On the *k*-error Joint Linear Complexity and Error Multisequence over F_q (*char* $F_q = p$, prime)

M. Sindhu and M. Sethumadhavan*

TIFAC CORE in Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore, India
{m_sindhu,m_sethu}@cb.amrita.edu

Abstract. Finding fast and efficient algorithms for computing the *k*-error joint linear complexity and error multisequence of multisequences is of great importance in cryptography, mainly for the security analysis of word based stream ciphers. There is no efficient algorithm for finding the error multisequence of a prime power periodic multisequence. In this paper we propose an efficient algorithm for finding the *k*-error joint linear complexity together with an error multisequence of *m* fold prime power periodic multisequences over F_q, where char $F_q = p$, a prime.

Keywords: Word based stream ciphers, Multisequences, Error Joint Linear Complexity, Error multisequence, Generalized Stamp-Martin Algorithm.

1 Introduction

Complexity measures for keystream sequences over finite fields, such as the linear complexity and the *k*-error linear complexity, is of great relevance to cryptology, in particular, to the area of stream ciphers. Stream ciphers uses deterministically generated pseudorandom sequences to encrypt the message stream. The keystream should be a truly random sequence of elements of a finite field. Security of stream ciphers depends on the quality of the keystreams and the keystream must possess various properties such as having good statistical randomness properties and a high linear complexity in suitable sense, so that the keystream sequence cannot be predicted from a small portion of its terms of the sequence.

The vast majority of proposed keystream generators are based on the use of linear feedback shift registers (LFSR). The length of the shortest LFSR which generates the given sequence is known as the linear complexity of the sequence. A necessary requirement for unpredictability of keystream sequence is long period, which can be attained by large linear complexity. Developments in stream ciphers point towards an interest in word based stream ciphers which require the study of complexity theory of multisequences i.e., of parallel streams of finitely many sequences, and of their complexity properties ([6-8], [10], [12], [14]). A cryptographically strong sequence should not only have a large linear complexity, but also changing a few terms should

* Adjunct Scientist, Society for Electronic Transactions and Security (SETS), Chennai, Tamil Nadu, India.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 512–519, 2014.

not cause any significant decrease of the linear complexity. This unfavorable property leads to the concept of k-error linear complexity ([9], [13]). Many authors studied various properties of k-error linear complexity of single and multisequences ([1-5], [7], [10 -14]). In [13] Stamp and Martin gave an efficient algorithm for finding the k-error linear complexity of 2^n periodic binary sequences. This algorithm was later modified by Kaida [2] and he found out the corresponding error vector together with the k-error linear complexity. Kaida et al. in [3] further extended this algorithm to the case of sequences with period p^n over F_q; *Char* $F_q = p$.

There is no efficient algorithm for finding the error multisequence of a prime power periodic multisequence. In this paper we propose an efficient algorithm for the computation of error multisequence e together with the k-error joint linear complexity of m fold multisequences over F_q of period p^n. In other words, we find k-error joint linear complexity and an m fold p^n periodic multisequence $e = (e^{(0)}, e^{(1)}, ..., e^{(m-1)})$ such that

(i) $\sum_{i=0}^{m-1}\sum_{j=0}^{N-1} e_j^{(i)} \leq k, where\ e^{(i)} = (e_j^{(i)}), 0 \leq j \leq N-1, 0\leq i\leq m-1, 0 \leq k \leq mp^n$

(ii) The joint linear complexity of the multisequence $S+e$ instead of S is $L_{N,k}(S)$ - the k-error joint linear complexity of S.

Let $S = (S^{(0)}, S^{(1)}, ..., S^{(m-1)})$ with $S^{(h)} = (s_0^{(h)}, s_1^{(h)}, ..., s_{N-1}^{(h)})$, where $s_i^{(h)} \in F_q$, for $0 \leq h \leq m-1$,$0 \leq i \leq N-1$ be an m fold $N = p^n$ periodic multisequence over the finite field $F_q = \{\alpha_0 = 0, \alpha_1, ..., \alpha_{q-1}\}$ of q elements. The joint linear complexity of this multisequence S is defined as the smallest integer $L \geq 0$ for which there exists coefficients $d_1, d_2, ..., d_L$ in F_q such that $s_N^{(h)} + d_1 s_{N-1}^{(h)} + ... + d_L s_{N-L}^{(h)} = 0$ for each $0 \leq h \leq M-1$ and for every $N \geq L$.

An m fold N periodic multisequence S can be interpreted as an $m \times N$ matrix over F_q. For defining the k-error joint linear complexity of multisequences, we need the following definition of term distance [6].

