Vulnerability of Nonlinear Filter Generators Based on Linear Finite State Machines

Jin Hong[1], Dong Hoon Lee[1], Seongtaek Chee[1], and Palash Sarkar[2]

[1]National Security Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{jinhong,dlee,chee}@etri.re.kr

[2]Indian Statistical Institute
203, B.T. Road, Kolkata 700108, India
palash@isical.ac.in

Abstract. We present a realization of an LFSM that utilizes an LFSR. This is based on a well-known fact from linear algebra. This structure is used to show that a previous attempt at using a CA in place of an LFSR in constructing a stream cipher did not necessarily increase its security. We also give a general method for checking whether or not a nonlinear filter generator based on an LFSM allows reduction to one that is based on an LFSR and which is vulnerable to Anderson information leakage.

Keywords : stream cipher, nonlinear filter model, LFSR, CA, Anderson information leakage.

1 Introduction

Linear feedback shift registers (LFSR) are one of the most useful building blocks for constructing stream ciphers. There are classical models of memoryless synchronous stream ciphers that utilize LFSR's : the nonlinear filter model (NF) and the nonlinear combiner model (NC).

For the NF model, building on previous works[0,0], Anderson[0] showed that much information about the state of the LFSR may be obtained from the output key stream, if the distribution of possible states of LFSR's in relation to output stream blocks is not uniform. And for a random NF model, this is usually quite irregular.

In the paper [0], presented at CRYPTO 2002, a model that combines the NF and NC models was introduced. This model utilized a cellular automaton (CA) instead of an LFSR to eliminate the above mentioned information leakage of NF models. In the paper, it is claimed that this non-uniform distribution stems from the fact that a particular state bit of an LFSR affects the output key stream several times. This would be unavoidable in an NF generator based on an LFSR. It is also claimed that this property can be avoided through the use of a CA, thus removing Anderson information leakage. In this paper, we show this claim to be incorrect.

CA is a special case of linear finite state machines (LFSM) and can be viewed as a one-dimensional array of cells. The cells change state at each clock tick, and the new state of a cell is completely determined by its present state and those of its left and right neighbors. CA's have been applied to various fields such as biological system, fault-tolerant computation, VLSI design, and cryptography. (See [0] for a survey on the general theory of CA.) In the cryptographical field, CA's have been used in designing hash functions and stream ciphers[0,0]. It was believed that from the security perspective, a CA would give properties better than those of an LFSR. However, we shall show that the use of a CA in place of an LFSR does not necessarily increase security.

Recalling a well-known fact from linear algebra, we give a way to realize an LFSM, utilizing an LFSR. In short, the realization is done by attaching a linear map to an LFSR. We understand that, due to its simplicity, this could have been known to experts of this field. But we could not find any references, and it seems that this fact was not looked at from the security perspective.

The realization could be of interest in its own. For example, it gives a natural way of running a CA in the reverse direction, something which was thought to be a complex procedure. But as will be shown through the examination of arguments in [0], this realization also has grave consequences in the use of LFSM's as cryptographic building blocks.

The paper is organized as follows. Section shall present the simple mathematical fact that is the starting point of this paper. This is used in Section in realizing an LFSM using an LFSR and a linear map. In Section , we review the notion of Anderson information leakage and examine the system given in [0]. Using the realization of a CA which utilizes an LFSR, we shall argue that the system did not achieve its design goal. The section that follows presents an explicit example confirming these arguments. Next, in Section we give a general method for checking whether or not a given NF generator based on an LFSM admits a reduction to a NF generator based on an LFSR that is vulnerable to Anderson information leakage. The last section closes the paper with some concluding remarks. Some readers might want to read Appendix , which contains remarks on what further developments the basic idea of this paper might bring.

2 Basic facts and definitions

In this section we shall recall some elementary facts from linear algebra and introduce two classes of linear finite state machines, CA and LFSR.

