On Binary Sequences Generated by Self-clock Controlled LFSR

Michele Elia, Guglielmo Morgari and Maria Spicciola

Abstract— **The paper considers some peculiar properties of binary sequences generated by self-clocked linear feedback shift registers of maximum length, and compares these properties with those of truly random sequences. In particular it examines their periods, their 0-1 distributions, and their linear complexity profiles.**

I. Introduction

The cryptographic qualities of stream ciphers depend on the mechanisms used to generate long binary sequences starting from short blocks of bits. These mechanisms are usually described in terms of finite state machines. The literature concerning the generation of binary sequences is vast and wide-ranging with many profound and elegant results, see [1], [4], [6], [12] and the references therein, however, the quest for finite state machines that generate sequences demonstrated to satisfy all cryptographic requirements is far from being complete. In [13], Rueppel investigated the properties of self-decimated sequences produced by Linear Feedback Shift Registers (LFSR), and drew the conclusion that these sequences may have some applications in cryptography and spread spectrum communication. A slightly different approach to studying self-decimation is described in [3], where emphasis is set on the decimation of the LFSR states, rather than of the sequence itself; these two approaches are closely related, but not totally equivalent. This paper focus on state decimation, looking at its interplay with linear codes and their "puncturing", and compares some peculiar properties of self-clocked binary generated sequences with those of truly random sequences. In particular it examines their *period*, their 0-1 distributions, and their *linear complexity profile* (LCP), which is defined as the length of the shortest LFSR that generates the subsequence up to length n for every n.

Let $\mathbf{u}(n)^T = [u_1(n), \ldots, u_m(n)]$ be a m-dimensional binary vector representing the state of a shift register at step n. The state transition of a self-clock controlled LFSR, hereafter shortened as scLFSR, is described by an $m \times m$ square matrix $S(n)$, which eventually depends on n and the current machine state $\mathbf{u}(n)$. The state evolution of a scLFSR is the sequence of states obtained as

$$\mathbf{u}(n+1) = S(n)\mathbf{u}(n) \quad .$$

Assume that the binary sequence

$$\mathbf{X} = [x(0), x(1), \ldots, x(L-1), \ldots]$$

is produced by an output function defined as $x(n) = u_k(n)$, where k is the index ($1 \le k \le m$) of a LFSR cell, called *output cell*, thus $\mathbf{X}$ is ultimately periodic of period L, same period as the sequence of states $\mathbf{u}(n)$. A LFSR, with primitive generator polynomial $g(z)$ of degree m, and transition matrix G (the companion matrix of $g(z)$), produces a sequence $\mathbf{Y}$ such that any block of $L = 2^m - 1$ consecutive symbols is a code word of a dual Hamming code $(2^m - 1, m, 2^{m-1})$ [9]. In this situation, the linear complexity of the sequence $\mathbf{Y}$ is m, and both generator polynomial and current state of the LFSR are easily computed from $2m$ consecutive symbols of $\mathfrak{Y}$, by means of the Berlekamp-Massey algorithm [10]. This linear complexity profile is (cryptographically speaking) an exceedingly small constant, but it can be definitively improved by self-clock controlling the LFSR; the bit used to control the clock can be taken from any register cell in a fixed position h ($1 \le h \le m$), which is called the *control cell*, and we assume

$$S(n) = \begin{cases} G & if \quad \text{the } h\text{-entry in } G\mathbf{u}(n) \text{ is } 0 \\ G^2 & if \quad \text{the } h\text{-entry in } G\mathbf{u}(n) \text{ is } 1 \end{cases} \quad .$$

The visible result of this expedient is a cancellation of elements from the sequence $\mathbf{Y}$, an operation, called puncturing, that yields a sequence $\mathbf{X}$ with shorter period but, in general, with a good linear complexity profile. To study these properties the paper is organized as follows. Next Section collects, for easy reference, general notions and preliminary results concerning self-clock controlled LFSR that are mainly drawn from the literature. Section 3 presents the main results concerning 0-1 distributions in particular for Fibonacci and Galois structures. Section 4 deals with the linear complexity profiles of truly random sequences and self-clocked LFSR sequences. Section 5 collects numerical results and mentions some open problems.

