LINEAR CIPHERS AND RANDOM SEQUENCE GENERATORS WITH MULTIPLE CLOCKS

James L. Massey and Rainer A. Rueppel
Insitute of Telecommunications
Swiss Federal Institute of Technology
8092 Zurich, Switzerland

Abstract

A construction is given for perfect linear ciphers that uses two digits of key per plaintext digit, which appears to be the minimum possible. The construction utilizes two shift-registers that are clocked at different speeds, and suggests a new type of random sequence generator in which two linear feedback shift-registers are clocked at different speeds and their contents combined at the lower clock rate. The effects of variable speed are analyzed, and the linear complexity of the sequences produced by such generators is determined.

T. Beth, N. Cot, and I. Ingemarsson (Eds.): Advances in Cryptology - EUROCRYPT '84, LNCS 209, pp. 74-87, 1985.

1. Introduction

We begin this paper by considering how much key is required in a perfect linear cipher. We show in Section 2 that two digits of key per plaintext digit suffice, and we conjecture that this much key is also necessary. The perfect linear cipher constructed in Section 2 utilizes two shift-registers that are clocked at different speed, a "trick" that we have borrowed from convolutional coding lore.

The perfect linear ciphers of Section 2 suggest a promising structure for random sequence generation, which we propose in Section 3, that utilizes two linear feedback shift-registers (LFSR's) clocked at different speeds. In Section 4, we investigate analytically the effects of such variable speed in LFSR's. These results are then used in Section 5 to determine the linear complexity of the sequences produced by the previously suggested random sequence generator. We close the paper with some additional observations and suggestions for generalizations.

2. Perfect Linear Ciphers and Convolutional Codes

Suppose that one wishes to use a ciphering system of the additive type in which the ciphertext digit y_j is determined by the plaintext digit x_j in the manner (reminiscent of a stream cipher) that

$$y_j = x_j + z_j \qquad j = 0,1,2,\ldots \qquad (1)$$

where the digit z_j is determined in some prescribed manner by the key K and the previous plaintext digits. [All digits and operations are assumed to be in F_q, the finite field of q elements, unless specified otherwise.] Suppose further that, for whatever reason, one demands that the enciphering be linear in the plaintext with memory M so that

$$z_j = \sum_{i=1}^{M} c_i(j,K)x_{j-i} \qquad j = 0,1,2,\ldots \qquad (2)$$

where the coefficients $c_j(j,K)$ depend both on the time instant j and the key K. (We suppose that the initial conditions $x_{-1}, x_{-2},\ldots, x_{-M}$ required in (2) are dummy plaintext digits that may be chosen as convenient.) Suppose finally we demand that the enciphering be perfect in the sense that, for some appropriate probability measure over the keys,

one has, for every choice of β in F_q and every $j \geq 0$,

$$\Pr(z_j = \beta | z_{j-1},\ldots,z_0,x_{j-1},\ldots,x_0,\ldots,x_{-M}) = \frac{1}{q}. \qquad (3)$$

In other words, we require that, for each allowable plaintext sequence, the additive sequence $z_0,z_1,z_2,..$ be a completely random q-ary sequence.

From (2), it follows that

$$\Pr(z_j = 0 | x_{j-1} = \ldots = x_{j-M} = 0) = 1 \qquad (4)$$

so that perfect secrecy as specified by (3) is impossible without some plaintext restriction. From (4), we see that the least plaintext restriction compatible with perfect secrecy in such a linear cipher is

$$[x_{j-1},x_{j-2},\ldots,x_{j-M}] \neq [0,0,..,0], \qquad j = 0,1,2,... \qquad (5)$$

which we hereafter assume to be the only restriction on the plaintext. [For q = 2 and M = 1, we see that (5) implies $x_j = 1$, all j, so that no interesting system is possible; for all M > 1, however, the plaintext restriction admits interesting systems.]

