计算有限域 GF(q)上 $2p^n$-周期序列的 k-错线性复杂度及其错误序列的算法

牛志华*①②　　孔得宇①
①(上海大学计算机工程与科学学院　上海　200444)
②(上海大学先进研究院　上海　200444)

摘　要：序列的 k-错线性复杂度是序列线性复杂度稳定性的重要评价指标。在求得一个序列 k-错线性复杂度的同时，也需要求出是哪些位置的改变导致了序列线性复杂度的下降。该文提出一个在 GF(q) 上计算 $2p^n$-周期序列 s 的 k-错线性复杂度以及对应的错误序列 e 的算法，这里 p 和 q 是素数，且 q 是一个模 p^2 的本原根。该文设计了一个追踪代价向量的 trace 函数，算法通过 trace 函数追踪最小的代价向量来求出对应的错误序列 e，算法得到的序列 e 使得 $(s+e)$ 的线性复杂度达到 k-错线性复杂度的值。

关键词：密码学；周期序列；线性复杂度；k-错线性复杂度；错误序列
中图分类号：TN918.1　　**文献标识码**：A　　**文章编号**：1009-5896(2018)07-1723-08
DOI: 10.11999/JEIT170972

Algorithm for Computing the k-error Linear Complexity and the Corresponding Error Sequence of $2p^n$-periodic Sequences over GF(q)

NIU Zhihua①②　　KONG Deyu①
①(*School of Computer Engineering and Science*, *Shanghai University*, *Shanghai* 200444, *China*)
②(*Shanghai Institute for Advanced Communication and Data Science*, *Shanghai University*, *Shanghai* 200444, *China*)

Abstract: The k-error linear complexity of a sequence is a fundamental concept for assessing the stability of the linear complexity. After computing the k-error linear complexity of a sequence, those bits that make the linear complexity reduced also need to be computed. For $2p^n$-periodic sequence over GF(q), where p and q are odd primes and q is a primitive root modulo p^2, an algorithm is presented, which not only computes the k-error linear complexity of a sequence s but also gets the corresponding error sequence e. A function is designed to trace the vector **cost** called "trace function", so the error sequence e can be computed by calling the "trace function", and the linear complexity of $(s+e)$ reaches the k-error linear complexity of the sequence s.

Key words: Cryptography; Periodic sequence; Linear complexity; k-error linear complexity; Error sequence

1　引言

线性复杂度是密钥流序列随机性的一个非常重要的指标。线性复杂度低的序列是不安全的，因为只要知道连续 2LC(s) 个比特，就可以通过 Berlekamp-Massey 算法[1]恢复出整个序列。然而高线性复杂度的序列也可能是不安全的，比如当改变某些序列的几位之后，序列的线性复杂度大幅下降，这样线性复杂度不稳定的序列，作为密钥序列不安全。针对这一问题，国内外学者相继提出了球体复杂度、重量复杂度[2,3]、k-错线性复杂度[4]的概念，后来人们更习惯用 k-错线性复杂度的概念来描述序列线性复杂度的稳定性。

近年来，针对周期序列线性复杂和 k-错线性复杂度的研究，有了大量的研究进展[5−19]。

对于 2^n-周期的序列，文献[5]提出了一个计算序列线性复杂度的快速算法，文献[4]提出了一个计算序列 k-错线性复杂度的快速算法，结合文献[4]和文献[5]，文献[6]提出了计算序列线性复杂度谱的算法。

对于 p^n-周期的序列，文献[7]提出了一个计算序列线性复杂度的算法，在文献[7]算法的基础上，文献[8]和文献[9]提出了计算序列 k-错线性复杂度的算法，文献[10]提出了一个计算序列线性复杂度谱的算法。

收稿日期：2017-10-20；改回日期：2018-01-15；网络出版：2018-04-08
*通信作者：牛志华　zhniu@staff.shu.edu.cn
基金项目：上海市自然科学基金 (16ZR1411200, 17ZR1409800)，国家自然科学基金(61772022, 61572309, 61462077)
Foundation Items: Shanghai Natural Science Foundation (16ZR1411200, 17ZR1409800), The National Nature Science Foundation of China (61772022, 61572309, 61462077)

对于$2p^n$-周期的序列，文献[11]提出了计算序列线性复杂度的算法，文献[12]提出了计算序列k-错线性复杂度的算法，文献[13]在文献[12]算法的基础上，提出了改进的计算序列k-错线性复杂度的算法。更多关于线性复杂度和k-错线性复杂度的研究成果见文献[14–19]。

k-错线性复杂度是评价序列线性复杂度的稳定性的重要指标，我们知道了一个序列改变了若干位之后，它的线性复杂度下降到了多少，还需要知道是哪些位的改变使得序列的线性复杂度发生下降。即我们在前人计算序列的k-错线性复杂度的基础上[5–19]，要计算出达到该k-错线性复杂度的错误序列e。目前，对于周期序列的错误序列的研究，文献[20]提出了一个计算2^n周期序列的错误序列算法，文献[21]提出了一个计算p^n周期序列的错误序列算法。本文给出了一个在GF(q)上计算$2p^n$-周期序列的k-错线性复杂度以及对应错误序列e的算法，这里p和q是素数，且q是一个模p^2的本原根。本文设计了一个追踪代价向量的trace函数，算法通过trace函数追踪最小的代价向量$\mathbf{cost}[i,i+l,h_0,h_1]^{(2)}$，$0\le h_0<q, 0\le h_1<q$来求出对应的错误序列$e$，算法得到的序列$e$使得$(s+e)$的线性复杂度等于序列$s$的$k$-错线性复杂度。

