Cryptogr. Commun. (2018) 10:1075–1090
https://doi.org/10.1007/s12095-017-0265-2

Nonlinear vectorial primitive recursive sequences

Sartaj Ul Hasan[1] · **Daniel Panario**[2] · **Qiang Wang**[2]

Received: 30 November 2016 / Accepted: 3 November 2017 / Published online: 15 November 2017

Abstract We discuss nonlinear vectorial primitive recursive sequences. First we consider the nonlinearly filtered multiple-recursive matrix generator for producing pseudorandom vectors based on some nonlinear schemes and give lower bounds for their componentwise linear complexity. Moreover, we obtain certain results concerning the jump multiple-recursive matrix generator and establish that sequences generated by them have better period and componentwise linear complexity as compared to usual multiple-recursive matrix generator sequences. We also include analogous results for transformation shift registers for generating pseudorandom vectors.

Keywords Multiple-recursive matrix generator · Transformation shift register · Linear complexity

Mathematics Subject Classification (2010) 94A55 · 94A60

This article is part of the Topical Collection on *Special Issue on Sequences and Their Applications*

✉ Sartaj Ul Hasan
sartajulhasan@gmail.com

Daniel Panario
daniel@math.carleton.ca

Qiang Wang
wang@math.carleton.ca

[1] Scientific Analysis Group, Defence Research and Development Organisation, Metcalfe House, Delhi 110054, India

[2] School of Mathematics and Statistics, Carleton University, Ottawa, Ontario K1S 5B6, Canada

1 Introduction

Linear feedback shift registers (LFSRs) are used as basic building blocks in most of the modern stream ciphers. These ciphers produce only a single bit per clock and hence are often referred to as bit-oriented ciphers. Moreover, apart from having good cryptographic properties, these ciphers also have very low cost of implementation in hardware. Since the manipulation of several bits is required to produce just a single bit, bit-oriented ciphers are sluggish when it comes to their software implementation.

It is quite natural to ask if we can design feedback shift registers (FSRs) that produce a word in each clock instead of a bit so that software efficiency in various applications such as high speed link encryption may be achieved. This question was stated by Preneel in [21] and in fact, he asked if we can design fast and secure FSRs with the help of the word operations of modern processors and the techniques of parallelism.

Niederreiter [15–18] gave a solution to Preneel's problem even before it was formally stated in the form of his multiple-recursive matrix method for producing pseudorandom vectors, which we will refer to as "multiple-recursive matrix generator (MRMG)" throughout this paper. We remark that multiple-recursive matrix generator is indeed a generalization of matrix congruential generators studied by Franklin [5] and Grothe [10]. Zeng, Han and He [25] introduced the notion of σ-LFSR and suggested a way to further improve the efficiency of MRMG by imposing a restriction on the choice of its coefficient matrices from a special set of matrices that are compatible with word operations of modern processors. It may be noted that the notion of σ-LFSR is exactly the same as Niederreiter's MRMG. We refer to [1, 2, 6, 7] for more on primitive MRMG.

The sequences generated by MRMGs are prone to attacks based on algebraic techniques due to their inherent linearity. Moreover, linear complexity plays a crucial role in determining the security of the keystream generated by MRMGs. The higher linear complexity we have, the better security we achieve. Thus, for all practical purposes, we have to not only destroy the linear structure of MRMGs, but also need to have better linear complexity. One way of doing this is to employ some nonlinear functions on the contents of MRMGs. In fact, one such scheme based on Langford arrangement was discussed in [1]. Motivated by this, we introduce the notion of nonlinearly filtered MRMGs in Section 3 and prove that certain lower bounds for the linear complexity can be guaranteed for the sequences generated by them.

The other way of increasing the linear complexity of MRMG sequences, while still maintaining large period and good statistical properties, is to apply clock control, that is, to irregularly step the MRMG through its successive states. As in the case of regularly clocked LFSRs [13], the regularly clocked MRMGs could also be susceptible to correlation and fast correlation attacks. Thus irregular clocking would help in making the MRMGs immune to correlation attacks. However, as pointed out in [13], key stream generators that use irregular clocking are prone to timing and power attacks. In order to avoid these side-channel attacks while still preserving all the advantages of irregular clocking, we discuss jump multiple-recursive matrix generator in Section 4 and prove that sequences generated by the jump multiple-recursive matrix generator have much better period and linear complexity as compared to usual MRMGs. It may be noted that the notion of word-oriented cascade jump σ-LFSR was introduced in [26], particularly, to study those cascade jump σ-LFSRs whose matrix polynomial has only three terms. Our approach to the jump multiple-recursive matrix generator is much more unified and general. Moreover, our results are completely different from those obtained in [26].

Tsaban and Vishne [24] have introduced the notion of transformation shift register (TSR) for generating pseudorandom vectors that provides yet another solution to Preneel's problem. It turns out that a TSR is indeed a particular case of MRMG. The reader is referred to [3, 11, 14, 22] for some recent progress concerning TSRs. In order to blot out the linear structure of TSRs and to enhance linear complexity, we introduce the notion of nonlinearly filtered TSRs in Section 5. We also introduce the notion of jump transformation shift registers in Section 6. The results discussed in Sections 5 and 6 are quite analogous to those obtained in Sections 3 and 4, but we include them for the sake of completeness.

The main objective of this paper is to introduce "nonlinearity" on MRMGs as well as on TSRs that have not been yet reported in the literature and to bring them to the notice of the larger crypto community. These "nonlinear word-oriented linear feedback shift registers" have relatively better cryptographic properties as compared to their linear counterparts. Moreover, nonlinear word-oriented linear feedback shift registers have an added advantage over their LFSR counterparts in terms of software efficiency as they produce a word instead of a bit. Thus, nonlinear word-oriented linear feedback shift registers may be a useful choice for designing stream ciphers, particularly, when software efficiency is required.

Part of this work has been presented at the International Conference on SEquences and Their Applications (SETA-2016) held at Southwest Jiaotong University, Chengdu, China during October 09–14, 2016.

2 Preliminaries

We recall some definitions and results described in [1, 16] concerning MRMGs. As we know, in the majority of practical applications, one generally uses finite fields with characteristic 2. Henceforth, we shall restrict ourselves to fields with characteristic 2 and their extensions. However, some of the results discussed in this paper may directly be extended to a finite field with odd characteristic. We shall denote, as usual, by $\mathbb{F}_2$ the finite field with 2 elements, by $\mathbb{F}_{2^m}$ the extension field of $\mathbb{F}_2$ of degree m and by $\mathbb{F}_2[X]$ the ring of polynomials in one variable X with coefficients in $\mathbb{F}_2$.

Given any ring R and any positive integer d, let $M_d(R)$ denote the set of all $d \times d$ matrices with entries in R. Throughout this paper, we fix positive integers m and n, and a vector space basis $\{\alpha_0, \ldots, \alpha_{m-1}\}$ of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$. Given any $s \in \mathbb{F}_{2^m}$, there are unique $s_0, \ldots, s_{m-1} \in \mathbb{F}_2$ such that $s = s_0\alpha_0 + \cdots + s_{m-1}\alpha_{m-1}$, and we shall denote the corresponding co-ordinate vector $(s_0, \ldots, s_{m-1})$ of s by $\mathbf{s}$. Evidently, the association $s \longmapsto \mathbf{s}$ gives a vector space isomorphism of $\mathbb{F}_{2^m}$ onto $\mathbb{F}_2^m$. Elements of $\mathbb{F}_2^m$ may be thought of as row vectors and so $\mathbf{s}C$ is a well-defined element of $\mathbb{F}_2^m$ for any $\mathbf{s} \in \mathbb{F}_2^m$ and $C \in M_m(\mathbb{F}_2)$.