We first make the quite trivial observation that perfect linear ciphers exist for every M and every F_q. One can simply choose the coefficients $c_i(j,K)$ independently at random from a uniform distribution over F_q; the plaintext restriction (5) guarantees that one of the independent "key digits" will then appear with a non-zero multiplier on the right in (2) so that (3) will be satisfied. This perfect linear ciphering system, however, requires M digits of key for each digit of plaintext. This large key requirement appears quite unsatisfactory (particularly for large M as would be desirable to ease the restriction specified by (5)) when one reflects that a perfect one-time system (which is an additive cipher in which the additive sequence is itself the random key) requires only one digit of key for each digit of plaintext. The question we now pose is: What is the least amount of key (measured in digits of key per plaintext digit) required for a perfect linear cipher as specified by (1), (2) and (3) with the plaintext restriction (5) ? We now

show that two digits of key per plaintext digit is always sufficient, and we conjecture that this much key is also necessary for all $M \geq 2$.

To prove our claim, we consider the specific linear cipher system shown in Fig. 1 consisting of a random key generator (whose outputs are independently chosen from a uniform distribution over F_q) that drives a shift-register that is clocked at a rate d times faster than the shift-register driven by the plaintext source.

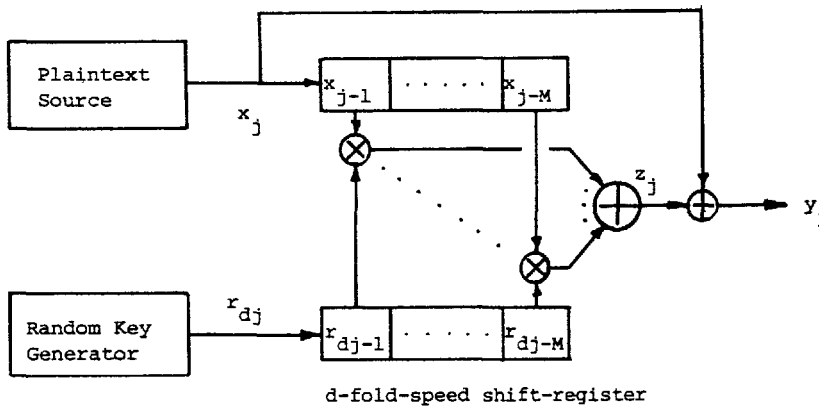

Fig. 1: A perfect linear cipher system, conjectured to use a minimum of random key digits when d = 2.

To show that (3) holds for the system of Fig. 1 under the restriction (5), consider at time j the leftmost non-zero digit in the upper shift-register, say x_{j-i}. From Fig. 1, we see that its multiplying coefficient $c_i(j,K)$ is just the random digit r_{dj-i}. But, provided that d > 1 so that the lower shift-register is shifting faster to the right than is the upper one, it follows because r_{dj-i} has just come abreast of x_{j-i} at time j that this same random digit at earlier time instants could have multiplied only digits that are to the left of x_{j-i} in the upper shift-register. But, as all these latter digits must be zeroes, it follows that the earlier generated digits $z_{j-1}, z_{j-2}, \ldots$ are all independent of r_{dj-i}; hence the fact that $x_{j-i}\, r_{dj-i}$ with $x_{j-i} \neq 0$ is a component of z_j implies that (3) is satisfied, as was to be shown. The linear cipher

of Fig. 1 requires d digits of key per plaintext digit, and we have shown it to be perfect for all $d \geq 2$. The least key, of course, is used when $d = 2$.

In fact, we have borrowed our answer to the linear cipher problem posed above from our earlier solution [1, pp.19-21] to a problem in error-correcting codes. The problem there was to find the smallest ensemble of time-varying codes such that the codewords enjoy pairwise independence -- this coding problem is formally identical to the linear cipher problem, and our "double speed ensemble" solution to this coding problem remains the smallest ensemble known to suffice.

It is well known [2, pp. 680-683] that the least amount of key required for perfect secrecy in any type of ciphering system is one key digit per plaintext digit (when the plaintext is irredundant). It would thus be interesting if one could prove that a perfect linear cipher requires at least two key digits per plaintext digit, as this would give some theoretical force to the rubric that "linearity is the curse of the cryptographer".