2 准备工作

周期为N的序列s的k-错线性复杂度定义为，当改变s的一个周期中至多$k(0\le k<N)$位之后，得到的所有序列的线性复杂度中的最小值，记为$\mathrm{LC}_k(s)$。即

$$\mathrm{LC}_k(s)=\min\left\{\mathrm{LC}(s+e)|w_H(e)\le k\right\}$$

这里e是周期为N的序列，常被称为错误序列、错误类型，$w_H(e)$表示序列e的一个周期中不为零的元素个数。显然，至少存在一个错误序列e，使得$\mathrm{LC}(s+e)=\mathrm{LC}_k(s)$，我们称使得$(s+e)$的线性复杂度等于$k$-错线性复杂度的错误序列$e$为$k$-错线性复杂度对应的错误序列。

设$s=(s_0,s_1,s_2,\cdots)$是在有限域GF(q)上周期为$2p^n$的序列，p和q是素数，且q是一个模p^2的本原根。设$s^N=\left(s_0,s_1,s_2,\cdots,s_{2p^n-1}\right)$是$s$的一个周期，$N=2p^n$，$s$的线性复杂度可以通过文献[11]算法计算出来。

文献[12]定义代价向量$\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}$是在原始序列$s$中，改变当前元素$s_i$到$h_0$以及$s_{i+l}$到$h_1$的改变位数的最小值，这里$h_0=0,1,\cdots,q-1$，$h_1=0,1,\cdots,q-1$，$2l$是当前元素的数量。

在每次循环中，算法把序列s分成$2p$行，记为$A_1,A_2,\cdots,A_{2p}$。在计算k-错线性复杂度的时候，我们使用下面的符号：

T_A：使得$A_1=A_3=\cdots=A_{2p-1}$且$A_2=A_4=\cdots=A_{2p}$的代价；

T_B：使得$A_1+A_{p+1}=A_2+A_{p+2}=\cdots=A_p+A_{2p}$的代价；

T_C：使得$A_1-A_{p+1}=(-1)^{t+1}\left(A_t-A_{p+t}\right)$，for $1\le t\le p$的代价。

根据Wei-Xiao-Chen算法[11]，在每次循环中，$A_1,A_2,\cdots,A_{2p}$有4种可能的情况，c的值(代表线性复杂度的变化)将会按照下面的情况发生变化：

情况 1 $A_1=A_3=\cdots=A_{2p-1}$且$A_2=A_4=\cdots=A_{2p}$，此时$T_A\le k$，在这种情况下，c不变；

情况 2 $A_1+A_{p+1}=A_2+A_{p+2}=\cdots=A_p+A_{2p}$，此时$T_A>k, T_B\le k$，$c=c+(p-1)l$；

情况 3 $A_1-A_{p+1}=(-1)^{t+1}\left(A_t-A_{p+t}\right)$, for $1\le t\le p$，此时$T_A>k, T_B>k, T_C>k, c=c+(p-1)l$；

情况 4 此时$T_A>k, T_B>k, T_C>k$，在这种情况下，$c=c+2(p-1)l$。

T_A,T_B和T_C按照式(1)~式(3)计算：

$$T_A=\sum_{i=0}^{l-1}\min\nolimits_{0\le h_0,h_1<q}\left\{\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}\right\} \tag{1}$$

$$T_B=\sum_{i=0}^{l-1}\min\nolimits_{0\le h<q}\left\{\sum_{j=0}^{p-1}\mathrm{bcost}[i+jl,h]^{(pl)}\right\},$$

$$\mathrm{bcost}[i,h]^{(pl)}=\min\nolimits_{0\le d_1,d_2<q}\left\{\mathbf{cost}[i,i+pl,d_1,d_2]^{(2pl)}\middle| d_1+d_2=h\right\} \tag{2}$$

$$T_C=\sum_{i=0}^{l-1}\min\nolimits_{0\le h<q}\left\{\sum_{j=0}^{p-1}\mathrm{ccost}[i+jl,h]^{(pl)}\right\},$$

$$\mathrm{ccost}[i+jl,h]^{(pl)}=\min\left\{\mathbf{cost}\left[i+jl,\ i+jl+pl,h_{i+jl}^0,h_{i+jl}^1\right]^{(2pl)}\middle|(-1)^j\cdot\left(h_{i+jl}^1-h_{i+jl}^0\right)=h\right\} \tag{3}$$

文献[12]没有考虑到$T_A>k$时，$T_B\le k$且$T_C\le k$的情况，文献[13]中，用两个分支分别计算$T_A>k$，$T_B\le k$以及$T_A>k$，$T_C\le k$的情况，每个分支求得一个c，而最小的c即是正确的k-错线性复杂度。本文结合文献[12]和文献[13]，介绍计算$2p^n$周期序列s的k-错线性复杂度的算法。