Definition 1 Let $C_0, C_1, \ldots, C_{n-1} \in M_m(\mathbb{F}_2)$. Given any n-tuple $(\mathbf{s}_0, \ldots, \mathbf{s}_{n-1})$ of elements of $\mathbb{F}_{2^m}$, let $(\mathbf{s}_i)_{i=0}^{\infty}$ denote the infinite sequence of elements of $\mathbb{F}_{2^m}$ determined by the following linear recurrence relation:

$$\mathbf{s}_{i+n} = C_0\mathbf{s}_i + C_1\mathbf{s}_{i+1} + \cdots + C_{n-1}\mathbf{s}_{i+n-1} \quad i = 0, 1, \ldots. \tag{1}$$

The system (1) is called a *multiple-recursive matrix generator (MRMG)* of order n over $\mathbb{F}_{2^m}$, while the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is referred to as the *sequence generated by the MRMG* (1). The n-tuple $(\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$ is the *initial state* of the MRMG (1) and the polynomial $I_m X^n - C_{n-1}X^{n-1} - \cdots - C_1 X - C_0$ with matrix coefficients is the *matrix polynomial* of the MRMG (1). The sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is *ultimately periodic* if there are integers r, n_0 with

$r \geq 1$ and $n_0 \geq 0$ such that $\mathbf{s}_{j+r} = \mathbf{s}_j$ for all $j \geq n_0$. The least positive integer r with this property is the *period* of $(\mathbf{s}_i)_{i=0}^{\infty}$ and the corresponding least nonnegative integer n_0 is the *preperiod* of $(\mathbf{s}_i)_{i=0}^{\infty}$. The sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is *periodic* if its preperiod is 0.

The following result gives some basic facts about MRMG.

Proposition 1 [6, Proposition 4.2] *For the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ generated by the MRMG (1) of order n over $\mathbb{F}_{2^m}$, we have*

(i) *$(\mathbf{s}_i)_{i=0}^{\infty}$ is ultimately periodic, and its period is no more than $2^{mn} - 1$;*
(ii) *if C_0 is nonsingular, then $(\mathbf{s}_i)_{i=0}^{\infty}$ is periodic; conversely, if $(\mathbf{s}_i)_{i=0}^{\infty}$ is periodic whenever the initial state is of the form $(b, 0, \ldots, 0)$, where $b \in \mathbb{F}_{2^m}$ with $b \neq 0$, then C_0 is nonsingular.*

An MRMG of order n over $\mathbb{F}_{2^m}$ is *primitive* if for any choice of nonzero initial state, the sequence generated by that MRMG is periodic of period $2^{mn} - 1$.

In view of Proposition 1 if $I_m X^n - C_{n-1}X^{n-1} - \cdots - C_1 X - C_0 \in M_m(\mathbb{F}_2)[X]$ is the matrix polynomial of primitive MRMG, then the matrix C_0 is necessarily nonsingular.

Corresponding to a matrix polynomial $I_m X^n - C_{n-1}X^{n-1} - \cdots - C_1 X - C_0 \in M_m(\mathbb{F}_2)[X]$, we can associate a (m, n)-block companion matrix $\mathsf{C}_{mrmg} \in M_{mn}(\mathbb{F}_2)$ of the following form

$$\mathsf{C}_{mrmg} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & C_0 \\ I_m & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & C_1 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & I_m & \mathbf{0} & C_{n-2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & I_m & C_{n-1} \end{pmatrix}, \tag{2}$$

where I_m denotes the $m \times m$ identity matrix over $\mathbb{F}_2$, while $\mathbf{0}$ indicates the zero matrix in $M_m(\mathbb{F}_2)$. Using a Laplace expansion or a suitable sequence of elementary column operations, it is easy to see that $\det \mathsf{C}_{mrmg} = \pm \det(C_0)$. Consequently,

$$\mathsf{C}_{mrmg} \in \mathrm{GL}_{mn}(\mathbb{F}_2) \quad \text{if and only if} \quad C_0 \in \mathrm{GL}_m(\mathbb{F}_2), \tag{3}$$

where $\mathrm{GL}_m(\mathbb{F}_2)$ is the general linear group of $m \times m$ nonsingular matrices over $\mathbb{F}_2$.

It may be noted that the block companion matrix (2) is the state transition matrix for the MRMG (1). Indeed, the k-th state $\mathbf{S}_k := (\mathbf{s}_k, \mathbf{s}_{k+1}, \ldots, \mathbf{s}_{k+n-1}) \in \mathbb{F}_{2^m}^n$ of the MRMG (1) is obtained from the initial state $\mathbf{S}_0 := (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1}) \in \mathbb{F}_{2^m}^n$ by $\mathbf{S}_k = \mathbf{S}_0 \mathsf{C}_{mrmg}^k$, for any $k \geq 0$. We can identify MRMG (1) with the block companion matrix (2).

The following lemma reduces the calculation of an $mn \times mn$ determinant to an $m \times m$ determinant.

Lemma 1 [6, Lemma 5.1] *Let C_{mrmg} be an MRMG as given as in (2) and also let $M(X) \in M_m(\mathbb{F}_2[X])$ be defined by $M(X) := I_m X^n - C_{n-1}X^{n-1} - \cdots - C_1 X - C_0$. Then the characteristic polynomial of C_{mrmg} is equal to $\det(M(X))$.*

The following characterization of primitive MRMG can be easily extracted from the results given in [6]; see also [15, Theorem 4].

Proposition 2 [6] *Let $C_0 \in \mathrm{GL}_m(\mathbb{F}_2)$. Then the following are equivalent:*

(i) *an MRMG (1) of order n over $\mathbb{F}_{2^m}$ is primitive;*

(ii) $o(\mathbf{C}_{mrmg}) = q^{mn} - 1$, *where* $o(\mathbf{C}_{mrmg})$ *denotes the multiplicative order of* $\mathbf{C}_{mrmg}$ *in* $\mathrm{GL}_{mn}(\mathbb{F}_2)$;
(iii) $\det(M(X))$ *is a primitive polynomial over* $\mathbb{F}_2$ *of degree* mn, *where* $M(X)$ *is same as defined in Lemma 1.*

We recall a result that enables us to determine the linear complexity of sequences generated by a primitive MRMG.

Lemma 2 [16, Lemma 1] *Let* $\mathbf{s}_i = \left(s_i^{(1)}, \ldots, s_i^{(m)}\right) \in \mathbb{F}_2^m \simeq \mathbb{F}_{2^m} \quad i = 0, 1, \ldots,$ *be an arbitrary recursive vector sequence and let* $h(X) \in \mathbb{F}_2[X]$ *be the characteristic polynomial of the matrix* $\mathbf{C}_{mrmg}$ *in* (2). *Then for each* $1 \leq j \leq m$ *the sequence* $s_0^{(j)}, s_1^{(j)}, \ldots$ *of the* j^{th} *coordinates is a linear recurring sequences in* $\mathbb{F}_2$ *with characteristic polynomial* $h(X)$.

The following lemma trivially follows from Lemma 2 and gives the component-wise linear complexity of the sequences generated by a primitive MRMG.