2.1 Linear Algebra

Let us denote by I the $n \times n$ identity matrix. The characteristic polynomial of a matrix M with entries in the binary field $\mathbf{F}_2$ is defined to be the polynomial

$$\mathrm{char}(M) = \det(xI - M) \in \mathbf{F}_2[x]. \tag{1}$$

We define the *companion matrix* of a monic polynomial

$$p(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + x^n \in \mathbf{F}_2[x] \tag{2}$$

to be the matrix

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & & & & & 0 \\ 0 & 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & a_2 & \cdots & \cdots & a_{n-2} & a_{n-1} \end{pmatrix}. \tag{3}$$

We shall accept the following statement as a fact.

Fact 1 *Let $p(x)$ be the characteristic polynomial of a square matrix M. Denote by L, the companion matrix of the monic polynomial $p(x)$. If $p(x)$ is irreducible, then there exists an invertible (basis transition) matrix T satisfying*

$$TMT^{-1} = L.$$

This is the only fact from linear algebra we shall need in this paper. Readers familiar with linear algebra can look up Appendix to see justification for this fact.

Remark 1. The matrix T appearing in this fact is not unique. If the size of the square matrix M is n, there can be up to $2^n - 1$ of them.

2.2 Linear finite state machine

An n-bit *linear finite state machine* (LFSM), denoted by $\mathcal{M}$, is a pair $(\mathbf{F}_2^n, M)$, where M is an $n \times n$ matrix. The internal state of $\mathcal{M}$ is described by an n-bit vector $\mathbf{v} = (v_0, \ldots, v_{n-1}) \in \mathbf{F}_2^n$. The evolution of $\mathcal{M}$ over discrete time $t \geq 0$ is described by a sequence of n-bit vectors $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots$, satisfying

$$\mathbf{v}^{(t+1)} = M\mathbf{v}^{(t)}. \tag{4}$$

Here, $\mathbf{v}^{(0)}$ is the initial state. For $t \geq 0$, we shall write

$$\mathbf{v}^{(t)} = (v_0^{(t)}, v_1^{(t)}, \ldots, v_{n-1}^{(t)}).$$

It is well known that if the characteristic polynomial of M is *primitive* over $\mathbf{F}_2$, then each of the sequences

$$\mathbf{v}_i = (v_i^{(t)})_{t \geq 0} \tag{5}$$

has period $2^n - 1$ [0]. This is the maximum possible period obtainable for the state sequence of an LFSM. The most popular subclasses of the LFSM's are CA's and LFSR's.

2.3 Cellular automaton

A *cellular automaton* (CA) is an LFSM with a defining matrix M which is tridiagonal. If the upper and lower subdiagonal entries of M are all equal to 1, then it is called a 90/150 CA. Visually, the general matrix defining a 90/150 CA will be of the form

$$\begin{pmatrix} c_0 & 1 & 0 & 0 \cdots & \cdots & 0 \\ 1 & c_1 & 1 & 0 \cdots & \cdots & 0 \\ 0 & 1 & c_2 & 1 & & 0 \\ \vdots & & & & & \vdots \\ & & & & 1 & 0 \\ 0 & \cdots & & 1 & c_{n-2} & 1 \\ 0 & \cdots & & 0 & 1 & c_{n-1} \end{pmatrix}, \tag{6}$$

where each c_i is either 0 or 1. We shall only consider 90/150 CA's in this paper. The sequences obtained from such a CA satisfies the following relation. For each $0 \le i \le n-1$ and $t \ge 0$,

$$v_i^{(t+1)} = v_{i-1}^{(t)} \oplus c_i v_i^{(t)} \oplus v_{i+1}^{(t)},$$

where we take $v_{-1}^{(t)} = v_n^{(t)} = 0$.

2.4 Linear feedback shift register

A *linear feedback shift register* (LFSR) corresponding to a monic polynomial $p(x)$ given by (2) is an LFSM with the defining matrix set to the companion matrix of $p(x)$ given by (3). So if we write the internal state of the LFSR at time $t \ge 0$ as $\mathbf{v}^{(t)} = (v_0^{(t)}, v_1^{(t)}, \ldots, v_{n-1}^{(t)})$, we have

$$v_i^{(t+1)} = v_{i+1}^{(t)}$$

for each $0 \le i \le n-2$, and

$$v_{n-1}^{(t+1)} = a_0 v_0^{(t)} \oplus a_1 v_1^{(t)} \oplus \cdots \oplus a_{n-1} v_{n-1}^{(t)}.$$

Hence, register contents will be shifted to the *left* by one cell during the evolution process.