3. Variable Speed in Random Sequence Generation

A perfect secrecy system of the additive type is of course an ideal random number generator, i.e., its additive sequence $z_0, z_1, z_2, \ldots$ is a sequence of digits drawn independently at random from a uniform distribution over F_q. This suggests that the basic structure of Fig. 1 may be of use in random sequence generation. For this purpose, it is natural to replace the plaintext source of Fig. 1 by an M-stage linear feedback shift-register (LFSR) started in some non-zero state, as this automatically enforces the "plaintext restriction" (5) as well as introduces some element of pseudo-randomness. It is a natural next step to replace the random key generator of Fig. 1 by a second LFSR of length $L (L \geq M)$, also started in some non-zero state but clocked at a speed d times that of the first LFSR to produce the pseudo-random sequence which further "randomizes" the "plaintext" to produce the desired "random" sequence $z_0, z_1, z_2, \ldots$ The resulting random sequence generator is shown in Fig. 2 Such a device might be used as a random number generator or as a key stream generator in a conventional stream cipher.

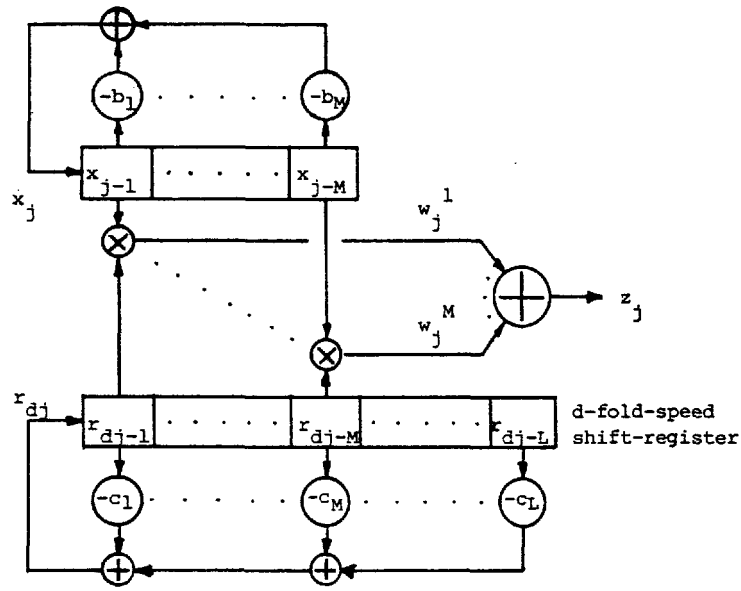

Fig. 2: A random sequence generator employing multiple speed shift-registers.

In the following sections, we analyze the effect of the speed factor d on the sequence produced by the generator of Fig. 2. Our interest is in the new phenomena that result when the speed factor is treated as an additional variable in shift-register sequence generation.

4. The Effects of Variable Speed

The sequence $\underset{\sim}{r} = r_0, r_1, r_2, \ldots$ of digits from F_q produced by the lower LFSR in Fig. 2 satisfies the homogeneous linear recursion

$$r_k + c_1 r_{k-1} + \ldots + c_L r_{k-L} = 0 \qquad k = 0,1,2,\ldots \tag{6}$$

(where k denotes the time instants of the high speed clock for this LFSR) whose characteristic polynomial is

$$c(x) = x^L + c_1 x^{L-1} + \ldots + c_L. \tag{7}$$

In what follows, we shall assume (mainly for analytic convenience) that c(x) is irreducible in $F_q[x]$. The roots of c(x) then lie in the extension field F_{q^L}, or $GF(q^L)$ to use the more usual notation. Let α be a root of c(x), (which implies that c(x) is the minimum polynomial over GF(q) of α). Then, for every A in $GF(q^L)$, the sequence $\hat{\underset{\sim}{r}}$ defined by

$$\hat{r}_k = A\alpha^k \qquad k = 0,1,2,\ldots \tag{8}$$

is a solution of (6) as can be seen by direct substitution; however, the digits in $\hat{\underset{\sim}{r}}$ lie in the extension field $GF(q^L)$ rather than in GF(q) as required for $\underset{\sim}{r}$. One remedies this by introducing the <u>trace operator</u> which maps $GF(q^L)$ into GF(q) in the manner

