文章编号:1007-2985(2011)06-0027-06

周期为 $2p^n$ 的 q 元序列 m 紧错线性复杂度*

周建钦[1,2],上官成[2]

(1. 杭州电子科技大学通信工程学院,浙江 杭州 310018;2. 安徽工业大学计算机学院,安徽 马鞍山 243002)

摘 要:结合 k 错线性复杂度曲线和最小错误的理论,提出 m 紧错线性复杂度的概念来研究序列线性复杂度的稳定性.首先优化魏 -肖 -陈算法的结构,即 GF(q) 上求周期为 $2p^n$ 的 q 元序列线性复杂度的快速算法;然后通过采用联合代价的方法,给出一个 GF(q) 上求周期为 $2p^n$ 的 q 元序列 k 错线性复杂度的快速算法;接着给出周期为 $2p^n$ 的 q 元序列的 m 紧错线性复杂度快速算法,其中 p 和 q 是奇素数,q 为模 p^2 的一个本原根.

关键词:流密码;序列;线性复杂度;k 错线性复杂度;m 紧错线性复杂度

中图分类号:TN918.1 **文献标志码**:A

1 问题的提出

密钥序列的线性复杂度是流密码强度的一个重要度量指标.序列的高线性复杂度不一定能保证序列的安全性,虽然有的序列线性复杂度很大,但是改变其中 1 位或几位,其线性复杂度会发生较大变化而变成一个密码学意义上的弱序列.例如 2 元序列 $S=\overbrace{0,0,\cdots,0,1}^{n}$,其线性复杂度为 n,但只要最后 1 位 1 改为 0,则该序列的线性复杂度降为 0.因此,序列的线性复杂度的稳定性与序列的不可预测性密切相关,密钥序列的线性复杂度不仅要足够大,而且必须有很好的稳定性.

丁 -肖 -单[1] 最早注意到这个问题,率先创立了流密码的稳定性理论,并提出了重量复杂度、球体复杂度等流密码稳定性度量指标.随后,Stamp M[2] 也引入了类似"球体复杂度"的线性复杂度稳定性度量指标——k 错线性复杂度,同时给出了 k 错线性复杂度曲线的概念.改变序列的一个周期中至多 k 位后得到的最小线性复杂度,称为 k 错线性复杂度.

任何一个序列的 k 错线性复杂度也可以反复运用 B-M 算法计算出来,但为了计算周期为 N 的 2 元序列的 k 错线性复杂度,要将这个算法运用 $\sum_{j=0}^{k}\binom{N}{j}$ 次.对于周期为 N 的 2 元序列,虽然已有了一些特殊周期序列线性复杂度的快速算法,但如果没有一个有效的算法来计算 k 错线性复杂度,也得将这些快速算法运用 $\sum_{j=0}^{k}\binom{N}{j}$ 次.即使 N 和 k 不是太大,计算量也是相当大的.在 Games-Chan 算法[3] 的基础上,Stamp M 等[2] 提出一个确定周期为 2^n 的 2 元序列的 k 错线性复杂度的快速算法,魏仕民等[4] 提出了确定周期为 $2p^n$ 的 q 元序列的线性复杂度的一个快速算法,戴小平等[5] 给出了确定周期为 $2p^n$ 的 2 元序列的 k 错线性复杂度的一个快速算法.

利用文献[4] 中计算 GF(q) 上周期为 $2p^n$ 的序列线性复杂度的魏 -肖 -陈算法模式,笔者设计出文中的计算 GF(q) 上周期为 $2p^n$ 的序列 k 错线性复杂度的新算法.新算法采用了联合代价,这个联合代价不同

* 收稿日期:2011-08-02
基金项目:浙江省自然科学基金资助项目(Y1100318,R1090138);国家自然科学基金委员会与中国工程物理研究院联合基金资助(10776077);上海市信息安全综合管理技术研究重点实验室开放课题(AGK2009007)
作者简介:周建钦(1963-),男,山东巨野人,教授,硕士,主要从事通信、密码学与理论计算机科学研究.