II. Preliminaries

Let $g(z) = z^m + g_1 z^{m-1} + \ldots + g_{m-1} z + g_m$ be a binary primitive polynomial with a companion $m \times m$ matrix

Michele Elia, Politecnico di Torino, Italy, elia@polito.it
Guglielmo Morgari, Maria Spicciola, Telsy, Torino, Italy, guglielmo.morgari@telsy.it, maria.spicciola@telsy.it

ISBN 978-963-311-370-7

G that can be written in the form

$$G = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ g_m & g_{m-1} & g_{m-2} & \cdots & g_2 & g_1 \end{bmatrix} .$$

The Hamming weight w_g of $g(z)$ is the number of its non-zero coefficients. Assuming that $\mathbf{u}(n)$ is the state of a LFSR at step n, then it is said to be of Fibonacci-type or Galois-type, if the next state is obtained as $\mathbf{u}(n+1) = G\mathbf{u}(n)$, or $\mathbf{u}(n+1) = G^T\mathbf{u}(n)$, respectively.
The set $\mathfrak{Y}$ of sequences $\mathbf{Y}$ generated by a LFSR of length m is a vector space of dimension m, which has the same dimension of the vector space of blocks of $L = 2^m - 1$ consecutive bits which are code words $Y(z) = \sum_{n=0}^{L-1} y(n)z^n$ of a dual Hamming code with generator polynomial

$$g^{\perp}(z) = \frac{z^L - 1}{g(z)} .$$

Since a dual Hamming code is a cyclic code, the cyclic shift of any code word is still a code word, if we consider the generator polynomial $g^{\perp}(z)$, the m code words obtained as $z^j g^{\perp}(z) \bmod (z^L - 1)$ for $j = 1, \ldots, m$ are linearly independent, thus form a basis for the code, and correspondingly for the sequence set $\mathfrak{Y}$. These m code words are generated by a Fibonacci-type LFSR with generator polynomial $g^{\perp}(z)$, and initial states $\mathbf{u}_i(0)$, each vector having every entry $u_k(0)$ equal zero for $k \neq i$, except $u_i(0)$ which is equal to 1.
An equivalent basis $\mathfrak{B}$, produced by a Galois-type LFSR requires the definition of a set of polynomials $t_j(z)$, $j = 0, \ldots, m-1$, where each $t_j(z)$ is the product of z^j by the polynomial obtained truncating $g(z)$ at the z-power j. We have

$$Y_1(z) = t_0(z)Y_0(z), Y_2(z) = t_1(z)Y_0(z) \bmod (z^L - 1), \ldots$$
$$Y_m(z) = t_{m-1}(z)Y_0(z) \bmod (z^L - 1)$$

where

$$Y_0(z) = z^{-m}Y_m(z).$$

Note that every $t_j(z)$ can be written as a power of $z^{\ell(t_j)}$ modulo $g(z)$. It is straightforward to check that $Y_1(z)$, $Y_2(z)$, ... , $Y_m(z)$, are linearly independent modulo $z^L - 1$, which in turn implies that $z^{\ell(t_0)}$, $z^{\ell(t_1)}$, ... , $z^{\ell(t_m)}$ are linearly independent modulo $g(z)$.
Every non-zero code word of a dual Hamming code has constant weight 2^{m-1}, and the number of zeros is $2^{m-1} - 1$. Important is the distribution of the runs of 1s and 0s in every code word. A run of 1s of length k in a binary sequence consists of k consecutive 1s between two 0s, and a run of 0s is similarly defined with the role of 0 and 1 exchanged. In [6] the 0-1 run distributions, which are the same in any code word of a dual Hamming code, are derived

- 1 run of length m of '1s', and
  0 runs of length m of '0s'
- 0 run of length $m-1$ of '1s', and
  1 runs of length $m-1$ of '0s'
- 2^{m-k-2} runs of length k, of either '0s' or '1s' for $1 \leq k \leq m-2$.

