在长度为 r 的周期内,1 的游程的数目为游程总数的 $\frac{1}{2}$,2 的游程的数目占游程总数的 $\frac{1}{2^2}, \dots, c$ 游程的数目占游程总数的 $\frac{1}{2^c}$,而且对于任意长度 0 的游程个数和 1 的游程个数相等。

(3) 异相自相关函数是个常数。

(1)和(2)的意义都较明显,(3)意味着通过对序列与其平移后的序列作比较,不能给出其他任何信息。

4.2 线性反馈移位寄存器

LFSR 是 Linear Feedback Shift Register 的缩写。

如图 4-2 所示标以 $a_1, a_2, \dots, a_n$ 的小方块是(0,1)二值的寄存器, $a_i(t+1)$ 表示 t 时刻 a_i 寄存器所含的值。

$$a_{i+1}(t) = a_i(t+1)$$

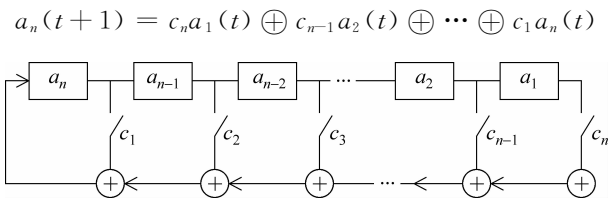


图 4-2 线性反馈移位寄存器(LFSR1)

$c_i=0$ 或 1 表示开关 c_i 分别是打开或闭合。

例如(见图 4-3 与表 4-1)：

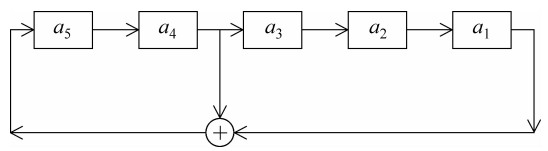


图 4-3 线性反馈移位寄存器(LFSR2)

表 4-1 图 4-3 对应表

i	a_5	a_4	a_3	a_2	a_1	i	a_5	a_4	a_3	a_2	a_1
0	1	0	1	0	1	17	1	1	0	0	1
1	1	1	0	1	0	18	0	1	1	0	0
2	1	1	1	0	1	19	1	0	1	1	0
3	0	1	1	1	0	20	0	1	0	1	1
4	1	0	1	1	1	21	0	0	1	0	1
5	1	1	0	1	1	22	1	0	0	1	0
6	0	1	1	0	1	23	0	1	0	0	1
7	0	0	1	1	0	24	0	0	1	0	0
8	0	0	0	1	1	25	0	0	0	1	0
9	1	0	0	0	1	26	0	0	0	0	1
10	1	1	0	0	0	27	1	0	0	0	0
11	1	1	1	0	0	28	0	1	0	0	0
12	1	1	1	1	0	29	1	0	1	0	0
13	1	1	1	1	1	30	0	1	0	1	0
14	0	1	1	1	1	31	1	0	1	0	1
15	0	0	1	1	1	32	1	1	0	1	0
16	1	0	0	1	1	33	1	1	1	0	1

续表

i	a_5	a_4	a_3	a_2	a_1	i	a_5	a_4	a_3	a_2	a_1
34	0	1	1	1	0	37	0	1	1	0	1
35	1	0	1	1	1	38	0	0	1	1	0
36	1	1	0	1	1	39	0	0	0	1	1

从表 4-1 中可见 $i=31$ 时,状态恢复到 $i=1$ 的情况,该 LFSR 存在周期是 31,线性反馈寄存器中总假定 $c_1c_2\cdots c_n$ 中至少有一个系数不为 0,否则无论初始状态是什么,在最多 n 步后必然是 $00\cdots 0$,而且一直维持此现状。

如果仅有一个系数,设 a_j 项系数不为零,则有

$$a_n(t+1) = a_j(t), \quad a_j(t+1) = a_{j+1}(t)。 \quad i = 1, 2, \cdots, n-1$$

实际上是一个延迟装置。

引理 1: $c_n=1$ 的 n 级 LFSR,对于 a_i 的初值为 0, $i=1, 2, \cdots, n$,其输出是周期序列。

若 a_i 不是全部为 0, $i=1, 2, \cdots, n$,情况如何?