Lemma 3 *Let* $\mathbf{s}_i = \left(s_i^{(1)}, \ldots, s_i^{(m)}\right) \in \mathbb{F}_2^m \simeq \mathbb{F}_{2^m} \quad i = 0, 1, \ldots,$ *be a sequence generated by a primitive MRMG of order* n *over* $\mathbb{F}_{2^m}$. *Then for each* $1 \leq j \leq m$, *the linear complexity of the* j^{th} *coordinate sequence* $s_0^{(j)}, s_1^{(j)}, \ldots$ *over* $\mathbb{F}_2$ *is* mn.

We also recall some definitions and results from [11, 24] concerning transformation shift registers (TSR) for generating pseudorandom vectors. These results are similar to what we discussed for multiple-recursive matrix generator and will be used in the sequel.

Definition 2 [11, 24] Let $c_0, c_1, \ldots, c_{n-1} \in \mathbb{F}_2$ and $T \in M_m(\mathbb{F}_2)$. Given any n-tuple $(\mathbf{s}_0, \ldots, \mathbf{s}_{n-1})$ of elements of $\mathbb{F}_{2^m}$, let $(\mathbf{s}_i)_{i=0}^{\infty}$ denote the infinite sequence of elements of $\mathbb{F}_{2^m}$ determined by the following linear recurrence relation:

$$\mathbf{s}_{i+n} = \mathbf{s}_i(c_0 T) + \mathbf{s}_{i+1}(c_1 T) + \cdots + \mathbf{s}_{i+n-1}(c_{n-1} T) \quad i = 0, 1, \ldots. \tag{4}$$

The system (4) is a *transformation shift register (TSR)* of order n over $\mathbb{F}_{2^m}$, while the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is the *sequence generated by the TSR* (4). The n-tuple $(\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$ is the *initial state* of the TSR (4) and the polynomial $X^n - (c_{n-1}T)X^{n-1} - \cdots - (c_1 T)X - (c_0 T)$ with matrix coefficients is the *tsr-polynomial* of the TSR (4). The sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is *ultimately periodic* if there are integers r, n_0 with $r \geq 1$ and $n_0 \geq 0$ such that $\mathbf{s}_{j+r} = \mathbf{s}_j$ for all $j \geq n_0$. The least positive integer r with this property is the *period* of $(\mathbf{s}_i)_{i=0}^{\infty}$ and the corresponding least nonnegative integer n_0 is the *preperiod* of $(\mathbf{s}_i)_{i=0}^{\infty}$. The sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is *periodic* if its preperiod is 0.

The basic properties of TSRs are stated in the following proposition.

Proposition 3 [11] *For the sequence* $(\mathbf{s}_i)_{i=0}^{\infty}$ *generated by the TSR* (4) *of order* n *over* $\mathbb{F}_{2^m}$, *we have*

(i) $(\mathbf{s}_i)_{i=0}^{\infty}$ *is ultimately periodic, and its period is* $\leq 2^{mn} - 1$;
(ii) *if* $c_0 \neq 0$ *and* T *is nonsingular, then* $(\mathbf{s}_i)_{i=0}^{\infty}$ *is periodic; conversely, if* $(\mathbf{s}_i)_{i=0}^{\infty}$ *is periodic whenever the initial state is of the form* $(b, 0, \ldots, 0)$, *where* $b \in \mathbb{F}_{2^m}$ *with* $b \neq 0$, *then* $c_0 T$ *is nonsingular.*

A TSR of order n over $\mathbb{F}_{2^m}$ is *primitive* if for any choice of nonzero initial state, the sequence generated by that TSR is periodic of period $2^{mn}-1$.

In view of Proposition 3, if $X^n-(c_{n-1}T)X^{n-1}-\cdots-(c_1T)X-(c_0T)\in M_m(\mathbb{F}_2)[X]$ is the tsr-polynomial of a primitive TSR, then the matrix c_0T is necessarily nonsingular.

Corresponding to a tsr-polynomial $X^n-(c_{n-1}T)X^{n-1}-\cdots-(c_1T)X-(c_0T)\in M_m(\mathbb{F}_2)[X]$, we can associate a (m,n)-block companion matrix $\mathsf{C}_{tsr}\in M_{mn}(\mathbb{F}_2)$ of the following form

$$\mathsf{C}_{tsr}=\begin{pmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & c_0T\\ I_m & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & c_1T\\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & I_m & \mathbf{0} & c_{n-2}T\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & I_m & c_{n-1}T\end{pmatrix}, \tag{5}$$

where $c_0,c_1,\ldots,c_{n-1}\in\mathbb{F}_2$, $T\in M_m(\mathbb{F}_2)$ and I_m denotes the $m\times m$ identity matrix over $\mathbb{F}_2$, while $\mathbf{0}$ indicates the zero matrix in $M_m(\mathbb{F}_2)$.

Note that the block companion matrix (5) is the state transition matrix for the TSR (4) and we can identify TSR (4) with block companion matrix (5).

The following lemma is used to compute the characteristic polynomial of a TSR (5).

Lemma 4 [24, Proposition 3.1] *Let C_{tsr} be a TSR as given in (5) and also let $\widetilde{M}(X)\in M_m(\mathbb{F}_2[X])$ be defined by $\widetilde{M}(X):=I_mX^n-(c_{n-1}T)X^{n-1}-\cdots-(c_1T)X-(c_0T)$. Then the characteristic polynomial of C_{tsr} is equal to* $\det\left(\widetilde{M}(X)\right)$.

Primitive TSRs admit the following characterization.

Proposition 4 [11] *Let c_0 be non-zero and $T\in\mathrm{GL}_m(\mathbb{F}_2)$. Then the following are equivalent:*

(i) *a TSR (4) of order n over $\mathbb{F}_{2^m}$ is primitive;*
(ii) *$o(\mathsf{C}_{tsr})=2^{mn}-1$, where $o(\mathsf{C}_{tsr})$ denotes the multiplicative order of T in $\mathrm{GL}_{mn}(\mathbb{F}_2)$;*
(iii) *$\det(F(X))$ is a primitive polynomial of degree mn over $\mathbb{F}_2$, where $\widetilde{M}(X)$ is same as defined in Lemma 4.*

The following lemma ensures that the characteristic polynomial of each of the component sequences is same as the TSR itself.

Lemma 5 [11] *Let $\mathbf{s}_i=\left(s_i^{(1)},\ldots,s_i^{(m)}\right)\in\mathbb{F}_2^m\simeq\mathbb{F}_{2^m}\quad i=0,1,\ldots,$ be an arbitrary recursive vector sequence and let $g\in\mathbb{F}_2[x]$ be the characteristic polynomial of the matrix C_{tsr} in (5). Then for each $1\le j\le m$ the sequence $s_0^{(j)},s_1^{(j)},\ldots$ of the j-th coordinates is a linear recurring sequences in $\mathbb{F}_2$ with characteristic polynomial g.*

The following lemma gives linear complexity of the componentwise sequences generated by a primitive TSR.

Lemma 6 [11, Corollary 1] *Let $\mathbf{s}_i=\left(s_i^{(1)},\ldots,s_i^{(m)}\right)\in\mathbb{F}_2^m\simeq\mathbb{F}_{2^m}\quad i=0,1,\ldots,$ be a sequence generated by a primitive TSR of length n over $\mathbb{F}_{2^m}$. Then for each $1\le j\le m$, the linear complexity of the j-th coordinate sequence $s_0^{(j)},s_1^{(j)},\ldots$ over $\mathbb{F}_2$ is mn.*

3 Nonlinearly filtered primitive MRMGs

First we recall some definitions and classical results about the linear complexity of nonlinearly filtered primitive LFSRs. We then use these to define nonlinearly filtered primitive MRMGs.

