Generation of Binary Sequences with Controllable Complexity and Ideal r-Tupel Distribution

Thomas Siegenthaler
Amstein Walthert Kleiner*

Réjane Forré
Inst. for Communications Technology †

Abstract

A key stream generator is analyzed which consists of a single linear feedback shift register (LFSR) with a primitive connection polynomial and a nonlinear feedforward logic. It is shown, how, for arbitrary integers n and r and a binary LFSR of length $L = n \cdot r$ the linear complexity of the generated keystream can be determined for a large class of nonlinear feedforward logics. Moreover, a simple condition imposed on these logics ensures an ideal r-tupel distribution for these keystreams. Practically useful solutions exist where the keystream has linear complexity $n \cdot r^{n-1}$ together with an ideal r-tupel distribution.

1 Introduction

A common type of keystream generator consists of a single binary linear feedback shift register (LFSR) and a feedforward logic (see Fig. 1). If the sequence produced by the LFSR has period p, all binary (key-stream-) sequences of length p are generated by suitable feedforward logics. This makes the keystream generator of Fig. 1 attractive from the theoretical point of view. The type shown in Fig. 1 is also of considerable practical interest because it needs only a single (instead of several) LFSR. However, in the general case the analysis of this type of keystream generator has shown to be rather difficult [1]. Groth [2] proposed a layered structure for the feedforward logic to control the linear complexity of the generated keystream. This arrangement generates keystreams of large linear complexities, however, the statistics of these keystreams are hard to control. Rueppel suggested [3] a simply realisable and therefore practically useful class of feedforward logics such that a lower bound for the keystream's linear complexity is guaranteed. A closely related structure had independently been proposed by Günther/Bernasconi [4] which is also simple realisable and also guarantees a minimal linear complexity of the keystream. The latter two methods are based on the existence of one or several

*Information Systems Engineering AG, Leutschenbachstr. 45, 8050 Zürich, Switzerland
†ETH Zentrum, Sternwartstr. 7, 8092 Zürich, Switzerland

D. Chaum and W.L. Price (Eds.): Advances in Cryptology - EUROCRYPT '87, LNCS 304, pp. 15-23, 1988.

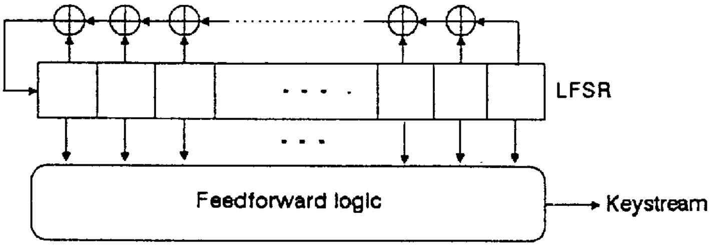

Figure 1: A common type of keystream generator

high order products in the corresponding algebraic normalform of the feedforward logic. A new approach [8] is proposed here. First, a number of "well chosen" delayed replicas (called "phases") of the sequence generated by the LFSR are picked, then every nonlinear feedforward logic is allowed. The analysis uses the theory of finite fields GF(2^n). The approach is strongly based on an interpretation of two results recently obtained by Brynielsson. It is assumed that the LFSR of Fig. 1 has a primitive connection polynomial.

2 Synthesis of keystream generators

In finite fields, every function f : GF(q) $\longrightarrow$ GF(q) with $x \longmapsto f(x)$ can be expressed as a polynomial [5]:

$$f(x) = \sum_{i=0}^{q-1} a_i x^i \tag{1}$$

with coefficients

$$a_i = \sum_{x \neq 0} [f(0) - f(x)] x^{-i}, \quad a_i \in \mathrm{GF}(q). \tag{2}$$

Definition 1 *If the symbol y_k of the sequence $\{y_k\}$ over* GF(q) *is obtained as $y_k = f(x_k)$ where f denotes the polynomial in (1) and $\{x_k\}$ is a sequence over* GF(q)*, then $\{y_k\}$ is called a* **polynomial sequence**.

