Analysis of the Linear Complexity in Pseudorandom Sequence Generators*

Amparo Fúster-Sabater

Information Security Institute, C.S.I.C.
Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es

Abstract. In this paper, binary sequences generated by nonlinearly filtering maximal length sequences are studied. Specifically, the parameter linear complexity of the filtered sequences has been considered and analyzed. In fact, a method of computing all the nonlinear filters that generate sequences with a cryptographically large linear complexity has been developed. The procedure is based on the concept of equivalence classes of nonlinear filters and on the addition of filters from different classes. Three distinct representations of nonlinear filters have been systematically addressed. The method completes the class of nonlinear filters with guaranteed linear complexity found in the cryptographic literature.

Keywords: pseudorandom sequence, linear complexity, encryption function, nonlinear filter, cryptography.

1 Introduction

Sequence generators based on Linear Feedback Shift Registers (LFSR) are very commom procedures to generate pseudorandom sequences for multiple applications: computer simulation, circuit testing, error-correcting codes or cryptography (stream ciphers).

The encryption procedure in stream ciphers tries to imitate the mythic *one-time pad cipher* [1] that remains as the only known perfectly secure or absolutely unbreakable cipher. This encryption procedure is designed to generate from a short key a long sequence (*keystream sequence*) of seemingly random bits. Some of the most recent designs in stream ciphers can be found in [2, 3]. Typically, a stream cipher consists of a keystream generator whose output sequence is XORed with the plaintext (in emission) in order to obtain the ciphertext or with the ciphertext (in reception) in order to recover the original plaintext. References [4–7] provide a solid introduction to the study of stream ciphers.

Most keystream generators are based on maximal-length LFSRs [8] whose output sequences, the so-called m-sequences, are combined in a nonlinear way

* Research partially supported by CDTI (Spain) under Project Cenit- HESPERIA as well as by Ministry of Science and Innovation and European FEDER Fund under Project TIN2011-25452/TSI.

B. Murgante et al. (Eds.): ICCSA 2013, Part V, LNCS 7975, pp. 407–420, 2013.

(by means of nonlinear filters, nonlinear combinators, irregularly decimated generators, typical elements from block ciphers, etc) to produce sequences of cryptographic application. Desirable properties for such sequences can be enumerated as follows:

1. Long Period
2. Good statistical properties
3. Large Linear Complexity (LC).

One general technique for building a keystream generator is to use a nonlinear filter, i.e. a nonlinear function applied to the stages of a single maximal-length LFSR. That is the output sequence is generated as the image of a nonlinear Boolean function F in the LFSR stages. Period and statistical properties of the filtered sequences are characteristics deeply studied in the literature, see [9], [10] and the references above mentioned. In addition, such sequences have to pass all 19 DIEHARD tests [11] to be accepted as cryptographic sequences.

Regarding the third requirement, linear complexity of a sequence is defined as the amount of known sequence necessary to reconstruct the entire sequence. In cryptographic terms, LC must be as large as possible in order to prevent the application of the Berlekamp-Massey algorithm [12]. A recommended value for LC is about half the sequence period. Although several contributions to the linear complexity of nonlinearly filtered sequences can be found in [6], [13] and [14], the problem of determining the exact value of the linear complexity attained by any nonlinear filter is still open [15]. For an efficient calculation of Vandermonde matrices, the interested reader is referred to [16–18]

In this paper, a method of computing all the nonlinear filters applied to a LFSR with $LC \geq \binom{L}{k}$ (where L is the LFSR length and k the order of the filter) has been developed. The procedure is based on the concept of equivalence classes of nonlinear filters and on the handling of such filters from different classes. No restriction is imposed on the parameters of the nonlinear filtering function. The method completes the families of nonlinear filters with guaranteed LC given in [6].

The paper is organized as follows. Basic concepts and specific notation is introduced in Section 2. Three different representations of nonlinear filters are given in Sections 3 as well as an equivalence relationship for nonlinear filters is defined in Section 4. The construction of all possible filters preserving the cosets of weight k is developed in Section 5. Discussion on numerical features and an example is given in Section 6. Finally, conclusions in Section 7 end the paper.

2 Basic Concepts and Notation

Specific notation and different basic concepts are introduced as follows:

A *m-sequence*. Let $\{s_n\}$ be the binary output sequence of a maximal-length LFSR of L stages, that is a LFSR whose characteristic polynomial $P(x) = \sum_{j=0}^{L} p_j\, x^j$ with $p_j \in \{0, 1\}$ is primitive of degree L, see [6], [8]. In that case, the

output sequence is a m-sequence of period $2^L - 1$. Moreover, $\{s_n\}$ is completely determined by the LFSR initial state and the characteristic polynomial $P(x)$. The sequence $\{s_n\}$ satisfies the linear recursion:

$$\sum_{j=0}^{L} p_j \, s_{n+j} = 0,$$

that allows one to express any term of the sequence as a linear combination of the previous L terms.