(1)

A. Clock-controlling and puncturing

In [13], Rueppel considered $[d, k]$ self-decimated sequences of a binary LFSR, which determines its own clock in the following way:

> *whenever the output symbol is '0', d clock pulses are applied to the LFSR, and, in the case the output symbol id '1', k clock pulses are applied to the LFSR.*

Given a sequence $\mathbf{Y} = \{y(0), y(1), \ldots, y(n), \ldots\}$ produced by a LFSR, the effect of decimation is to keep every d or k bits depending on the value of the last bit taken. In practice the generated sequence of bits is altered, and the period length is shortened. When d=1 and k=2, a different way to look at the clocked sequence is puncturing, that is to exclude every bit that follows a '1' whenever '1' occurs. An equivalent look is to implement the self-clock control considering the sequence of states $\{\bar{\mathbf{u}}(0), \bar{\mathbf{u}}(1), \ldots, \bar{\mathbf{u}}(n), \ldots\}$ of the LFSR and puncturing the states that correspond to excluded bits. Punctured states are thus excluded in the state sequence $\{\mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(n), \ldots\}$ of a scLFSR, yielding a shorter periodic sequence of states. An advantage of this view is that the output bit and the control bit can be taken from different cells of the register. Assuming that the cells are numbered from right to left, let the output cell be in position I, and the control cell be in position J. Let $c(0), c(1), \ldots, c(n), \ldots$, be the control sequence, then, denoting with $G\mathbf{u}(n)|_J$ the J-entry in vector $G\mathbf{u}(n)$, at step n, the clock-control bit is $c(n) = G\mathbf{u}(n)|_J$, and the output bit is $x(n) = u_I(n)$.

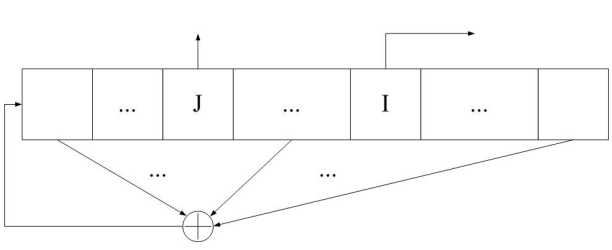

Fig. 1. Clock-controlled LFSR Fibonacci-type: I output cell, J clock control cell

It follows that at step n the state $\mathbf{u}(n)$ of a self-clock controlled (1-2) LFSR is

$$\mathbf{u}(n) = [(1 - c(n-1))G + c(n-1)G^2]\mathbf{u}(n-1),$$

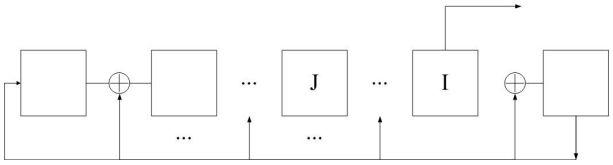

Fig. 2. Clock-controlled LFSR Galois-type: I output cell, J clock control cell

which leads to the expression

$$\mathbf{u}(n) = \prod_{i=0}^{n-1} [(1 - c(n-1))G + c(n-1)G^2]\mathbf{u}(0) \qquad (2)$$

in terms of the initial state $\mathbf{u}(0)$, and the matrix G. Therefore, a self-clock controlled LFSR is equivalent to a non-linear feedback shift register, with eventually a shorter period T that can be computed elementarily as follows.

Since the period length does depend neither on the control cell nor on the output cell, for counting purposes, we assume $J = I = m$. Therefore the only punctured positions are occupied by '1', and the number of punctured states is equal to the number of discarded '1' from the original sequence. Then, the number of '0's in the punctured sequence is $C_0 = 2^{m-1} - 1$. Since the punctured positions in a run of '1's are the first, the third, and generally any odd position, using the run distributions of '1' given in (1), it is immediately seen that the number of punctured '1' is

$$\begin{aligned} D_1 &= \sum_{k=1}^{m-3} 2^{m-k-2} \left\lceil \frac{k}{2} \right\rceil + \left\lceil \frac{m}{2} \right\rceil + \left\lceil \frac{m-2}{2} \right\rceil \\ &= \frac{2^{m-1} - 2 - 2\delta}{3} \quad , \end{aligned}$$

where δ is equal to $2^m - 1$ modulo 3, that is $\delta = 0$ if m is even, and $\delta = 1$ if m is odd, thus the number of '1' in the punctured sequence is

$$C_1 = 2^{m-1} - D_1 = \frac{2^m - 1 + 2\delta}{3} \quad .$$

In conclusion, the period N of the punctured sequence is

$$N = \frac{2}{3}(2^m - 1) - \frac{2}{3}\delta = \frac{2}{3}\left(2^m - \frac{3 - (-1)^m}{2}\right) \quad .$$

III. 0-1 Distributions in clock-controlled LFSR sequences

The numbers of '0' and '1' in a sequence, generated by a self-clock controlled LFSR, depend on both the relative position of control and output cells, and the implementation LFSR-type, namely Fibonacci or Galois.