定义 4-1: 若序列除开始若干项后的其余部分是周期序列,则称此序列为准周期序列。

如图 4-4 所示的 LFSR 输出序列是准周期序列,假定 r 是周期, m 是满足 $a_{m+t} = a_{m+r+t}$ 的最小正整数, t 是一切正整数,即 $a_m \neq a_{m+r}$;第 $m, m+1, m+r, m+r+1$ 时的状态分别是

$$\begin{aligned} &a_m a_{m+1} \cdots a_{m+n-1} \\ &a_{m+1} a_m \cdots a_{m+n} \\ &a_{m+r} a_{m+r-1} \cdots a_{m+r+n-1} \\ &a_{m+r+1} a_{m+r+2} \cdots a_{m+r+n} \end{aligned}$$

由假设

$$\begin{aligned} a_{m+n} &= c_n a_m + c_{n-1} a_{m-1} + \cdots + c_1 a_{m+n-1} \\ a_{m+n+r} &= c_n a_{m+r} + c_{n-1} a_{m+r-1} + \cdots + c_1 a_{m+r+n-1} \end{aligned}$$

但 $a_{m+n} = a_{m+r+n}$,从而 $c_n a_m = c_n a_{m+r}$, $c_n = 1, a_m = a_{m+r}$ 与假设矛盾,故 $m=0$,即 $\{a_i\}$ 是周期序列。

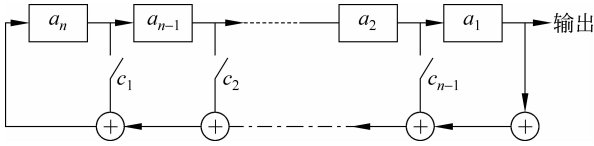


图 4-4 n 级 LFSR

引理 2: n 级 LFSR 对于 0 输入的输出是周期性的,其周期 $r \leq 2^n - 1$ 。

证 根据引理 1, n 级 LFSR 对于零输入的输出是周期性的,若初始状态为 $a_n = a_{n-1} = \cdots = a_1 = 0$,则输出是 0。否则其状态就不会出现 $00\cdots 0$,状态最多为 $2^n - 1$,后一状态和前一状态由当前状态决定,而且是唯一决定, n 级 LFSR 的输出序列周期达到最大的 $2^n - 1$ 。

4.3 Golomb 随机性概念

本节将讨论 LFSR 满足 Golomb 的随机性的三个公设。

G1: 由线性移位寄存器结构相同时输出的序列也相同, 所产生 m 序列的过程中必须遍历 $2^n - 1$ 个非零状态的每一个, 然后才出现重复。这 $2^n - 1$ 个状态中 a_1 位有 $2^n - 1$ 个是 1, 其余 $2^n - 1$ 个是 0, 这就满足第一个公设。

G2: 由于寄存器中不会出现全 0 状态, 所以不会出现 0 的 n 游程, 而且必然有一个 1 的 n 游程, 但也不可能有长度更大的 1 游程。因为若出现 1 的 $n+1$ 游程, 必然有两个 1 状态相邻, 这是不可能的, 1 的游程必出现在如下的串中:

$$0 \underbrace{11\cdots 10}_{n\text{位}}$$

当这 $n+2$ 位通过 LFSR 时, 依次出现

$$0 \underbrace{1\cdots 1}_{n-1\text{位}} \underbrace{1\cdots 1}_{n\text{位}} \underbrace{1\cdots 10}_{n-1\text{位}}$$

所以不会出现 1 的 $n-1$ 游程, 会出现 0 的 $n-1$ 游程:

$$1 \underbrace{0\cdots 01}_{n-1}$$

它产生 $1 \underbrace{0\cdots 0}_{n-1}$ 和 $\underbrace{0\cdots 01}_{n-1}$ 两个状态。

如果 $n=2$, 即 $n=2$ 级的 LFSR, 满足 Golomb 公设。

如果 $n>2$, 则 r 为不超过 $n-2$ 的任一正整数, 任何 1 的 r 游程意味着存在串 $0 \underbrace{1\cdots 10}_r$,

为了计算 1 的 r 游程的数目, 只要计算左边是这样的 $r+2$ 位的状态数目, 任何一个 1 的 r 游程总会在通过 LFSR 时处在这样的位置, 其余的 $n-r-1$ 位可以由 0, 1 构成的任何状态, 所以 1 的 r 游程的数目为 2^{n-r-1} , 于是在每一循环中出现 1 游程的数目为 $1 +$

$$\sum_{r=1}^{n-2} 2^{n-r+2} = 2^{n-2}; 0 \text{ 的游程数目也是 } 2^{n-2}。$$

小结:

