Annals of University of Craiova, Math. Comp. Sci. Ser.
Volume 36(1), 2009, Pages 49–54
ISSN: 1223-6934

Linear Feedback Shift Register Optimizations

Nicolae Constantinescu

Abstract. One actual problem in cryptography is to find a generator which must carry out some conditions asked by the beneficiary. There are presented some interesting and new results concerning the complexity of the combinations of linear feedback shift registers (Schneier [7]). These combination can be described in terms of boolean function theory using the logical operators like sum and product. There are also introduced the notion of spliting (a generalization of classical term of decimation) and the inverse operator called interleave. The presented results have applications in cryptography, more exactly, to the construction of some chipher system which are used in simetric-key encryption for high-level safety-comunications(Zeng [3]). Also, the present exposure approaches the power of the generator to attack.

2000 Mathematics Subject Classification. Primary 94A55; Secondary 11T71, 68P25.
Key words and phrases. LFSR. Berlekamp Massey Cryptographyc attack.

1. Introduction

The basement of symmetric cryptography consiste of the cipher system evaluation problem (Massey [4] and [5]). The evaluation problems are different from the cracking problems in the following way (Preda [6]):

- the evaluator want to find the *minimal* size of the output information from which one can find, using powerful mathematics tools, *some information* regarding the cipher algorithm, the key used and/or the plain text
- the cracker want to find the *maximum* size of the information from which he can find the plain text

The basic problem of the encryption system is the RND (pseudorandom number generator). The construction of this generator must be framing in some parameters (compute restrictions caused by the calculation system or the execution times). Here the term of minimize and maximize has a generic significance. In fact the problem of evaluation is a *multicriterial problem*: some objective functions must be maximized (size of the key, nonlinearity degree, the equivalent linear complexity, period of the pseudorandom generator if we have one) and other functions must be minimized (the redundance of the key generator). These functions are conditioned by the knowledge of the cryptographic system (the adversary has completed knowledge about the system we use)(Koblitz [1], Scambray [8]).

Therefore we have the connection written in vectorial form:

$$\mathbf{c} = \mathbf{f}(\mathbf{m}; \mathbf{k}_t) \qquad (1)$$

where $\mathbf{f}$ is the encryption operator. If $\mathbf{k}_t = \mathbf{k}$ for every $t \in T$ (T is the ciphering time period which is a finite set) then we can rewrite the above

Received: 1 July 2009.

$$\mathbf{c} = \mathbf{f}(\mathbf{m};\mathbf{k}) \qquad (2)$$

where $\mathbf{f}$ is the encryption operator. In this case we say that we have a codification of the information (the role of the coding theory is to protect the information from errors which can appear in the communication channel; the role of the cryptography is to protect the information from the evedroper).In the codification case, after solving some nonlinear system, we can write

$$\mathbf{m} = \mathbf{h}(\mathbf{c};\mathbf{k}) \qquad (3)$$

Thus the knowledge of $\mathbf{f}(.;.)$ allows us to find $\mathbf{m}$ form $\mathbf{c}$. The system (1), which is a stochastic system, is much difficult to solve then the system (2),which is a deterministic system, because the time parameter t is involved. Thus the solution of system (2), given by (3), is a particular solution of the system (1) in the case $\mathbf{k}_t = \mathbf{k}$. Many times we have the encryption function $\mathbf{f}$ given in scalar form like

$$c_i = f(m_i, k_i), \forall i \in \mathrm{N}$$

where k_i is the i^{th} key derived from base key $\mathbf{k}_t$. If f can be factorized like

$$f(m_i, k_i) = m_i \oplus g(k_i),$$

then the encryption scheme is called *stream encryption* and we call the function g *pseudorandom generator.*

2. A variant of Linear feedback shift-registers

2.1. Berlekamp Massey algorithm. Mathematical Backround. Berlekamp-Massey algorithm is an efficient algorithm for computing the linear complexity of a p-ary finite sequence s^n of length n. The algorithm has n iterations and at N iteration is computing the linear complexity of the subsequence s^N consisting of the first N terms of s^n. For $p = 2$ we get the a binary sequence.

Definition 2.1. *(Next discrepancy). Let us consider the finite p-ary sequence* $s^{N+1} = s_0, s_1, ..., s_{N-1}, s_N$. *For* $C(D) = 1 + c_1 D + ... + c_L D^L$. *Let* $< L, C(D) >$ *be an* ***LFSR*** *which generates the subsequence* $s^N = s_0, s_1, ..., s_{N-1}$. *The next discrepancy* d_N *is the difference between* s_N *and the* $(N+1)$*-st term generated by* ***LFSR*** *:*

$$d_N = (s_N + \sum_{i=1}^{L} c_i s_{N-i}) \ mod\ p.$$

Theorem 2.1. *(Increasing of the linear complexity). Let us consider the finite p-ary sequence* $s^N = s_0, s_1, ..., s_{N-1}$ *of complexity* $L = L(s^N)$ *and let* $< L, C(D) >$ *be an* ***LFSR*** *which generates* s^N.