A. Fibonacci-type LFSR sequences

A Fibonacci-type LFSR is shown in Figure 1, the output sequence does not depend on the cell because the output of any cell is a delayed version of the output of the first cell. Considering the clock controlled version, the period of every generated sequence does not depend on the clock control position, although, the number of zeros and ones depend on the relative positions of the output cell and the clock control cell. For computational purposes, it is convenient to fix the control cell position $J = m$, and letting the output cell position I to vary from m to 1.

Let N_{0I} and N_{1I} denote the number of 0s and 1s in a period of a sequence, respectively; obviously, we have $N_{0I} + N_{1I} = T$, where T is the period. The approach is to count the number n_{0I} of 0s in punctured positions of an underlying maximum length sequence that has $2^{m-1} - 1$ zeros, then $N_{0I} = 2^{m-1} - 1 - n_{0I}$.

If $I = m$, then $n_{0m} = 0$ because punctured positions are occupied only by "1"s, thus

$$N_{0m} = 2^{m-1} - 1 \quad .$$

If $I = m - 1$, then 0 occupies a punctured position if the pattern 01 occurs in positions $[m-1, m]$, since the number of patterns 01 in a m-sequence is 2^{m-2}, this is also the number of punctured 0s, thus $n_{0m-1} = 2^{m-2}$, and

$$N_{0m-1} = 2^{m-1} - 1 - 2^{m-2} \quad .$$

If $I = m-2$, then 0 occupies a punctured position if the pattern 001 occurs in positions $[m-2, m-1, m]$, and no other pattern determines a puncturing of a 0, since the number of patterns 001 in a m-sequence is 2^{m-3}, this also is the number of punctured 0s, thus $n_{0m-2} = 2^{m-3}$, and

$$N_{0m-2} = 2^{m-1} - 1 - 2^{m-3} \quad .$$

If $I = m - 3$, then 0 occupies a punctured position if one of the patterns 0001, 0101, and 0111 occur in positions $[m-3, m-2, m-1, m]$, since the number of each pattern of four bits in a m-sequence is 2^{m-4}, the number of punctured 0s is thus $n_{0m-2} = 3 \cdot 2^{m-4}$, and

$$N_{0m-3} = 2^{m-1} - 1 - 3 \cdot 2^{m-4} \quad .$$

Clearly, the general form of the number of zeros is

$$N_{0I} = 2^{m-1} - 1 - a(m-I) \cdot 2^{m-(m-I)-1}$$

where $a(I)$ is the number of patterns of length $m-I-2$ that can be formed with patterns of the form 0, 01, and 11, thus satisfies the recurrence

$$a(k) = a(k-1) + 2a(k-2)$$

with initial conditions $a(0) = 0$, $a(1) = 1$. The sequence is thus a Jacobsthal sequence [14], and has the form

$$a(k) = \frac{2^k - (-1)^k}{3}$$

In conclusion we have the closed expressions

$$\begin{cases} N_{0,m-I} = \dfrac{1}{3}2^m - 1 + \dfrac{1}{6}(-1)^{m-I}2^I \\ \qquad\qquad\qquad\qquad\qquad\qquad I = 0,\dots,m-1 \\ N_{1,m-I} = \dfrac{1}{3}2^m + \dfrac{1}{3}(-1)^m - \dfrac{1}{6}(-1)^{m-I}2^J \end{cases}$$

where $J = m$ is the clock control cell position, and I is the output cell position.
Using the closed form of the number of zeros and ones, it is immediately seen that the clocked sequence is perfectly balanced, i.e. $N_{0I} = N_{1I}$, if and only if $I = 1$ if m is odd, and $I = 2$ if m is even.

