应用密码学作业 #3

XXX：202XX80XXXXXXXX

2023 年 4 月 14 日

线性移位寄存器

特征多项式 $f(x)=1+x+x^3+x^4$ 对应的线性移位寄存器图如下：

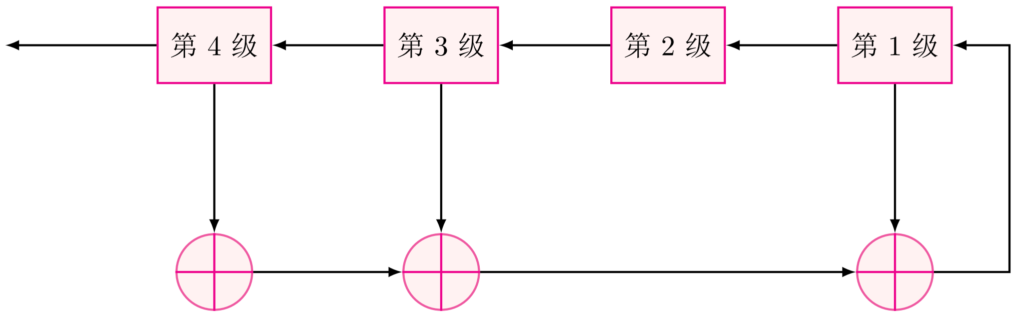

图 1: 线性移位寄存器

若其初始状态为 1101，则输出为 110 110 110……, 其周期为 3. 而 $f(x)=1+x+x^3(1+x)=(1+x)(1+x^3)=(1+x)(1+x)(1+x+x^2)$。故序列的最小生成多项式可能为 $f(x)=1+x$，$f(x)=(1+x)^2$ 或 $f(x)=(1+x+x^2)$。其中前两者都只有一项参加反馈，故不可能。而第三者有两级参加反馈，即取序列当前项的前两项异或可得到当前项。验证可生成该序列。其最短线性移位寄存器为

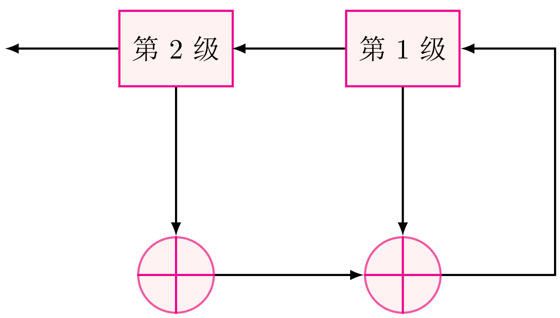

图 2: 最短线性移位寄存器

破译线性移位寄存器密码系统

由明文密文可以得到密钥流为 1110100111，设该 3 级线性移位寄存器的特征多项式为 $f(x)=x^3+c_1x^2+c_2x+c_3$，根据密钥流序列，可得到下

列方程

$$\begin{cases} c_3 + c_2 + c_1 = 0 \\ c_3 + c_2 = 1 \\ c_3 + c_1 = 0 \end{cases}$$