Definition 3 Let $(s_i)_{i=0}^{\infty}$ be a sequence generated by a primitive LFSR. If a sequence $(z_i)_{i=0}^{\infty}$ is produced by any non-zero linear combination of finitely many (say, L) products of k different shifts of the sequence $(s_i)_{i=0}^{\infty}$, that is,

$$z_i = \sum_{t=0}^{L-1} c_t s_{i+\delta_0^t} s_{i+\delta_1^t} \cdots s_{i+\delta_{k-1}^t},$$

where, for each $j = 0, \ldots, k-1$, δ_j^t denote the distance at which sequence has been shifted, then the sequence $(z_i)_{i=0}^{\infty}$ is obtained by a *k-th order filtering* of the sequence $(s_i)_{i=0}^{\infty}$.

We recall the following classical result by Rueppel [23] which gives a lower bound for the linear complexity of filtered sequences.

Proposition 5 [23, Corollary 5.7] *Let $(s_i)_{i=0}^{\infty}$ be a sequence generated by a primitive LFSR of length n over $\mathbb{F}_2$ and let $(z_i)_{i=0}^{\infty}$ be produced by any non-zero linear combination of L consecutive k-th order products $(k < n)$ of equidistant shifts of the sequence $(s_i)_{i=0}^{\infty}$, that is*

$$z_i = \sum_{t=0}^{L-1} c_t s_{i+t} s_{i+t+\delta} \cdots s_{i+t+(k-1)\delta},$$

where $\gcd(\delta, 2^n - 1) = 1$. *Then, the linear complexity of the filtered sequence $(z_i)_{i=0}^{\infty}$ is at least* $\binom{n}{k} - (L-1)$.

Without essentially having the Rueppel's restriction that δ and $2^n - 1$ be coprime, the following result of Paterson [19] still gives a good lower bound on the linear complexity of the filtered sequences.

Proposition 6 [19, Theorem 1] *Let $(s_i)_{i=0}^{\infty}$ be a sequence generated by a primitive LFSR of length n over $\mathbb{F}_2$. Assume $0 < \delta < 2^n - 1$ and u is the least positive integer such that $(2^n - 1) | \delta(2^u - 1)$. Let $(z_i)_{i=0}^{\infty}$ be produced by any non-zero linear combination of L consecutive k-th order products $(k < n)$ of equidistant shifts of the sequence $(s_i)_{i=0}^{\infty}$, that is*

$$z_i = \sum_{t=0}^{L-1} c_t s_{i+t} s_{i+t+\delta} \cdots s_{i+t+(k-1)\delta}.$$

Then, the linear complexity of $(z_i)_{i=0}^{\infty}$ is at least $\binom{u}{k} \left(\frac{n}{u}\right)^k - (L-1)$.

Analogous to the notion of nonlinearly filtered primitive LFSRs, we introduce the notion of nonlinearly filtered primitive MRMGs and study their linear complexity.

Consider $(\mathbf{s}_i)_{i=0}^{\infty}$, where $\mathbf{s}_i = \left(s_i^{(1)}, \ldots, s_i^{(m)}\right) \in \mathbb{F}_2^m \cong \mathbb{F}_{2^m}$, $i = 0, 1, \ldots$, is a sequence over $\mathbb{F}_{2^m}$ generated by a primitive MRMG of order n. Then the "product" and "sum" of

the two different shifts $\mathbf{s}_i$ and $\mathbf{s}_{i+\delta}$ of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is defined as the componentwise multiplication and componentwise addition respectively:

$$\mathbf{s}_i\mathbf{s}_{i+\delta} = \left(s_i^{(1)}s_{i+\delta}^{(1)}, \ldots, s_i^{(m)}s_{i+\delta}^{(m)}\right), \ \mathbf{s}_i + \mathbf{s}_{i+\delta} = \left(s_i^{(1)} + s_{i+\delta}^{(1)}, \ldots, s_i^{(m)} + s_{i+\delta}^{(m)}\right). \quad (6)$$

Definition 4 Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be a sequence generated by a primitive MRMG. If a sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ is produced by any non-zero linear combination of finitely many (say, L) products of k different shifts of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$, that is,

$$\mathbf{z}_i = \sum_{t=0}^{L-1} c_t \mathbf{s}_{i+\delta_0^t}\mathbf{s}_{i+\delta_1^t}\cdots\mathbf{s}_{i+\delta_{k-1}^t},$$

then the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ is obtained by a *k-th order filtering* of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$.

The lower bound for the componentwise linear complexity of nonlinearly filtered MRMGs is given by the following proposition.

Proposition 7 *Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be a sequence generated by a primitive MRMG of length n over $\mathbb{F}_{2^m}$ and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be produced by any non-zero linear combination of L consecutive k-th order products $(k < n)$ of equidistant shifts of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$, that is*

$$\mathbf{z}_i = \sum_{t=0}^{L-1} c_t \mathbf{s}_{i+t}\mathbf{s}_{i+t+\delta}\cdots\mathbf{s}_{i+t+(k-1)\delta},$$

where $\gcd(\delta, 2^{mn}-1) = 1$. *Then for each* $1 \le j \le m$, *the linear complexity of the j-th coordinate sequence* $z_0^{(j)}, z_1^{(j)}, \ldots$ *is at least* $\binom{mn}{k} - (L-1)$.

Proof From (6), it follows that for each $1 \le j \le m$, we have

$$z_i = \sum_{t=0}^{L-1} c_t s_{i+t}^{(j)}s_{i+t+\delta}^{(j)}\cdots s_{i+t+(k-1)\delta}^{(j)} \quad i = 0, 1, \ldots.$$

Moreover, by Lemma 3, the linear complexity of the sequences $s_i^{(j)}$, $i = 0, 1, \ldots$, is mn. Thus, by Proposition 5, the linear complexity of the sequence $(z_i^{(j)})_{i=0}^{\infty}$ is at least $\binom{mn}{k} - (L-1)$. □

By relaxing the condition $\gcd(\delta, 2^{mn}-1) = 1$, the following proposition still gives a good lower bound for the componentwise linear complexity of nonlinearly filtered primitive MRMGs.

Proposition 8 *Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be a sequence generated by a primitive MRMG of length n over $\mathbb{F}_{2^m}$. Assume $0 < \delta < 2^{mn}-1$ and u is the least positive integer such that $(2^{mn}-1)|\delta(2^u-1)$. Let a sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ be produced by any non-zero linear combination of L consecutive k-th order products $(k < mn)$ of equidistant shifts of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ defined as*

$$\mathbf{z}_i = \sum_{t=0}^{L-1} c_t \mathbf{s}_{i+t}\mathbf{s}_{i+t+\delta}\cdots\mathbf{s}_{i+t+(k-1)\delta}.$$

Then for each $1 \leq j \leq m$, *the linear complexity of the* j*-th coordinate sequence* $z_0^{(j)}, z_1^{(j)}, \ldots$ *is at least* $\binom{u}{k}\left(\frac{mn}{u}\right)^k - (L-1)$.

Proof It follows by similar arguments as the ones given in the proof of Proposition 7 and by using Proposition 6. □

Our discussions so far provide only generic framework for introducing nonlinearity on the contents of MRMGs that ensures a minimum value for the componentwise linear complexity. However, one might use cryptographically well-studied nonlinear functions, like the one suggested in [1], on the contents of MRMGs so as to produce sequences with possibly explicit and better linear complexity.