B. Galois-type LFSR sequences

A Galois type implementation of a LFSR is shown in Figure 2, the output sequence depends on the cell in the sense that it is a delayed version the sequence obtained from another cell, but the amount of the delay depends on the tap distribution. Also the 0-1 distribution depends on the positions of the taps.
Let $1 < i_1 < i_2 < \cdots < i_s < m$ be the positions of the taps, and let J be the position of the bit specifying the clock control, therefore

- If $i_s < J \leq m$, the distribution (N_{0J}, N_{1J}) is the same as in the case of Fibonacci-type LFSR.
- If $i_1 > J \geq 1$, the distribution (N_{0J}, N_{1J}) is the same as in the case of Fibonacci-type LFSR.
- If $J = 1$ then $\big(N_{0J} = T - N_{1J}, N_{1J} = 2^{m-1} - a(m-1)\big)$ as for Fibonacci type.
- If $J = m$ then $\big(N_{0J} = 2^{m-1} - 1, N_{1J} = T - N_{0J}\big)$ as for Fibonacci type.
- If $i_1 \leq J \leq i_s$ the distribution (N_{0J}, N_{1J}) is almost balanced, with very small discrepancies, but we have not been able to find closed form expressions for the related distributions.

IV. LCPs of binary sequences

An analysis on the importance of the Linear Complexity Profile (LPC) of a binary stream can be found in [12], and we refer to that an exposition for the main properties of LPC.
Let $\mathfrak{X} = x(0), x(1), \dots, x(n), \dots$ be a binary sequence, possibly periodic of period T.
If $\mathfrak{X}$ is a truly random binary sequence, its LCP is a step function with steps of variable height and variable duration, such that the function grows oscillating around a straight line of slope $\frac{1}{2}$. Of course, this behavior is an average behavior, therefore, it is useful to know the expected LCP of a truly random sequence.

A. Expected LCP of random sequences

Let $\mathfrak{X}_n$ be a subsequence of length n of an infinite truly random sequence $\mathfrak{X}$, and let $\ell = \ell(\mathfrak{X}_n)$ denote the length of the shortest recurrence generating $\mathfrak{X}_n$. Let $g_\ell(x) = x^\ell + a_1 x^{\ell-1} + \cdots + a_{\ell-1}x + a_\ell$ be the generator polynomial of $\mathfrak{X}_n$.

The linear complexity profile of $\mathfrak{X}_n$ is defined as the function $\ell(n,k) = \ell(\mathfrak{X}_k)$ for every k from 0 up to n. In order to compute the expectation of $\ell(\mathfrak{X}_n)$, for every n, it is necessary to define a probability measure over $\mathcal{X}_n = \{\mathfrak{X}_n\}$ the set of all binary sequences of length n. We say that a sequence $\mathfrak{X}_n$ is randomly generated if it is picked at random from $\mathcal{X}_n$ with probability $p\{\mathfrak{X}_n\} = \frac{1}{2^n}$, since $|\mathcal{X}_n| = 2^n$. This definition is tantamount to consider a sequence $\mathcal{X}_n$ as produced bit by bit with bit probability $\frac{1}{2}$.
Let $c(n,k)$ denote the number of sequences in $\mathcal{X}_n$ that are generated by a recurrence of order k, therefore, the expectation of $\ell(\mathfrak{X}_n)$ can be written as

$$E[\ell(\mathfrak{X}_n)] = \sum_{\mathfrak{X}_n \in \mathcal{X}_n} \frac{1}{2^n}\ell(\mathfrak{X}_n) = \frac{1}{2^n}\sum_{k=0}^{n} k c(n,k) \quad .$$

The last summation is easily computed, taking into account the following observations:

- Every generator polynomial of degree k is allowed, including x^k, [10], which is assumed to generate the all zero sequence.
- The sequence $0\cdots 0,1$ composed of $n-1$ zeros followed by a 1 is necessarily generated by a recurrence of degree n, [10], therefore $c(n,n) = 1$ since any other sequence of length n is generated by some LFSR of length less than n [10].
- $c(n,0) = 1$ is a consequence of the previous observation.
- $c(1,1) = 1$ since we have only the sequence "1" given that the sequence "0" is generated by a recurrence of order 0, and $c(n,1) = 2$, $n > 1$ since the recurrence with generator polynomial $x + 1$ generates two sequences, namely the all zeros and the all ones sequences, but the all zeros sequence is generated by a recurrence of order 0 by definition, and the sequence 01 cannot be generated by a recurrence of order 1.
- if $n > 2k$ and $k \geq 0$, then $c(n,k) = c(n-1,k) = c(2k,k)$, because any periodic sequence generated by a LFSR of length k is specified by its first $2k$ digits and the sequences longer than $2k$ are the periodic extensions of some sequence generated by a LFSR of length k.
- $c(2,2) = 1$ and accounts for the sequence 01. $c(3,2) = 4$ is obtained as difference