The roots of $P(x)$ are α^{2^i} $(i = 0, 1, \ldots, L-1)$ where α is a primitive element in $GF(2^L)$ that is an extension of the binary field $GF(2)$ with 2^L elements [19]. Any generic element of the sequence, s_n, can be written in terms of the roots of $P(x)$ as:

$$s_n = Tr(C\,\alpha^n) = \sum_{j=0}^{L-1} (C\alpha^n)^{2^j}, \quad n \geq 0 \tag{1}$$

where $C \in GF(2^L)$. Furthermore, the $2^L - 1$ nonzero choices of C result in the $2^L - 1$ distinct shifts of the same m-sequence. If $C = 1$, then $\{s_n\}$ it is said to be in its *characteristic phase*.

Nonlinear filter. It is a Boolean function $F(x_0, x_1, \ldots, x_{L-1})$ in L variables of degree k. For a subset $A = \{a_0, a_1, \ldots, a_{r-1}\}$ of $\{0, 1, \ldots, L-1\}$ with $r \leq k$, the notation $x_A = x_{a_0}\, x_{a_1} \ldots x_{a_{r-1}}$ is used. The Boolean function can be written as [20]:

$$F(x_0, x_1, \ldots, x_{L-1}) = \sum_{A} c_A \, x_A, \tag{2}$$

where $c_A \in \{0, 1\}$ and the summation is taken over all subsets A of $\{0, 1, \ldots, L-1\}$.

Filtered sequence. The sequence $\{z_n\}$ is the keystream or output sequence of the nonlinear filter F applied to the L stages of the LFSR. The keystream bit z_n is computed by selecting bits from the m-sequence such that

$$z_n = F(s_n, s_{n+1}, \ldots, s_{n+L-1}).$$

Cyclotomic coset. Let Z_{2^L-1} denote the set of integers $[1, \ldots, 2^L - 1]$. An equivalence relation R is defined on its elements $q_1, q_2 \in Z_{2^L-1}$ such as follows: $q_1 R\, q_2$ if there exists an integer j, $0 \leq j \leq L-1$, such that

$$2^j \cdot q_1 = q_2 \bmod 2^L - 1.$$

The resultant equivalence classes into which Z_{2^L-1} is partitioned are called the *cyclotomic cosets* mod $2^L - 1$, see [8]. All the elements q_i of a cyclotomic coset have the same number of 1's in their binary representation; this number is called the *coset weight*. The leader element, E, of every coset is the smallest integer in such an equivalence class. Moreover, the cardinal of any coset is L or a proper divisor of L.

Characteristic polynomial of a cyclotomic coset. It is a polynomial $P_E(x)$ defined by $P_E(x) = (x + \alpha^E)(x + \alpha^{2E}) \ldots (x + \alpha^{2^{(r-1)}E})$, where the degree r ($r \leq L$) of $P_E(x)$ equals the cardinal of the cyclotomic coset E.

Characteristic sequence of a cyclotomic coset. It is a binary sequence $\{S_n^E\}$ defined by the expression $\{S_n^E\} = \{\alpha^{En} + \alpha^{2En} + \ldots + \alpha^{2^{(r-1)}En}\}$ with $n \geq 0$. Recall that the previous sequence $\{S_n^E\}$ satisfies the linear recurrence relationship given by $P_E(x)$, see [8], [19]. Moreover, $\{S_n^E\}$ is a decimation of the m-sequence $\{s_n\}$ obtained from such a sequence by taking one out of E terms.

3 Different Representations of Nonlinear Filters

According to the previous section, nonlinear filters can be characterized by means of different representations:

3.1 Algebraic Normal Form (ANF)

The equation (2) describes the ANF of a nonlinear filter $F(s_n, s_{n+1}, \ldots, s_{n+L-1})$. That is F is represented as the sum of distinct products in the variables $(s_n, s_{n+1}, \ldots, s_{n+L-1})$. For each nonlinear filter the ANF representation is unique. The algebraic degree, k, of the Boolean function F is the highest degree of a monomial in F. This representation of Boolean functions is currently used by the designer of nonlinear filters.

3.2 Bit-Wise Sum of the Characteristic Sequences

Now, if all the variables s_{n+j} $(0 \leq j \leq L-1)$ in the ANF representation of F are substituted by their corresponding expressions in (1) and the resulting terms grouped, then the generic element z_n of the filtered sequence $\{z_n\}$ can be written as:

$$\begin{gathered} z_n = F(s_n, s_{n+1}, \ldots, s_{n+L-1}) = \\ C_1\alpha^{E_1 n} + (C_1\alpha^{E_1 n})^2 + \ldots + (C_1\alpha^{E_1 n})^{2^{(r_1-1)}} + \\ C_2\alpha^{E_2 n} + (C_2\alpha^{E_2 n})^2 + \ldots + (C_2\alpha^{E_2 n})^{2^{(r_2-1)}} + \\ \vdots \\ C_N\alpha^{E_N n} + (C_N\alpha^{E_N n})^2 + \ldots + (C_N\alpha^{E_N n})^{2^{(r_N-1)}}, \end{gathered} \tag{3}$$

where r_i is the cardinal of coset E_i, the subindex i ranges in the interval $1 \leq i \leq N$ and N is the number of cosets of weight $\leq k$.