1.1 Definition 1

Let $S = (S^{(0)}, S^{(1)}, ..., S^{(m-1)})$ and $T = (T^{(0)}, T^{(1)}, ..., T^{(m-1)})$ be two m fold N periodic multisequences over F_q .We define the term distance $\delta_T(S,T)$ between S and T as the number of entries in S that are different from the corresponding entries in T.

1.2 Definition 2

Let $S = (S^{(0)}, S^{(1)}, ..., S^{(m-1)})$ be an m fold N periodic multisequences over F_q. For an integer k ($0 \leq k \leq mN$), the k-error joint linear complexity $L_{N,k}(S)$ of S is defined as the smallest possible joint linear complexity obtained by changing *k* or fewer terms of *S* in its first period of length *N* and then continuing the changes periodically with period *N*. In other words

$$L_{N,k}(S) = \min_T L(T) \tag{1}$$

where the minimum is taken over all *m* fold *N* periodic sequence *T* over F_q with term distance $\delta_T(S,T) \leq k$.

2 Algorithm for Computing the *k*-error Joint Linear Complexity and Error Multisequence

Let $S = (S^{(0)}, S^{(1)}, ..., S^{(m-1)})$ be an *m* fold *N* periodic multisequences over F_q. Let $N = p^n = pM$ and we write one period of the multisequence *S* as

$$S_N = (S(0)_M, S(1)_M, ..., S(p-1)_M)$$

where

$$S(j)_M = (S^{(0)}(j)_M, S^{(1)}(j)_M, ..., S^{(m-1)}(j)_M)$$

with

$$S^{(h)}(t)_M = (S^{(h)}_{tM+1}, S^{(h)}_{tM+2}, ..., S^{(h)}_{(t+1)M}), \quad 0 \leq t \leq p-1, 0 \leq h \leq m-1.$$

Let $a = (a_{0,} a_1, ..., a_{p-1}) \in F_q^{\ p}$. Define a function F_u on $F_q^{\ p}$ for $0 \leq u \leq p-1$ as follows [14]

$$F_u(a) = \sum_{t=0}^{p-u-1} \binom{p-t-1}{u} a_t \tag{2}$$

Now define a multisequence $b_{M,u}$, $0 \leq u \leq p-1$ where each single sequence is of length *M* as

$$b_{M,u} = (b^{(0)}_{M,u}, b^{(1)}_{M,u}, ..., b^{(m-1)}_{M,u})$$

where

$$b^{(j)}_{M,u} = F_u(S^{(0)}(j)_M, S^{(1)}(j)_M, ..., S^{(p-1)}(j)_M) \tag{3}$$

and F_u is computed component wise.

For the computation of *k*-error joint linear complexity of *S*, we are forcing the value of ω as large as possible in the algorithm, so that $(p\text{-}\omega)M$ becomes as small as possible in Step II(4) of our algorithm, under the assumption that the necessary and sufficient condition for minimum number of changes in the original multisequence is less than or equal to *k*, obtaining the minimal case ω. This criterion can be achieved with introduction of the cost matrix $A_N = (A^{(0)}_N, A^{(1)}_N, ..., A^{(m-1)}_N)$ where $A^{(h)}_N$ is a matrix of size $q \times N$ defined as

$$A_N^{(h)} = \left\lfloor A^{(h)}(l,j)_N \right\rfloor, 0 \le l \le q-1, 0 \le h \le m-1, 0 \le j \le N-1 \tag{4}$$

Initially the cost matrix is defined as $A_N^{(h)} = \left\lfloor A^{(h)}(l,j)_N \right\rfloor$ with

$$A^{(h)}(l,j)_N = \begin{cases} 0 & if\ \alpha_l = s_j^{(h)} \\ 1 & if\ \alpha_l \ne s_j^{(h)} \end{cases} \tag{5}$$

Proposed algorithm has *n* rounds. In each round, a multisequence $S_M = (S^{M(0)}, S^{M(1)}, \ldots, S^{M(m-1)})$ where $S^{M(h)} = (s_0^{M(h)}, s_1^{M(h)}, \ldots, s_{M-1}^{M(h)})$, with each sequence of length *M* is computed from a multisequence S_{pM} with each sequence of length pM where $M = p^{n-r}, 1 \le r \le n$. Also new cost matrices $\left(A_M^{(h)}\right)$ with $A_M^{(h)} = [A^{(h)}(l,j)_M]$ are computed in each step where $A^{(h)}(l,j)_M$ is the minimum number of changes required in the original sequence $S^{(h)}$ of length *N* for changing $s_j^{M(h)}$ to $s_j^{M(h)} + \alpha_l$ in the sequence $S^{M(h)}$ without altering the previous results.