3 Reducing an LFSM to an LFSR

In this section, we shall see how the contents of the previous section relate to each other. We give a way to realize an LFSM, utilizing an LFSR and a linear map. It seems that the method we are going to give is known to the experts of this field. But we could not find any references, so it is explained here for completeness.

Let us be given an LFSM $\mathcal{M}$ defined by a matrix M. Denote the characteristic polynomial of M by $p(x)$ and the companion matrix of $p(x)$ by L. Notice that the matrix L defines an LFSR. We say that this LFSR is *associated* with the LFSM $\mathcal{M}$.

Suppose that $p(x)$ is irreducible. Then we know from Fact 1 that there exists some invertible matrix T such that

$$TMT^{-1} = L. \tag{7}$$

Now, recalling that the evolution of LFSM internal state is given by (4), if the initial state of the LFSM $\mathcal{M}$ was $\mathbf{v}$, the state $\mathbf{v}^{(t)}$ of the LFSM at time $t \geq 0$ will be given by

$$\mathbf{v}^{(t)} = M^t \mathbf{v}.$$

Here, M^t denotes M multiplied t times and not the transpose of M. Similarly, if the initial state of the LFSR defined by the matrix L was $\mathbf{w}$, the state $\mathbf{w}^{(t)}$ of the LFSR at time $t \geq 0$ will be given by

$$\mathbf{w}^{(t)} = L^t \mathbf{w}.$$

Now, if we had $\mathbf{w} = T\mathbf{v}$, using (7), we can easily check the following sequence of equalities.

$$\mathbf{v}^{(t)} = M^t \mathbf{v} = (T^{-1}LT)^t \mathbf{v} = T^{-1}L^t T\mathbf{v} = T^{-1}L^t \mathbf{w} = T^{-1}\mathbf{w}^{(t)}. \tag{8}$$

This shows that an LFSM is intimately related to the LFSR defined by its characteristic polynomial.

Proposition 1. *The current internal state* $\mathbf{v}^{(t)}$ *of an LFSM which starts at the initial state* $\mathbf{v}^{(0)}$ *may be calculated using the internal state* $\mathbf{w}^{(t)}$ *of the associated LFSR through the simple linear equation*

$$\mathbf{v}^{(t)} = T^{-1}\mathbf{w}^{(t)} \tag{9}$$

by initializing the LFSR with $\mathbf{w}^{(0)} = T\mathbf{v}^{(0)}$.

So, even though an LFSM seems much more complicated than an LFSR, the two are only apart by a simple linear transformation.

The only hypothesis on the LFSM we have used in this section is that its characteristic polynomial be irreducible. In most cryptographic applications of an LFSM, the characteristic polynomial will be taken to be primitive, in order to achieve maximal period, so this is not a very restricting assumption. Hence any cryptographic system that bases its safety on the complexity of an LFSM, compared to an LFSR, may not be as safe as it seems at first sight. Since CA's are just a special type of LFSM's the same can be said of systems using CA's.

4 Security of nonlinear filter models utilizing a CA

In this section, we shall present a system which has tried to use a CA in place of an LFSR in order to remove some unwanted property of a stream cipher. We shall apply the theory of Section to show that the attempt did not succeed in achieving its goal.

4.1 The NF-CA model

In the paper [0], a memoryless synchronous stream cipher called the *filter-combiner* (FC) model was introduced. We shall not present the whole FC model in this paper, but use only a small part of the model in explaining one of the main arguments of that work. The referenced paper contains more than what is presented here.

Let $\mathcal{M} = (\mathbf{F}_2^n, M)$ be a CA. We assume that the characteristic polynomial $p(x)$ of the CA is primitive. It is known that each of the n sequences given by (5) are all exactly the same periodic sequence with only the starting points different. Hence they are relative shifts of each other.

We apply a nonlinear filter f with good properties, for example, high resiliency and nonlinearity, on the cells of the CA to obtain a stream cipher. The system is to satisfy the following loosely stated constraints. We refer the reader to the original paper [0] for exact statements.