得

$$\begin{cases} c_1 = 1 \\ c_2 = 0 \\ c_3 = 1 \end{cases}$$

故该密码系统使用的 3 级线性移位寄存器特征多项式是 $f(x) = x^3 + x^2 + 1$。

BM 算法

计算 $d_1 = 1$，$m = 0$。$f_2(x) = f_1(x) + x^{1-0} f_0(x)$

$f_2(x) = 1 + x + x^1(1)$

$f_2(x) = 1$

$l_2 = max(l_1, 2 - l_1) = 1$

计算 $d_2 = 0$，$m = 0$。$f_3(x) = f_2(x) = 1$。$l_3 = l_2 = 1$

计算 $d_3 = 1$，$m = 0$。$f_4(x) = f_3(x) + x^{3-0} f_0(x)$

$f_4(x) = 1 + x^3(1)$

$f_4(x) = 1 + x^3$

$l_4 = max(l_3, 4 - l_3) = 3$

计算 $d_4 = 1$，$m = 3$。$f_5(x) = f_4(x) + x^{4-3} f_3(x)$

$f_5(x) = 1 + x^3 + x^1(1)$

$f_5(x) = 1 + x + x^3$

$l_5 = max(l_4, 5 - l_4) = 3$

计算 $d_5 = 1$，$m = 3$。$f_6(x) = f_5(x) + x^{5-3} f_3(x)$

$f_6(x) = 1 + x + x^3 + x^2(1)$

$f_6(x) = 1 + x + x^2 + x^3$
$l_6 = max(l_5, 6 - l_5) = 3$

计算 $d_6 = 1$，$m = 3$。$f_7(x) = f_6(x) + x^{6-3}f_3(x)$
$f_7(x) = 1 + x + x^2 + x^3 + x^3(1)$
$f_7(x) = 1 + x + x^2$
$l_7 = max(l_6, 7 - l_6) = 4$

计算 $d_7 = 0$，$m = 6$。$f_8(x) = f_7(x) = 1 + x + x^2$。$l_8 = l_7 = 4$

计算 $d_8 = 0$，$m = 6$。$f_9(x) = f_8(x) = 1 + x + x^2$。$l_9 = l_8 = 4$

计算 $d_9 = 1$，$m = 6$。$f_{10}(x) = f_9(x) + x^{9-6}f_6(x)$
$f_{10}(x) = 1 + x + x^2 + x^3(1 + x + x^2 + x^3)$
$f_{10}(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6$
$l_{10} = max(l_9, 10 - l_9) = 6$

计算 $d_{10} = 1$，$m = 9$。$f_{11}(x) = f_{10}(x) + x^{10-9}f_9(x)$
$f_{11}(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^1(1 + x + x^2)$
$f_{11}(x) = 1 + x^4 + x^5 + x^6$
$l_{11} = max(l_{10}, 11 - l_{10}) = 6$

计算 $d_{11} = 1$，$m = 9$。$f_{12}(x) = f_{11}(x) + x^{11-9}f_9(x)$
$f_{12}(x) = 1 + x^4 + x^5 + x^6 + x^2(1 + x + x^2)$
$f_{12}(x) = 1 + x^2 + x^3 + x^5 + x^6$
$l_{12} = max(l_{11}, 12 - l_{11}) = 6$

计算 $d_{12} = 1$，$m = 9$。$f_{13}(x) = f_{12}(x) + x^{12-9}f_9(x)$
$f_{13}(x) = 1 + x^2 + x^3 + x^5 + x^6 + x^3(1 + x + x^2)$
$f_{13}(x) = 1 + x^2 + x^4 + x^6$
$l_{13} = max(l_{12}, 13 - l_{12}) = 7$

计算 $d_{13} = 1$，$m = 12$。$f_{14}(x) = f_{13}(x) + x^{13-12}f_{12}(x)$

$f_{14}(x) = 1 + x^2 + x^4 + x^6 + x^1(1 + x^2 + x^3 + x^5 + x^6)$
$f_{14}(x) = 1 + x + x^2 + x^3 + x^7$
$l_{14} = max(l_{13}, 14 - l_{13}) = 7$

计算 $d_{14} = 0$，$m = 12$。$f_{15}(x) = f_{14}(x) = 1 + x + x^2 + x^3 + x^7$。$l_{15} = l_{14} = 7$

计算 $d_{15} = 1$，$m = 12$。$f_{16}(x) = f_{15}(x) + x^{15-12} f_{12}(x)$
$f_{16}(x) = 1 + x + x^2 + x^3 + x^7 + x^3(1 + x^2 + x^3 + x^5 + x^6)$
$f_{16}(x) = 1 + x + x^2 + x^5 + x^6 + x^7 + x^8 + x^9$
$l_{16} = max(l_{15}, 16 - l_{15}) = 9$

计算 $d_{16} = 1$，$m = 15$。$f_{17}(x) = f_{16}(x) + x^{16-15} f_{15}(x)$
$f_{17}(x) = 1 + x + x^2 + x^5 + x^6 + x^7 + x^8 + x^9 + x^1(1 + x + x^2 + x^3 + x^7)$
$f_{17}(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^7 + x^9$
$l_{17} = max(l_{16}, 17 - l_{16}) = 9$