Let $CB_{(M)}^{(h)}$ denote the costs of $b_{M,u}^{(h)}, 0 \le u \le p-2, 0 \le i \le M-1, 0 \le h \le m-1$ and it is given by $CB_{(M)}^{(h)} = \left\lfloor \lambda^{(h)}(u,i)_M \right\rfloor$ where $\lambda^{(h)}(u,i)_M$ is the minimum number of changes in $S^{N(h)}$ necessary and sufficient for making

$$b^{(h)}(0)_{M,u} = b^{(h)}(1)_{M,u} = \ldots = b^{(h)}(u)_{M,u} = 0,\ 0 \le i \le M-1.$$

The total cost of $b_{M,u}^{(h)}$ is defined as $TB_{M,u}^{(h)} = \sum_{i=0}^{M-1} \lambda^{(h)}(u,i)_M$ with

$$\lambda^{(h)}(u,i)_M = \min\left\{ \sum_{j=1}^{p-1} A^{(h)}(e_j^{(h)}, jM+i)_{pM} \,\middle|\, e^{(h)} \in D_j^{(h)}(M,u) \right\} \tag{6}$$

where

$$D_j^{(h)}(M,u) = \left\{ e^{(h)} \,\middle|\, F_t(e^{(h)} + b_j^{(h)}(M,t)) \right\} = 0, 0 \le t \le u \tag{7}$$

is the set of all $e^{(h)}$ which can make $b^{(h)}(0)_{M,u} = b^{(h)}(1)_{M,u} = \ldots = b^{(h)}(i)_{M,u} = 0$. Now define

$$TB_{M,u} = \sum_{h=1}^{M} TB_{M,u}^{(h)}, 0 \le u \le p-2 \tag{8}$$

Using $TB_{M,u}$ and the value *k* of allowed term changes, choose the case ω as large as possible to make the increment $(p\text{-}\omega)M$ to the *k*-error joint linear complexity of *S* as minimum as possible in Step II(4) . If we can force case ω to happen, then the change vector of S_M is recorded as $(VC_0^{(0)}(M), VC_0^{(1)}(M), \ldots, VC_0^{(m-1)}(M))$, where

$$VC_0^{(h)}(M) = (V^{(h)}(0,0)_M, V^{(h)}(0,1)_M, \ldots, V^{(h)}(0, pM-1)_M) \tag{9}$$

such that

$$\sum_{j=0}^{p-1} A^{(h)}(V^{(h)}(0, jM+i)_M, jM+i)_{pM} = \begin{cases} \lambda^{(h)}(p-\omega-1)_M, if\ 1 \le \omega \le p-1 \\ \min\left\{ \sum_{j=0}^{p-1} A^{(h)}(e_j^{(h)}, jM+i)_{pM} \middle| e^{(h)} \in F_q^p \right\}, \\ if\ \omega = p \end{cases} \tag{10}$$

Here $V^{(h)}(0, jM+i)_M$ computes the change value of $S^{pM(h)}$ with the minimum number of changes in the sequences $S^{(h)}$ so that the case ω to happen is not getting altered in the algorithm. We also computes the change matrix $V_M^{(h)} = \lfloor V^{(h)}(l,i)_M \rfloor$ such that

$$A^{(h)}(l,i)_M = \sum_{j=0}^{p-1} A^{(h)}(V^{(h)}(l, jM+i)_{pM} \tag{11}$$

During the computation, the values are changed as

$$A^{(h)}(l,i)_M = \min\left\{ \sum_{j=0}^{p-1} A^{(h)}(e_j^{(h)}, jM+i)_{pM} \middle| e^{(h)} \in \hat{D}^{(h)}(l,i)_{M,\omega} \right\} \tag{12}$$

where

$$\hat{D}^{(h)}(l,i)_{M,j} = \left\{ \begin{array}{l} e^{(h)} \middle| F_s(e_0^{(h)}, e_1^{(h)}, \ldots., e_{p-1}^{(h)}) + b^{(h)}(s,i)_M = 0,\ 0 \le s\ p-\omega-1\ and \\ F_{p-\omega}(e_0^{(h)}, e_1^{(h)}, \ldots, e_{p-1}^{(h)}) = F_{p-\omega}(V^{(h)}(0,i)_M, \ldots, V^{(h)}(0,(p-1)M+i)_M) + \alpha_l \end{array} \right\}, \tag{13}$$
$$if\ 1 \le \omega \le p-1$$