对于$l=p^m$，$\mathbf{cost}[i,i+l,s_i,s_{i+l}]^{(2l)}$的初始值是0，$\mathbf{cost}[i,i+l,s_i+\alpha,s_{i+l}]^{(2l)}$和$\mathbf{cost}[i,i+l,s_i,s_{i+l}+\beta]^{(2l)}$

的初始值是 1，$\mathbf{cost}\left[i,i+l,s_i+\alpha,s_{i+l}+\beta\right]^{(2l)}$ 的初始值是 2，$i=0,1,\cdots,l-1,\alpha=0,1,\cdots,q-1,\beta=0,1,\cdots,q-1$。

在每次循环中，代价向量 **cost** 将按照式(4)~式(7)计算，同样有 4 种情况：

$$\begin{aligned}&\text{update cost } T_A:\\&\quad\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}\\&=\sum_{j=0}^{(p-1)/2}\mathbf{cost}\left[i+2jl,i+pl+2jl,h_0,h_1\right]^{(2pl)}\\&\quad+\sum_{j=0}^{(p-1)/2-1}\mathbf{cost}\left[i+(2j+1)l,i+pl\right.\\&\quad\left.+(2j+1)l,h_1,h_0\right]^{(2pl)},\ h_0=0,1,\cdots,q-1;\\&\quad h_1=0,1,\cdots,q-1;\ i=0,1,\cdots,l-1\end{aligned}\tag{4}$$

$$\begin{aligned}&\text{update cost } T_B:\\&\quad\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}\\&=\min\left\{\sum_{j=0}^{p-1}\mathbf{cost}\left[i+jl,i+pl+jl,h_{i+jl}^0,h_{i+jl}^1\right]^{(2pl)}\right.\\&\quad\left|h_{i+jl}^0+h_{i+jl}^1=h_i^0+h_i^1,j=1,2,\cdots p-1,\right.\\&\quad\left.\sum_{j=0}^{p-1}(-1)^jh_{i+jl}^0=h_0,\sum_{j=0}^{p-1}(-1)^jh_{i+jl}^1=h_1\right\},\\&\quad h_0{=}0,1,\cdots,q-1;h_1=0,1,\cdots,q-1;\\&\quad i=0,1,\cdots,l-1\end{aligned}\tag{5}$$

$$\begin{aligned}&\text{update cost } T_C:\\&\quad\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}\\&=\min\left\{\sum_{j=0}^{p-1}\mathbf{cost}\left[i+jl,i+pl+jl,h_{i+jl}^0,h_{i+jl}^1\right]^{(2pl)}\right.\\&\quad\left|(-1)^j\left(h_{i+jl}^1-h_{i+jl}^0\right)=h_i^1-h_i^0,\right.\\&\quad\left.j=1,2,\cdots,p-1,\ \sum_{j=0}^{p-1}h_{i+jl}^0=h_0,\ \sum_{j=0}^{p-1}h_{i+jl}^1=h_1\right\},\\&\quad h_0=0,1,\cdots,q-1;\ h_1=0,1,\cdots,q-1;\\&\quad i=0,1,\cdots,l-1\end{aligned}\tag{6}$$

$$\begin{aligned}&\text{update cost } T_D:\\&\quad\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}\\&=\min\left\{\sum_{j=0}^{(p-1)/2}\mathbf{cost}\left[i{+}2jl,i{+}pl+2jl,h_{i+2jl}^0,h_{i+2jl}^1\right]^{(2pl)}\right.\\&\quad+\sum_{j=0}^{(p-1)/2-1}\mathbf{cost}\left[i+(2j+1)l,i+pl+(2j+1)l,\right.\\&\quad\left.h_{i+(2j+1)l}^1,h_{i+(2j+1)l}^0\right]^{(2pl)}\left|\sum_{j=0}^{p-1}h_{i+jl}^0=h_0,\right.\\&\quad\left.\sum_{j=0}^{p-1}h_{i+jl}^1=h_1\right\},\ h_0=0,1,\cdots,q-1;\\&\quad h_1=0,1,\cdots,q-1;i=0,1,\cdots,l-1;\\&\quad j=1,2,\cdots,p-1\end{aligned}\tag{7}$$

在上面的公式中，$2pl$ 是当前元素的个数，$2l$ 是下次循环时的元素的个数。

文献[12]中证明了计算代价向量 **cost** 的正确性，关于 **cost** 向量的计算可以在文献[12]中找到依据。当出现 $T_B\le k,T_C\le k$ 的情况时，我们将计算分成两个分支，关于这种情况下的计算及正确性可以在文献[13]中找到依据。计算 GF(q) 上 $2p^n$-周期序列的 k-错线性复杂度的算法可参考文献[12,13]。

3　计算 GF(q) 上 $2p^n$-周期序列的 k-错线性复杂度及其错误序列的算法

本节将提出 GF(q) 上计算 $2p^n$-周期序列的 k-错线性复杂度 $\mathrm{LC}_k(s)$ 以及达到 k-错线性复杂度对应的错误序列 e 的算法。我们将在定理 1 中证明错误序列的存在性以及可以通过追踪最小的代价向量 **cost** 来计算出达到 k-错线性复杂度的错误序列 e；并通过归纳定理 1 中追踪最小的代价向量得到错误序列的过程，得出 trace 函数；最后通过归纳计算 GF(q) 上计算 $2p^n$-周期序列的 k-错线性复杂度算法和 trace 函数得出算法 1，即计算 $2p^n$-周期序列的 k-错线性复杂度以及达到该 k-错线性复杂度的错误序列 e 的算法。