于 Stamp-Martin 算法中计算 GF(2) 上周期为 2^n 的序列 k 错线性复杂度的代价模式. 在 Stamp-Martin 算法中,cost[i] 是用来衡量在不影响前面结果的情况下改变当前某个位置上 a_i 的代价. 由于魏 -肖 -陈算法的特殊逻辑结构,用文献[2]中的 cost 模式很难得到一个计算 k 错线性复杂度的快速算法. 由于周期为 $2p^n$ 的序列的线性复杂度在元素 a_i 和 a_{i+p^n} 间的代价和或差是有联系的,因此笔者提出应用联合代价 cost[i,$i+p^n$] 作为同时改变 a_i 和 a_{i+p^n} 的代价,该算法无论是逻辑结构还是描述都是相当简单的.

Kurosawa K 等[6] 引入了最小错误 minerror(S)(S 为 2 元序列)的概念来描述周期为 2^n 的 2 元序列的线性复杂度的稳定性,将其定义为使得序列线性复杂度下降所至少要改变的元素个数,即 k 错线性复杂度曲线第 1 个跃变点对应的 k 值,并给出了 k=minerror(S) 时 k 错线性复杂度的上界. Lauder A 等[7] 给出了确定周期为 2^n 的 2 元序列 k 错线性复杂度曲线算法.

笔者结合序列的 k 错线性复杂度曲线和最小错误的理论,提出 m 紧错线性复杂度来研究流密码的稳定性. m 紧错线性复杂度是一个 2 元组(k_m,LC_m),序列 S 的 k 错线性复杂度曲线的第 m 个跃变点对应的 k_m 值和对应的 k_m 错线性复杂度 LC_m 称为 m 紧错线性复杂度. 自然,0 紧错线性复杂度即为(0,LC_0),其中 LC_0 为线性复杂度. 对于 1 紧错线性复杂度(k_1,LC_1),k_1 为使得序列线性复杂度下降所至少要改变的元素个数,即 Kurosawa K 等[6] 定义的最小错误 minerror(S),LC_1 为 k_1 错线性复杂度.

2 魏 -肖 -陈算法

首先优化魏 -肖 -陈算法的结构. 设 $S=s_0,s_1,\cdots$ 为周期 $N=2p^n$ 的 q 元序列,p 和 q 是奇素数,q 是一个模 p^2 的本原根,$S^N=s_0,s_1,\cdots,s_{N-1}$ 是 S 的第 1 个周期,$l=p^{n-1}$,$A_i=a_{(i-1)l}a_{(i-1)l+1}\cdots a_{il-1}(i=1,2,\cdots,2p)$.

引理 1 如果 $A_{p+1}+A_1=A_{p+2}+A_2=\cdots=A_{2p}+A_p$,$A_{p+1}-A_1=(-1)^{i+1}(A_{p+i}-A_1)$,$i=1,2,\cdots,p$,那么 $A_1=A_3=\cdots=A_p=A_{p+2}=\cdots=A_{2p-1}$,$A_{p+1}=A_{p+3}=\cdots=A_{2p}=A_2=\cdots=A_{p-1}$,$(A_1,A_2)=(A_1,A_{p+1})$.

证明 因为 $A_{p+1}+A_1=A_{p+i}+A_i$,$A_{p+1}-A_1=-A_{p+i}+A_i$,$i=2,4,\cdots,p-1$,所以 $A_1=A_{p+i}$,$A_{p+1}=A_i$,$i=2,4,\cdots,p-1$. 因为 $A_{p+1}+A_1=A_{p+i}+A_i$,$A_{p+1}-A_1=A_{p+i}-A_i$,$i=1,3,\cdots,p$,所以 $A_1=A_i$,$A_{p+1}=A_{p+i}$,$i=1,3,\cdots,p-1$. 因此 $A_1=A_3=\cdots=A_p=A_{p+2}=\cdots=A_{2p-1}$,$A_{p+1}=A_{p+3}=\cdots=A_{2p}=A_2=\cdots=A_{p-1}$,$(A_1,A_2)=(A_1,A_{p+1})$.

引理 2 如果 $A_{p+1}+A_1=A_{p+2}+A_2=\cdots=A_{2p}+A_p$,那么 $A_i-A_{i+1}=-(A_{p+i}-A_{p+i+1})$,$i=2,4,\cdots,p-1$,$(\sum_{i=1}^{p}(-1)^{i+1}A_i,\sum_{i=1}^{p}(-1)^{i+1}A_{i+1})=(\sum_{i=1}^{p}(-1)^{i+1}A_i,\sum_{i=1}^{p}(-1)^{i+1}A_{p+i})$.