and

$$\hat{D}^{(h)}(l,i)_{M,p} = \left\{ e^{(h)} \middle| F_0(e_0^{(h)}, e_1^{(h)}, \ldots, e_{p-1}^{(h)}) = F_0(V^{(h)}(0,i)_M, \ldots, V^{(h)}(0,(p-1)M+i)_M) + \alpha_l \right\}, \tag{14}$$
$$\text{if } \omega = p$$

and

$$S_M^{(h)} = F_{p-\omega}(S^{(h)}(0)_{pM} + VC_0^{(h)}(M), \ldots, S^{(h)}(p-1)_{pM} + VC_{p-1}^{(h)}(M)) \tag{15}$$

with

$$VC_j^{(h)}(M) = (V^{(h)}(0, jM)_M, V^{(h)}(0, jM+1)_M, \ldots, V^{(h)}(0,(j+1)M-1)_M)$$
$$for\ 0 \le j \le p-1 \tag{16}$$

For the computation of error multisequence e, we compute the error matrix as

$$E(M) = (E^{(h)}(l,i)_M), 0 \le h \le m-1, 0 \le i \le N-1, 0 \le j \le q-1 \tag{17}$$

where $E^{(h)}(l,i)_M$ is defined as follows. For changing $s_i^{M(h)}$ to $s_i^{M(h)} + \alpha_l$ under the situation that the happening of case ω at M^{th} step is not altered in the algorithm, we can change elements $s^{N(h)}_{jM+i}$ by $E^{(h)}(l, jM+i)$ in the original sequence $S^{(h)}$. But from equation (11) we get

$$E^{(h)}(l, pyM+jM+i)_M = E^{(h)}(V^{(h)}(l, jM+i)_M, pyM+jM+i)_{pM} \quad , \qquad (18)$$

$$0 \le l \le q-1, 0 \le j \le p-1, 0 \le h \le m-1, 0 \le y \le \frac{N}{pM}-1, 0 \le i \le M-1$$

For each $0 \le h \le m-1$ the error sequences are initialized as $E^{(h)}(l,i)_N = \alpha_l$. Now we are presenting the algorithm.

Algorithm:

```
I. (a) Initialize $N = p^n$ , $S = S_N = (S(0)_N, S(1)_N, ..., S(p-1)_N)$ , $M = N,\ L_{N,k} = 0$
   (b) Cost matrix initialization, $A^{(h)}_N = \lfloor A^{(h)}(l,j)_N \rfloor$ using equation (5)
   (c) Error matrix initialization, $E(N) = (E^{(h)}(l,i)_N)\ with\ E^{(h)}(l,i)_N = \alpha_l$ using
       equation (17)
II. Round $r$ : for $1 \le r \le n$
    1. $M = \frac{M}{p}$
    2. For $0 \le h \le m-1$ and $1 \le u \le p-2$
        (i) Compute $b^{(h)}_{M,u}$ from $S(i)_{pM}$ using equation (3)
        (ii) Compute $TB^{(h)}_{M,u}$ from $S(i)_{pM}$ and $A^{(h)}_M$ using equation (6)
        (iii) Compute $TB_{M,u}$ using equation (8)
    3. Select one of the following $p$ cases
        (i) Case 1 : $if\ k < TB_{M,0}$
        (ii) Case $\omega$: if $TB_{M,\omega-2} \le k \le TB_{M,\omega-1}$, $2 \le \omega \le p-1$
        (iii) Case $p$ : if $TB_{M,p-2} \le k$
    4. If Case $\omega$: $L_{N,k} = L_{N,k} + (p-\omega)M$
    5. If Case $\omega$ : $\omega$ = 1 or $p$ ,then $S^{(h)}_M = b^{(h)}_{M,\omega-1}$ and $A^{(h)}_M = A^{(h)}_{M,\omega}$ ,
       $0 \le h \le m-1$ using equation (12)
    6. If Case $\omega$: $2 \le \omega \le p-1$
       For $0 \le h \le m-1$,
            if $TB^{(h)}_{M,\omega-2} < TB^{(h)}_{M,\omega-1}$ then $S^{(h)}_M = b^{(h)}_{M,\omega-1}$ and $A^{(h)}_M = A^{(h)}_{M,\omega}$ ,
            $0 \le h \le m-1$ using equation (12)
```

else $S_M^{(h)} = 0$, $A_M^{(h)} = 0$ and $k = k - TB_{M,w-2}^{(h)}$

Compute the error matrix $E(M)$ from $E(pM)$ and $V_M^{(h)}$ using equation (18)