Thus a nonlinear filter $F(s_n, s_{n+1}, \ldots, s_{n+L-1})$ can be represented in terms of the N characteristic sequences $\{S_n^{E_i}\}$ that appear in this sequential decomposition in cosets shown in equation (3).

At this point different features can be pointed out. Note that the i-th row of (3) corresponds to the nth-term of the sequence $\{C_i\alpha^{E_i n} + (C_i\alpha^{E_i n})^2 + \ldots + (C_i\alpha^{E_i n})^{2^{(r_i-1)}}\}$, where the coefficient $C_i \in GF(2^{r_i})$ determines the starting

point of such a sequence. In fact, as long as C_i ranges in its corresponding extension field we shift along the sequence $\{S_n^{E_i}\}$. If the corresponding characteristic polynomial $P_{E_i}(x)$ is a primitive polynomial, then the characteristic sequence $\{S_n^{E_i}\}$ is a m-sequence.

If $C_i = 0$, then $\{S_n^{E_i}\}$ would not contribute to the filtered sequence $\{z_n\}$. In that case, the cyclotomic coset E_i would be degenerate. Linear complexity of the filtered sequence is related to the number of coefficients C_i different from zero as the contribution to LC of any nondegenerate coset equals the cardinal of such a coset.

3.3 A N-tuple of Coefficients

This is a representation very close to the previous one. In fact, a nonlinear filter $F(s_n, s_{n+1}, \ldots, s_{n+L-1})$ can be represented in terms of a N-tuple of coefficients $(C_1, C_2, \ldots, C_N)$ with $C_i \in GF(2^{r_i})$ where each coefficient determines the starting point of the sequence $\{S_n^{E_i}\}$ with reference to its characteristic phase and N denotes, as before, the number of cosets of weight $\leq k$.

In this work, the three representations will be indistinctly used.

4 Equivalence Classes for Nonlinear Filters

The idea of grouping nonlinear filters in equivalence classes for their handling has been already developed in the literature, see [21]. This is the technique followed in this section to design filters with specific properties.

Let G be the set of the kth-order nonlinear filters applied to a LFSR of length L. We are going to group the elements of G producing the filtered sequence $\{z_n\}$ or a shifted version of $\{z_n\}$, notated $\{z_n\}^*$. From equation (3), it is clear that if we substitute C_i for $C_i \cdot \alpha^{E_i}$ $\forall i$, then we will obtain $\{z_{n+1}\}$. In general,

$$C_i \to C_i \cdot \alpha^{jE_i} \quad \forall i \Rightarrow \{z_n\} \to \{z_{n+j}\}.$$

This fact enables us to define an equivalence relationship $\sim$ on the set G as follows: $F_0 \sim F_1$ with $F_0, F_1 \in G$ if

$$\{F_0(s_n, \ldots, s_{n+L-1})\} = \{F_1(s_n, \ldots, s_{n+L-1})\}^*.$$

Therefore, two different nonlinear filters F_0, F_1 in the same equivalence class will produce shifted versions of the same filtered sequence. In addition, it is easy to see that the relation defined above is an equivalence relationship. Making use of the third representation for nonlinear filters (N-tuple of coefficients) in the previous section, we see that the coefficients associated with F_0, F_1, notated $(C_{E_i}^0)$ and $(C_{E_i}^1)$ respectively, satisfy

$$C_{E_i}^1 = C_{E_i}^0 \cdot \alpha^{j\,E_i} \quad \forall i. \tag{4}$$

Clearly, the number of elements in every equivalence class equals the period of the filtered sequence, T, so that in (4) the index j verifies $1 \leq j \leq T-1$.

Definition 1. *Two nonlinear filters F_0 and F_1 in the same equivalence class are consecutive if they satisfy the equation (4) with $j = 1$ or equivalently*

$$F_1(s_n, \ldots, s_{n+L-1}) = F_0(s_{n+1}, \ldots, s_{n+L}).$$

Let $E_1, E_2, \ldots, E_M$ be the leaders of the nondegenerate cosets of weight at most k in $\{z_n\}$ and $r_1, r_2, \ldots, r_M$ their corresponding cardinals. Several results can be pointed out.