引理 3 如果 $A_{p+1}-A_1=(-1)^{i+1}(A_{p+i}-A_i)$,$i=1,2,\cdots,p$,那么 $A_i+A_{i+1}=A_{p+i}+A_{p+i+1}$,$i=2,4,\cdots,p-1$,$(\sum_{i=1}^{p}A_i,\sum_{i=1}^{p}A_{i+1})=(\sum_{i=1}^{p}A_i,\sum_{i=1}^{p}A_{p+i})$.

有了上面的引理,魏 -肖 -陈算法可以转化为下面的算法 1.

算法 1 求 GF(q) 上周期为 $2p^n$ 序列线性复杂度的算法.

```
l=p^n, a=S^N, c=0;
while l>1 do
    l=l/p;
    A_i=a_{(i-1)l}a_{(i-1)l+1}…a_{il-1};(i=1,2,…,2p)
    if A_{p+1}+A_1=A_{p+2}+A_2=…=A_{2p}+A_p then
      if A_{p+1}-A_1=(-1)^{i+1}(A_{p+i}-A_i)(i=1,2,…,
      2p) then
        a=(A_1,A_{p+1});
      else
        c=c+(p-1)l;
        a=(Σ_{i=1}^{p}(-1)^{i+1}A_i,Σ_{i=1}^{p}(-1)^{i+1}A_{p+i});
      end if
    else
      if A_{p+1}-A_1=(-1)^{i+1}(A_{p+i}-A_i)(i=1,2,
      …,2p) then
        c=c+(p-1)l;
        a=(Σ_{i=1}^{p}A_i,Σ_{i=1}^{p}A_{p+i});
      else
        c=c+2(p-1)l;
        a=(Σ_{i=1}^{p}A_{2i-1},Σ_{i=1}^{p}A_{2i});
      end if
    end if
end while
if a≠(0,0) then
```

```
if a0 = a1 then
  c = c + 1;
else
  if a0 + a1 = 0 then
    c = c + 1;
  else
    c = c + 2;
  end if
end if
end if
```

算法 1 对魏-肖-陈算法的改变，并不影响算法的结构，与原算法的时间复杂度相同. 由魏-肖-陈算法难以得到计算 k 错线性复杂度的快速算法，而这样修改后有利于将该算法推广为计算 k 错线性复杂度的算法，并由此得出计算 m 紧错线性复杂度的快速算法.

3　确定 k 错线性复杂度的快速算法

由于有条件 $A_{p+1}+A_1=A_{p+2}+A_2=\cdots=A_{2p}+A_p$ 和 $A_{p+1}-A_1=(-1)^{i+1}(A_{p+i}-A_i)$，$i=1,2,\cdots,p$，因此 a_i 和 a_{l+i} 是相互关联的，故采用联合代价的方法来表示同时改变 a_i 和 a_{l+i} 所需要改变的原始序列位数.

在 Stamp-Martin 算法中，cost 只计算了改变当前 a_i 的代价，事实上，即使在保持当前位置元素不变时 cost 也可能不为 0. 在算法中，用 $\mathrm{cost}[i,i+l,h_0,h_1]^{(2l)}$ 来表示同时改变 a_i 为 h_0，a_{i+l} 为 h_1 时所需要至少改变的原始序列 S 的位数，这里 $h_0=0,1,\cdots,q-1$，$h_1=0,1,\cdots,q-1$，$2l$ 为当前元素的个数. $l=p^n$ 时，$\mathrm{cost}[i,i+l,a_i,a_{i+l}]^{(2l)}$ 的初始值为 0，$\mathrm{cost}[i,i+l,a_i,a_{i+l}+\beta]^{(2l)}$ 和 $\mathrm{cost}[i,i+l,a_i+\alpha,a_{i+l}]^{(2l)}$ 的初始值为 1，$\mathrm{cost}[i,i+l,a_i+\alpha,a_{i+l}+\beta]^{(2l)}$ 的初始值为 2，这里 $i=0,1,\cdots,l-1$，$\alpha=1,2,\cdots,q-1$，$\beta=1,2,\cdots,q-1$.