$$\mathrm{Tr}(\beta) = \sum_{i=0}^{L-1} \beta^{q^i}. \tag{9}$$

The trace is a linear operator with respect to the "scalar" field GF(q), i.e., for a_1 and a_2 in GF(q) and for β_1 and β_2 in $GF(q^L)$,

$$\mathrm{Tr}(a_1\beta_1 + a_2\beta_2) = a_1\,\mathrm{Tr}(\beta_1) + a_2\,\mathrm{Tr}(\beta_2). \tag{10}$$

It now follows from (10) that the GF(q) sequence $\underset{\sim}{r}$ with

$$r_k = \mathrm{Tr}(A\alpha^k) \qquad k = 0,1,2,\ldots \tag{11}$$

is a solution of (6) for every A in $GF(q^L)$. In fact, since each choice of A gives a different sequence $\underset{\sim}{r}$, (11) gives <u>all</u> the GF(q) solutions of (6), as there are exactly q^L such solutions corresponding to the q^L choices of the initial conditions $r_{-1}, r_{-2}, \ldots, r_{-L}$ in (6). It is convenient to associate A with the corresponding initial state $[r_{-1}, r_{-2}, \ldots, r_{-L}]$ of the lower LFSR in Fig. 2.

Now consider the sequence

$$\underset{\sim}{r}[d] = r_0, r_d, r_{2d}, \ldots \tag{12}$$

that appears at the input tap of the lower LFSR at the (slower) clock

times j of the surrounding logic. We see from (12) that $\underset{\sim}{r}[d]$ is just the d-th decimation of the sequence $\underset{\sim}{r}$ and moreover that

$$(\underset{\sim}{r}[d])_j = r_{dj} = \mathrm{Tr}(A\alpha^{dj}). \tag{13}$$

It is illuminating to write (13) as

$$(\underset{\sim}{r}[d])_j = \mathrm{Tr}(A\beta^j) \tag{14a}$$

where $$\beta = \alpha^d. \tag{14b}$$

The period T of the original LFSR is the smallest positive integer t such that c(x) divides x^t-1; equivalently, T is the multiplicative order of α in $GF(q^L)$ and thus $\alpha, \alpha^2, .., \alpha^{T-1}, \alpha^T = 1$ are the T distinct roots of x^T-1. By proper choice of d ($1 \leq d \leq T$), it follows from (14b) that β can be selected as any root of x^T-1 and hence as a root of any monic irreducible polynomial that divides x^T-1. The multiplicative order of β in $GF(q^L)$ will be T/gcd(d,T). The following proposition, which is a mild generalization of known results for the decimation of maximal-length sequences, is now an immediate consequence of (14a).

Proposition 1: If the sequence $\underset{\sim}{r}$ produced by an L-stage LFSR of period T, whose characteristic polynomial is the minimum polynomial over GF(q) of the element α, is observed at intervals of d clock cycles, then this observed sequence $\underset{\sim}{r}[d]$ is a sequence producible by the LFSR of period T/gcd(d,T), whose characteristic polynomial is the minimum polynomial over GF(q) of $\beta = \alpha^d$. Moreover, every sequence producible by the latter LFSR is equal to $\underset{\sim}{r}[d]$ for some choice of the initial state of the former LFSR.

The practical import of Proposition 1 is that multiple-clocking provides a means by which a single LFSR with fixed feedback connections can be used to generate sequences that appear to be produced by LFSR's with different feedback connections. We shall call the LFSR, whose characteristic polynomial is the minimum polynomial over GF(q) of $\beta = \alpha^d$, the LFSR simulated by the LFSR, whose characteristic polynomial is the minimum polynomial over GF(q) of α, when the latter LFSR is shifted at d times the observation rate.