III. If $\sum_{h=0}^{m-1} A^{(h)}(l_t,1)_1 \le k$ and $S_1 = (\alpha_{l_0}, \alpha_{l_1}, ..., \alpha_{l_{m-1}})$ for $0 \le l_t \le q-1$, then $L_{N,k}(S) = L_{N,k}$ and $e = (e^{(h)})$ *where* $e^{(h)} = (E^{(h)}(t,0)_1, E^{(h)}(t,1)_1, ...,$ $E^{(h)}(t, N-1)_1)$ where e is an error multisequence

IV. Else if $\sum_{h=0}^{m-1} A^{(h)}(l_t,1)_1 > k$ then $S_1 = (\alpha_{l_0}, \alpha_{l_1}, ..., \alpha_{l_{m-1}})$, $L_{N,k}(S) = L_{N,k} + 1$ and $e = (e^{(h)})$ *where* $e^{(h)} = (E^{(h)}(0,0)_1, E^{(h)}(0,1)_1, ..., E^{(h)}(0, N-1)_1)$ where e is an error multisequence

V. The final $L_{N,k}(S)$ is the k-error Joint Linear Complexity of given m fold N periodic multisequence and e is an error multisequence

From the above discussion we can see that the correctness of this algorithm follows from that of the Generalized Stamp Martin Algorithm [2]. The time complexity of this algorithm is m times that of the time complexity of Generalized Stamp Martin Algorithm when applied on a single sequence.

3 Conclusion

There is no efficient algorithm for finding the error multisequence of a prime power periodic multisequence. In this paper we derived an algorithm for finding the k-error joint linear complexity and an error multisequence of an m fold prime power periodic multisequence over F_q . Finding the error joint linear complexity spectrum and its properties of periodic multisequences over a finite field is also of related interest.

References

1. Ding, C., Xiao, G., Shan, W. (eds.): The Stability Theory of Stream Ciphers. LNCS, vol. 561. Springer, Heidelberg (1991)
2. Kaida, T.: On Algorithms for the k-Error Linear Complexity of Sequences over *GF(p^m) with Period* p^n, Ph. D. Thesis, Kyusu Institute of Tech. (1999)
3. Kaida, T., Uehara, S., Imamaura, K.: An Algorithm for the k-Error Linear Complexity of Sequences over GF(p^m) with period p^n, p a prime. Information and Computation 151(1-2), 137–147 (1999)
4. Kaida, T.: On the generalized Lauder Paterson algorithm and profiles of the k-error linear complexity for exponent periodic sequences. In: Helleseth, T., Sarwate, D., Song, H.-Y., Yang, K. (eds.) SETA 2004. LNCS, vol. 3486, pp. 166–178. Springer, Heidelberg (2005)

5. Lauder, A.G.B., Paterson, K.G.: Computing the Error Linear Complexity Spectrum of a Binary Sequence of Period 2^n. IEEE Transactions on Information Theory 49(1), 273–281 (2003)
6. Meidl, W.: Discrete Fourier Transform, Joint Linear Complexity and Generalised Joint Linear Complexity of Multisequences. In: Helleseth, T., Sarwate, D., Song, H.-Y., Yang, K. (eds.) SETA 2004. LNCS, vol. 3486, pp. 101–112. Springer, Heidelberg (2005)
7. Meidl, W., Niederreiter, H., Venkateswarlu, A.: Error linear complexity Measures for Multisequences. Journal of Complexity 23(2), 169–192 (2007)
8. Meidl, W., Niederreiter, H.: The expected value of the joint linear complexity of periodic multisequences. Journal of Complexity 19(1), 61–72 (2003)
9. Niederreiter, H.: Linear Complexity and Related Complexity Measures for Sequences. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 1–17. Springer, Heidelberg (2003)
10. Sethumadhavan, M., Sindhu, M., Srinivasan, C., Kavitha, C.: An algorithm for k-error joint linear complexity of binary multisequences. Journal of Discrete Mathematical Sciences & Cryptography 11(3), 297–304 (2008)
11. Sethumadhavan, M., Yogha Laxmie, C., Vijaya Govindan, C.: A construction of p-ary balanced sequence with large k-error linear complexity. Journal of Discrete Mathematical Sciences and Cryptography 9(2), 253–261 (2006)
12. Sindhu, M., Sethumadhavan, M.: Linear Complexity Measures of Binary Multisequences. International Journal of Computer Applications 16, 6–10 (2013)
13. Stamp, M., Martin, C.F.: An Algorithm for the k-Error Linear Complexity of Binary Sequences with Period 2^n. IEEE Trans. Inf. Theory 39, 1398–1407 (1993)
14. Venkateswarlu, Studies on error linear complexity measures for multisequences, PhD thesis (2007)