计算 $d_{17} = 0$，$m = 15$。$f_{18}(x) = f_{17}(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^7 + x^9$。
$l_{18} = l_{17} = 9$

计算 $d_{18} = 0$，$m = 15$。$f_{19}(x) = f_{18}(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^7 + x^9$。
$l_{19} = l_{18} = 9$

计算 $d_{19} = 1$，$m = 15$。$f_{20}(x) = f_{19}(x) + x^{19-15} f_{15}(x)$
$f_{20}(x) = 1 + x^3 + x^4 + x^5 + x^6 + x^7 + x^9 + x^4(1 + x + x^2 + x^3 + x^7)$
$f_{20}(x) = 1 + x^3 + x^9 + x^{11}$
$l_{20} = max(l_{19}, 20 - l_{19}) = 11$

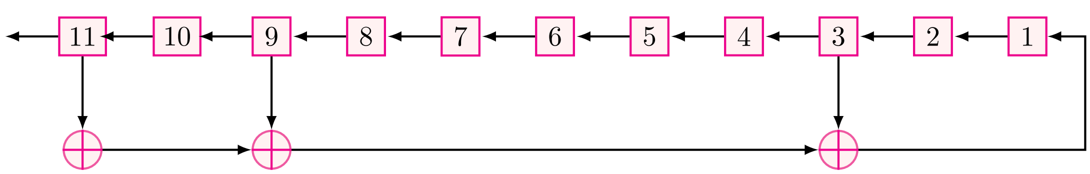

图 3: 线性移位寄存器

生成 BM 算法计算过程的 latex 代码的如下：

```python
seq="10011011000111010100"

# if the i-th bit of the number f[i] in a binary form is 1,
# then term x^i is in f(x)
# example:  if n-th function fn(x)=x^3 + x^2 + 1,
#        then fn(x) will be encoded into the n-th element of array f,
#        that is f[n] = int("0b1101",2)
f = [0 for i in range(21)]
l = [0 for i in range(21)]
d = [0 for i in range(21)]
f[0]=1
f[1]=0b11 #f1(x)=x+1
d[0]=1
l[0]=0
l[1]=1
min = 0

# @params f: the encoded decimal form of a function
# @params n: the degree of the function f
# convert f to a string
# example:  if f = 3, then f=0b11, the return string will be "1 + x"
#         if f = 11, then f=0b1011, the return string will be "1 + x^2 + x^3"
def ftostr(f,n):
  fstr = "1 "
  for i in range (1, len(bin(f))-2):
    mask = 1 << i
    if bin(f & mask) != '0b0':
      if i > 1 : fstr+="+ x^{%d} " % i
      else: fstr+="+ x ".format(i)
  fstr+=""
  return fstr

#to generate latex code of bm algorithm
for i in range(1,20):
  flen = len(bin(f[i]))-2
  d[i] = f[i] & int("0b" + seq[i-flen+1:i+1],2)
  print("compute $d_{%d}={%d}$," % (i, bin(d[i]).count('1') % 2))
  print("$m={0}$.".format(min))
  if bin(d[i]).count('1') % 2 == 1:
    f[i+1] = f[i] ^ (f[min] << (i - min))
    print("$f_{%d}(x) = f_{%d}(x) + x^{%d-%d}f_{%d}(x)$\\\\" % (i+1, i, i, min, min))
    print("$f_{%d}(x) =%s + x^{%d}(%s)$\\\\" % (i+1, ftostr(f[i],i), i-min, ftostr(f[min],min)))
    print("$f_{%d}(x) =%s$\\\\" % (i+1, ftostr(f[i+1],i+1)))
    l[i+1] = max(l[i], i+1-l[i])
    print("$l_{%d}=max(l_{%d}, %d-l_{%d})=%d$" % (i+1, i, i+1, i, l[i+1]) )
    if i+1 - l[i] > l[i] :
      min = i
  else:
    f[i+1] = f[i]
    print("$f_{%d}(x) = f_{%d}(x) =%s." % (i+1, i, ftostr(f[i+1],i+1)))
    l[i+1] = l[i]
    print("$l_{%d}=l_{%d} =%d$" % (i+1, i ,l[i]))
  print("\n~\\\\")
```

bm.py