The following theorem is shown to be crucial for the computation of the linear complexity of the keystream produced by a generator as given in Fig. 1.

Theorem 1 (Brynielsson [6]) *Let $\{x_k\}$ be a maximum length sequence over* GF(2^n) *with a primitive characteristic polynomial of degree r and let $H(i)$ denote the Hamming weight of the integer i. The polynomial sequence $\{y_k\}$ with $y_k = f(x_k)$ has linear complexity* LK($\{y_k\}$)*:*

$$\mathrm{LK}(\{y_k\}) = \sum_{a_i \neq 0} r^{H(i)}, \quad a_i \in \mathrm{GF}(2^n), \tag{3}$$

where the a_i's denote the coefficients in (1).

At a first glance, polynomial sequences together with Theorem 1 seem not to have any connection to the system of Fig. 1. Next, this connection is worked out with the help of Lemma 1 and Lemma 2. We consider a maximum length sequence $\{x_k\}$ over GF(2^n). Symbols x_k from GF(2^n) may be written as

$$x_k = x_{n-1,k}\alpha^{n-1} + x_{n-2,k}\alpha^{n-2} + \cdots + x_{1,k}\alpha + x_{0,k} \tag{4}$$

where the $x_{i,k}$'s belong to GF(2) and where α denotes a primitive element of GF(2^n). The n binary sequences $\{x_{i,k}\}$, $i = 0, 1, \ldots, n-1$, in (4) are called the **binary subsequences** of $\{x_k\}$.

Lemma 1 (Brynielsson [7]) *Let $\{x_k\}$ be a maximum length sequence over* GF(2^n) *with (primitive) characteristic polynomial $p(x)$ of degree r. The binary subsequences $\{x_{i,k}\}$, $i = 0, 1, \ldots, n-1$, of $\{x_k\}$ are linear independent and fulfil the* **same linear recursion** *with an associated (primitive) characteristic polynomial $q(x)$ of degree $L = r \cdot n$.*

Therefore, the subsequences $\{x_{1,k}\}$ differ only by delays of each other. The polynomial $q(x)$ can be determined [7,8]. The following Lemma 2 is well known.

Lemma 2 *Let $\{z_k\}$ be a binary maximum length sequence with (primitive) characteristic polynomial $q(x)$ of degree L. Every delayed version $\{z_{k-d}\}$, where d denotes an integer in the range $[0, \ldots, 2^L - 1]$ of $\{z_k\}$ can be obtained by some linear combination of the sequences $\{z_{k-1}\}$,$\{z_{k-2}\}$, ..., $\{z_{k-L}\}$.*

This means that every phase of the maximum length sequence generated by the LFSR of Fig. 1 can be obtained as a linear combination of the sequences from the L stages of this LFSR. We are now ready to establish the connection between Theorem 1, Lemma 1, Lemma 2 and a system as given in Fig. 1. Consider a maximum length sequence $\{x_k\}$ over GF(2^n). Choose any of the binary subsequences $\{x_{i,k}\}$ mentioned in Lemma 1, say $\{x_{0,k}\}$. This binary subsequence is generated by a binary LFSR of length L, its feedback connections are known from $q(x)$. The binary subsequences $\{x_{i,k}\}$, $i = 1, 2, \ldots, n-1$, are only phase shifts of $\{x_{0,k}\}$ (Lemma 1) and can be obtained as linear combinations of the sequences at the L stages of the LFSR that generates $\{x_{0,k}\}$ due to Lemma 2. (Instead of generating the maximum length sequence $\{x_k\}$ over GF(2^n) by a corresponding LFSR of length r with feedback connections due to $p(x)$, $\{x_k\}$ is generated by a (binary) LFSR of length L with feedback connections due to $q(x)$ and linear combinations of the sequences occuring at the L stages of this LFSR.) Every feedforward logic can now be applied to the n binary sequences $\{x_{0,k}\}$, $\{x_{1,k}\}$, ..., $\{x_{n-1,k}\}$ to produce the binary keystream $\{y_k\}$. This feedforward logic is then described as a polynomial f : GF(2^n) $\longrightarrow$ GF(2) with $y_k = f(x_k)$ as given in expression (1). The linear