- *The feedback shift register* ***LFSR*** $< L, C(D) >$ *generates* $s^{N+1} = s_0, s_1, ..., s_{N-1}, s_N$ *if and only if the next discrepancy* d_N *is* 0
- *if* $d_N = 0$ *then* $L(s^{N+1}) = L$
- *Let suppose that* $d_N \neq 0$. *Let* m *be the largest integer smaller than* N *such that* $L(s^m) < L(s^N)$ *and let* $< L(s^m), B(D) >$ *be a shift register* ***LFSR*** *of length* $L(s^m)$ *which generates* s^m. *Then* $< L', C'(D) >$ *is a feedback shift register* ***LFSR*** *of smallest length which generates* s^{N+1} *where:* $L' = L$ *for* $L > N/2$ *and* $L' = N + 1 - L$ *if* $L \leq N/2$ *and* $C'(D) = C(D) + B(D)D^{N-m}$.

2.2. Berlekamp-Massey Implementation. Above results allow us to implement the algorithm of computing the equivalent linear complexity of a p-ary sequence. For the portability of the algorithm we give like parameter p the field characteristic.

Input: p-ary sequence $s^n = s_0, s_1, s_2, ..., s_{n-1}$ of length n.

Output: linear complexity $L(s^n)$ of s^n, $0 \leq L(s^n) \leq n$.

1. Initialization.

$$C(D) = 1, B(D) = 1, L = 0$$

$$m = -1, N = 0, b = 1,$$

$$p = 2 \ (field\ characteristic)$$

2. While $(N < n)$ do:

2.1 Compute the next discrepancy

$$d: \ d = (s_N + \sum_{i=1}^{L} c_i s_{N-i}) \ mod\ p.$$

2.2 If $d \neq 0$ do:

$$T(D) = C(D), \quad C(D) = C(D) + db^{-1}B(D)D^{N-m}$$

If $L \leq \frac{N}{2}$ then

$$L = N + 1 - L, \ m = N, \ B(D) = T(D), d = b.$$

2.3 $N = N + 1$.

3. Return(L).

2.3. Method Description.

Definition 2.2. *(Vaduva* [2]*). A RND (pseudorandom number generator) is the structure:* $G = (S, \mu, f, U, g)$ *where* S *is a finit lots of states;* μ *is a distribution of probability on* S *named initialy distribution;* $f : S \to S$ *is a transition function;* U *is lots of output symbols;* $g : S \to U$ *is an output function.*

Definition 2.3. *A LFSR consist in* n *memory locations and a feedback function wich express any new element* $a(t)$*, with* $t \leq n$*, of the string in dependence of the elements previous generating* $a(t-n), a(t-n+1), ...$ *where the feedback function must be unsingular, which it means:* $a(t) = g(a(t-1), ..., a(t-n+1)) \oplus a(t-n)$ *where* $\oplus$ *means exclusiv OR operation.*

One of the most popular of this type is Geffe generator showed in fig. 1, with the formula

$$y(t) = x_1(t) * x_3(t) \oplus \ \bar{x}_1 \ (t) * x_2(t).$$

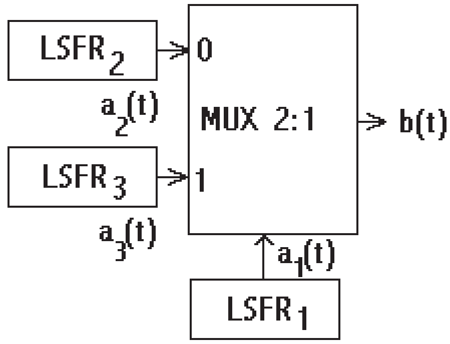

FIGURE 1. Linear Feedback Shift Register Generally Scheme

2.4. Algorithm. In this work it is proposed a new type of Gollman generators (based on many shift waterfalling registers, like [13, 14, 15]).The basic element of the algoritm is Linear feeedback shift register. As a matter of fact the output of the algoritm is a boolean function of the output from each R_i register. The first register R_0 have a constant tact and the register R_i $(i = 1, ...n-1)$ is turning by a variable number (given by the value of the tact cell of previous register $tact_{i-1}$ and by the function F_i) and the output is given by cell out. Therefore the setting of the algorithms consist in:

a) Parameters

- the number of the removal registers R by the algorithm notated with n
- the degree of the feedback polinom deg $f_i(X)$, $i = 0, 1, ..., n-1$.
- the feedback formulas $f_i(X)$ of the register R_i, $i = 0, 1, ..., n-1$. This are primitive polinoms of n-degree in modulo 2 algebra.
- the size of the cell noted with k.
- the maximnumber of rotations doing by the registerin the course of one step notated with p (minimum of the value of p is 2)
- the function of the tact for the next register is

$F_i = \begin{cases} 1 & if \quad MSB[tact_{i-1}] = 0 \\ 2 & in \quad other \quad case \end{cases}$

- the last register has the tact given by the formula:

$F_i = \begin{cases} 1 & if \quad MSB[tact_{n-2} \oplus MSB[R_{n-1}[\deg -1]]] = 0 \\ 2 & in \quad other \quad case \end{cases}$