  $$c(3,2) = 2^3 - [c(3,0) + c(3,1) + c(3,3)] = 4 \quad ,$$

  moreover $c(n,2) = c(4,2) = 8$ for every $n \geq 4$, where $c(4,2)$ is obtained by direct counting, or repeating the same argument used above for evaluating $c(3,2)$.
- We have the recurrence $c(2k,k) = 4c(2(k-1),k-1)$ because adding one cell to a LFSR we have at disposal one more initial condition and one more tap, therefore

  $$c(2k,k) = 2^{2k-1} \quad k \geq 1 \quad .$$

- If $2k > n$, then $c(n,k) = 4^{n-k}$ for every $\lfloor \frac{n}{2} \rfloor + 2 \leq k \leq n-1$, $n > 2$

An initial set of values of $c(n,k)$ are reported in Table 1.

$n\backslash k$	0	1	2	3	4	5
1	1	1	-	-	-	-
2	1	2	1	-	-	-
3	1	2	4	1	-	-
4	1	2	8	4	1	-
5	1	2	8	16	4	1

The average (or expectation) of $\ell(\mathfrak{X}_n)$ is obtained as

$$E[\ell(\mathfrak{X}_n)] = \frac{1}{2^n}\sum_{k=0}^{n} kc(n,k) =$$

$$= \begin{cases} \frac{n}{2} + \frac{4}{18} - \frac{3n+2}{9}2^{-n} & \text{even } n \\ \frac{n}{2} + \frac{5}{18} - \frac{3n+2}{9}2^{-n} & \text{odd } n\ , \end{cases}$$

while the mean square deviation σ_n^2, computed through the average of the squares $E[\ell(\mathfrak{X}_n)^2]$, is

$$\sigma_n^2 = \begin{cases} \frac{86}{81} - \frac{42n+82}{81}2^{-n} - \frac{(3n+2)^2}{81}2^{-2n} & \text{even } n \\ \frac{86}{81} - \frac{39n+80}{81}2^{-n} - \frac{(3n+2)^2}{81}2^{-2n} & \text{odd } n\ . \end{cases}$$

It is immediately seen that the mean square deviation for separately even and odd m is monotonically increasing, and in both cases asymptotically approaches $\sigma = \frac{86}{81}$. Whereas, the average profile asymptotically slightly depends on n parity

$$E[\ell(\mathfrak{X}_n)] \asymp \begin{cases} \frac{n}{2} + \frac{4}{18} & \text{even } n \\ \frac{n}{2} + \frac{5}{18} & \text{odd } n\ . \end{cases}$$

Let us remark that the square deviation of a linear complexity profile of a truly random sequence of length n from the straight line is approximately $n\frac{86}{81}$.

B. LCP of self-clocked LFSR sequences

A "random" periodic sequence $\mathfrak{R}$ of period T is obtained by repeating the same block of T consecutive truly random bits. The LCP of $\mathfrak{R}$ is a step function that grows around a straight line of slope $\frac{1}{2}$, for n from 1 up to $2T$, then it assumes the constant value T for every $n > 2T$.

This typical LCP can be assumed as a reference profile for evaluating the goodness, for cryptographic purposes, of the LCP of sequences deterministically generated.

Let $\mathfrak{Y} = y(0), y(1), \ldots, y(n), \ldots$ be a binary sequence generated by a LFSR of length m, its LCP is a step function that grows linearly with the length n up to the value m, as n varies from 1 to $2m$, thus it assumes the constant value m for every $n > 2m$.

An heuristic argument suggests that the LCP of binary sequences, generated by self-clock controlled LFSR, are nearly optimal, in the sense that have the same trend as the LCP of a random periodic sequence. Let $\mathfrak{X} = x(0), x(1), \ldots, x(n), \ldots$ be the clocked sequence obtained puncturing $\mathfrak{Y}$, then $x(n) = y(t(n))$, we may argue as follows:

> suppose that $y(t(n)+1)$ is not punctured then $x(n+1) = y(t(n)+1)$ with probability $1/2$ because the sequence $\mathfrak{Y}$ appears to be a random sequence. If $x(n+1) = y(t(n)+1)$ then $L_n = L_{n-1}$, if $x(n+1) = y(t(n)+1)+1 \bmod 2$, then $L_n = n+1-L_{n-1}$ by Massey criterion [10]. Therefore on the average we expect to have
>
> $$E[L_n] = \frac{1}{2}L_{n-1} + \frac{1}{2}(n+1-L_{n-1}) = \frac{n}{2}.$$

Thus, the expected LCP is a step function that growths around a straight line of slope $\frac{1}{2}$ on the average. This fact has also been verified numerically as reported in our conclusions; however heuristic observations and numerical facts have suggested and stimulated the search for a theoretical proof.

