The Linear Complexity of the LFSR Based Generalized Shrinking-Multiplexing Generator

Todor D. Tashev [1], Borislav Y. Bedzhev[2] and Zhaneta N. Tasheva[3]

***Abstract* – An architecture of Generalized Shrinking-Multiplexing Generator (*GSMG*), based on Linear Shift Feedback Registers (*LFSRs*), is investigated in the paper. The linear complexity of its output binary pseudo random sequences is established. Some linear complexity analysis is given. The established *GSMG* properties show that the proposed architecture allows producing binary pseudo random sequences with good properties like uniform distributions of 1s and 0s, unpredictable nonlinearity, enormous period and large linear complexity.**

***Keywords* – Cryptography, Pseudo Random Number Generator, Stream Cipher, Clock Controlled Generators, *LFSR*, Linear Complevity.**

I. INTRODUCTION

Nowadays, the clock controlled Pseudo Random Number Generators (*PRNGs*) are an important tool for development of stream ciphers, applied in the communication information systems. On the one hand, their high performance velocity and cost-effective implementation is based on their simple architecture which combines fast and cheap elements like Linear Feedback Shift Registers (*LFSRs*) and Feedback with Carry Shift Registers (*FCSRs*) with some nonlinear functions [2, 3, 5]. On the other hand, the performance quality of the clock controlled *PRNGs* [6, 7, 8] depends on their crypto resistance, which is connected with its ability to generate nonlinear Pseudo Random Sequence (*PRS*) with enormous period, uniform distribution and large linear complexity.

Due to this reason the aim of this paper is to investigate the linear complexity of a *LFSR* based *Generalized Shrinking-Multiplexing Generator* (*GSMG*). The paper is organised as follows. First, the *LFSR* based *GSMG* architecture is described. Second, the linear complexity of the *LFSR* based *GSMG* is established. After that some linear complexity analysis and a comparison with the Shrinking Generator are given. Finally, the advantages and possible application areas of the *LFSR* based *GSMG* are discussed.

[1] Eng. Todor D. Tashev, PhD. Student in Communication Network and Systems, Shumen University, Shumen, Bulgaria, Phone: +359 54 980 422, e-mail: todor_tashev@yahoo.com.

[2] Assoc. Prof. DSc. Eng. Borislav Y. Bedzhev, NMU "V. Levski", Faculty of Artillery, Air Defence and Communication-Information Systems, Shumen, Bulgaria, Phone: +359 54 88 64 38, e-mail: bedzhev@mail.pv-ma.bg.

[3] Assoc. Prof. PhD. Eng. Zhaneta N. Savova-Tasheva, NMU "V. Levski", Faculty of Artillery, Air Defence and Communication-Information Systems, Shumen, Bulgaria, Phone: +359 54 980 422, e-mail: tashevi86@yahoo.com.

II. THE LFSR BASED GSMG ARCHITECTURE

The proposed general architecture of the *GSMG* [9] can be realized by means of linear or nonlinear pseudo random sequences. There are eight possible variants of the *GSMG* architecture depending on the linear constructive elements, which most often are fast and cheap *LFSRs* and *FCSRs*. Most of these *GSMG* architectures are statistically analyzed by the authors of [11, 12, 13] but strong mathematical analysis have not been made yet. Here the fifth architecture, proposed in [9], is analyzed.

The *LFSR* based *GSMG* architecture (Fig. 1) uses as building modules *LFSRs*.