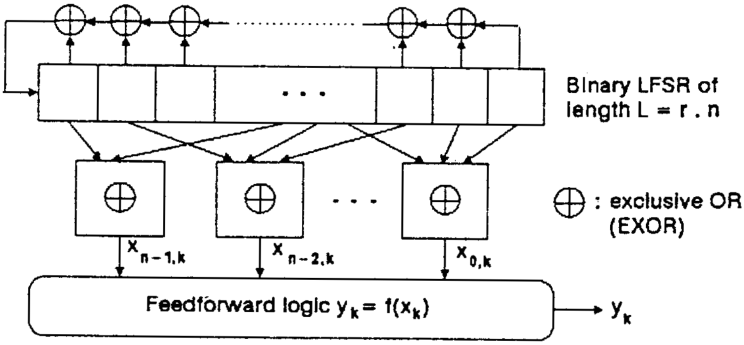

Figure 2: Synthesis of keystream generators

complexity of $\{y_k\}$ is computed by theorem 1. The corresponding system is shown in Fig. 2.

Because of the required EXOR blocks, the system of Fig. 2 is a slightly restricted version of that shown in Fig. 1. But the linear complexity can be exactly determined for arbitrary feedforward logics as given in Fig. 2.

So far we have not mentioned the $r-$tupel distribution of the keystream $\{y_k\}$. The $r-$tupel $\underline{y_i}$ is defined as a sequence $[y_i, y_{i+1}, \ldots, y_{i+r-1}]$ of successing symbols of $\{y_k\}$. The set $\underline{Y} = [\underline{y_0}, \underline{y_1}, \ldots, \underline{y_{p-1}}]$ contains all $r-$tupels of the sequence $\{y_k\}$ of period p. The following definition is useful:

Definition 2 *A binary sequence $\{y_k\}$ of period $p = 2^L - 1$ exhibits an* **ideal $r-$tupel distribution** *$\underline{Y}$, $1 \leq r \leq L$, if exactly one of the 2^r possible and disjoint binary $r-$tuples occurs $2^{L-r} - 1$ times in a period of $\{y_k\}$ and each of the others occurs 2^{L-r} times.*

Lemma 3 *An ideal $r-$tupel distribution of $\underline{Y}$ implies ideal $r'-$tupel distribution of $\underline{Y}$ for all r' with $1 \leq r' \leq r$.*

> **Proof:** From an ideal $r-$tupel distribution follows that exactly one of the 2^r possible and disjoint binary $r-$tupels occurs $2^{L-r} - 1$ times and each of the others 2^{L-r} times. Therefore, exactly one $r'-$tupel, $1 \leq r' \leq r$, occurs $(2^{L-r} - 1) + 2^{L-r} \cdot (2^{r-r'} - 1) = 2^{L-r'} - 1$ times and each of the others occurs $2^{L-r} \cdot 2^{r-r'} = 2^{L-r'}$ times, as was to be shown.

Theorem 2 *Let $\{x_k\}$ denote a maximum length sequence over* GF(2^n) *of period $2^{nr} - 1$ and f a polynomial f :* GF$(2^n) \longrightarrow$ GF(2). *A polynomial sequence $\{y_k\} = f(\{x_k\})$ exhibits an ideal $r'-$tupel distribution for all r' with $1 \leq r' \leq r$ for $x \in$* GF(2^n) *if and only if*

$$|\{x : f(x) = 1\}| = 2^{n-1}. \tag{5}$$

where $|\{.\}|$ denotes the cardinality of the enclosed set $\{.\}$.