- (1) 任一循环含 2^{n-1} 个 1, $2^{n-1} - 1$ 个 0。
- (2) 1 的 n 游程有一个, 没有 0 的 n 游程。
- (3) 没有 1 的 $n-1$ 游程, 有一个 0 的 $n-1$ 游程。
- (4) 若 $1 \leq r \leq n-2$, 则 1 和 0 的 r 游程各 2^{n-r-1} 个。
- (5) 每一循环有 2^{n-2} 个 1 的游程和 2^{n-2} 个 0 的游程。

G3: 自相关函数 $c(t)$ 对于某整数 K , 满足

$$nc(t) = \sum_{i=0}^{n-1} (2s_i - 1)(2s_{i+t} - 1) = \begin{cases} n, & t = 0 \\ K, & i \leq t \leq n-1 \end{cases}$$

这个自相关函数 $c(t)$ 是用来度量序列 s 和 s 移位 t 位的相似性。

若 s 是随机的周期 N 序列, 则 $|Nc(t)|$ 可能是非常小的 ($0 < t < n$)。

例 4-1 周期 $n=15, s^{15}=0,1,1,0,0,1,0,0,0,1,1,1,1,0,1$, 可证其满足 Golomb 随机公设。

- (1) s^{15} 中 0 的数目为 7, 1 的数目是 8。
- (2) s^{15} 有 8 轮, 有 4 轮只为 1, 两轮为长度 2, 一轮为长度 3, 一轮长为 4。
- (3) 自相关函数 $c(t)$ 有两个值: $c(0)=1, c(t)=\frac{1}{15}, 1\leq t\leq 4$ 。

定理 4-1 (换行定理): 周期为 2^n-1 的 m 序列, 其异相自相关函数等于 $\frac{-1}{2^n-1}$ 。

证 设 $\{a_t\}$ 是周期为 2^n-1 的 m 序列, 对正整数 $\tau, 0<\tau<2^n-1, \{a_t\}+\{a_{t+\tau}\}$ 在一个周期内为 0 的位的数目正好是序列 $\{a_t\}$ 和 $\{a_{t+\tau}\}$ 对应的位相同的位的数目, 其中 $\{a_{t+\tau}\}$ 是将 $\{a_t\}$ 序列平行 τ 位而成。

设序列 $\{a_t\}$ 满足

$$\begin{aligned} a_{h+n} &= c_1 a_{h+n-1} + c_2 a_{h+n-2} + \cdots + c_n a_h \\ a_{h+n+\tau} &= c_1 a_{h+n+\tau-1} + c_2 a_{h+n+\tau-2} + \cdots + c_n a_{h+\tau} \\ a_{h+n} + a_{h+n+\tau} &= c_1 (a_{h+n-1} + a_{h+n+\tau-1}) + c_2 (a_{h+n-2} + a_{h+n+\tau-2}) + \cdots + c_n (a_h + a_{h+\tau}) \end{aligned}$$

令 $b_j=a_j+a_{j+\tau}$ 故

$$b_{h+n} = c_1 b_{h+n-1} + c_2 b_{h+n-2} + \cdots + c_n b_h$$

$\{b_j\}$ 实际上也是 m 序列, 为了计算 $R(\tau)$, 只要将 $\{b_j\}$ 在某一循环中 0 的个数减去 1 的个数, 再除以 2^n-1 , 即

$$R(\tau) = \frac{2^{n-1} - 1 - 2^{n-1}}{2^n - 1} = \frac{-1}{2^n - 1}$$

4.4 非线性移位寄存器举例

先看一个 JK 触发器的例子。

如图 4-5 所示, LFSR1 是 m 级, $\{a\}$ 是它的输出, LFSR2 是 n 级, $\{b\}$ 是其输出, 它的工作用表 4-2 表示。

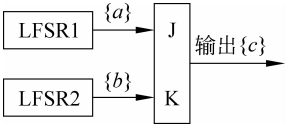


图 4-5 JK 触发器

表 4-2 图 4-5 对应表

J	K	c_n
0	0	c_{n-1}
0	1	0
1	0	1
1	1	c_{n-1}

这种体制随机性方面较好, 关系比较简单, 容易从 $\{c_i\}$ 对 $\{a_i\}, \{b_j\}$ 作出推断。这种体制的随机性比较好, 然而只要知道序列的一部分便可推导出这些方程的解。