As alluded to in the introduction, nonlinearly filtered MRMGs output a word per clock instead of a bit and hence, they should have efficient software encryption unlike nonlinearly filtered LFSRs.

4 Jump controlled multiple-recursive matrix generator

In stream cipher design, one can achieve better linear complexity of the sequences by using cascading clock controlled feedback shift registers. In this generator, the output of one FSR is obtained by clocking or stepping through its state space once or multiple times depending upon the output of another FSR. The keystream generators that use clock-controlled FSRs are usually stepped a few times to produce just a single bit of the sequence. As a consequence, we have relatively less rate of sequence generation, which makes them not so attractive for high speed implementations. In order to achieve better efficiency in such cases, let an FSR move to a state more than one step ahead without actually traversing consecutive intermediate states. This phenomenon is called jumping and the notion of bit-oriented jump controlled LFSR was first introduced by Jansen in [12]. It was later used in designing a stream cipher called Pomaranch [13].

Analogous to the notion of jump controlled LFSR, we discuss jump controlled multiple-recursive matrix generator, in which output of one MRMG is clock controlled by another MRMG. The jump multiple-recursive matrix generator shall render double efficiency, one because of its jumping and other because it produces a word per clock unlike jump LFSRs. In this section, we also study period and componentwise linear complexity of sequences generated by jump controlled multiple-recursive matrix generator.

Let us suppose that the state transition matrix C_{mrmg} as described in (2) has maximum possible multiplicative order in the corresponding general linear group, the output sequence then achieves the maximum possible period $2^{mn}-1$. Now from Lemma 1, we have that the characteristic polynomial $\chi(X)$ of C_{mrmg} is given by

$$\chi(X) = \det\left(I_m X^n - C_{n-1}X^{n-1} - \cdots - C_1 X - C_0\right)$$

and in view of Proposition 2, $\chi(X)$ is primitive of degree mn over $\mathbb{F}_2$.

Since $\chi(\mathsf{C}_{mrmg}) = 0$, C_{mrmg} may be viewed as a root of χ, that is, the order of C_{mrmg} is $2^{mn}-1$. Now $\mathsf{C}_{mrmg}+I$ being an element of $\mathbb{F}_2^{mn}$ is equal to C_{mrmg}^J for some positive integer J. In fact, the identity $\mathsf{C}_{mrmg}^J = \mathsf{C}_{mrmg} + I$ is essentially equivalent to $X^J = X + 1 \pmod{\chi(X)}$. The integer J is the *Jump index* of χ (cf. 12). It is easy to conclude from here that by changing state transition matrix of MRMG from C_{mrmg} to $\mathsf{C}_{mrmg} + I$, we are effectively making J steps through the state space of the original MRMG, that is, jumping J steps

ahead of the original state. It is clear that we achieve the same effect as multiplying a state vector either by C_{mrmg}^{J} or by $\mathsf{C}_{mrmg}+I$. Moreover, it is much more efficient to go from C_{mrmg} to $\mathsf{C}_{mrmg}+I$ rather than going from C_{mrmg} to C_{mrmg}^{J}. Thus making a MRMG jump is indeed efficient and attractive. To see it more directly, let us assume that at a particular instance t, the t-th state of MRMG is $\mathbf{S}_t := (\mathbf{s}_t, \mathbf{s}_{t+1}, \ldots, \mathbf{s}_{t+n-1}) \in \mathbb{F}_{2^m}^n$. Now in order to traverse it J times through the state space, we need to multiply it by C_{mrmg}^{J} as given in the following expression:

$$\mathbf{S}_t \mathsf{C}_{mrmg}^{J} = \mathbf{S}_t \left(\mathsf{C}_{mrmg} + I\right) = \mathbf{S}_t \mathsf{C}_{mrmg} + \mathbf{S}_t = \mathbf{S}_{t+1} + \mathbf{S}_t. \tag{7}$$

Thus, in practice, jumping can be achieved simply by adding the current state to the next state.

Let the characteristic polynomial of the modified transition matrix $\mathsf{C}_{mrmg}+I$ be denoted by $\chi^{\perp}(X)$; it follows that $\chi^{\perp}(X) = \det(XI + \mathsf{C}_{mrmg} + I) = \chi(X+1)$. It may be remarked that the dual $\chi^{\perp}(X)$ is not necessarily a primitive polynomial even though $\chi(X)$ is primitive. As noted in [12], if the dual polynomial $\chi^{\perp}(X)$ is also primitive, its jump index $J^{\perp}$ always exists and is given by

$$J^{\perp} = J^{-1} \pmod{2^{mn}-1}.$$

It is clear that the jump index of the dual polynomial only exists if J is relatively prime with order of χ, i.e., $\gcd(J, 2^{mn}-1) = 1$. Moreover, the jump index J^* of the reciprocal polynomial $\chi^*(X) = X^{mn}\chi(1/X)$ of $\chi(X)$ is given by

$$J^* = 1 - J \pmod{2^{mn}-1}.$$

After having these basic facts at our disposal, we now turn our focus toward jump controlled MRMGs. We discuss here the basic building block for designing a key stream generator that consists of a control MRMG, say C-MRMG and a clock-controlled generating MRMG, say G-MRMG. For achieving better period and better linear complexity of sequences, we take both C-MRMG and G-MRMG to be primitive MRMGs of length n over $\mathbb{F}_{2^m}$. We take the output of G-MRMG controlled by the binary sequence generated by C-MRMG in the following manner: if the output of C-MRMG is 0, we take the output from G-MRMG only after clocking it once, and if the output of C-MRMG is 1, we take the output from G-MRMG only after jumping it J times, where J is the jump index of G-MRMG.

In order to achieve maximum period of the component sequences generated by G-MRMG when clocked controlled under C-MRMG, we impose the condition $\gcd(J-1, 2^{mn}-1) = 1$. The reader is referred to [4, 12] for more details.

The following proposition guarantees the maximum period for component sequences generated by a G-MRMG when controlled under C-MRMG.

Proposition 9 *Let G-MRMG be a jump MRMG controlled under C-MRMG as described above. Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be the sequence generated by G-MRMG when clocked regularly and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be the sequence produced by G-MRMG when clocked under the control of C-MRMG. Then for each $1 \le j \le m$, the period of the j-th coordinate sequence $z_0^{(j)}, z_1^{(j)}, \ldots$ is $(2^{mn}-1)^2$.*

Proof It is clear that each component sequence $z_i^{(j)}$, $i = 0, 1, \ldots$ would traverse a step-or-J steps through the state space depending on whether C-MRMG produces 0 or 1, respectively. Moreover, in view of Lemmas 2 and 3, each component sequence $s_i^{(j)}$, $i = 0, 1, \ldots$, of G-MRMG when clocked regularly is primitive with period $2^{mn}-1$ and linear complexity mn.

Thus, it follows from [9, Theorem 4] together with the condition $\gcd(J-1, 2^{mn}-1) = 1$ that the period of each component sequence $z_i^{(j)}$, $i = 0, 1, \ldots$ is $(2^{mn}-1)^2$. □

Proposition 9 clearly exhibits that there is a significant improvement in the period of the component sequences of jump controlled MRMGs. In fact, it has increased from the period $2^{mn}-1$ of usual MRMGs to the period $(2^{mn}-1)^2$ of the jump controlled MRMGs.