算法 2　周期为 $2p^n$ 的 q 元序列 k 错线性复杂度的快速算法.

```
l = p^n, a = S^n, c = 0;
cost[i,i+l,a_i,a_{i+1}]^(2l) = 0,
cost[i,i+l,a_i,a_{i+1}+α]^(2l) = 0,
cost[i,i+l,a_i+α,a_{i+1}]^(2l) = 1,
cost[i,i+l,a_i+α,a_{i+1}+β]^(2l) = 2,
for i = 0,1,…,l-1,α = 0,1,…,q-1,β = 0,1,…,q-1;
while l > 1 do
    l = l/p;
    A_i = a_{(i-1)l} a_{(i-1)l+1} … a_{il-1}; (i = 1,2,…,2p)
    B_i = A_i + A_{p+i}; (i = 1,2,…,p)
    bcost[i,h]^(pl) = min{cost[i,i+l,d1,d2]^(2pl) | d1 + d2 = h},
    for h = 0,1,…,q-1, i = 0,1,…,pl-1;
    bcost[i,h]^(l) = Σ_{j=0}^{p-1} bcost[i,h]^(pl),
    for h = 0,1…,q-1, i = 0,1,…,l-1;
    T_B = Σ_{i=00≤h<q}^{l-1} min{bcost[i,h]^(l)};
    if T_B ≤ k then
        cost[i,i+l,h0,h1]^(2l) = Σ_{j=0}^{(p-1)/2} cost[i+2jl,i+2jl+pl,h0,h1]^(2pl) + Σ_{j=0}^{(p-1)/2-1} cost[i+(2j+1)l,i+(2j+1)l+pl,h1,
                                 h0]^(2pl)
        for h0 = 0,1,…,q-1, h1 = 0,1,…,q-1, i = 0,1,…,l-1;
        T_C = Σ_{i=00≤h0<q, 0≤h1<q}^{l-1} min {cost[i,i+l,h0,h1]^(2l)};
        if T_C ≤ k then
            a = (A1, A_{p+1});
        else
            c = c + (p-1)l;
```

$a=(\sum_{i=1}^{p}(-1)^{i+1}A_i,\sum_{i=1}^{p}(-1)^{i+1}A_{p+i})$;

$cost[i,i+l,h_0,h_1]^{(2l)}=\min\{\sum_{j=0}^{p-1}cost[i+jl,i+jl+pl,h_{i+jl}^0,h_{i+jl}^1]^{(2pl)}\mid h_{i+jl}^0+h_{i+jl}^1=h_i^0+h_i^1,\sum_{j=0}^{p-1}(-1)^j h_{i+jl}^0=h_0,\sum_{j=0}^{p-1}(-1)^j h_{i+jl}^1=h_1,j=0,1,\cdots,p-1\}$

for $h_0=0,1,\cdots,q-1,h_1=0,1,\cdots,q-1,i=0,1,\cdots,l-1$;

end if

else

$D_i=(-1)^{i+1}(A_{p+i}-A_i);(i=1,2,\cdots,p)$

$dcost[i+jl,h]^{(pl)}=\min\{cost[i+jl,i+jl+pl,h_{i+jl}^0,h_{i+jl}^1]^{(2pl)}\mid(1-)^j(h_{i+jl}^1-h_{i+jl}^0)=h\}$,

for $h=0,1,\cdots,q-1,i=0,1,\cdots,l-1,j=0,1,\cdots,p-1$;

$dcost[i,h]^{(l)}=\sum_{j=0}^{p-1}dcost[i+jl,h]^{(2pl)}$,

for $h=0,1,\cdots,q-1,i=0,1,\cdots,l-1$;

$T_D=\sum_{i=00\leqslant h<q}^{l-1}\min\{dcost[i,h]^{(l)}\}$;

if $T_D\leqslant k$ then

$c=c+(p-1)l$;

$a=(\sum_{i=1}^{p}A_i,\sum_{i=1}^{p}A_{p+i})$;

$cost[i,i+l,h_0,h_1]^{(2l)}=\min\{\sum_{j=0}^{p-1}cost[i+jl,i+jl+pl,h_{i+jl}^0,h_{i+jl}^1]^{(2pl)}\mid(-1)^j(h_{i+jl}^1-h_{i+jl}^0)=h_i^1-h_i^0,j=0,2,\cdots,p-1,\sum_{j=0}^{p-1}h_{i+jl}^0=h_0,\sum_{j=0}^{p-1}h_{i+jl}^1=h_1\}$,

for $h_0=0,1,\cdots,q-1,h_1=0,1,\cdots,q-1,i=0,1,\cdots,l-1$;

else

$c=c+2(p-1)l$;

$a=(\sum_{i=1}^{p}A_{2i-1},\sum_{i=1}^{p}A_{2i})$;