Lemma 1. *If p nonlinear filters in the same equivalence class are chosen*

$$(C_{E_i}), (C_{E_i} \cdot \alpha^{q_1 E_i}), (C_{E_i} \cdot \alpha^{q_2 E_i}), \ldots, (C_{E_i} \cdot \alpha^{q_{p-1} E_i}) \tag{5}$$

($q_1, q_2, \ldots, q_{p-1}$ being integers) in such a way that no characteristic polynomial $P_{E_i}(x)$ $(1 \leq i \leq M)$ divides the polynomial

$$Q(x) = (1 + x^{q_1} + \ldots + x^{q_{p-1}}), \tag{6}$$

then the nonlinear filter characterized by the coefficients

$$\tilde{C}_{E_i} = C_{E_i}(1 + \alpha^{q_1 E_i} + \ldots + \alpha^{q_{p-1} E_i}) \ \ (1 \leq i \leq M) \tag{7}$$

preserves the same cosets E_i as those of the filters defined in (5).

Proof. The result follows from the fact that the coefficients of the new nonlinear filter verify

$$\tilde{C}_{E_i} = C_{E_i}(1 + \alpha^{q_1 E_i} + \ldots + \alpha^{q_{p-1} E_i}) \neq 0 \ \ (1 \leq i \leq M)$$

as no α^{E_i} is a root of $Q(x)$. □

Therefore an easy way to guarantee the presence of all the cosets E_i in the new filter is just summing $p \leq r_{min}$ consecutive nonlinear filters in the same equivalence class (r_{min} being the least cardinal of all the cosets E_i) as $deg\, Q(x) < deg\, P_{E_i}(x)$ $(1 \leq i \leq M)$.

Lemma 2. *The sum of nonlinear filters satisfying the conditions of Lemma 1 gives rise to a new nonlinear filter in a different equivalent class.*

Proof. We proceed by contradiction. Suppose that the new filter belongs to the same equivalence class. Then,

$$\tilde{C}_{E_i} = C_{E_i}(1 + \alpha^{q_1 E_i} + \ldots + \alpha^{q_{p-1} E_i}) = C_{E_i} \cdot \alpha^{q_1 E_i} \ \ \forall i. \tag{8}$$

For simplicity reasons, assume that coset E_i=coset 1. Therefore, according to (8)

$$(1 + \alpha^{q_1} + \ldots + \alpha^{q_{p-1}}) = \alpha^j$$

and

$$(1 + \alpha^{q_1 E_i} + \ldots + \alpha^{q_{p-1} E_i}) = \alpha^{j E_i} \ \ (2 \leq i \leq M).$$

Thus, it follows that

$$(1 + \alpha^{q_1} + \ldots + \alpha^{q_{p-1}})^{E_i} = (1 + \alpha^{q_1 E_i} + \ldots + \alpha^{q_{p-1} E_i}) \ \ (2 \leq i \leq M).$$

Nevertheless, it is a well known fact that in $GF(2^L)$ this equality only holds for E_i of the form 2^m (i.e. the elements of coset 1) but not for the leaders of any coset $E_i \neq$ coset 1. □

4.1 Practical Design of Nonlinear Filters with Guaranteed Linear Complexity

According to the previous results, a method of constructing nonlinear filters with large linear complexity can be stated as follows:

1. Start from a nonlinear filter F_0 whose number of nondegenerate cosets is known to be large, for instance, a nonlinear filter with a unique term of order k and equidistant stages [6], which preserves all the cosets of weight k.
2. Sum two consecutive nonlinear filters in this class $F_0 + F_1$ in order to jump into a different equivalence class preserving all the cosets E_i of the previous class.
3. Repeat step 2 in order to generate as many different equivalence classes as desired.

If at least one of the nondegenerate cosets has as characteristic sequence a m-sequence, then we can jump into 2^L-1 different equivalence classes before coming back to the original class. In this way, the sum operation $F_0 + F_1$ is a simple source of generation of nonlinear filters that preserve the k-weight cosets.

5 Construction of All Possible Nonlinear Filters with Cosets of Weight k: A Specific Algorithm

In order to generate all the nonlinear filters with guaranteed cosets of weight k, we start from a filter with a unique term product of k equidistant phases of the form:

$$F_0(s_n, s_{n+1}, \ldots, s_{n+L-1}) = s_n s_{n+\delta} \ldots s_{n+(k-1)\delta} \tag{9}$$

with $1 \leq k \leq L$ and $gcd(\delta, 2^L - 1) = 1$. According to [6], the sequence obtained from this type of filters includes all the k-weight cosets.

Given F_0 in ANF, the computation of its N_k-tuple is carried out via the root presence test described in [6]. That is the computation of Vandermonde determinants for which there is a simple formula. Next, the N_k-tuple representations for $F_1 = \mathcal{S}(F_0)$ and $F_0 + F_1$ are easily computed too. The key idea in this construction method is shifting the filter $F_0 + F_1$ through its equivalence class and summing it with F_0 in order to cancel the successive components of its N_k-tuple.