Lemma 1: In a (n,k,d) cyclic code any k consecutive positions of the code words can be taken as information positions. Equivalently, any $k \times k$ sub-matrix of the generator matrix formed with k consecutive rows has full rank.

PROOF. The first statement follows from two facts

- a cyclic code with generator polynomial $g(x)$ of degree $r = n-k$ admits a systematic encoding as $c(x) = I(x)x^r + r(x)$, where $I(x)$ is a polynomial of information symbols of degree $k-1$, and $r(x)$ is the remainder of the division of $I(x)x^r$ by $g(x)$.
- the k bit in the highest position can be shifted to occupy any k consecutive positions, starting from position $t \geq k$, by the multiplication $x^t c(x) \bmod x^n - 1$.

The second statement is an immediate consequence of the first statement. ■

The following Lemma is an immediate consequence of the definition of cyclic codes.

Lemma 2: Any sub-code (n,k',d') of a cyclic code (n,k,d) is cyclic.

Let $\mathcal{C}^{(m)} = (2^m - 1, m, 2^{m-1})$ be a dual Hamming code of dimension m. Consider the punctured code $\mathcal{P}^{(m)} = (\frac{2}{3}(2^m - \frac{3-(-1)^m}{2}), m, d)$ obtained self-clocking $\mathcal{C}^{(m)}$, and let $\mathcal{C}_n$ be a code, of length n, whose code words $\mathfrak{X}_n$ are the initial part of the code words of $\mathcal{P}^{(m)}$.

Suppose that $\mathcal{C}_n$ is a cyclic code of length n, dimension k_n, and minimum distance d_n, then $k_n \leq m$, because the entries in each code word are linear combination of at most m entries as in the code $\mathcal{C}^{(m)}$.

Let us preliminary recall that the number of primes dividing $2^{k_n}-1$ or $2^{k_n}-2$ is less than $2k_n$, then the number of primes dividing $2^{k_n}-1$ or $2^{k_n}-2$ for every $k_n \leq m$ is not greater than $m(m+1)$, thus these primes are a very small fraction of $\frac{2}{3}2^m$.

Theorem 1: Let $\mathfrak{X}$ be a binary sequence obtained by puncturing a sequence $\mathfrak{Y}$ produced by a linear feedback shift register of length m and primitive generator polynomial $g(x)$. The LCP of $\mathfrak{X}$ is very close to the LCP of a random binary sequence.

PROOF. The sequence $\mathfrak{X}$ is a code word of $\mathcal{P}^{(m)}$, and its periodic version is certainly generated by a LFSR of length $T = \frac{2}{3}\left(2^m - \frac{3-(-1)^m}{2}\right)$. It will turn out that such a length cannot be smaller, proving that the LCP of $\mathfrak{X}$ is close to that of a random sequence.

The LCP of $\mathfrak{X}$ is obtained by computing the linear complexity of $\mathfrak{X}_n$, for every n from 1 up to T.

Assume that $n > 2m$, since the LCP for small n is not relevant. If $\mathfrak{X}_n$ is generated by a shift register of length $L_n < n/2$, then $\mathfrak{X}_n$ is a code word of a cyclic code $\mathcal{C}^{L_n}$ of dimension L_n, however, it belongs also to a subcode $\mathcal{C}_n$ of dimension m. Therefore, considering this subcode $\mathcal{C}_n$, and letting n to run in the sequence of prime numbers up to T, then $n|2^m-1$ or $n|2^m-2$. These conditions are a consequence of Theorem 8.14 [11, p.246] stating that in a cyclic code of length n we have $iA_i = n\beta_i$, where A_i is the number of code words of Hamming weight i. If n is prime then necessarily $n|A_i$, unless $i=0$ or $i=n$, in which case we have $A_0 = 1$, and possibly $A_n = 1$. Thus, if the code dimension is m, then n is a divisor of either 2^m-1 or possibly 2^m-2 Since only few primes, precisely less than $2m$, are factors of either 2^m-1 or 2^m-2, it follows that L_n cannot be less than $n/2$, for the majority of primes n less than T. The relatively regular growth is guaranteed by the prime distribution as a consequence of the so-called Bertrand's Postulate, that is if $n > 1$, there is at least one prime p such that $n < p < 2n$ [7, p.343]. This fact forces the LCP to be close to $n/2$ for almost every n, although the deviation from $n/2$ may be large. ■