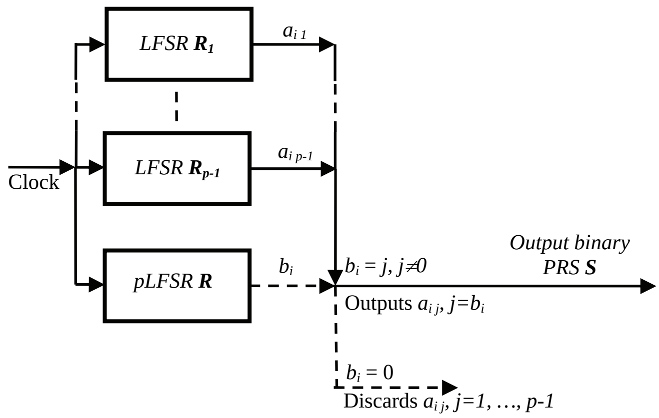

Fig. 1. The LFSR Based Generalized Shrinking – Multiplexing Generator

Definition 1: A *LFSR* based *GSMG* comprises a *pLFSR* ***R*** of length L which produces one p-ary number in a time and $p-1$ slaved *LFSRs* of length $L_1, L_2, \ldots, L_{p-1}$. The clock controls the movement of a data in all used *LFSRs*.

The algorithm of *LFSR* based *GSMG* consists of the following steps:

1. All slaved *LFSRs* and control *pLFSR* are clocked.
2. If the p-ary output of the control *pLFSR* ***R*** at moment i is non-zero ($b_i = j, j \neq 0$), the binary output of the slaved *LFSRs* $\boldsymbol{R_j}$ forms a part of the *LFSR* based *GSMG* output *PRS* ***S***.
3. Otherwise, if the output of the control *pLFSR* ***R*** is equal to 0 ($b_i = 0$), the outputs of all slaved *LFSRs* $\boldsymbol{R_1}, \boldsymbol{R_2}, \ldots, \boldsymbol{R_{p-1}}$ are discarded.

Therefore, the produced binary *PRS* is a *shrunken* version of the slaved binary *PRSs*, generated by the *LFSRs* $\boldsymbol{R_1} \div \boldsymbol{R_{p-1}}$, when the output of the control *pPRS* ***B*** is zero, and a *mixed* version of the slaved *PRSs*, when the output is nonzero. Due to this reason the output *LFSR* based *GSMG* sequence ***S*** is nonlinear and unpredictable with more complexity than the sequences, produced by the slaved *LFSRs*. The nonlinearity in the *LFSR* based *GSMG* architecture is a result of the fact that the linear algebraic structure of the slaved *LFSR* sequences is destroyed by means of the shrinking and multiplexing.

III. The Linear Complexity of the Sequences Generated by the LFSR Based GSMG

In this section it is proved the exponential bounds of the linear complexity of sequences, generated by the *LFSR* based *GSMG*. The importance of the exponentially large *PRS* linear complexity follows from the strong necessity of avoiding some popular attacks on *PRSs* or stream chippers. There is no need to know the way a *PRS* is generated in order to break it through its linear complexity. In fact, any *PRS* with linear complexity λ can be easily reconstructed if 2λ bits are known by the Berlekamp-Massey algorithm [1, 4], which in time $O(\lambda^2)$ finds the shortest *LFSR* generating this *PRS*.

Here it should be mentioned that the high linear complexity is only necessary but not sufficient condition *PRNG* to have good cryptographic properties. There are many others conditions like period; uniform distribution of *d*-tuples for a large range of *d*; good, usually lattice-liked, structure in high dimensions; good statistical properties; resistance to known attacks and so on.

A. Theoretical analysis

The following symbols are used when the linear complexity of the *LFSR* based *GSMG* sequence is established:

- L_i, $i = 1, 2, \ldots, p-1$ – length of the slaved *LFSR* $\boldsymbol{R_i}$;
- L - length of the control *pLFSR* ***R***;
- T_i, $i = 1, 2, \ldots, p-1$ – period of the slaved *LFSR* $\boldsymbol{R_i}$;
- T – period of the control *pLFSR* ***R***;
- T_s – period of the generated by the *LFSR* based *GSMG* sequence ***S***;
- $N_{(\neq 0)}$ – quantity of the nonzero elements in a period of the control *p*PRS sequence;
- $a_j(i)$ – the *i*-th element of the sequence $\boldsymbol{A_j}$, $j = 1, \ldots, p-1$, generated by the slaved *LFSR* $\boldsymbol{R_j}$;
- $b(i)$ – the *i*-th element of the sequence ***B***, generated by the control *pLFSR* ***R***;
- $s(i)$ – the *i*-th element of the sequence ***S***, generated by the *LFSR* based *GSMG*;
- $k_{i\,j}$ – *i*-th position with value *j* in the sequence ***B***, generated by the control *pLFSR* ***R***.