定理 1　假设在计算 $2p^n$-周期序列的 k-错线性复杂度的最后，计算出了 $\mathbf{cost}\left[0,1,h_0,h_1\right]^{(2)}$，$0\le h_0<q,0\le h_1<q$，和 k-错线性复杂度 $\mathrm{LC}_k(s)$，令 $\min_{0\le h_0<q,0\le h_1<q}\left\{\mathbf{cost}\left[0,1,h_0,h_1\right]^{(2)}\right\}=k'$，则存在一个错误序列 e^N，且 $w_H\left(e^N\right)=k'$，使得 $(s+e)^N$ 的线性复杂度等于 $\mathrm{LC}_k(s)$，并且可以通过追踪最小的 $\mathbf{cost}\left[0,1,h_0,h_1\right]^{(2)}$ 计算出错误序列。

证明　代价向量 $\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}$ 是在原始序列 s 中，改变当前元素 s_i 到 h_0 以及 s_{i+l} 到 h_1 的改变位数的最小值，$0\le h_0<q,0\le h_1<q$，$2l$ 是当前元素的数量。

我们使用向量 **flag** 存储每次循环中所执行的分支。在第 $j(1\le j\le n)$ 步，

如果 $T_A\le k$，对应情况 1，则 $\mathbf{flag}[l]=1$；

如果 $T_A>k,T_B\le k$，对应情况 2，则 $\mathbf{flag}[l]=2$；

如果 $T_A>k,T_C\le k$，对应情况 3，则 $\mathbf{flag}[l]=3$；

如果 $T_A>k,T_B>k,T_C>k$，对应情况 4，则 $\mathbf{flag}[l]=4$。

我们将从最小的 $\mathbf{cost}\left[0,1,h_0,h_1\right]^{(2)}$ 来开始追踪，因为代价向量代表改变原始序列的最小位数。最小的代价向量意味着我们可以得到需要改变最少的原始序列的位数。

这里只证明 $\mathbf{flag}[l]=4$ 的情况，其他情况相似：

如果 $\mathbf{flag}[l]=4$，按照式(7)计算 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}$。

在计算之后，将知道是哪些代价向量 $\mathbf{cost}^{(2pl)}$ 相加得到了 $\mathbf{cost}^{(2l)}$。之后，把这些 h^0_{i+2jl} 和 h^1_{i+2jl} $(i=0,1,\cdots,l-1; j=1,2,\cdots,p-1)$ 的值存储在向量 **sflag** 中，以便于在下一步中继续追踪这些代价向量 $\mathbf{cost}^{(2pl)}$。

如果 $pl<N/2$，将依次追踪这些代价向量；否则，意味着已经追踪到了代价向量的初始值，此时可以通过这些代价向量计算出错误序列，计算方法如下：

如果 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}\neq 0$，根据 **cost** 的定义，可以得知 a_i 或者 a_{i+l} 肯定从原始序列中发生了改变。因此，如果 $a_i\neq h_0$，错误序列的对应位置 $\mathbf{error}[i]=h_0-a_i$；同样，如果 $a_{i+l}\neq h_1$, $\mathbf{error}[i+l]=h_1-a_{i+l}$。

在追踪完所有的代价向量之后，将得到完整的错误向量 **error**，即所求的原始序列 s 所对应的错误序列 e，且 $(s+e)^N$ 的线性复杂度等于 $\mathrm{LC}_k(s)$。证毕

通过定理 1，得到下面的 trace 函数，trace 函数可以通过追踪最小的代价向量 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2)}$, $0\le h_0<q$, $0\le h_1<q$ 计算出对应 k-错线性复杂度的错误序列。trace 函数的详细过程如表 1 所示。

trace 函数是通过追踪代价向量 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}$ 来计算出序列的错误序列。首先，通过向量 $\mathbf{flag}[l]$ 判断执行哪一分支，然后通过再次计算 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}$ 来得到是哪些代价向量在上一步相加得到的 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}$，最后继续追踪这些代价向量，直到追踪到了代价向量的初始值，就可以计算出对应 k-错线性复杂度的错误序列。

根据计算 GF(q) 上 $2p^n$-周期序列的 k-错线性复杂度的算法和追踪代价向量的 trace 函数，我们归纳出表 2 的算法 1，算法输入序列 s 的一个周期 s^N 和一个固定的整数 k，输出序列 s 的 k-错线性复杂度 $\mathrm{LC}_k(s)$ 以及达到该 k-错线性复杂度的错误序列 e：

算法在步骤 16，通过 trace 函数追踪最小的 $\mathbf{cost}[0,1,h_0,h_1]^{(2)}, 0\le h_0<q, 0\le h_1<q$ 来求出对应 k-错线性复杂度的错误序列。trace 函数先从 $\mathbf{cost}^{(2)}$ 开始追踪，在第 1 次追踪之后可以得到 p 个代价向量 $\mathbf{cost}^{(2p)}$，第 2 步依次追踪这 p 个 $\mathbf{cost}^{(2p)}$ 将得到 p^2 个 $\mathbf{cost}^{(2p^2)}$，最终通过 n 步的追踪，将追踪到 p^n 个 $\mathbf{cost}^{(2p^n)}$，根据 **cost** 的定义，我们即可通过这 p^n 个 $\mathbf{cost}^{(2p^n)}$ 求出对应 k-错线性复杂度的错误序列。