$cost[i,i+l,h_0,h_1]^{(2l)}=\min\{\sum_{j=0}^{(p-1)/2}cost[i+2jl,i+2jl+pl,h_{i+2jl}^0,h_{i+2jl}^1]^{(2pl)}+\sum_{j=0}^{(p-1)/2-1}cost[i+(2j+1)l,i+(2j+1)l+pl,h_{i+(2j+1)l}^0,h_{i+(2j+1)l}^1]^{(2pl)}\mid\sum_{j=0}^{p-1}h_{i+jl}^0=h_0,\sum_{j=0}^{p-1}h_{i+jl}^1=h_1\}$

for $h_0=0,1,\cdots,q-1,h_1=0,1,\cdots,q-1,i=0,1,\cdots,l-1$;

end if

end if

end while

if $cost[0,1,0,0]^{(2)}>k$ then

if $\min_{0<h<q}\{cost[0,1,h,h]^{(2)}\}\leqslant k$ then

$c=c+1$;

else

if $\min_{h_0+h_1=0}\{cost[0,1,h_0,h_1]^{(2)}\}\leqslant k$ then

$c=c+1$;

else

$c=c+2$;

end if

end if

end if

算法 2 在 $k=0$ 时等价于魏 -肖 -陈算法. 由下面的命题易知算法 2 的正确性.

命题 1 **设** $cost(l)=\sum_{i=1}^{l-1}\min_{\substack{0\leqslant h_0<q,\\0\leqslant h_1<q}}\{cost[i,i+l,h_0,h_1]^{(2l)}\}$,**则** $cost(l)\leqslant k,l=p^n,\cdots,p,l$.

证明 **当** $l=p^n$ **时,显然** $cost(l)=0$.

在第 j 步：若 $T_B \leqslant k$ 且 $T_C \leqslant k$，则 $\mathrm{cost}(l)=T_C \leqslant k$；若 $T_B \leqslant k$ 且 $T_C > k$，则 $\mathrm{cost}(l)=T_B \leqslant k$；若 $T_B > k$ 且 $T_D \leqslant k$，则 $\mathrm{cost}(l)=T_D \leqslant k$；若 $T_B > k$ 且 $T_D > k$，则

$$\mathrm{cost}[i,i+l,h_0,h_1]^{(2l)}=\min\{\sum_{j=0}^{(p-1)/2}\mathrm{cost}[i+2lj,i+pl+2lj,h^0_{i+lj},h^1_{i+lj}]^{2pl}+\sum_{j=0}^{(p-1)/2-1}\mathrm{cost}[i+(2j+1)l,i+pl+(2j+1)l,h^1_{i+(2j+1)l},h^0_{i+(2j+1)l}]^{(2pl)}\mid\sum_{j=0}^{p-1}h^0_{i+lj}=h_0,\sum_{j=0}^{p-1}h^1_{i+lj}=h_1\}$$

$$h_0=0,1,\cdots,q-1;h_1=0,1,\cdots,q-1;i=0,1,\cdots,l-1.$$

因此，

$$\mathrm{cost}(l)=\sum_{i=1}^{l-1}\min_{\substack{0\leqslant h_0<q,\\0\leqslant h_1<q}}\{\mathrm{cost}[i,i+l,h_0,h_1]^{(2l)}\}=\sum_{i=1}^{pl-1}\min_{\substack{0\leqslant h_0<q,\\0\leqslant h_1<q}}\{\mathrm{cost}[i,i+pl,h_0,h_1]^{(2pl)}\}=\mathrm{cost}(pl)\leqslant k.$$

4　确定 m 紧错线性复杂度的快速算法

在算法 2 的基础上，易求周期为 $2p^n$ 的 q 元序列 S 的 m 紧错线性复杂度. 首先对算法 2 作如下修改：在 While 循环前增加“$T_{\min}=2l$”；在第 1 个“$c=c+(p-1)l$” 之前增加“if $T_C<T_{\min}$ then $T_{\min}=T_C$”；在“$D_i=(-1)^{i+1}(A_{p+i}-A_i)$” 之前增加“if $T_B<T_{\min}$ then $T_{\min}=T_B$”；在“$c=c+2(p-1)l$” 之前增加“if $T_D<T_{\min}$ then $T_{\min}=T_D$”；在“if $\mathrm{cost}[0,1,0,0]^{(2)}>k$ then” 后增加“if $\mathrm{cost}[0,1,0,0]^{(2)}<T_{\min}$ then $T_{\min}=\mathrm{cost}[0,1,0,0]^{(2)}$”；在“$c=c+2$” 之前增加“if $\min(\min_{0<h<q}\{\mathrm{cost}[0,1,h,h]^{(2)}\},\min_{h_0+h_1=0}\{\mathrm{cost}[0,1,h_0,h_1]^{(2)}\})<T_{\min}$ then $T_{\min}=\min(\min_{o<h<q}\{\mathrm{cost}[0,1,h,h]^{(2)}\},\min_{h_0+h_1=0}\{\mathrm{cost}[0,1,h_0,h_1]^{(2)}\})$”.