Proof: Assume $\mid \{x : f(x) = 1\} \mid = b$ and $\mid \{x : f(x) = 0\} \mid = c$ with $b + c = 2^n$ and $f(0) = 0$. All r–tupels $\underline{x_i} = [x_i, x_{i+1}, \ldots, x_{i+r-1}]$ for $i = 0, 1, \ldots, 2^{nr} - 1$ in the maximum length sequence $\{x_k\}$ are disjoint and every possible 2^n–ary nonzero r–tupel occurs exactly once. Binary r–tupels in $\{y_k\}$ occur from $\underline{x_i} = [x_i, x_{i+1}, \ldots, x_{i+r-1}]$ as $\underline{y_i} = [y_i, y_{i+1}, \ldots, y_{i-r-1}]$ with $y_i = f(x_i)$. First, we note that the 1-tupel distribution of $\underline{Y}$ is ideal iff $b = c = 2^{n-1}$. Lemma 3 implies that none of the r'–tupel distributions for $1 \leq r'$ is ideal if the 1–tupel distribution of $\underline{Y}$ is not. Therefore, (5) is a necessary condition for an ideal r–tupel distribution of $\underline{Y}$. This condition is also sufficient as is shown now. First, nonzero r–tupels $\underline{y_i}$ are considered. From the assumption $f(0) = 0$ follows that for nonzero $\underline{y_i}$'s the involved $\underline{x_i}$'s are nonzero too. From (5) follows that $(2^{n-1})^r - 1$ or $2^{L-r} - 1$ (for $L = n \cdot r$) r–tupels $\underline{x_i}$ are mapped into $\underline{y_1} = \underline{0}$, where the -1 accounts for the missing r–tupel $\underline{x_i} = \underline{0}$ in the maximum length sequence $\{x_k\}$. This completes the proof. If $f(0) = 1$ is assumed, a similar proof exists.

3 Nonlinear feedforward logic

From theorem 2 follows that a system as given in Fig. 2 generates a keystream $\{y_k\}$ with an ideal r–tupel distribution iff the polynomial $f : \mathrm{GF}(2^n) \longrightarrow \mathrm{GF}(2)$ which describes the feedforward logic of Fig. 2 fulfils condition (5). The designer of such a system prefers polynomials f as given in (1) such that the following properties hold

(a) $f : \mathrm{GF}(2^n) \longrightarrow \mathrm{GF}(2)$ (produces a binary sequence)

(b) f such that $\mid \{x : f(x) = 1\} \mid = \mid \{x : f(x) = 0\} \mid$ (ideal r–tupel distribution)

(c) f produces a keystream of large linear complexity

(d) f is easy to implement.

Solutions which fulfil all of the above requirements are described in [9] and will be discussed hereafter. As the polynomial f has to map $\mathrm{GF}(2^n)$ onto $\mathrm{GF}(2)$, it makes sense to use the so-called "trace" function.

Definition 3 *For $\alpha \in \mathrm{GF}(q^m)$, the* **trace** *of α over $\mathrm{GF}(q)$ is defined by*

$$\mathrm{Tr}_{\mathrm{GF}(q^m)/\mathrm{GF}(q)}(\alpha) = \alpha + \alpha^q + \alpha^{q^2} + \cdots + \alpha^{q^{m-1}}. \tag{6}$$

It can be shown [10, Theorem 2.23] that $\mathrm{Tr}_{\mathrm{GF}(q^m)/\mathrm{GF}(q)}$ is a linear transformation from $\mathrm{GF}(q^m)$ onto the subfield $\mathrm{GF}(q)$. Moreover, considering the special case $q = 2$, one can prove [9] that the function $\mathrm{Tr}_{\mathrm{GF}(q^m)/\mathrm{GF}(q)}(\alpha)$ computed for all the α's in $\mathrm{GF}(2^n)$ takes on the value 0 exactly 2^{n-1} times, and the value 1 consequently