The componentwise linear complexity of the sequences generated by jump controlled MRMG is given by the following proposition.

Proposition 10 *Let G-MRMG be a jump MRMG controlled under C-MRMG as described above. Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be the sequence generated by G-MRMG when clocked regularly and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be the sequence produced by G-MRMG when clocked under the control of C-MRMG. Then for each $1 \leq j \leq m$, the componentwise linear complexity of the j-th coordinate sequence $z_0^{(j)}, z_1^{(j)}, \ldots$ is $mn\,(2^{mn}-1)$.*

Proof It follows by similar arguments as given in the proof of Proposition 9 and by [9, Corollary 1] together with the condition $\gcd(J-1, 2^{mn}-1) = 1$. □

It is easy to see from Proposition 10 that the componentwise linear complexity of jump controlled MRMGs is $(2^{mn}-1)$ times more than that of the usual MRMGs.

The sequences generated by jump controlled MRMG when viewed over $\mathbb{F}_2$ are useful in various applications. The following theorem gives explicit period and greatest possible value of linear complexity of the sequences generated by jump controlled MRMGs when viewed over $\mathbb{F}_2$.

Theorem 1 *Let G-MRMG be a jump MRMG controlled under C-MRMG as described above. Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be the sequence generated by G-MRMG when clocked regularly and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be the sequence produced by G-MRMG when clocked under the control of C-MRMG. Also, let $\gcd(J-1, 2^{mn}-1) = 1$. If the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ in $\mathbb{F}_{2^m}$ generated by G-MRMG when clocked under the control of C-MRMG is viewed as a sequence in $\mathbb{F}_2$, then its period and the greatest possible value of its linear complexity are given by $(2^{mn}-1)^2$ and $m^2n\,(2^{mn}-1)$, respectively.*

Proof It is easy to see from Lemma 2 that the minimal polynomial over $\mathbb{F}_2$ of each of the component sequences of G-MRMG when clocked regularly is primitive of degree mn and is given by $\chi(X)$. In other words, each component sequences of G-MRMG when clocked regularly is of period $2^{mn}-1$ and is essentially a shifted version of other components. Therefore, when G-MRMG is clocked under the control of C-MRMG, each component sequence $z_i^{(j)}$, $i = 0, 1, \ldots$ has the same minimal polynomial (say, $g(X)$) over $\mathbb{F}_2$ and has degree $mn\,(2^{mn}-1)$ in view of the Proposition 10.

We can think of each component sequence $z_i^{(j)}$, $i = 0, 1, \ldots$ as an m-decimated sequence of $(\mathbf{z}_i)_{i=0}^{\infty}$ when viewed as a sequence over $\mathbb{F}_2$. Thus, the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ when viewed as a sequence over $\mathbb{F}_2$ can be obtained by interleaving these m decimated sequences $z_i^{(j)}$, $j = 0, \ldots, m-1$. If $f(X)$ denotes the minimal polynomial of the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ when viewed as a sequence over $\mathbb{F}_2$, then by [8, Theorem 1], we have that $f(X) \mid g(X^m)$. Hence, the greatest possible value of the linear complexity of the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ when viewed as a sequence over $\mathbb{F}_2$ is $m^2n\,(2^{mn}-1)$, which is the degree of the polynomial $g(X^m)$. Moreover, it follows from Proposition 9 that the period of the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ when viewed as a sequence over $\mathbb{F}_2$ is $(2^{mn}-1)^2$. □

It is clear from (7) that to make a G-MRMG traverse through J steps in the state space, we do not really need the explicit value of J. However, in order to maximize the period of jump controlled MRMGs, the greatest common divisor of $J-1$ and the period $2^{mn}-1$ must be equal to 1. By taking ideas from Pohlig-Hellman method [20], we prove the following theorem that gives an alternative way to check this condition without actually computing J.

Theorem 2 *Let $h(X)$ be a primitive polynomial of degree N over $\mathbb{F}_2$ with period $P = 2^N - 1$ and jump index J. Moreover, assume that $p_1, p_2, \ldots, p_r$ are the distinct prime factors of P, where each factor occurs only once. Then* $\gcd(J-1, P) = 1$ *if and only if* $X^{\frac{P}{p_i}} \neq (X+1)^{\frac{P}{p_i}} \pmod{h(X)}$ *for each* $i = 1, \ldots, r$.

Proof It is easy to see that $\gcd(J-1, P) = 1$ if and only if $J \neq 1 \pmod{p_i}$ for each $i = 1, \ldots, r$. By the definition of jump index, we have

$$X^J = X + 1 \pmod{h(X)}. \tag{8}$$

For each i, we can reduce the computation of J as a discrete logarithm in $\mathbb{F}_{2^N}^*$ to computation of $J \pmod{p_i}$ as a discrete logarithm in the cyclic groups of order p_i generated by $X^{\frac{P}{p_i}}$ as indicated in the following equation.

$$(X^{\frac{P}{p_i}})^{J \pmod{p_i}} = (X+1)^{\frac{P}{p_i}} \pmod{h(X)}. \tag{9}$$

It follows from (9) that $J \neq 1 \pmod{p_i}$ if and only if $X^{\frac{P}{p_i}} \neq (X+1)^{\frac{P}{p_i}} \pmod{h(X)}$. This completes the proof. □

In practice, if the factors of P are known, it is easier to check the equivalent conditions of Theorem 2 as compared to computing J first and then testing whether or not $\gcd(J-1, P) = 1$.

5 Nonlinearly filtered primitive TSRs

Motivated by nonlinearly filtered primitive MRMGs, we introduce the notion of nonlinearly filtered primitive TSRs in this section and obtain some results analogous to those discussed in Section 3.

Consider $(\mathbf{s}_i)_{i=0}^{\infty}$, where $\mathbf{s}_i = \left(s_i^{(1)}, \ldots, s_i^{(m)}\right) \in \mathbb{F}_2^m \cong \mathbb{F}_{2^m}$, $i = 0, 1, \ldots$, is a sequence over $\mathbb{F}_{2^m}$ generated by a primitive TSR of order n. Then the "product" and "sum" of the two different shifts $\mathbf{s}_i$ and $\mathbf{s}_{i+\delta}$ of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ is defined as the componentwise multiplication and componentwise addition respectively:

$$\mathbf{s}_i\mathbf{s}_{i+\delta} = \left(s_i^{(1)}s_{i+\delta}^{(1)}, \ldots, s_i^{(m)}s_{i+\delta}^{(m)}\right), \ \mathbf{s}_i + \mathbf{s}_{i+\delta} = \left(s_i^{(1)} + s_{i+\delta}^{(1)}, \ldots, s_i^{(m)} + s_{i+\delta}^{(m)}\right). \tag{10}$$

Definition 5 Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be a sequence generated by a primitive TSR. If a sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ is produced by any non-zero linear combination of finitely many (say, L) products of k different shifts of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$, that is,

$$\mathbf{z}_i = \sum_{t=0}^{L-1} c_t \mathbf{s}_{i+\delta_0^t}\mathbf{s}_{i+\delta_1^t}\cdots\mathbf{s}_{i+\delta_{k-1}^t},$$

then the sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ is obtained by a *k-th order filtering* of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$.

The following proposition gives lower bound for the componentwise linear complexity of nonlinearly filtered TSRs.