Now consider the sequence $\underset{\sim}{r}^i[d]$ observed in the i-th stage of the lower LFSR of Fig. 2 at the slower clock times j of the surrounding logic. Then

$$(\underset{\sim}{r}^i[d])_j = r_{dj-i} = \mathrm{Tr}(A\alpha^{dj-i})$$

$$= \mathrm{Tr}(A\alpha^{-i}\beta^j) \qquad (15)$$

where β is given by (14b). (Note that $\underset{\sim}{r}[d] = \underset{\sim}{r}^0[d]$.) From (15), we see that this sequence is again a sequence producible by the LFSR simulated by the faster-shifting LFSR. We now consider the relationship between the sequences observed in adjacent stages of the faster-shifting LFSR.

If $\underset{\sim}{s} = s_0, s_1, s_2, \ldots$ is any periodic sequence, we shall call the sequence $\theta^n\underset{\sim}{s} = s_n, s_{n+1}, s_{n+2}, \ldots$ the <u>n-th phase</u> of the sequence $\underset{\sim}{s}$. If $\underset{\sim}{s}$ can be described as

$$s_j = \mathrm{Tr}(C\gamma^j) \qquad j = 0,1,2,..$$

then it follows that

$$(\theta^n\underset{\sim}{s})_j = \mathrm{Tr}(C\gamma^n\gamma^j) \qquad j = 0,1,2,.. \qquad (16)$$

so that the phase shift n can be read off by comparing the multipliers of γ^j in the trace descriptions.

In general, the sequences $\underset{\sim}{r}^i[d]$, for $i = 1,2,..,L$, will not be phase shifts of one another; rather, they will be "cyclically distinct" sequences producible by the simulated LFSR. However, when gcd(d,T) = 1 so that the simulated and simulating LFSR's have the same period, the sequences $\underset{\sim}{r}^i[d]$ will be phase shifts of one another. To see this, we note that gcd(d,T) = 1 means that d has a multiplicative inverse e modulo T, i.e., there exists an integer e ($1 \leq e < T$) such that

$$de = QT + 1$$

and hence

$$\beta^e = \alpha^{de} = \alpha^{QT+1} = \alpha. \qquad (17)$$

In this case, we can write (15) as

$$(\underset{\sim}{r}^i[d])_j = \mathrm{Tr}(A\beta^{-ie}\beta^j) \qquad (18)$$

The following proposition now follows from (18) and (16).

Proposition 2: When gcd(d,T) = 1, then the sequence $\underset{\sim}{r}^{i}[d]$ observed every d clock cycles in the i-th stage of an LFSR of length L and period T with an irreducible characteristic polynomial over GF(q) and non-zero initial loading is the e-th phase of the sequence $\underset{\sim}{r}^{i+1}[d]$ observed every d clock cycles in the (i+1)-st stage, where e ($1 \leq e < T$) is the multiplicative inverse of d modulo T.

The practical import of Proposition 2 is that simulating an LFSR by multiple-clocking of another LFSR gives simultaneous access to widely separated phases of the sequence produced by the simulated LFSR, rather than only to consecutive phases as when this sequece is produced by the actual LFSR being simulated.

Proposition 3: The sequences $\underset{\sim}{r}^{i}[d]$, (i = 1,2,..,L), described in Proposition 2 are linearly independent over GF(q).

To prove this propsition, it suffices to show that the initial constants $A\beta^{-ie}$ (i = 1,2,..,L) in the trace descriptions of the L sequences are linearly independent over GF(q). If not, there would exist a_i (i = 1, 2,..,L) in GF(q) not all zero such that $a_1\beta^{-e} + a_2\beta^{-2e} + \ldots + a_L\beta^{-Le}=0$, and hence β^e would be the root of a non-zero polynomial over GF(q) with degree less than L. But this is impossible since $\beta^e = \alpha$ has a minimum polynomial of degree L.

The practical import of Proposition 3 is that any sequence producible by the simulated LFSR can be obtained by a linear combination of the contents of the faster-shifting LFSR taken at the slower observation times. But this is not too surprising since any such sequence could also be produced by linear combinations of the contents of the actual LFSR being simulated. It does show, however, that no flexibility is lost when the LFSR is simulated by a faster LFSR observed under a slower clock, rather than directly implemented.