得到的算法记为算法 3. 首先用 $k=0$ 调用算法 3，得到原始序列 0 错线性复杂度 c_0，即 0 紧错线性复杂度 $(0,c_0)$，同时得到 $T_{\min}$，记为 k_1；用 $k=k_1$ 调用算法 3，则得到原始序列的 k_1 错线性复杂度 c_1，如果 $c_1>c_0$ 即 1 紧错线性复杂度 (k_1,c_1)，同时得到 $T_{\min}$，记为 k_2；用 $k=k_2$ 调用算法 3，则得到原始序列的 k_2 错线性复杂度 c_2，如果 $c_2>c_1$ 即 2 紧错线性复杂度 (k_2,c_2)，同时得到 $T_{\min}$，记为 k_3；类似递归调用算法 3，即得到原始序列的 m 紧错线性复杂度 (k_m,LC_m).

从 0 错线性复杂度开始重复 k 错线性复杂度的计算，并在每次求 k 错线性复杂度的同时，计算使序列线性复杂度再次下降最少需要改变原始序列的位数 $T_{\min}$. 在当前 k 错线性复杂度计算过程每步循环中，在 $T_B\leqslant k,T_C>k$ 和 $T_B>k,T_D\leqslant k$ 的情况下不能通过改变原始序列不大于 k 位而使 c 降低 $2(p-1)l$，可以求出使 c 降低 $2(p-1)l$ 所需改变的原始序列的最小位数 T_C；在 $T_B>k,T_D>k$ 的情况下不能通过改变原始序列不大于 k 位而使 c 降低，可以求出使 c 降低所需改变的原始序列的最小位数 $\min(T_B,T_D)$. 所有使每步循环中的 c 降低所需改变原始序列的最小位数中的最小值就是 $T_{\min}$，最后，当 $\mathrm{cost}[0,1,0,0]^{(2)}>k$ 时还需计算一次 $T_{\min}$. 在当前 k 错线性复杂度计算完毕后将 $k=T_{\min}$ 代入下次 k 错线性复杂度的计算，设当前 $k=k_1$，计算 k_1 错线性复杂度 c_1，同时得到 $T_{\min}$；将 $k_2=T_{\min}$，再计算 k_2 错线性复杂度 $c_2(c_1>c_2)$，同时得到 $T_{\min}$；类似递归调用算法 3，最终得到原始序列的 m 紧错线性复杂度.

例 1　求 3 元序列 S 的紧错线性复杂度：

$$S^{50}=12101000121202102202112211212121210121012102110100.$$

12 次调用算法 3，得到的结果为 $(0,45),(5,28),(7,25),(9,25),(11,22),(13,21),(21,8),(23,5),(24,5),(26,4),(29,1),(37,0)$. **由于在每次 k 错线性复杂度计算中，当 $T_B>k,T_D\leqslant k$ 的情况下改动 T_B 位不能使序列的线性复杂度下降，而需要改动 T_C 位才能使线性复杂度下降，因此去掉** $(9,25),(24,5)$，**依次得到紧错线性复杂度为** $(0,45),(5,28),(7,25),(11,22),(13,21),(21,8),(23,5),(26,4),(29,1),(37,0)$.

5　结语

在 k 错线性复杂度相关理论基础上提出 m 紧错线性复杂度的概念来研究流密码稳定性，优化魏 -肖-陈算法的结构，通过联合代价的方法给出了确定周期为 $2p^n$ 的 q 元序列的 k 错线性复杂度算法，并由 k 错线性复杂度算法给出了确定周期为 $2p^n$ 的 q 元序列的 m 紧错线性复杂度算法，其中 p 和 q 为奇素数，q 为模 p^2

的一个本原根.