Proposition 11 *Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be a sequence generated by a primitive TSR of length n over $\mathbb{F}_{2^m}$ and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be produced by any non-zero linear combination of L consecutive k-th order products $(k < n)$ of equidistant shifts of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$, that is*

$$\mathbf{z}_i = \sum_{t=0}^{L-1} c_t \mathbf{s}_{i+t}\mathbf{s}_{i+t+\delta} \cdots \mathbf{s}_{i+t+(k-1)\delta},$$

where $\gcd(\delta, 2^{mn} - 1) = 1$. Then for each $1 \leq j \leq m$, the linear complexity of the j-th coordinate sequence $z_0^{(j)}, z_1^{(j)}, \ldots$ is at least $\binom{mn}{k} - (L-1)$.

Proof It follows by similar arguments as given in the proof of Proposition 7 and by using Proposition 5. □

After we relax the restriction $\gcd(\delta, 2^{mn} - 1) = 1$ in Proposition 11, the following proposition still ensures a good lower bound for the componentwise linear complexity of the nonlinear filtered TSRs.

Proposition 12 *Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be a sequence generated by a primitive TSR of length n over $\mathbb{F}_{2^m}$. Assume $0 < \delta < 2^{mn}-1$ and u is the least positive integer such that $(2^{mn}-1)|\delta(2^u-1)$. Let a sequence $(\mathbf{z}_i)_{i=0}^{\infty}$ is produced by any non-zero linear combination of L consecutive k-th order products $(k < mn)$ of equidistant shifts of the sequence $(\mathbf{s}_i)_{i=0}^{\infty}$ defined as*

$$\mathbf{z}_i = \sum_{t=0}^{L-1} c_t \mathbf{s}_{i+t}\mathbf{s}_{i+t+\delta} \cdots \mathbf{s}_{i+t+(k-1)\delta}.$$

Then for each $1 \leq j \leq m$, the linear complexity of the j-th coordinate sequence $z_0^{(j)}, z_1^{(j)}, \ldots$ is at least $\binom{u}{k} \left(\frac{mn}{u}\right)^k - (L-1)$.

Proof It follows by similar arguments as given in the proof of Proposition 7 and by using Proposition 6. □

The results of this section are quite general in nature. It is important to note that for practical purposes, one has to employ a nonlinear function on the contents of TSRs that significantly increases the linear complexity. For instance, a nonlinear scheme based on Langford arrangement was given in [11], which is similar yet different from what has been recently proposed in [1] and gives enhanced componentwise linear complexity.

Unlike their LFSRs counterparts, nonlinearly filtered TSRs produce a word per clock and hence are efficient in software implementations.

6 Jump controlled transformation shift registers

In this section, we introduce the notion of jump controlled transformation shift registers, in which output of one TSR is clock controlled by another TSR and study period and componentwise linear complexity of sequences generated by jump controlled transformation shift

registers. Like jump controlled MRMG, the jump controlled transformation shift registers shall also provide greater efficiency unlike jump LFSRs.

We begin with a TSR whose state transition matrix C_{tsr} as given in (5) has order $2^{mn}-1$ so as to obtain maximum period of the corresponding sequence. In view of Lemma 4, the characteristic polynomial $\widetilde{\chi}(X)$ of C_{tsr} admits the following form

$$\widetilde{\chi}(X) = \det\left(I_m X^n - (c_{n-1}T)X^{n-1} - \cdots - (c_1 T)X - (c_0 T)\right),$$

and Proposition 4 clearly implies that $\widetilde{\chi}(X)$ is primitive of degree mn over $\mathbb{F}_2$.

As pointed out in [12], $\widetilde{\chi}(X)$ being primitive, the jump index of $\widetilde{\chi}$ shall always exits. Let $\widetilde{J}$ be the jump index of $\widetilde{\chi}$. By definition of jump index, we have, $\mathsf{C}_{tsr}^{\widetilde{J}} = \mathsf{C}_{tsr} + I$, which is indeed equivalent to $X^{\widetilde{J}} = X + 1 \pmod{\widetilde{\chi}(X)}$.

In order to traverse $\widetilde{J}$ step ahead through the state space of the original TSR, we shall multiply the state vector by $\mathsf{C}_{tsr}^{\widetilde{J}}$, which, in turn, is same as multiplying by $\mathsf{C}_{tsr} + I$. In fact, if at a particular instance t, the t-th state of TSR is $\mathbf{S}_t := (\mathbf{s}_t, \mathbf{s}_{t+1}, \ldots, \mathbf{s}_{t+n-1}) \in \mathbb{F}_{2^m}^n$, then the effect of jumping is achieved simply by adding the current state to the next state as is evident from the following equation:

$$\mathbf{S}_t \mathsf{C}_{tsr}^{J} = \mathbf{S}_t\,(\mathsf{C}_{tsr} + I) = \mathbf{S}_t \mathsf{C}_{tsr} + \mathbf{S}_t = \mathbf{S}_{t+1} + \mathbf{S}_t. \tag{11}$$

Our basic set-up for a keystream generator consists of a control TSR, say C-TSR and a clock-controlled generating TSR, say G-TSR. We take both C-TSR and G-TSR to be primitive TSRs of length n over $\mathbb{F}_{2^m}$ for achieving high period and better linear complexity. We take the output of G-TSR controlled by the binary sequence generated by C-TSR in the following manner: if the output of C-TSR is 0, we take the output from G-TSR only after clocking it once, and if the output of C-TSR is 1, we take the output from G-TSR only after jumping it $\widetilde{J}$ times, where $\widetilde{J}$ is the jump index of G-TSR. To maximize the period of the component sequences in this set-up, we also impose a restriction that $\gcd(\widetilde{J}-1, 2^{mn}-1) = 1$ or equivalently, $\gcd(\widetilde{J}^*, 2^{mn}-1) = 1$, where $\widetilde{J}^* = 1 - \widetilde{J} \pmod{2^{mn}-1}$ is jump index of the reciprocal polynomial $\widetilde{\chi}^*(X) = X^{mn}\widetilde{\chi}\,(1/X)$ of $\widetilde{\chi}(X)$.

The following proposition is analogous to Proposition 9 and gives period of component sequences generated jump TSR.

Proposition 13 *Let G-TSR be a jump TSR controlled under C-TSR as described above. Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be the sequence generated by G-TSR when clocked regularly and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be the sequence produced by G-TSR when clocked under the control of C-TSR. Then for each $1 \le j \le m$, the period of the j-th coordinate sequence $z_0^{(j)}, z_1^{(j)}, \ldots$ is $(2^{mn}-1)^2$.*

Proof By using Lemmas 5 and 6, the proof follows by similar arguments as given in Proposition 9 together with the condition $\gcd(\widetilde{J}-1, 2^{mn}-1) = 1$. □

It clear from Proposition 13 that the period has increased from the period $2^{mn}-1$ of usual TSRs to the period $(2^{mn}-1)^2$ of the jump controlled TSRs.

The following proposition gives the componentwise linear complexity of the sequences generated by jump controlled TSRs.

Proposition 14 *Let G-TSR be a jump TSR controlled under C-TSR as described above. Let $(\mathbf{s}_i)_{i=0}^{\infty}$ be the sequence generated by G-TSR when clocked regularly and let $(\mathbf{z}_i)_{i=0}^{\infty}$ be the sequence produced by G-TSR when clocked under the control of C-TSR. Then for*

each $1 \leq j \leq m$, *the componentwise linear complexity of the* j*-th coordinate sequence* $z_0^{(j)}, z_1^{(j)}, \ldots$ *is* $mn\,(2^{mn}-1)$.