- the cell which indicates the output from the first register is given by $out_0 \in \{0, ..., \deg -1\}$.
- the output cell of the last register is

$$y_{n-1} = R_{n-1}[R_{n-2}[out_{n-1}] + R_{n-1}[\deg -1] >> (k - \log_2(\deg))moddeg]$$

where

$R_{n-1}[\deg -1] >> (k - \log_2(\deg))$ are the first 7 bits of the cell deg-1 of the last register.

b) Randomising

- the variable of the output cell from each R_i register depends of $out_0 \in \{0, ..., \deg -1\}$. for $i = 0, 1, ..., n-1$.
- the tact cell for the next R_i register is $tact_{i-1} \in \{0, ..., \deg -1\}$ for $i = 0, 1, ..., n-1$.

c) The generate algorithm of a k-bit

RotateR_0. Take of from the adress out_0 the output from R_0 register notated by $y_0 = R_0[out_0]$ in accordance with the value finding in $tact_0$ cell meaning $R_0[tact_0]$ compute the next tact

$F_i(tact_0) \in \{1, ..., p\}$.

Start

for i=1 to n-1 execute

- rotate R_i register by $F_i(tact_i)$ times.
- take of the output y_i of the R_i register, from the adress $R_{i-1}[out_{i-1}]\ mod$ deg meaning

$$y_i = R_i[R_{i-1}[out_i] mod \deg]$$

- in accordance with the value finding in $tact_i$ cell, meaning $R_i[tact_i]$ compute the next tact

$$F_i(tact_i) \in \{1, ..., p\}$$

endfor.

Compute output $y = f(y_1, ..., y_n)$

Stop

d) Explanations over the parametric functions

- the feedback relations are doing in modulo 2^k algebra.
- function $f(y_1, ..., y_n) = y_1 \oplus ... \oplus y_n$.
- the tact functions given by the next formula:

$F_i = \{.1 \quad if \quad LSB[tact_{i-1}] = 02 \quad in \quad other \quad case \forall i = 1, ..., n.$

3. Conclusions

The domain of aplication for these algoritms are large. The problems related with (pseudo) random number generation have cryptanalitic nature, because the most public algoritms have been considered "weak" owing of the output calculating methods. Thus, the testing of these are doing not only with clasic tests, using adapting attacks. These are useful becouse its show practical degree of safety. In this order, the most used test is the attack based on Berlekamp Massey construction (Maurer [9], Menicocci [10], Meyerm [11], Golic [12]). The future research is based on the Berlekamp-Massey attack study, which conclude to optimize the register length and initial values from them related to the attack complexity.

References

[1] Neal Koblitz, *Algebric Aspects of Cryptography*, Springer-Verlag, 1997.

[2] Ion Vaduva, *Modele de simulare cu Calculatorul*, Ed. Tehnica, 1977.

[3] K.C. Zeng, C.H. Yang, T.R.N. Rao, An improved linear syndrome algorithm in cryptanalysis with application, Proceedings Crypto'89, *Springer-Verlag Lecture Notes in Computer Science*, **435**, 1989.

[4] J.L. Massey, Shift-register synthesis and BCH decoding, *IEEE Trans. Information Theory*, IT-15, (1), 122-127.

[5] J.L. Massey, R.A. Rueppel, Linear ciphers and random sequence generators with multiple clocks, Proceedings Eurocrypt'84, *Springer-Verlag Lecture Notes in Computer Science*, 209, 74-87, 1984.

[6] Vasile Preda, *Teoria Deciziilor Statistice*, Ed. Academiei, 1992.

[7] Bruce Schneier, *Applied Cryptography-Second Edition (Protocols, Algorithms and Source Code in C)*, John Wiley & Sons, Inc., 1996.

[8] Joel Scambray, Suart McClure, George Kurtz, *Hacking Exposed - Network Security Secrets & Solutions; SecondEdition*, Osborne/McGraw-Hill-2001.

[9] U. Maurer, Cascade Chipers:The importance of being first, *J. of Cryptology*, 6, 1993.

[10] R. Menicocci, A systematic Attack on clocked controlled cascades, *Adv. in Cryptology-Eurocrypt*, 1994.

[11] W. Meyerm, O. Staffelback, Fast corellation attacks on certain stream ciphers, *Journal of Cryptology*, 1, 159-176, 1989.

[12] J. Golic, Cryptanalysis of the Alleged A5 Stream Cipher, *Adv. in Cryptology-Eurocrypt*, 1997.

[13] C. Jansen, D. Boekee, The shortest feedback shift register that can generate a given sequence, *Adv. in Cryptology -Crypto*, 90-99, 1990.

[14] B. Arazi, On the synthesis of De Bruijn sequences, *Information and Control*, 49,(**2**), 81-90, 1981.

[15] A. Lempel, On a homomorphism of De Bruijn Graph and its application to the design of Feedback shift registers, *IEEE Transactions on Computers*, C-19, (**12**), 1204-1209, 1970.

(Nicolae Constantinescu) University of Craiova, Faculty of Mathematics and Computer Science, Department of Computer Science, 13 Alexandru Ioan Cuza Street, Craiova, 200585, Romania

E-mail address: nikyc@central.ucv.ro