算法在计算 k-错线性复杂度时，循环 n 次，每次循环中，计算 T_A,T_B,T_C 的最大时间复杂度为 $O(pl)$，更新代价向量 **cost** 的最大时间复杂度为 $O(l)$，因此计算 k-错线性复杂度的时间复杂度为 $O(npl)$。追踪错误序列时，循环 n 次，每次循环时间复杂度为 $O(l)$，因此计算错误序列的时间复杂度为 $O(nl)$。算法总的时间复杂度为 $O(npl)$。举个例子，使用算法 1 计算一个周期长 1250 的序列的错误序列所需时间大约为 1.5 s。

4 数值实验

本节用一个例子来说明，使用算法 1 计算 GF(q) 上 $2p^n$-周期序列 k-错线性复杂度以及达到该 k-错线性复杂度对应的错误序列 e 的详细过程。

设 s 是在有限域 GF(3)上，周期为 $N=2\times 5^2$ 的序列，s 的一个周期是：

$$s^N=\text{11201 02010 12011 20101 21110 12011}$$
$$\text{01021 01100 02101 02210}$$

我们将用算法 1 来计算 $k=8$ 时的错误序列：

初值：$a=s^N; l=5^2; c=0; \mathbf{error}[N]=0;$

$$\left(\mathbf{cost}[i,i+l,h_0,h_1]^{(50)}\right)=\begin{pmatrix}22112\ 02021\ 12111\ 11202\ 12220\\ 12201\ 11120\ 21022\ 21111\ 22211\\ 21212\ 12111\ 22122\ 20212\ 21121\\ 11121\ 12112\ 02200\ 12111\ 11111\\ 01210\ 21211\ 11111\ 22020\ 21102\\ 10221\ 22202\ 12211\ 21121\ 20012\\ 22022\ 11122\ 11211\ 02212\ 02221\\ 12111\ 20221\ 20122\ 12121\ 12212\\ 21122\ 21212\ 21222\ 11222\ 11122\end{pmatrix}$$

矩阵的第 $i(0\le i<5^2)$ 列，代表 $\mathbf{cost}[i,i+l,h_0,h_1]^{(2l)}$，$h_0$和$h_1$ 分别等于 (0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)。

第 1 次循环：$l=5$, $A_1=11201$, $A_2=02010$, $A_3=12011$, $A_4=20101$, $A_5=21110$, $A_6=12011$, $A_7=01021$, A_8=01100, $A_9=02101$, $A_{10}=02210$。

$T_A=23>k$, $T_B=12$, $T_C=11$

$\mathbf{flag}[l]=4$, $a=\left(\sum_{i=1}^{p}A_{2i-1},\sum_{i=1}^{p}A_{2i}\right)$, $c=40$

表 1　trace 函数详细过程

trace$\left(\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)},m\right)$：

如果 $(\mathbf{flag}[l]=4)$

按照 update cost T_D 计算 $\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}$；

$\mathbf{sflag}\left[i+2jl\right]=h^0_{i+2jl}$, $\mathbf{sflag}\left[i+2jl+p\right]=h^1_{i+2jl}$, for $j=0,1,\cdots,\dfrac{p-1}{2}$；

$\mathbf{sflag}\left[i+(2j+1)l\right]=h^1_{i+(2j+1)l}$, $\mathbf{sflag}\left[i+(2j+1)l+p\right]=h^0_{i+(2j+1)l}$, for $j=0,1,\cdots,\dfrac{p-1}{2}-1$；

for $j=0$ to $p-1$ 执行

rpart $=\mathbf{sflag}\left[j+p\right]$；

执行 computerror$(\mathbf{sflag},l,jl,j,m,\text{rpart})$；

end for

如果 $(\mathbf{flag}[l]=3)$

按照 update cost T_C 计算 $\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}$；

$\mathbf{sflag}\left[i+jl\right]=h^0_{i+jl}$, $\mathbf{sflag}\left[i+jl+p\right]=h^1_{i+jl}$, for $j=0,1,\cdots,p-1$；

for $j=0$ to $p-1$ 执行

rpart $=(-1)^j\left(h_1-h_0\right)+\mathbf{sflag}\left[j\right]$；

执行 computerror$(\mathbf{sflag},l,jl,j,m,\text{rpart})$；

end for

如果 $(\mathbf{flag}[l]=2)$

按照 update cost T_B 计算 $\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}$；

$\mathbf{sflag}\left[i+jl\right]=h^0_{i+jl}$, $\mathbf{sflag}\left[i+jl+p\right]=h^1_{i+jl}$, for $j=0,1,\cdots,p-1$

for $j=0$ to $p-1$ 执行

rpart $=h_0+h_1-\mathbf{sflag}\left[j\right]$；

执行 computerror$(\mathbf{sflag},l,jl,j,m,\text{rpart})$；

end for

如果 $(\mathbf{flag}[l]=1)$

按照 updatecost T_A 计算 $\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2l)}$；