The final result is:

1. A set of N_k basic filters of the form $(0, 0, \ldots, d_i, \ldots, 0, 0)$ $(1 \leq i \leq N_k)$ with $d_i \in GF(2^L), d_i \neq 0$.
2. Their corresponding ANF representations.

The combination of all these basic filters with d_i $(1 \leq i \leq N_k)$ ranging in $GF(2^L)$ (with the corresponding ANF representations) gives rise to all the possible terms of order k that preserve the cosets of weight k. Later, the addition of terms of order $< k$ in ANF permits the generation of all the nonlinear filters of order k that guarantee a linear complexity $LC \geq \binom{L}{k}$.

An algorithm for computing the basic nonlinear filters with cosets of weight k is depicted in Fig. 1. The employed notation is now introduced:

```
Input: One nonlinear filter with guaranteed k-weight cosets,
       F_0(s_n, ..., s_{L-1}) → (C_1^0, ..., C_i^0, ..., C_{N_k}^0),

 Compute F_1 = S(F_0(s_n, ..., s_{L-1})) → (C_1^1, ..., C_i^1, ..., C_{N_k}^1),
 for j = N_k to 2 do
     Step 1: Addition of the two filters: F_0 + F_1 = F_01 →
               (C_i^0) + (C_i^1) = (C_i^2)
     Step 2: Comparison F_0 : F_01
               (C_1^0, ..., C_i^0, ..., C_{N_k}^0) : (C_1^2, ..., C_i^2, ..., C_{N_k}^2)
     Step 3: Shifting of (C_1^2, ..., C_i^2, ..., C_{N_k}^2) through its equivalence class
               until C_j^2 = C_j^0
               (C_1^2, ..., C_j^2, ..., C_{N_k}^2) → (C_1^3, ..., C_j^0, ..., 0)
     Step 4: Addition
               (C_i^0) + (C_i^3) = (C_i^4) = (C_1^4, ..., 0, ..., 0)
               keep (I_i^{j-1}) = (C_i^4)
     Step 5: Substitution
               (C_i^0) ← (C_i^4)
 end for

 (B_i^1) = (I_i^1); Display the ANF.
 for j = 2 to N_k do
     Step 6: Comparison (B_i^1), ..., (B_i^{j-1}) : (I_i^j)
        for l = 1 to j - 1 do
            Step 7: Shifting of (B_i^l)
                      until B_l^l = I_l^j
        end for
     Step 8: Addition
               sum_{l=1}^{j-1} (B_i^l)' + (I_i^j) = (B_i^j)
               Display the ANF.
 end for

Output: N_k basic filters (B_i^j) = (0, 0, ..., d_j, ..., 0) to generate
        all the nonlinear filters preserving the k-weight cosets
        and their ANF representations.
```

Fig. 1. Pseudo-code of the algorithm to generate N_k basic filters

- F_0 is the initial filter with guaranteed cosets of weight k. Its N_k-tuple coefficient representation can be written as:

$$F_0 = (C_1^0, C_2^0, \ldots, C_{N_k}^0) = (C_i^0) \ \ (1 \leq i \leq N_k).$$

- $F_1 = \mathcal{S}(F_0)$ is the consecutive filter in the same equivalence class. Its N_k-tuple coefficient representation can be written as:

$$F_1 = (C_1^1, C_2^1, \ldots, C_{N_k}^1) = (C_i^1) \ \ (1 \leq i \leq N_k).$$

- $F_{01} = F_0 + F_1$ is a new filter in a different equivalence class whose N_k-tuple coefficient representation is:

$$F_{01} = (C_1^2, C_2^2, \ldots, C_{N_k}^2) = (C_i^2) \ \ (1 \leq i \leq N_k).$$

– The filter (C_i^2) ranges in its equivalence class until the j-th component $(C_j^2) = (C_j^0)$. The resulting filter is:

$$(C_1^3, C_2^3, \ldots, C_j^0, \ldots, 0) = (C_i^3) \ \ (1 \leq i \leq N_k),$$

where $C_l^3 = 0$ for $(j+1 \leq l \leq N_k)$.

– The filter (C_i^4) is the sum of:

$$(C_i^0) + (C_i^3) = (C_i^4) = (C_1^4, C_2^4, \ldots, 0, \ldots, 0) \ \ (1 \leq i \leq N_k),$$

where $C_l^4 = 0$ for $(j \leq l \leq N_k)$.

– (I_i^j) is an intermediate filter where (C_i^4) is stored for the corresponding value of the index j.

$$(I_1^j, I_2^j, \ldots, I_{N_k}^j) = (I_i^j) \ \ (1 \leq i \leq N_k).$$

– (B_i^j) is a basic filter whose components are 0 except for the j-th component $d_j \neq 0$.

$$(B_1^j, B_2^j, \ldots, B_{N_k}^j) = (B_i^j) = (0, 0, \ldots, d_j, \ldots, 0) \ \ (1 \leq i \leq N_k).$$

The symbol $(B_i^j)'$ means that the initial filter (B_i^j) has been shifted through its equivalence class.