(1) 若 $c_n=c_{n-1}=0$, 则 $a_{n+1}=0$ 。

- (2) 若 $c_n=0, c_{n-1}=1$, 则 $a_{n+1}=1$ 。
- (3) 若 $c_n=1, c_{n+1}=0$, 则 $b_{n+1}=1$ 。
- (4) 若 $c_n=c_{n+1}=1$, 则 $b_{n+1}=0$ 。

即从 c_n 和 c_{n+1} 便可对 a_{n+i} 和 b_{n+i} 中的 i 作出判定。故这个体制是非安全的。

例 4-2 LFSR1 $a_4(t+1)=a_2(t)+a_1(t)$ 初态(11111)

LFSR2 $a_3(t+1)=a_2(t)+a_1(t)$ 初态(1111)

由 LFSR1(见图 4-6)的输出和 LFSR2(见图 4-7)的输出的和构成输出列 $\{c\}$, 如表 4-3 所示。

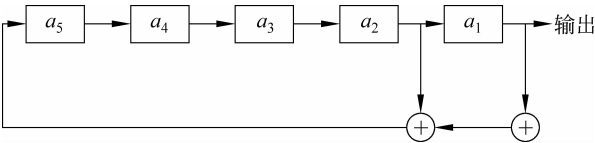


图 4-6 LFSR1

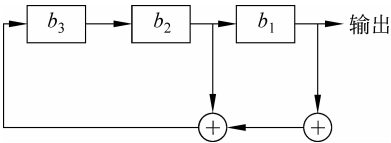


图 4-7 LFSR2

表 4-3 输出列 $\{c\}$

a_4	a_3	a_2	a_1		b_3	b_2	b_1		c		a_4	a_3	a_2	a_1		b_3	b_2	b_1		c
1	1	1	1		1	1	1		0		0	1	1	1		1	0	0		1
0	1	1	1		0	1	1		0		0	0	1	1		0	1	0		1
0	0	1	1		0	0	1		0		0	0	0	1		1	0	1		0
0	0	0	1		1	0	0		1		1	0	0	0		1	1	0		0
1	0	0	0		0	1	0		0		0	1	0	0		1	1	1		1
0	1	0	0		1	0	1		1		0	0	1	0		0	1	1		1
0	0	1	0		1	1	0		0		1	0	0	1		0	0	1		0
1	0	0	1		1	1	1		0		1	1	0	0		1	0	0		0
1	1	0	0		0	1	1		1		0	1	1	0		0	1	0		0
0	1	1	0		0	0	1		1		1	0	1	1		1	0	1		0
1	0	1	1		1	0	0		1		0	1	0	1		1	1	0		1
0	1	0	1		0	1	0		1		1	0	1	0		1	1	1		1
1	0	1	0		1	0	1		1		1	1	0	1		0	1	1		0
1	1	0	1		1	1	0		1		1	1	1	0		0	0	1		1
1	1	1	0		1	1	1		1		1	1	1	1		1	0	0		1
1	1	1	1		0	1	1		0		0	1	1	1		0	1	0		1
0	1	1	1		0	0	1		0		0	0	1	1		1	0	1		0
0	0	1	1		1	0	0		1		0	0	0	1		1	1	0		1
0	0	0	1		0	1	0		1		1	0	0	0		1	1	1		1

续表

a_4	a_3	a_2	a_1		b_3	b_2	b_1		c		a_4	a_3	a_2	a_1		b_3	b_2	b_1		c
1	0	0	0		1	0	1		1		0	1	0	0		0	1	1		1
0	1	0	0		1	1	0		0		0	0	1	0		0	0	1		1
0	0	1	0		1	1	1		1		1	0	0	1		1	0	0		1
1	0	0	1		0	1	1		0		1	1	0	0		0	1	0		0
1	1	0	0		0	0	1		1		1	0	1	1		1	1	0		1
0	1	1	0		1	0	0		0											
1	0	1	1		0	1	0		1		0	1	0	1		1	1	1		0
0	1	0	1		1	0	1		0		1	0	1	0		0	1	1		1
1	0	1	0		1	1	0		0		1	1	0	1		0	0	1		0
1	1	0	1		1	1	1		0		1	1	1	0		1	0	0		0
1	1	1	0		0	1	1		1		1	1	1	1		0	1	0		1
1	1	1	1		0	0	1		0		0	1	1	1		1	0	1		0