V. NUMERICAL RESULTS AND CONCLUSIONS

In this paper we have presented several properties of the 0-1 distributions in sequences generated by self-clock controlled LFSR. In conclusion, the effect of puncturing $\mathfrak{Y}$, by a self-clocked mechanism, is to shorten the sequence from $L = 2^m - 1$ to $T \approx \frac{2}{3}2^m$, and contemporarily to increase the linear complexity from m to T, with a linear passage from m to $\frac{2}{3}2^m$. These properties have been either theoretically proved or motivated by heuristic arguments, and have been numerically checked for LFSRs of length up to 22. The LCP as a step function is made of steps of different size, and floors also of different extension, then we have reported the distribution of steps from 0 up to an observed maximum of 23, and the floor size from the minimum value 1 up to an observed maximum of 26. We believe that this synthetic description is quite significant, moreover we have observed that its shape is independent from the register length.

The empirical 0-1 distributions are in perfect agreement with the theoretical ones found in Section 3. Also the LCPs of binary self-clocked sequences as described inSection 4 have been checked for sequence lengths up to 2^{30} and several register lengths as large as 80, and the theoretical results are empirically confirmed. Note that, the periods of practical sequences are very large, thus the properties theoretically proved cannot be exhaustively checked. In Figures 3 and 4 are shown the linear complexity profile for register lengths 5 and 22. In particular, the LCP for the register length 22 is indistinguishable from the LCP of a random sequence of the same length, and due to the scale squeezes, it is practically a straight line. Steps and floors are visible in the LCP of a register of length 5.

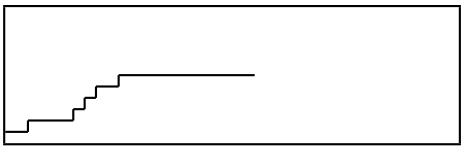

Fig. 3. LCP for a Fibonacci-type scLFSR of length 5

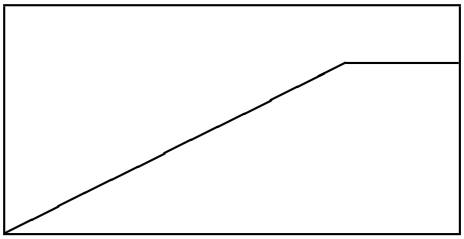

Fig. 4. LCP for a Fibonacci-type scLFSR of length 22

REFERENCES

[1] J.P. Allouche, and J. Shallit, *Automatic Sequences*, Cambridge University Press, Cambridge, 2003.

[2] M. Coppola, private communication

[3] M. Elia, E. Viterbo, "Linear sequences and Punctured linear codes", *Proceedings SIC' 2000*, 23-28 July 2000, Orlando, FLORIDA, pp.660-665.

[4] G. Everest et alii, *Recurrence Sequences*, AMS, Providence, 2003.

[5] J. Dj. Golić, private communication

[6] S.W. Golomb, *Shift Register Sequences*, Aegean Park Press, Laguna Hills, 1982.

[7] G.H. Hardy, E.M. Wright, *An Introduction to the Theory of Numbers*, Oxford Press, 1971.

[8] D.E. Knuth, *The Art of Computer Programming*, Seminumerical algorithms, vol. II, Addison-Wesley, Reading Massachussetts, 1981.

[9] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North Hol., New York, 1977.

[10] J.L. Massey, Shift-Register Synthesis and BCH decoding, *IEEE Trans. on Inform. Th.*, IT-15, 1969, pp.122-127.

[11] W.W. Peterson, E.J. Weldon, *Error-Correcting Codes*, MIT Press, Cambridge (Mass.), 1981.

[12] R.A. Rueppel, *Analysis and Design of Stream Cipher*, Springer, New York, 1986.
[13] R.A. Rueppel, *When Shift Registers clock themselves*, Eurocrypt, 1987.
[14] N. Sloane, "The On Line Encyclopedia of sequences", www.research.att.com/∼njas/sequences/