5. Linear Complexity of the Random Sequence Generator

The linear complexity $\Lambda(\underset{\sim}{z})$ of a periodic sequence $\underset{\sim}{z}$ is the degree L of the characteristic polynomial of smallest degree among those LFSR's that produce the sequence $\underset{\sim}{z}$, i.e., the length of the shortest LFSR that produces $\underset{\sim}{z}$. Linear complexity is widely used in cryptographic analysis

despite its limitations as a "true complexity" measure for sequences. We now compute the linear complexity of the sequence $\underset{\sim}{z}$ produced by the generator of Fig. 2 when the two component LFSR's have irreducible characteristic polynomials and relatively prime lengths L and M.

As we shall be dealing with extensions $GF(q^n)$ of $GF(q)$ for different n, we shall denote the trace operator from $GF(q^n)$ to $GF(q)$ by T_n so that, for γ in $GF(q^n)$,

$$T_n(\gamma) = \sum_{i=0}^{n-1} \gamma^{q^i}. \tag{19}$$

We shall make key use of the following identity, which is of some independent interest.

Lemma 1: If γ and δ are in $GF(q^L)$ and $GF(q^M)$, respectively, where $\gcd(L,M) = 1$, then

$$T_L(\gamma)T_M(\delta) = T_{LM}(\gamma\delta). \tag{20}$$

Note that $GF(q^L)$ and $GF(q^M)$ are both subfields of $GF(q^{LM})$ so that the product in (20) of $\gamma\delta$ is well-defined in $GF(q^{LM})$. To prove (20), we first note from (19) that

$$T_{LM}(\gamma\delta) = \sum_{i=0}^{LM-1} \gamma^{q^i}\delta^{q^i}. \tag{21}$$

Next, we observe that, because $\gamma \in GF(q^L)$ and $\delta \in GF(q^M)$,

$$\gamma^{q^i} = \gamma^{q^{i \bmod L}} \tag{22a}$$

and

$$\delta^{q^i} = \delta^{q^{i \bmod M}} \tag{22b}$$

where "i mod n" denotes the remainder when i is divided by n. Because $\gcd(L,M) = 1$, the Chinese remainder theorem implies that (i mod L, i mod M) takes on each pair (j,k) with $0 \leq j < L$ and $0 \leq k < M$ exactly once as i ranges from 0 to LM-1. Thus (21) and (22) imply

$$T_{LM}(\gamma\delta) = \sum_{j=0}^{L-1} \sum_{k=0}^{M-1} \gamma^{q^j} \delta^{q^k}$$

$$= \sum_{j=0}^{L-1} \gamma^{q^j} \sum_{k=0}^{M-1} \delta^{q^k}$$

which we recognize now from (19) to be the desired identity (20).

The following result is proved in [3] and is a simple consequence of the fact that the degree of the minimum polynomial over GF(q) of γ is the least positive integer t such that $\gamma^{q^t} = \gamma$.

Lemma 2: If the minimum polynomials of β and γ over GF(q) have degrees L and M, respectively, and gcd(L,M) = 1, then the minimum polynomial of $\beta\gamma$ over GF(q) has degree LM.

Now suppose that the characteristic polynomials c(x) and b(x) of the two LFSR's of Fig. 2 are irreducible, that α is a root of c(x) and γ is a root of b(x), that $\beta = \alpha^d$ has the same multiplicative order in $GF(q^L)$ as α, and that the degrees L and M satisfy gcd(L,M) = 1. Then the i-th input sequence $\underset{\sim}{w}^i$ to the adder forming $\underset{\sim}{z}$ in Fig. 2 is given according to (18) by

$$(\underset{\sim}{w}^i)_j = T_L(A\beta^{-ie}\beta^j)T_M(B\gamma^{-i}\gamma^j), \tag{23}$$

where A and B are non-zero elements of $GF(q^L)$ and $GF(q^M)$ if, as we now assume, the initial states of the LFSR's are both non-zero. Using (20), we obtain

$$(\underset{\sim}{w}^i)_j = T_{LM}(AB\beta^{-ie}\gamma^{-i}(\beta\gamma)^j). \tag{24}$$