2^{n-1} times. Therefore, the conditions (a) and (b) will both be satisfied by a function

$$f(x) = \mathrm{Tr}_{\mathrm{GF}(2^n)/\mathrm{GF}(2)}(g(x)), \quad x \in \mathrm{GF}(2^n) \tag{7}$$

where $g(x)$ permutes the finite field $\mathrm{GF}(2^n)$. In [10, Theorem 7.8] it is shown that the monomial x^b is a permutation polynomial of $\mathrm{GF}(q)$ if and only if b and $q-1$ are relatively prime. Consider the function

$$f(x) = \mathrm{Tr}_{\mathrm{GF}(2^n)/\mathrm{GF}(2)}(x^b) \tag{8}$$

where

$$\gcd(b, 2^n - 1) = 1. \tag{9}$$

From definition (2) we get

$$f(x) = x^b + x^{b\cdot 2} + x^{b\cdot 2^2} + \cdots + x^{b\cdot 2^{n-1}}. \tag{10}$$

Theorem 1 can be employed to compute the linear complexity of the polynomial sequence $\{y_k\}$ with $y_k = f(x_k)$:

$$\mathrm{LK}(\{y_k\}) = \sum_{i:a_i \neq 0} r^{H(i)} \tag{11}$$

Nonzero a_i's only occur for indices $i = b, b\cdot 2, b\cdot 2^2, \ldots, b\cdot 2^{n-1}$. All of these indices are simply obtained by shifting the binary representation of the integer b. Thus

$$H(i) = H(b), \quad \text{for } i = b, b\cdot 2, b\cdot 2^2, \ldots, b\cdot 2^{n-1} \tag{12}$$

In order to obtain a keystream-sequence of large linear complexity, one should choose an integer b of large Hamming weight $H(b)$. On the other hand, b and $2^n - 1$ must be relatively prime, according to (9). Thus, the choice $b = 2^n - 1$ (which would provide the maximal Hamming weight $H(b) = n$ is excluded, and therefore

$$b = 2^n - 2 = -1 \bmod (2^n - 1), \quad \text{with } H(b) = n - 1 \tag{13}$$

is optimal. Let

$$f(x) = \mathrm{Tr}_{\mathrm{GF}(2^n)/\mathrm{GF}(2)}(x^{-1}) \tag{14}$$

be the filtering polynomial applied to the symbols x_k of the maximum length sequence $\{x_k\}$ over $\mathrm{GF}(2^n)$. According to Theorem 1, the polynomial sequence $\{y_k\} = f(\{x_k\})$ has linear complexity

$$\mathrm{LK}(\{y_k\}) = \sum_{i:a_i \neq 0} r^{n-1} = r^{n-1} \cdot n. \tag{15}$$

The maximal linear complexity reachable for given integers r and n is easily computed by considering the case where all the a_i's in (1) are different from zero [9]:

$$\mathrm{LK}_{max}(\{y_k\}) = \sum_{a_i} r^{H(i)} = \sum_{i=0}^{n} \binom{n}{i} r^i = (r+1)^n \tag{16}$$

From (2) follows that all coefficients a_i of the polynomial

$$f_{max}(x) = \begin{cases} 0, & \text{when } x = 1 \\ 1, & \text{else} \end{cases} \tag{17}$$

are nonzero and therefore this polynomial reaches the maximal linear complexity as given in (16). However, note that this polynomial does not fulfil Theorem 2 and that the statistical properties of the generated polynomial sequence are quite disastrous.

The ratio

$$\frac{\mathrm{LK}(\{y_k\})}{\mathrm{LK}_{max}(\{y_k\})} = \frac{nr^{n-1}}{(r+1)^n} = \rho$$

can be optimized with respect to r for any given integer n by means of a simple derivation. The value $r = n - 1$ turns out to be optimal, and we obtain

$$\rho_{max}(n) = \left(\frac{n-1}{n}\right)^{n-1} = \left(1 - \frac{1}{n}\right)^{n-1} \tag{18}$$

For large values of n, and consequently of r, this ratio converges to e^{-1}. This means that the proposed structure can provide a pseudo-random sequence with a linear complexity of about 1/3 of the reachable maximal linear complexity together with an ideal $(n-1)-$tupel distribution.