Proof It follows by similar arguments as given in Proposition 9 and by [9, Corollary 1] together with the condition $\gcd(\widetilde{J}-1, 2^{mn}-1) = 1$. □

Proposition 14 confirms that the componentwise linear complexity of jump controlled TSRs has significantly increased from the usual TSRs.

The period and greatest possible value of linear complexity of the sequences generated by jump controlled TSRs when viewed over $\mathbb{F}_2$ is given by the following theorem.

Theorem 3 *Let G-TSR be a jump TSR controlled under C-TSR as described above. Let* $(\mathbf{s}_i)_{i=0}^{\infty}$ *be the sequence generated by G-TSR when clocked regularly and let* $(\mathbf{z}_i)_{i=0}^{\infty}$ *be the sequence produced by G-TSR when clocked under the control of C-TSR. Also, let* $\gcd(\widetilde{J}-1, 2^{mn}-1) = 1$. *If the sequence* $(\mathbf{z}_i)_{i=0}^{\infty}$ *in* $\mathbb{F}_{2^m}$ *generated by G-TSR when clocked under the control of C-TSR is viewed as a sequence in* $\mathbb{F}_2$, *then its period and the greatest possible value of its linear complexity are given by* $(2^{mn}-1)^2$ *and* $m^2n\,(2^{mn}-1)$, *respectively.*

Proof The proof is exactly along the similar lines as of Theorem 1. □

Remark 1 In order to maximize the period of the jump TSRs, we must ensure the condition $\gcd(\widetilde{J}-1, 2^{mn}-1) = 1$. This can be done by checking the equivalent conditions of Theorem 2.

7 Conclusions

We study nonlinearly filtered multiple-recursive matrix generator for producing pseudorandom vectors based on nonlinear schemes and give lower bounds for their componentwise linear complexity. We also study the jump multiple-recursive matrix generator and establish that sequences generated by them have better period and componentwise linear complexity when compared to usual multiple-recursive matrix generator sequences. Analogous results are given for transformation shift registers for generating pseudorandom vectors. Table 1 summarizes our findings in terms of periods and linear complexity.

Table 1 Summary of our results

	Generators	Period	Linear complexity
Propositions 9 and 10	G-MRMG clocked by C-MRMG	$(2^{mn}-1)^2$	$mn(2^{mn}-1)$
Theorem 1	G-MRMG clocked by C-MRMG over $\mathbb{F}_2$	$(2^{mn}-1)^2$	$\leq m^2n(2^{mn}-1)$
Propositions 13 and 14	G-TSR clocked by C-TSR	$(2^{mn}-1)^2$	$mn(2^{mn}-1)$
Theorem 3	G-TSR clocked by C-TSR over $\mathbb{F}_2$	$(2^{mn}-1)^2$	$\leq m^2n(2^{mn}-1)$

Acknowledgements Daniel Panario and Qiang Wang are partially supported by NSERC of Canada.

References

1. Bishoi, S.K., Haran, H.K., Hasan, S.U.: A note on the multiple-recursive matrix method for generating pseudorandom vectors. Discrete Appl. Math. **222**, 67–75 (2017)
2. Chen, E., Tseng, D.: The splitting subspace conjecture. Finite Fields Appl. **24**, 15–28 (2013)
3. Cohen, S.D., Hasan, S.U., Panario, D., Wang, Q.: An asymptotic formula for the number of irreducible transformation shift registers. Linear Algebra Appl. **484**, 46–62 (2015)
4. Chambers, W.G.: Clock-controlled shift registers in binary sequence generators. IEE Proceedings E - Computers and Digital Techniques **135**(1), 17–22 (1988)
5. Franklin, J.N.: Equidistribution of matrix-power residues modulo one. Math. Comput. **18**(88), 560–568 (1964)
6. Ghorpade, S.R., Hasan, S.U., Kumari, M.: Primitive polynomials, singer cycles, and word-oriented linear feedback shift registers. Des. Codes Cryptogr. **58**(2), 123–134 (2011)
7. Ghorpade, S.R., Ram, S.: Block companion singer cycles, primitive recursive vector sequences, and coprime polynomial pairs over finite fields. Finite Fields Appl. **17**, 461–472 (2011)
8. Golić, J.D.: On decimation of linear recurring sequences. Fibonacci Quart. **33**(5), 407–411 (1995)
9. Gollmann, D., Chambers, W.G.: Clock-controlled shift registers: a review. IEEE J. Sel. Areas Commun. **7**(4), 525–533 (1989)
10. Grothe, H.: Matrix generators for pseudo-random vector generation. Stat Papers. **28**, 233–238 (1987)
11. Hasan, S.U., Panario, D., Wang, Q.: Word-oriented transformation shift registers and their linear complexity. In: Helleseth, T., Jedwab, J. (eds.) Proceedings of SEquences and Their Applications - SETA 2012. Lecture Notes in Comput. Sci., vol. 7280, pp. 190–202. Springer, Berlin (2012)
12. Jansen, C.J.A.: Stream cipher design based on jumping finite state machines. Cryptology eprint archive: Report 2005/267. http://eprint.iacr.org/2005/267 (2005)
13. Jansen, C.J.A., Helleseth, T., Kholosha, A.: Cascade jump controlled sequence generator and Pomaranch stream cipher (Version 2). eSTREAM, ECRYPT Stream Cipher Project Report 2006/006 (2006)
14. Jiang, Y., Yang, J.: On the number of irreducible linear transformation shift registers. Des. Codes Cryptogr. **83**, 445–454 (2017)
15. Niederreiter, H.: Factorization of polynomials and some linear-algebra problems over finite fields. Linear Algebra Appl. **192**, 301–328 (1993)
16. Niederreiter, H.: The multiple-recursive matrix method for pseudorandom number generation. Finite Fields Appl. **1**, 3–30 (1995)
17. Niederreiter, H.: Pseudorandom vector generation by the multiple-recursive matrix method. Math. Comp. **64**, 279–294 (1995)
18. Niederreiter, H.: Improved bound in the multiple-recursive matrix method for pseudorandom number and vector generation. Finite Fields Appl. **2**, 225–240 (1996)
19. Paterson, K.G.: Root counting, the DFT and the linear complexity of nonlinear filtering. Des. Codes Cryptogr. **14**, 247–259 (1998)
20. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. IEEE Trans. Inform. Theory **24**, 106–110 (1978)
21. Preneel, B.: Introduction to the Proceedings of the Second Workshop on Fast Software Encryption. Lecture Notes in Comput. Sci., vol. 1008, pp. 1–5. Springer, Berlin (1995)
22. Ram, S.: Enumeration of linear transformation shift registers. Des. Codes Cryptogr. **75**, 301–314 (2015)
23. Rueppel, R.A.: Analysis and Design of Stream Ciphers. Springer, Berlin (1986)
24. Tsaban, B., Vishne, U.: Efficient feedback shift registers with maximal period. Finite Fields Appl. **8**, 256–267 (2002)
25. Zeng, G., Han, W., He, K.: Word-oriented feedback shift register: σ-LFSR. Cryptology eprint archive: Report 2007/114. http://eprint.iacr.org/2007/114 (2007)
26. Zeng, G., Yang, Y., Han, W., Fan, S.: Word-oriented cascade jump Σ-LFSR. In: Bras-Amorós, M., Høholdt, T. (eds.) AAECC 2009. Lecture Notes in Comput. Sci., vol. 5527, pp. 127–136. Springer, Heidelberg (2009)