由表 4-3 可见序列 $\{c\}$ 的周期扩大了。

例 4-3 Press 体制。

如图 4-8 所示的 Press 体制可以用来生成伪随机序列,它由 4 组 JK 触发器组成,还要加上循环计数器,用来决定每一个时间脉冲作用下输出的单元。

这个体制的密钥是 8 个移位寄存器的初态以及输出单元的顺序。

如图 4-9 所示有 LFSR A 和 LFSR B。LFSR A 输出是二进制地址,复合序列输出的正是根据 LFSR A 所指定的地址从 LFSR B 中取出的内容,具体见图 4-10。

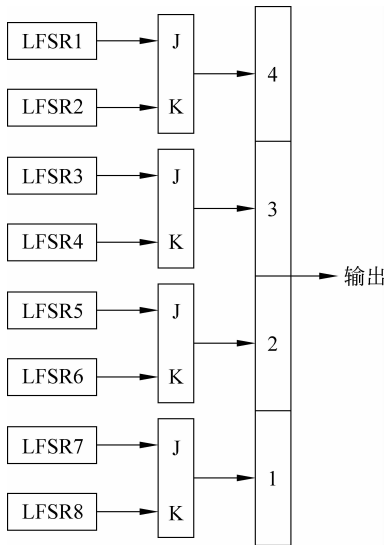


图 4-8 Press 1

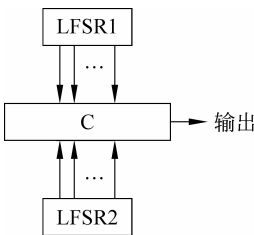


图 4-9 Press 2

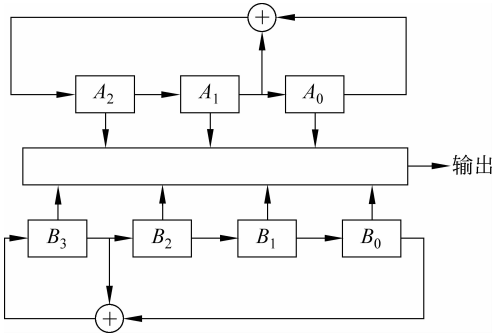


图 4-10 Press 3

复合序列的具体方案非常多。

例如： $A_i(i=1,2,3), B_j(j=1,2,3,4), \{A_i\}, \{B_j\}$ 可以加以重新编号。

设 $A_2A_1A_0$ 的初值为 011, $B_3B_2B_1B_0$ 的初值为 1000, 复合序列输出是 $B_N, N = (A_2A_1)$, 见表 4-4。

表 4-4 图 4-10 对应表

A_2	A_1	A_0	B_3	B_2	B_1	B_0	$N=(A_2A_1)$		B_N
1	1	1	1	0	0	0	1	1	1
0	1	1	1	1	0	0	0	1	0
0	0	1	1	1	1	0	0	0	0
1	0	0	1	1	1	1	1	0	1
0	1	0	0	1	1	1	0	1	1
1	0	1	1	0	1	1	1	0	0
1	1	0	0	1	0	1	1	1	0
1	1	1	1	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	0
0	0	1	0	1	1	0	0	0	0
1	0	0	0	0	1	1	1	0	0
0	1	0	1	0	0	1	0	1	0
1	0	1	0	1	0	0	1	0	1
1	1	0	0	0	1	0	1	1	0
1	1	1	0	0	0	1	1	1	0
0	1	1	1	0	0	0	0	1	0