Next a pseudo-code of the programmed algorithm is given in Fig. 1.

6 Discussion

Regarding the previous sections, distinct considerations must be taken into account.

Recall that the construction method described in the previous section to compute the basic filters $(0, 0, \ldots, d_i, \ldots, 0, 0), d_i \neq 0$ involves very simple operations:

– Sum operation: that is reduced to a logic sum of filters for the ANF representation or to a sum of elements of the extended field $GF(2^L)$ that expressed in binary representation is just an exclusive OR operation.
– Shifting operation through an equivalence class: that means an increment by 1 in all the indexes in the ANF representation or the multiplication of powers of α by their corresponding factors α^{E_i} in the N-tuple representation that just means the addition of exponents.

Consequently, the efficiency of the computation method is quite evident. In brief, we provide one with the complete class of nonlinear filters with $LC \geq \binom{L}{k}$ at the price of minimal computational operations.

In the case that the presence of more cosets of weight $\neq k$ were guaranteed, the procedure here described continues being applicable just enlarging the coefficient vector to new components corresponding to those new guaranteed cosets in the N-tuple representation.

Let us now see an illustrative example.

No.	A.N.F.	Coeff.
0	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(\alpha^5, 0)$
1	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_4$	$(\alpha^{12}, 0)$
2	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(\alpha^{19}, 0)$
3	$s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{26}, 0)$
4	$s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_3s_4$	$(\alpha^{2}, 0)$
5	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(\alpha^{9}, 0)$
6	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4$	$(\alpha^{16}, 0)$
7	$s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(\alpha^{23}, 0)$
8	$s_0s_1s_2 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{30}, 0)$
9	$s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3$	$(\alpha^{6}, 0)$
10	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_1s_2s_3 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{13}, 0)$
11	$s_0s_1s_2 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4$	$(\alpha^{20}, 0)$
12	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_1s_2s_3 \oplus s_1s_3s_4$	$(\alpha^{27}, 0)$
13	$s_0s_1s_2 \oplus s_0s_2s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(\alpha^{3}, 0)$
14	$s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_3$	$(\alpha^{10}, 0)$
15	$s_0s_1s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{17}, 0)$
16	$s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(\alpha^{24}, 0)$
17	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4$	$(1, 0)$
18	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4$	$(\alpha^{7}, 0)$
19	$s_0s_1s_2 \oplus s_0s_2s_3 \oplus s_2s_3s_4$	$(\alpha^{14}, 0)$
20	$s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_3s_4$	$(\alpha^{21}, 0)$
21	$s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_1s_3s_4$	$(\alpha^{28}, 0)$
22	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_3 \oplus s_2s_3s_4$	$(\alpha^{4}, 0)$
23	$s_0s_1s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(\alpha^{11}, 0)$
24	$s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{18}, 0)$
25	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_2s_3s_4$	$(\alpha^{25}, 0)$
26	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_3s_4 \oplus s_2s_3s_4$	$(\alpha, 0)$
27	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(\alpha^{8}, 0)$
28	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{15}, 0)$
29	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{22}, 0)$
30	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(\alpha^{29}, 0)$

Fig. 2. Class of nonlinear filters $(B_i^1) = (\alpha^5, 0)$

6.1 A Numerical Example

Let $(L, k) = (5, 3)$ be a nonlinear filter of third order applied to the stages of a LFSR of length $L = 5$ and primitive characteristic polynomial $P(x) = x^5 + x^3 + 1$ where α is a root of $P(x)$ so that $\alpha^5 = \alpha^3 + 1$. We have $N_3 = 2$ cyclotomic cosets of weight 3: coset 7= $\{7, 14, 28, 25, 19\}$ and coset 11= $\{11, 22, 13, 26, 21\}$. The initial filter with guaranteed cosets of weight 3 is $F_0(s_0, s_1, s_2) = s_0s_1s_2$. The algorithm described in Fig. 1 is applied.