Stamp-Martin 算法中,最先有可能增加的线性复杂度为 2^{n-1},其次分别为 $2^{n-2},\cdots,2,1,1$. 假如第 j 步有可能通过改变原始序列 s^N 中小于或等于 k 个元素而使线性复杂度减少 2^{n-j},后面所有步骤中有可能增加的线性复杂度为 $2^{n-j-1}+\cdots+2+1+1=2^{n-j}$,因此第 j 步中应该通过改变原始序列 s^N 中小于或等于 k 个元素而使线性复杂度减少 2^{n-j}. 由 Stamp-Martin 算法模式引出的其他特殊周期序列的 k 错线性复杂度算法,也可类似扩展为 m 紧错线性复杂度算法.

提出 m 紧错线性复杂度概念的另外一个重要原因是,迄今为止所有求 k 错线性复杂度的算法均满足 Stamp-Martin 模式. 而很多求线性复杂度的快速算法,如魏 -肖 -陈算法[8],不满足 Stamp-Martin 模式,故不容易扩展成为求 k 错线性复杂度的快速算法,详细论述参见文献[5]. 而对于确定的 m,求线性复杂度的快速算法一般都可以扩展成为求 m 紧错线性复杂度的快速算法.

Lauder A 等[7] 给出了一次确定周期为 2^n 二元序列 k- 错复杂度曲线的算法. 使用文中的算法求最小错误 minerror(S) 或 m 紧错线性复杂度(m 较小) 明显优于文献[7] 中的算法.

参考文献:

[1] DING Cun-sheng, XIAO Guo-zhen, SHAN Wei-juan. The Stability Theory of Stream Ciphers [M]. Springer Verlag: Lecture Notes in Computer Science, 1991: 561.

[2] STAMP M, MARTIN C F. An Algorithm for the k-Error Linear Complexity of Binary Sequences with Period 2^n [J]. IEEE Trans. on Information Theory, 1993, 39(4): 1 398-1 401.

[3] GAMES R A, CHAN A H. A Fast Algorithm for Determining the Complexity Pseudo Random Sequence with Period 2^n [J]. IEEE Trans. on Information Theory, 1983, 29(1): 144-146.

[4] WEI Shi-min, XIAO Guo-zhen, CHEN Zhong. A Fast Algorithm for Determining the Minimal Polynomial of a Sequence with a Period $2p^n$ Over GF(q) [J]. IEEE Trans. on Information Theory, 2002, 48(10): 2 754-2 758.

[5] 戴小平,周建钦. 求周期为 $2p^m$ 二元序列 k 错线性复杂度的快速算法 [J]. 兰州大学学报:自然科学版, 2008, 44(1): 65-70.

[6] KUROSAWA K, SATO F, SAKATA T, et al. A Relationship Between Linear Complexity and k-Error Linear Complexity [J]. IEEE Trans. on Information Theory, 2000, 46(2): 694-698.

[7] LAUDER A, PATERSON K. Computing the Error Linear Complexity Spectrum of a Binary Sequence of Period 2^n [J]. IEEE Trans. on Information Theory, 2003, 49(1): 273-280.

[8] 魏仕民,肖国镇,陈 钟. 确定周期为 $2^n p^m$ 二元序列线性复杂度的快速算法 [J]. 中国科学:E 辑, 2002, 32(3): 401-408.

m-Tight Error Linear Complexity of Sequences with Period $2p^n$ over GF(q)

ZHOU Jian-qin[1,2], SHANGGUAN Cheng[2]

(1. College of Telecommunication, Hangzhou Dianzi University, Hangzhou 310018, China; 2. College of Computer Science and Technology, Anhui University of Technology, Maanshan 243002, Anhui China)

Abstract: Using the theories of the minimum error and the k-error linear complexity profile of sequences, m-tight error linear complexity is presented to study the stability of the linear complexity of sequences. First, the structure of the Wei-Xiao-Chen algorithm for the linear complexity of sequences with period $2p^n$ over GF(q) is optimized, where p and q are odd primes and q is a primitive root (mod p^2). Second, the union cost is used, so that an efficient algorithm for computing the k-error linear complexity of a sequence with period $2p^n$ over GF(q) is derived, where p and q are odd primes and q is a primitive root (mod p^2). Finally, an efficient algorithm for computing m-tight error linear complexity of sequences with period $2p^n$ over GF(q) is given, where p and q are odd primes and q is a primitive root (mod p^2).

Key words: stream cipher; sequence; linear complexity; k-error linear complexity; m-tight error linear complexity

(责任编辑 向阳洁)