复合序列是周期的,其周期 r 满足

$$r \leqslant (2^m - 1)(2^n - 1)$$

其中, m 是 LFSR A 的级数, n 是 LFSR B 的级数。

当 $(m,n)=1$ 时 $r=(2^m-1)(2^n-1)$, 一般有 $r=\text{lcm}\{2^m-1,2^n-1\}$ 。

4.5 LFSR 的密码反馈

密码反馈的基本特性是将密文反馈给系统, 以进行后面的加密, 下面先举一简单的例子。

$m = \text{cryptography is the science of data security}$

用维吉利亚密码加密, 密钥是 redstar, 过程如下。

m : cryptographyisthescienceofdatasecurity

k : redstarredstarredstarredstarredstarred

c : TVBHMOXIESZPIJKLHKUIVEGHGYDRKEVWDUIZXB

redstar 第一次结束时, 紧接着用密文 TVBHMOX 进行加密。

m : cryptographyisthescienceofdatasecurity

k : redstarTVBHMOXKVQOKWPDCUGMGQTQEYURJTJEQ

c : TVBHMOXKVQOKWPDCUGMGQTQEYURJTJEQYTDKRJO

m :	cryptog	r	a	p	h	y	i	s
k :	redstar	$(r+T)$	$(e+V)$	$(d+B)$	$(s+H)$	$(t+M)$	$(a+O)$	$(r+X)$
c :	TVBHMOX	B	Z	T	G	D	W	G
	t	h	e	s	c	i	e	n
	$(r+B)$	$(e+Z)$	$(d+T)$	$(s+G)$	$(t+D)$	$(a+W)$	$(r+G)$	$(r+L)$
	L	K	A	Q	Y	E	B	P
	e	o	f	d	a	t	a	s
	$(d+A)$	$(s+Q)$	$(t+Y)$	$(a+E)$	$(r+B)$	$(r+P)$	$(e+Q)$	$(d+H)$
	H	W	W	H	S	Z	U	C
	c	u	r	i	t	y		
	$(t+W)$	$(a+H)$	$(r+S)$	$(r+Z)$	$(e+O)$	$(d+C)$		
	R	B	A	Y	R	D		

图 4-11 是利用 LFSR 进行密码反馈的示意图。

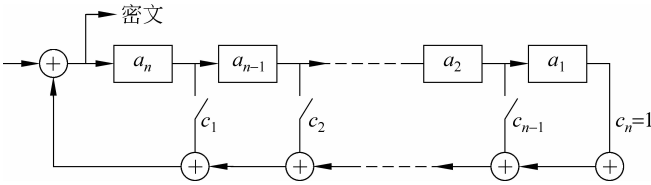


图 4-11 LFSR 1

密钥: $c_1, c_2, \dots, c_n, a_1, a_2, \dots, a_n$ 的初态 $s_1, s_2, \dots, s_n$

明文 $m=m_1m_2\cdots$

密文 $c=e_1e_2\cdots$

$$\begin{aligned} e_1 &= m_1 + \sum_{i=1}^n c_i s_{n-i+1} \\ e_2 &= m_2 + \sum_{i=2}^n c_i s_{n-i+2} + c_1 e_1 \\ e_3 &= m_3 + \sum_{i=3}^n c_i s_{n-i+3} + c_1 e_2 + c_2 e_1 \\ &\vdots \\ e_k &= m_k + \sum_{i=k}^n c_i s_{n-i+k} + \sum_{i=1}^{k-1} c_i e_{k-i} \\ e_{n+1} &= m_{n+1} + \sum_{i=1}^n c_{n-i+1} \\ &\vdots \\ e_{n+h} &= m_{n+h} + \sum_{i=1}^n c_{n-i+h} \end{aligned}$$

又如图 4-12 所示,移位寄存器的初态为 $S_1, S_2, \cdots, S_n$,系数是 $c_1, c_2, \cdots, c_n$ 时都与加密的 LFSR 相同,即解密密钥与加密密钥相同。这样当输入密文 $c_1c_2\cdots c_n$ 时便输出明文 $m=m_1m_2\cdots m_n$,实现了解密。

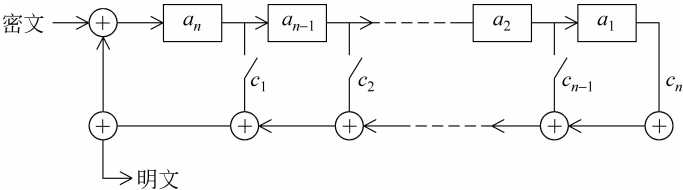


图 4-12 LFSR 2

习 题

如果 $t=0$ 时, $a_1a_2a_3a_4a_5$ 初态为 10101,求上述移位寄存器各个时间的状态。