No.	A.N.F.	Coeff.
0	$s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4$	$(0, \alpha^{13})$
1	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_1s_3s_4$	$(0, \alpha^{24})$
2	$s_0s_1s_2 \oplus s_0s_2s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{4})$
3	$s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3$	$(0, \alpha^{15})$
4	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(0, \alpha^{26})$
5	$s_0s_1s_2 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4$	$(0, \alpha^{6})$
6	$s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_1s_2s_3 \oplus s_2s_3s_4$	$(0, \alpha^{17})$
7	$s_0s_1s_2 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{28})$
8	$s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_3s_4$	$(0, \alpha^{8})$
9	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_1s_2s_3 \oplus s_1s_3s_4$	$(0, \alpha^{19})$
10	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_4$	$(0, \alpha^{30})$
11	$s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_1s_2s_3 \oplus s_1s_2s_4$	$(0, \alpha^{10})$
12	$s_0s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{21})$
13	$s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{1})$
14	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{12})$
15	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(0, \alpha^{23})$
16	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(0, \alpha^{3})$
17	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{14})$
18	$s_0s_1s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{25})$
19	$s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(0, \alpha^{5})$
20	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(0, \alpha^{16})$
21	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_4 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{27})$
22	$s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(0, \alpha^{7})$
23	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_2s_3s_4$	$(0, \alpha^{18})$
24	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4 \oplus s_2s_3s_4$	$(0, \alpha^{29})$
25	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_3s_4 \oplus s_1s_3s_4$	$(0, \alpha^{9})$
26	$s_0s_1s_2 \oplus s_0s_1s_4 \oplus s_0s_2s_4 \oplus s_2s_3s_4$	$(0, \alpha^{20})$
27	$s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_3s_4$	$(0, 1)$
28	$s_0s_1s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4$	$(0, \alpha^{11})$
29	$s_0s_1s_2 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_1s_2s_3$	$(0, \alpha^{22})$
30	$s_0s_1s_3 \oplus s_1s_2s_3 \oplus s_1s_3s_4 \oplus s_2s_3s_4$	$(0, \alpha^{2})$

Fig. 3. Class of nonlinear filters $(B_i^2) = (0, \alpha^{13})$

INPUT: The nonlinear filter $F_0(s_0, s_1, s_2) = s_0s_1s_2 \rightarrow (C_i^0) = (\alpha^{20}, \alpha^{13})$
Compute: $F_1(s_0, s_1, s_2) = s_1s_2s_3 \rightarrow (C_i^1) = (\alpha^{20} \cdot \alpha^7, \alpha^{13} \cdot \alpha^{11}) = (\alpha^{27}, \alpha^{24})$
Initialize: $(I_i^2) = (C_i^0) = (\alpha^{20}, \alpha^{13})$
for $j = N_3$ to 2

– Step 1: Addition of two filters $F_0 + F_1 = F_{01}$

$$(C_i^0) + (C_i^1) = (C_i^2)$$
$$(\alpha^{20}, \alpha^{13}) + (\alpha^{27}, \alpha^{24}) = (\alpha^5, \alpha^5)$$

– Step 2: Comparison $F_0 : F_1$

$$(C_i^0) : (C_i^2)$$
$$(\alpha^{20}, \alpha^{13}) : (\alpha^5, \alpha^5)$$

– Step 3: Shifting of (C_i^2) until $(C_2^2) = (C_2^0)$

$$(C_1^2, C_2^2) \rightarrow (C_1^3, C_2^0)$$
$$(\alpha^5, \alpha^5) \rightarrow (\alpha^{27}, \alpha^{13}) = (C_i^3)$$

– Step 4: Addition

$$(C_i^0) + (C_i^3) = (C_i^4)$$
$$(\alpha^{20}, \alpha^{13}) + (\alpha^{27}, \alpha^{13}) = (\alpha^5, 0)$$
$$(I_i^1) = (C_i^4)$$

end for
Introduce $(B_i^1) = (I_i^1) = (\alpha^5, 0)$
for $j = 2$ to N_3

– Step 6: Comparison $(B_i^1) : (I_i^2)$

$$(\alpha^5, 0) : (\alpha^{20}, \alpha^{13})$$

– Step 7: Shifting of (B_i^1) until $(B_1^1) = (I_1^2) = \alpha^{20}$

$$(B_i^1) \rightarrow (B_i^1)'$$
$$(\alpha^5, 0) \rightarrow (\alpha^{20}, 0)$$

– Step 8: Addition

$$(B_i^1)' + (I_i^2) = (B_i^2)$$
$$(\alpha^{20}, 0) + (\alpha^{20}, \alpha^{13}) = (0, \alpha^{13})$$
$$(B_1^2, B_2^2) = (0, \alpha^{13})$$

end for

OUTPUT: $N_3 = 2$ basic nonlinear filters and their corresponding ANF representations.

1. $(B_i^1) = (\alpha^5, 0)$
   ANF: $s_0s_1s_2 \oplus s_0s_1s_3 \oplus s_0s_1s_4 \oplus s_0s_2s_3 \oplus s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_3 \oplus s_1s_2s_4 \oplus s_1s_3s_4$.
2. $(B_i^2) = (0, \alpha^{13})$
   ANF: $s_0s_2s_4 \oplus s_0s_3s_4 \oplus s_1s_2s_4$.