$\mathbf{sflag}\left[2j\right]=h_0$, $\mathbf{sflag}\left[2j+1\right]=h_1$, for $j=0,1,\cdots,p-1$；

for $j=0$ to $p-1$ 执行

rpart $=\mathbf{sflag}\left[j+p\right]$；

执行 computerror$(\mathbf{sflag},l,jl,j,m,\text{rpart})$；

end for

computerror$(\mathbf{sflag},l,i,j,m,\text{rpart})$：

如果 $(pl<N/2)$

执行 trace$\left(\mathbf{cost}\left[i,i+pl,\mathbf{sflag}\left[j\right],\text{rpart}\right]^{(2pl)},i\right)$；

否则，如果 $\left(\mathbf{cost}\left[m+pj,m+pj+pl,\mathbf{sflag}\left[j\right],\text{rpart}\right]^{(2pl)}\neq 0\right)$

如果 $(a\left[m+pl\right]\neq\mathbf{sflag}\left[j\right])$

$\mathbf{error}\left[m+pj\right]=\mathbf{sflag}\left[j\right]-a\left[m+pl\right]$；

如果 $(a\left[m+pj+pl\right]\neq\text{rpart})$

$\mathbf{error}\left[m+pj+pl\right]=\text{rpart}-a\left[m+pl+pj\right]$；

否则，返回。

表 2 计算 GF(q) 上 $2p^n$ -周期序列的 k -错线性复杂度以及错误序列的算法

算法 1 计算 GF(q) 上 $2p^n$ -周期序列的 k-错线性复杂度以及错误序列的算法

输入：p,n,s^N,k

输出：$e,\mathrm{LC}_k(s)$

初始化：$N\leftarrow 2p^n;a\leftarrow s^N;l\leftarrow p^n;c\leftarrow 0;C_{\min}\leftarrow N;\mathbf{error}[N]\leftarrow 0$

$\mathbf{cost}\left[i,i+l,a_i,a_{i+l}\right]^{(2l)}\leftarrow 0$

$\mathbf{cost}\left[i,i+l,a_i+\alpha,a_{i+l}\right]^{(2l)}\leftarrow 1$

$\mathbf{cost}\left[i,i+l,a_i,a_{i+l}+\beta\right]^{(2l)}\leftarrow 1$

$\mathbf{cost}\left[i,i+l,a_i+\alpha,a_{i+l}+\beta\right]^{(2l)}\leftarrow 2$

for $i=0,1,\cdots,l-1$

$\alpha=1,2,\cdots,q-1;\beta=1,2,\cdots,q-1$

步骤 1 如果 $l>1$，转向步骤 2；否则，转向步骤 9；

步骤 2 $l\leftarrow l/p$, $A_i\leftarrow\left(a_{(i-1)l},a_{(i-1)l+1},\cdots,a_{il-1}\right)$ for $i=1,2,\cdots,2p$，计算 T_A，如果 $T_A\le k$，转向步骤 3；否则转向步骤 4；

步骤 3 $a\leftarrow\left(A_1,A_{p+1}\right)$；按照 update cost T_A 改变代价向量 **cost**，$\mathbf{flag}[l]\leftarrow 1$，转向步骤 1；

步骤 4 计算 T_B,T_C，如果 $T_B>k$ 且 $T_C>k$，转向步骤 5；否则转向步骤 6；

步骤 5 $c\leftarrow c+2(p-1)l,a\leftarrow\left(\sum_{i=1}^{p}A_{2i-1},\sum_{i=1}^{p}A_{2i}\right)$，按照 update cost T_D 改变代价向量 **cost**，$\mathbf{flag}[l]\leftarrow 4$，转向步骤 1；

步骤 6 $c\leftarrow c+(p-1)l$，如果 $T_B\le k$ 且 $T_C>k$，转向步骤 7；如果 $T_B>k$ 且 $T_C\le k$，转向步骤 8；如果 $T_B\le k$ 且 $T_C\le k$，在两个分支中转向步骤 7 和步骤 8；

步骤 7 $a\leftarrow\left(\sum_{i=1}^{p}(-1)^{i+1}A_i,\sum_{i=1}^{p}(-1)^{i+1}A_{p+i}\right)$，按照 update cost T_B 改变代价向量 **cost**, $\mathbf{flag}[l]\leftarrow 2$，转向步骤 1；

步骤 8 $a\leftarrow\left(\sum_{i=1}^{p}A_i,\sum_{i=1}^{p}A_{p+i}\right)$，按照 update cost T_C 改变代价向量 **cost**, $\mathbf{flag}[l]\leftarrow 3$，转向步骤 1；

步骤 9 如果 $\mathbf{cost}[0,1,0,0]^{(2)}>k$，转向步骤 10；否则，转向步骤 14；

步骤 10 如果 $\min_{0\le h<q}\left\{\mathbf{cost}[0,1,h,h]^{(2)}\right\}\le k$，转向步骤 12；否则，转向步骤 11；

步骤 11 如果 $\min_{h_0+h_1=0}\left\{\mathbf{cost}\left[0,1,h_0,h_1\right]^{(2)}\right\}\le k$，转向步骤 12；否则，转向步骤 13；