Also the following Theorem 1 determining the period T_s of the generated by the *LFSR* based *GSMG* sequence ***S*** is used. Refer to [10] to see the proof of the Theorem 1.

Theorem 1: If the generated by the slaved *LFSR* $\boldsymbol{R_j}$ sequence $\boldsymbol{A_j}$, $j = 1, \ldots, p-1$ and the generated by the control *pLFSR* ***R*** sequence ***B*** have maximal length (i.e. have primitive connections) and all periods T_i are co-prime with the period *T*, i.e. the greatest common devisor is $(T_i, T) = 1$ for $i = 1, 2, \ldots, p-1$, then the output shrinking and multiplexing sequence, generated by the *LFSR* based *GSMG*, has a maximal period defined by the equation:

$$T_S = N_{(\neq 0)} \cdot \prod_{i=1}^{p-1} T_i = N_{(\neq 0)} \cdot \prod_{i=1}^{p-1} 2^{L_j} - 1 \qquad (1)$$

The linear complexity λ_s of the sequence ***S*** generated by the *LFSR* based *GSMG* satisfies the following Theorem 2.

Theorem 2: If the generated by the slaved *LFSR* $\boldsymbol{R_j}$ sequence $\boldsymbol{A_j}$, $j = 1, \ldots, p-1$ and the generated by the control *pLFSR* ***R*** sequence ***B*** have maximal length (i.e. have primitive connections) and all periods T_i are co-prime with the period *T*, i.e. the greatest common devisor is $(T_i, T) = 1$ for $i = 1, 2, \ldots, p-1$, then the output shrinking and multiplexing sequence ***S***, generated by the *LFSR* based *GSMG*, has a linear complexity λ_s satisfied the inequality

$$\frac{(p-1)p^{L-1}}{2} \prod_{i=1}^{p-1} L_i < \lambda_S \leq (p-1)p^{L-1} \prod_{i=1}^{p-1} L_i \,. \qquad (2)$$

The next proposition, which follow from the definition of the *LFSR* based *GSMG*, is used in the proof.

Proposition 1: The integers $N_{(\neq 0)}$ and *T* are connected by equation

$$s(i + n.N_{(\neq 0)}) = a_j(k_i + nT), \quad \text{for} \ \ n = 0, 1, \ldots \,. \qquad (3)$$

Proof: To determine an *upper bound* on the linear complexity λ_s of the sequence ***S***, it is sufficed to find a polynomial $P(.)$, for which $P(s) = 0$, i.e. the coefficients of $P(.)$ represents a linear dependency satisfied by the elements of a sequence ***S***. Let $s^{N(\neq 0)}$ be the sequence $s(nN_{(\neq 0)}), n = 0, 1, \ldots$, i.e. the sequence ***S*** is decimated by $N_{(\neq 0)}$.

Proposition 1 states that this decimation results in transformations of every slaved sequence of the form $a_j(i + nT)$, $j = 1, 2, \ldots, p-1$. Since $(T_j, T) = 1$, $j = 1, 2, \ldots, p-1$, the above sequences $a_j(i + nT)$, $j = 1, 2, \ldots, p-1$ have maximal length and have the same linear complexity as the original sequences $a_j(i)$, $j = 1, 2, \ldots, p-1$. Therefore, the polynomials $Q_j(.)$ of degree L_i exist for which $Q_j(a_j) = 0$. But then the decimated sequence $s^{N(\neq 0)}$ satisfies polynomials $Q_j(.)$, i.e.

$$Q_j(s^{N(\neq 0)}) = 0, \ j = 1, 2, \ldots, p-1. \qquad (4)$$

Hence, a polynomial

$$P(s) = \prod_{j=1}^{p-1} Q_j(s^{N(\neq 0)}) \qquad (5)$$

of degree $N_{(\neq 0)} \prod_{i=1}^{p-1} L_i$, such that $P(s) = 0$, is found.