3.1 Connection to GMW-Sequences

After finishing this work our attention was drawn to the so called GMW-sequences (Gordon, Mills and Welch [11]). These binary sequences have correlation properties identical to those of maximum length sequences but possess a larger linear complexity. Some of these sequences $\{y_k\}$ can be specified as

$$y_k = \mathrm{Tr}_{GF(2^n)/GF(2)}\left([\mathrm{Tr}_{GF(2^{nr})/GF(2^n)}(\alpha^k)]^b\right) \tag{19}$$

where α is a primitive element of GF(2^{nr}) and b is any integer relatively prime to $2^n - 1$, r in the range $0 < r < 2^n - 1$. The interior Trace-function corresponds to a maximum length sequence over GF(2^n). This has been discussed in [12] together with an analysis of the tuple distribution, periodic autocorrelation and linear complexity of GMW-sequences as defined in (19). The results of our analysis with respect to the ideal tuple distribution and the linear complexity coincide with the results in [12]. However, the following difference concerning the derivation should be mentioned: Our analysis is based on Brynielsson's powerful Theorem 1 from which the linear complexity for every polynomial f applied to a maximum length sequence can be computed even if we use it only for a function as specified in (8). This function belongs to the same class of functions used in the GMW-construction according to expression (19). Moreover, Theorem 2 gives the necessary and sufficient condition for a polynomial $f(x)$ such that the corresponding polynomial sequence exhibits an ideal $r-$tuple distribution. The function $f(x)$

as defined in expressions (8) and (9) is a special case only which fulfils the requirements of Theorem 2. Finally, we proposed a practical implementation of these keystream generators which is completely different to the mechanization shown in [12].

Acknowledgement

The authors are grateful to L. Brynielsson for many stimulating discussions and private correspondence and also to I. Vajda and Prof. Dr. J. Massey for drawing our attention to reference [12].

References

[1] T. Herlestam, **On Functions of Linear Shift Register Sequences**, Advances in Cryptology, Eurocrypt '85, Lecture Notes in Computer Science, No. 219, Springer Verlag, 1985, p. 119-129.

[2] E.J. Groth, **Generation of Binary Sequences with Controllable Complexity**, IEEE Tr. on Inf. Theory, Vol. IT-17, No. 3, May 1971, p. 288-296.

[3] R.A. Rueppel **New Approaches to Stream Ciphers**, Diss. ETH No. 7714, Zürich, 1984.

[4] J. Bernasconi, C.G. Günther, **Analysis of a Nonlinear Feedforward Logic for Binary Sequence Generators**, Advances in Cryptology, Eurocrypt '85, Lecture Notes in Computer Science, No. 219, Springer Verlag, 1985, p. 161-168.

[5] B. Benjauthrit, I.S. Reed, **Galois Switching Functions and their Applications**, IEEE Tr. on Comp., Vol. C-25, No. 1, Jan. 1976, p. 78-86.

[6] L. Brynielsson, **On the Linear Complexity of Combined Shift Register Sequences**, Advances in Cryptology, Eurocrypt '85, Lecture Notes on Computer Science, No. 219, Springer Verlag, 1985, p. 156-160.

[7] -, **Entwurf und Analyse eines Kryptosystems über** GF(16), Kryptologie Aufbauseminar, J. Kepler Universität, Linz, 1985.

[8] Th. Siegenthaler, **Methoden für den Entwurf von Stream Cipher-Systemen**, Diss. ETH No. 8185, Dec. 1986.

[9] R. Forré, **Analyse eines Chiffriergenerators**, Diploma Project, Inst. for Communications Technology, ETH Zürich, Dec. 1986.

[10] R. Lidl, H. Niederreiter, **Finite Fields**, Addison-Wesley Publishing Company, 1983.

[11] B. Gordon, W. H. Mills, and L. R. Welch, **Some new difference sets**, Canad. J. Math., vol. 14, pp. 614-625, 1962.

[12] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, **Spread Spectrum Communications**, Vol. 1, Computer Science Press, 1985