步骤 12 $c\leftarrow c+1$，转向步骤 14；

步骤 13 $c\leftarrow c+2$，转向步骤 14；

步骤 14 如果 $C_{\min}>c$ 则 $C_{\min}\leftarrow c$, $\mathrm{LC}_k(s)\leftarrow C_{\min}$；

步骤 15 计算使得 $\mathbf{cost}\left[0,1,h_0,h_1\right]^{(2)},0\le h_0<q,\ \ 0\le h_1<q$ 最小的 h_0,h_1，转向步骤 16；

步骤 16 执行 $\mathrm{trace}\left(\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2)},m\right)$, $i=0,m=0$，转向步骤 17；

步骤 17 $e\leftarrow\mathbf{error}$，输出 e 和 $\mathrm{LC}_k(s)$，结束。

$$\left(\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(10)}\right)=\begin{pmatrix}1&2&2&1&2\\2&1&1&2&2\\2&2&2&2&1\\0&1&1&0&1\\1&0&0&1&1\\1&1&1&1&0\\1&2&2&1&2\\2&1&1&2&2\\2&2&2&2&1\end{pmatrix}\qquad\left(\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2)}\right)=\begin{pmatrix}8\\6\\8\\5\\3\\5\\9\\7\\9\end{pmatrix}$$

第 2 次循环：$l=1,\ A_1=1,\ A_2=1,\ A_3=1,\ A_4=1,\ A_5=1,\ A_6=0,\ A_7=1,\ A_8=1,\ A_9=0,\ A_{10}=2$。$T_A=3<k$，$a=\left(A_1,A_{p+1}\right)$, $\mathbf{flag}[l]=1$; $a=(1,0)$,$\mathbf{cost}[0,1,0,0]^{(2)}=8=k$，因此 $c=40$，$\mathrm{LC}_k(s)=40$。

计算出 $h_0=0,\ h_1=0$。执行 $\mathrm{trace}\left(\mathbf{cost}[0,1,0,0]^{(2)},0\right)$。开始追踪：$\mathbf{flag}[1]=1$，按照公式(4)计算 $\mathbf{cost}[0,1,0,0]^{(2)}$，在计算 $\mathbf{cost}[0,1,0,0]^{(2)}$ 中存储向量 **sflag**：做循环 for $j=0$ to $p-1$ 执行 $\mathrm{computerror}\left(\mathbf{sflag},l,jl,j,m,\mathbf{sflag}[j+p]\right)$。

举例，执行 $\mathrm{computerror}\left(\mathbf{sflag},1,0,0,0,0\right)$：此时，$pl<N/2$，执行 $\mathrm{trace}\left(\mathbf{cost}[0,5,0,0]^{(10)},0\right)$，$\mathbf{flag}[5]=4$，按照式(7)计算 $\mathbf{cost}[0,5,0,0]^{(10)}$，在计算 $\mathbf{cost}[0,5,0,0]^{(10)}$ 中存储向量 **sflag**：做循环 $j=0$ to $p-1$ 执行 $\mathrm{computerror}\left(\mathbf{sflag},l,jl,j,m,\mathbf{sflag}[j+p]\right)$。

举例，执行 $\mathrm{computerror}\left(\mathbf{sflag},5,0,0,0,h_0^1\right)$：此时 $pl=N/2,\ \mathbf{cost}[0,25,0,1]^{(25)}=1$，因为 $a[0]=1\neq\mathbf{sflag}[0]=0$，所以 $\mathbf{error}[0]=\left(\mathbf{sflag}[0]-a[0]+q\right)\bmod q=2$。

最终，可以得到，$e^N=$22202 00000 00000 00000 00000 02000 00020 00200 00000 00001，则 $(s+e)^N=$00100 02010 12011 20101 21110 11011 01011 01000 02101 02211。

我们使用算法 1 计算出序列 s 的 8-错线性复杂度为 $\mathrm{LC}_8(s)=40$ 以及对应的错误序列 e^N，用 Wei-Xiao-Chen[11]算法进行了验证，$(s+e)^N$ 的线性复杂度确实等于 s 的 8-错线性复杂度的值 40。

5　结束语

本文提出了在 GF(q) 上计算 $2p^n$-周期序列的 k-错线性复杂度以及达到该 k-错线性复杂度的错误序列 e 的算法，这里 p 和 q 是素数，且 q 是一个模 p^2 的本原根。

算法计算出的错误序列 e 使得 $(s+e)$ 的线性复杂度达到 k-错线性复杂度的值。算法是在计算代价向量的基础上，通过 trace 函数追踪最小的 $\mathbf{cost}\left[i,i+l,h_0,h_1\right]^{(2)},0\le h_0<q,0\le h_1<q$ 来求出对应的错误序列 e 的。最后，本文通过一个实例来说明计算 GF(q) 上 $2p^n$-周期序列 k-错线性复杂度以及通过 trace 函数求出对应的错误序列 e 的详细过程。

参 考 文 献

[1] MASSEY J L. Shift-register synthesis and BCH decoding [J]. *IEEE Transaction on Information Theory*, 1969, 15(1): 122–127. doi: 10.1109/TIT.1969.1054260.

[2] DING Cunsheng, XIAO Guozhen, and SHAN Weijuan. The Stability Theory of Stream Ciphers[M]. Berlin: Springer-Verlag, 1991.