It now follows from Lemma 2 and (11) that $\underset{\sim}{w}^i$ is a non-zero sequence produced by an LFSR with an irreducible characteristic polynomial of degree LM, and hence that the linear complexity of $\underset{\sim}{w}^i$ is

$$\Lambda(\underset{\sim}{w}^i) = LM. \tag{25}$$

In fact, we see from (24) that each sequence $\underset{\sim}{w}^i$ is produced by this same LFSR of length LM and hence so also is their sum

$$\underset{\sim}{z} = \sum_{i=1}^{M} \underset{\sim}{w}^i. \tag{26}$$

To show that

$$\Lambda(\underset{\sim}{z}) = LM, \tag{27}$$

it remains only to show that $\underset{\sim}{z}$ is not the all-zero sequence $\underset{\sim}{0}$. Now (10), (24) and (26) show that $\underset{\sim}{z} = \underset{\sim}{0}$ only if

$$\sum_{i=1}^{M} (\beta^{e}\gamma)^{i} = \sum_{i=1}^{M} (\alpha\gamma)^{i} = 0,$$

where we have made use of (17), which would require $\alpha\gamma$ to be the root of the polynomial $x^{M-1} + x^{M-2} + \ldots + x+1$ over GF(q). But this cannot be the case since Lemma 2 shows that the minimum polynomial of $\alpha\gamma$ over GF(q) has degree LM. We have thus proved our desired result, namely:

Proposition 4: When the two LFSR's in Fig. 2 have relatively prime lengths, irreducible connection polynomials and non-zero initial states, and when the speed factor d is relatively prime to the period T of the faster-shifting LFSR, then the output sequence $\underset{\sim}{z}$ will have linear complexity LM as will also each of the input sequences to the adder that forms $\underset{\sim}{z}$ in Fig. 2.

6. Remarks

One could of course utilize the sequence $\underset{\sim}{z}$ produced by the generator of Fig. 2 as the sequence "$\underset{\sim}{x}$" in another such "Fig. 2 generator", where the second LFSR would now be shifted at another speed factor d'. If this second LFSR has length N and gcd(LM,N) = 1, we see from Proposition 4 that we could obtain output sequences of linear complexity LMN. This process could be iterated as many times as desired.

One could also modify the Fig. 2 generator by also shifting the upper LFSR at another speed factor d'. The analysis of such generators is an obvious modification of that presented in this paper.

Finally, the reader may wonder why, in light of Proposition 4, one does not save hardware by using one of the $\underset{\sim}{w}^{i}$ sequences as the generator output since its linear complexity equals that of $\underset{\sim}{z}$. The answer is that the sequence $\underset{\sim}{w}^{i}$ may have a gross imbalance of 0's to 1's (when q = 2) and/or other short term "non-random" features. The intuitive argument of Section 3 that suggested the structure of the Fig. 2 generator also suggests that the short term statistics of $\underset{\sim}{z}$ will be much more "random" than those of $\underset{\sim}{w}^{i}$. It appears feasible to carry out an analysis to verify this suspicion, but such an analysis is beyond our aim in this paper, which was to show the many interesting features that mulitple clocks can introduce in sequence generators. When such sequences generators are used for cryptographic purposes, the various speed factors can be put

under control of the secret key. Thus, such multiple-clocking gives an added "dimension" to secure sequence generator design.

References

[1] J.L. Massey, "Error Bounds for Tree Codes, Trellis Codes, and Convolutional Codes with Encoding and Decoding Procedures", pp.1-57 in Coding and Complexity (Ed. G.Longo), CISM Courses and Lectures No. 216, Springer-Verlag, Vienna and New York, 1975.

[2] C.E. Shannon, "Communication Theory of Secrecy Systems", Bell System Tech. J., vol. 28, pp.656-715, October 1949.

[3] E.L. Key, "An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators", IEEE Trans. Info. Th., vol. IT-22, pp.732-736, Nov. 1976.