Consequently, the linear complexity λ_s of the sequence $\boldsymbol{S}$ generated by the *LFSR* based *GSMG* is at most

$$\lambda_S \le N_{(\neq 0)} \prod_{i=1}^{p-1} L_i = (p-1)p^{L-1} \prod_{i=1}^{p-1} L_i \ . \qquad (6)$$

To determine an *lower bound* on the linear complexity λ_s of the sequence $\boldsymbol{S}$, it is necessary to find the minimal polynomial $M(s)$ for which $M(s) = 0$. Since the sequence $\boldsymbol{S}$ satisfied the equation (4), then the polynomial $M(s)$ divides each polynomial $Q_j(s^{N(\neq 0)})$, $j = 1, 2, \ldots, p-1$. After putting the equation

$$N_{(\neq 0)} = (p-1)p^{L-1} \qquad (7)$$

in (4), the following equations are obtained

$$Q_j(s^{N(\neq 0)}) = Q_j(s^{(p-1)p^{p-1}}) =$$

$$= (Q_j(s))^{(p-1)p^{p-1}}, \ j = 1, 2, \ldots, p-1 . \qquad (8)$$

Therefore, the minimal polynomial $M(s)$ must be in the form $(Q_j(s))^r$ for $r \le (p-1)p^{p-1}$. The p is a prime number and hence, $p - 1$ is even.

The following assumptions will be made

$$r \le \frac{(p-1)p^{p-1}}{2} . \qquad (9)$$

Then the minimal polynomial $M(s)$ divides each polynomial $(Q_j(s))^{\frac{(p-1)p^{p-1}}{2}}$, $j = 1, 2, \ldots, p-1$. Since $Q_j(s)$, $j = 1, 2, \ldots, p-1$ are irreducible polynomials of degree ca L_j, they divide the polynomials $1 + x^{T_j}$ respectively.