Basic filters (B_i^1) and (B_i^2) range in their corresponding equivalence class (with $2^5 - 1$ filters per class) as it is shown in Fig. 2 and Fig. 3, respectively. Filter (B_i^1) includes a unique coset of weight 3 that is (coset 7) as so does (B_i^2) with (coset 11). None of the filters depicted in the previous figures attains the lower bound $LC \geq \binom{L}{k}$. Nevertheless, summing up each one of the ANF representations in Fig. 2 with every one of the ANF representations in Fig. 3, we get the 31×31 possible combinations of terms of order 3 that guarantee the cosets of weight 3 (coset 7 and coset 11). Next, the addition of terms of order < 3 in ANF representation permits us the generation of all the nonlinear filters of order 3 applied to the previous LFSR that guarantee a linear complexity $LC \geq \binom{5}{3}$.

7 Conclusions

In this paper, a method of computing all the nonlinear dynamical filters applied to a LFSR that guarantee the cosets of weight k has been developed. The procedure is based on the handling of nonlinear filters belonging to different equivalence classes. The method not only includes the nonlinear filters (e.g. filters obtained from equidistance phases or combination of equidistance phases) found in the literature but also it formally completes the class of filters with a guaranteed linear complexity. In brief, an easy way of designing keystream generators for stream cipher purposes has been provided.

References

1. Nagaraj, N.: One-Time Pad as a nonlinear dynamical system. Commun Nonlinear Sci. Numer Simulat. 17, 4029–4036 (2012)
2. eSTREAM, the ECRYPT Stream Cipher Project,The eSTREAM Portfolio (2012), http://www.ecrypt.eu.org/documents/D.SYM.10-v1.pdf
3. Robshaw, M., Billet, O. (eds.): New Stream Cipher Designs. LNCS, vol. 4986. Springer, Heidelberg (2008)
4. Menezes, A.J., et al.: Handbook of Applied Cryptography. CRC Press, New York (1997)
5. Paar, C., Pelzl, J.: Understanding Cryptography. Springer, Heidelberg (2010)
6. Rueppel, R.A.: Analysis and Design of Stream Ciphers. Springer, New York (1986)
7. Tan, S.K., Guan, S.U.: Evolving cellular automata to generate nonlinear sequences with desirable properties. Applied Soft Computing 7, 1131–1134 (2007)
8. Golomb, S.: Shift-Register Sequences, revised edn. Aegean Park Press, Laguna Hills (1982)
9. Fúster-Sabater, A., Caballero-Gil, P., Delgado-Mohatar, O.: Deterministic Computation of Pseudorandomness in Sequences of Cryptographic Application. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2009, Part I. LNCS, vol. 5544, pp. 621–630. Springer, Heidelberg (2009)
10. Fúster-Sabater, A., Caballero-Gil, P.: Chaotic modelling of the generalized self-shrinking generator. Appl. Soft Comput. 11, 1876–1880 (2011)
11. A. Marsaglia, Test of DIEHARD (1998), http://stat.fsu.edu/pub/diehard/
12. Massey, J.L.: Shift-Register Synthesis and BCH Decoding. IEEE Trans. Information Theory 15(1), 122–127 (1969)
13. Caballero-Gil, P., Fúster-Sabater, A.: A wide family of nonlinear filter functions with large linear span. Inform. Sci. 164, 197–207 (2004)
14. Limniotis, K., Kolokotronis, N., Kalouptsidis, N.: On the Linear Complexity of Sequences Obtained by State Space Generators. IEEE Trans. Inform. Theory 54, 1786–1793 (2008)
15. Kolokotronis, N., Limniotis, K., Kalouptsidis, N.: Lower Bounds on Sequence Complexity Via Generalised Vandermonde Determinants. In: Gong, G., Helleseth, T., Song, H.-Y., Yang, K. (eds.) SETA 2006. LNCS, vol. 4086, pp. 271–284. Springer, Heidelberg (2006)
16. Lee, K., O'Sullivan, M.E.: List decoding of Hermitian codes using Gröbner bases. Journal of Symbolic Computation 44(12), 1662–1675 (2009)

17. Respondek, J.S.: On the confluent Vandermonde matrix calculation algorithm. Applied Mathematics Letters 24(2), 103–106 (2011)
18. Respondek, J.S.: Numerical recipes for the high efficient inverse of the confluent Vandermonde matrices. Applied Mathematics and Computation 218(5), 2044–2054 (2011)
19. Lidl, R., Niederreiter, H.: Finite Fields. In: Enciclopedia of Mathematics and Its Applications, 2nd edn., vol. 20. Cambridge University Press, Cambridge (1997)
20. Ronjom, S., Helleseth, T.: A New Attack on the Filter Generator. IEEE Trans. Information Theory 53(5), 1752–1758 (2007)
21. Rønjom, S., Cid, C.: Nonlinear Equivalence of Stream Ciphers. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 40–54. Springer, Heidelberg (2010)