[3] 丁存生, 肖国镇. 流密码学及其应用[M]. 北京: 国防工业出版社, 1994: 1–275.
DING Cunsheng and XIAO Guozhen. Stream Cipher and Applications[M]. Beijing: National Defense Industry Press, 1994: 1–275.

[4] STAMP M and MARTIN C F. An algorithm for the k-error linear complexity of binary sequences with period 2^n [J]. *IEEE Transactions on Information Theory*, 1993, 39(4): 1398–1401. doi: 10.1109/18.243455.

[5] GAMES R A and CHAN A. A fast algorithm for determining the complexity of a binary sequence with period 2^n [J]. *IEEE Transaction on Information Theory*, 1983, 29(1): 144–146. doi: 10.1109/TIT.1983.1056619.

[6] LAUDER A G B and PATERSON K G. Computing the error linear complexity spectrum of a binary sequence of period 2^n [J]. *IEEE Transactions on Information Theory*, 2003, 49(1): 273–280. doi: 10.1109/TIT.2002.806136.

[7] XIAO Guozhen, WEI Shimin, LAM K Y, *et al*. A fast algorithm for determining the linear complexity of a sequence with period p^n over GF(q)[J]. *IEEE Transactions on Information Theory*, 2000, 46(6): 2203–2206. doi: 10.1109/18.868492.

[8] WEI Shimin, CHEN Zhong, and XIAO Guozhen. A fast algorithm for the k-error linear complexity of a binary sequence[C]. International Conferences on Info-tech and Info-net, Beijing, 2001: 152–157.

[9] MEIDL W. Linear Complexity and k-Error Linear Complexity for p^n-Periodic Sequences[M]. Basel, Birkhäuser, Coding, Cryptography and Combinatorics, 2004: 227–235.

[10] TANG Miao and ZHU Shixin. On the error linear complexity spectrum of p^n-periodic binary sequences[J]. *Applicable Algebra in Engineering, Communication and Computing*, 2013, 24(6): 497–505. doi: 10.1007/s00200-013-0210-3.

[11] WEI Shimin, XIAO Guozhen, and CHEN Zhong. A fast algorithm for determining the minimal polynomial of a sequence with period $2p^n$ over GF(q) [J]. *IEEE Transactions on Information Theory*, 2002, 48(10): 2754–2757. doi: 10.1109/TIT.2002.802609.

[12] ZHOU Jianqin. On the k-error linear complexity of sequences with period $2p^n$ over GF(q)[J]. *Designs Codes & Cryptography*, 2011, 58(3): 279–296. doi: 10.1007/s10623-010-9379-7.

[13] NIU Zhihua, LI Zhe, CHEN Zhixiong, *et al*. Computing the k-error linear complexity of q-ary sequences with period $2p^n$ [J]. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2012, E95-A(9): 1637–1641. doi: 10.1587/transfun.E95.A.1637.

[14] CHEN Zhixiong, NIU Zhihua, and WU Chenhuang. On the k-error linear complexity of binary sequences derived from polynomial quotients[J]. *Science China Information Sciences*, 2015, 58(9): 1–15. doi: 10.1007/s11432-014-5220-7.

[15] NIU Zhihua, CHEN Zhixiong, and DU Xiaoni. Linear complexity problems of level sequences of Euler quotients and their related binary sequences[J]. *Science China Information Sciences*, 2016, 59(3): 1–12. doi: 10.1007/s11432-015-5305-y.

[16] LIU Longfei, YANG Xiaoyuan, DU Xiaoni, *et al*. On the k-error linear complexity of generalised cyclotomic sequences [J]. *International Journal of High Performance Computing & Networking*, 2016, 9(5/6): 394–400. doi: 10.1504/IJHPCN. 2016.080411.

[17] LIU Longfei, YANG Xiaoyuan, DU Xiaoni, *et al*. On the linear complexity of new generalized cyclotomic binary sequences of order two and period pqr[J]. *Tsinghua Science & Technology*, 2016, 21(3): 295–301. doi: 10.1109/TST.2016. 7488740.

[18] PAN Wenlun, BAO Zhenzhen, LIN Dongdai, *et al*. The distribution of 2^n -periodic binary sequences with fixed k-error linear complexity[C]. International Conference on Information Security Practice and Experience, Zhangjiajie, China, 2016: 13–36.

[19] YU Fangwen, SU Ming, WANG Gang, *et al*. Error decomposition algorithm for approximating the k-error linear complexity of periodic sequences[C]. IEEE TrustCom/ BigDataSE/ISPA, Tianjin, China, 2016: 505–510.

[20] SALAGEAN A. On the computation of the linear complexity and the k-error linear complexity of binary sequences with period a power of two[J]. *IEEE Transaction on Information Theory*, 2005, 51(3): 1145–1150. doi: 10.1109/TIT.2004. 842769.

[21] TANG Miao. An algorithm for computing the error sequence of p^n -periodic binary sequences[J]. *Applicable Algebra in Engineering, Communication and Computing*, 2014, 25(3), 197–212. doi: 10.1007/s00200-014-0222-7.

牛志华： 女，1976 年生，副教授，研究方向为序列密码.

孔得宇： 男，1991 年生，硕士生，研究方向为序列密码.