TABLE I
LINEAR COMPLEXITY OF THE PRSS, GENERATED BY THE LFSR BASED GSMG, WITH P = 3

№	**Used *PRSs***	**Primitive Polynomials**	**Length**	**Period T_S**	**Linear Complexity of the *PRSs*, generated by the *LFSR* based *GSMG***		
					Lower Bound	**Upper Bound**	**Real**
1	*PRS* $\boldsymbol{A_1}$	$1 + x + x^3$	$L_1 = 2$	7.15.18 = 1890	3^2.3.4 = 108	108.2 = 216	126
	PRS $\boldsymbol{A_2}$	$1 + x + x^4$	$L_2 = 3$				
	3PRS $\boldsymbol{B}$	$1 + 2x^2 + x^3$	$L = 2$				
2	*PRS* $\boldsymbol{A_1}$	$1 + x + x^2$	$L_1 = 2$	3.15.18 / 3 = 270	3^2.2.4 = 72	72.2 = 144	108
	PRS $\boldsymbol{A_2}$	$1 + x + x^4$	$L_2 = 4$				
	3PRS $\boldsymbol{B}$	$1 + 2x^2 + x^3$	$L = 3$				
3	*PRS* $\boldsymbol{A_1}$	$1 + x + x^3$	$L_1 = 3$	7.3.18 = 378	3^2.2.3 = 54	54.2 = 108	90
	PRS $\boldsymbol{A_2}$	$1 + x + x^2$	$L_2 = 2$				
	3PRS $\boldsymbol{B}$	$1 + 2x^2 + x^3$	$L = 3$				
4	*PRS* $\boldsymbol{A_1}$	$1 + x + x^3$	$L_1 = 3$	7.15.6 = 630	3^1.3.4 = 36	36.2 = 72	42
	PRS $\boldsymbol{A_2}$	$1 + x + x^4$	$L_2 = 4$				
	3PRS $\boldsymbol{B}$	$2 + x + x^2$	$L = 2$				
5	*PRS* $\boldsymbol{A_1}$	$1 + x + x^2$	$L_1 = 2$	3.15.6 / 3 = 90	3^1.2.4 = 24	24.2 = 48	36
	PRS $\boldsymbol{A_2}$	$1 + x + x^4$	$L_2 = 4$				
	3PRS $\boldsymbol{B}$	$2 + x + x^2$	$L = 2$				
6	*PRS* $\boldsymbol{A_1}$	$1 + x + x^3$	$L_1 = 3$	7.3.6 = 126	3^1.2.3 = 18	18.2 = 36	30
	PRS $\boldsymbol{A_2}$	$1 + x + x^2$	$L_2 = 2$				
	3PRS $\boldsymbol{B}$	$2 + x + x^2$	$L = 2$				
7	*PRS* $\boldsymbol{A_1}$	$1 + x + x^3$	$L_1 = 3$	7.3.54 = 1134	3^3.2.3 = 162	162.2 = 324	270
	PRS $\boldsymbol{A_2}$	$1 + x + x^2$	$L_2 = 2$				
	3PRS $\boldsymbol{B}$	$2 + x + x^4$	$L = 4$				
8	*PRS* $\boldsymbol{A_1}$	$1 + x + x^3$	$L_1 = 3$	7.31.6 = 1302	3^1.3.5 = 45	45.2 = 90	48
	PRS $\boldsymbol{A_2}$	$1 + x^2 + x^5$	$L_2 = 5$				
	3PRS $\boldsymbol{B}$	$2 + x + x^2$	$L = 2$				
9	*PRS* $\boldsymbol{A_1}$	$1 + x + x^2$	$L_1 = 2$	3.31.6 = 558	3^1.2.5 = 30	30.2 = 60	42
	PRS $\boldsymbol{A_2}$	$1 + x^2 + x^5$	$L_2 = 5$				
	3PRS $\boldsymbol{B}$	$2 + x + x^2$	$L = 2$				

Consequently, polynomial $M(s)$ divides

$$(1+x^{T_j})^{\frac{(p-1)p^{p-1}}{2}} = 1 + x^{T_j \cdot \frac{(p-1)p^{p-1}}{2}} . \qquad (10)$$

But then the period of the sequence ***S***, generated by the *LFSR* based *GSMG*, is at most

$$T_S = \frac{(p-1)p^{L-1}}{2} . \prod_{i=1}^{p-1} (2^{L_i} - 1) . \qquad (11)$$

This contradicts to the Theorem 1, because

$$\frac{(p-1)p^{L-1}}{2} < N_{(\neq 0)} = \frac{(p^L - 1)(p-1)}{p} . \qquad (12)$$

Therefore, the assumption isn't true and $r > \frac{(p-1)p^{p-1}}{2}$, i.e. the lower bound on the linear complexity λ_s of the sequence ***S*** is

$$\lambda_S > \frac{(p-1)p^{L-1}}{2} \prod_{i=1}^{p-1} L_i . \qquad (13)$$

This conclusion ends the proof of the Theorem 2.

B. Practical analysis

The *LFSR* based *GSMG* architecture is modelled in Visual C++ environment. The linear complexity of the generated by the *LFSR* based *GSMG* sequences is practically analyzed by means of the Berlekamp-Massey algorithm. The theoretical lower and upper bounds of the linear complexity λ_s, given by the Theorem 2 and the found by the Berlekamp-Massey algorithm real λ_s are given in Table 1. The period of the output shrinking and multiplexing sequence ***S*** also is shown in the Table 1.

The practical analysis of the linear complexity and period of the sequences ***S***, generated by the *LFSR* based *GSMG*, confirm the theoretical results given by Theorem 1 and Theorem 2, i.e. the exponential period and exponential bounds of the linear complexity.

IV. CONCLUSION

In this paper the linear complexity of the *LFSR* based Generalized Shrinking-Multiplexing Generator is investigated mainly through algebraic techniques. It is proved the exponential lower and upper bounds of the linear complexity λ_s. Thus, the proposed *LFSR* based *GSMG* architecture allows to produce binary pseudo random sequences with good properties like uniform distributions of 1s and 0s, unpredictable nonlinearity, enormous period and large linear complexity. This shows that the elemental goals of the pseudo random number generators are achieved by *LFSR* based *GSMGs*. Consequently, they can be used as a part of a stream ciphers in the height-speed communication applications.

ACKNOWLEDGEMENT

We will be glad to thanks everyone who helps us to make some strong cryptanalysis of the *LFSR* based *GSMG*.

REFERENCES

[1] N. B. Atti, G. M. Diaz–Toca, H. Lombardi, The Berlekamp-Massey Algorithm revisited, Springer Berlin, Heidelberg, Volume 17, Number 1, April, 2006, http://www.springerlink.com/content/hk46h32538075152/.

[2] D. Coppersmith, H. Karwczayk, Y.Mansour, The Shrinking Generator, *Crypto'93*, http://imailab-www.iis.u-tocio.ac.jp/limit/ Papers/Crypto_Eurocrypt/ HTML/PDF/C93/22.pdf

[3] M. Goresky, A. Klapper, Fibonacci and Galois Representations of Feedback-With-Carry Shift Registers, *IEEE Trans. on Inform. Theory*, vol. 48, pp. 2826−2836, November 2002.

[4] S.Greenberg, N. Feldblum, G. Melamed, Implementation of the Berlekamp-Massey algorithm using a DSP, Proceedings of the 2004 11th IEEE International Conference Electronics, Circuits and Systems, ICECS 2004, ISBN: 0-7803-8715-5, pp. 358- 361, 2004.

[5] W. Meier, O. Staffelbach, The Self-Shrinking Generator, *Advances in Cryptology*, Eurocrypt 1994 (LNCS 950), 205-214, 1995, http://homes.esat.kuleuven.be/~jlano/stream/papers/ SSGms.pdf.

[6] A. Menezes, P. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, p. 780, 1997, www.cacr.math.uwaterloo.ca/hac.

[7] R. Ruepel, *Analysis and Design of Stream Ciphers*, Springer Verlag, N. Y., 1986.

[8] B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, 1996.

[9] T. Tashev, B. Bedzhev, Zh. Tasheva, The Generalized Shrinking-Multiplexing Generator, International Conference on Computer Systems and Technologies - *CompSysTech*'07, June 22-26, 2007, Bulgaria, under printing.

[10] T. Tashev, The Period of the LFSR Based Generalized Shrinking-Multiplexing Generator, International Conference on Computer Systems and Technologies - *CompSysTech*'07, June 22-26, 2007, Bulgaria, under printing.

[11] Zh. Tasheva, B. Bedzhev, V. Mutkov, A Shrinking Data Encryption Algorithm with p-adic Feedback with Carry Shift Register, *Conference Proc. of XII Int. Symp. on Theoretical Electrical Engineering*,ISTET'03,July 6-9, 2003, Warsaw, Poland, Volume II, pp.397–400, 2003.

[12] Zh. Tasheva, B. Bedzhev, B. Stoyanov, N-adic Summation-Shrinking Generator. Basic properties and empirical evidences, *Cryptology ePrint Archive*, Co-Editors: Mihir Bellare, UCSD Christian Cachin, IBM Zurich, Accepted and posted with Number 2005/068, http://eprint.iacr.org/2005/068.pdf.

[13] Zh. Tasheva, B. Bedzhev, B. Stoyanov, P-adic Shrinking–Multiplexing Generator, IEEE Third International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications IDAACS'2005, September 5-7, 2005 Sofia, Bulgaria, 2005.