1. The number r of cells used as inputs to f is small relative to the size n of the CA. ($r \leq \log_2 n$)
2. The starting points of the sequences obtained from the cells used as inputs to f is (almost) evenly distributed within the common periodic sequence.
3. The number of bits encrypted using the system does not come close to $2^n/r$.

We shall call this reduced model by the name NF-CA, a nonlinear filter model utilizing a CA. The paper claims that under these constraints the NF-CA is resistant to Anderson information leakage [0].

Anderson information leakage is an observation on the nonlinear filter model (a stream cipher that applies a nonlinear filter on an LFSR) that allows one to gather information on the initial state of the LFSR from the key stream. More explanation is given in the next subsection.

The author of [0] believed that Anderson information leakage was fundamentally due to using the same bit more than once as input to the nonlinear filter in obtaining the key stream. This reasoning was also somewhat vaguely stated in the paper [0]. Hence the main objective behind the above constrains was to remove the possibility of any part of the periodic sequence being used more than once.

4.2 Anderson information leakage

Consider the filter model of stream ciphers. This is a stream cipher that uses cell states of an LFSR as inputs to a nonlinear filter in obtaining a key stream. We shall write this model as NF-LFSR for short.

Suppose we use k consecutive cells of the LFSR as inputs to the nonlinear filter f. Let us take the convention, as given in Section , that the contents of the register are being shifted to the left at each step. If we fix the contents of k cells used as inputs to f, we can calculate one bit of output from the NF-LFSR. Similarly, if we fix contents of the k cells and also $(k-1)$ more cells that lie immediately to their right, we can calculate k output bits from the NF-LFSR.

Now, suppose we classify all possible $(2k-1)$-bit states according to the k-bit output key stream it will give. In the ideal case, each class will contain exactly 2^{k-1} elements. That this distribution usually is not uniform was investigated by Anderson [0] to show that much information about the state of the LFSR may be obtained from the output key stream.

He gives an explicit example using a 2-resilient nonlinear filter that uses 5 variables. The above mentioned table is constructed to show that it is indeed irregular. To show that actually useful information may be found, he lists all possible 9-bit initial states that can give the 5-bit output stream `11010`.

001010101	001110010	100110010	101110001	110110001
001110001	100110001	101001011	101110010	110110010

If we look closely at these values, we see that there is only a single `0` among all the 5th bits, and a single `1` among both 6th and 7th. In other words, if the key stream we obtain is `11010`, then at the starting point of this key stream, the state of the 5th cell of the LFSR would have been `1` with probability 0.9. Likewise, state of the 6th and 7th bit would have been `0` with probability 0.9.

Irregularity in the distribution of initial states classified according to output stream blocks contains potential for the NF-LFSR giving out information on the initial LFSR state.

We remark that some further developments of Anderson's idea appear in [0,0,0].

4.3 Information leakage of the NF-CA

Let the initial state of the NF-CA, or equivalently, that of the CA $\mathcal{M} = (\mathbf{F}_2^n, M)$ be denoted by $\mathbf{v}^{(0)}$. We shall add dummy variables to the nonlinear filter f and view it as defined on the whole CA. Then the t-th output key stream bit c_t of the NF-CA will be given by

$$c_t = f(\mathbf{v}^{(t)}). \tag{10}$$

We may follow through the arguments of Section in constructing an associated LFSR and finding a linear transformation T satisfying (7). And by applying (9) to the above equation, we may write

$$c_t = f \circ T^{-1}(\mathbf{w}^{(t)}), \tag{11}$$

where we have taken the initial state of the associated LFSR to be $\mathbf{w}^{(0)} = T\mathbf{v}^{(0)}$. Notice that since T^{-1} is a simple linear map, we may view the map $g = f \circ T^{-1}$ as just another normal nonlinear filter. That is, we have

$$c_t = g(\mathbf{w}^{(t)}). \tag{12}$$

We see that the right hand side is now the output of a normal NF-LFSR.

Proposition 2. *The NF-CA which uses nonlinear filter f on a CA initialized to $\mathbf{v}^{(0)}$ may be realized as an NF-LFSR. This is done by applying the nonlinear filter $g = f \circ T^{-1}$ to the associated LFSR and initializing it to $\mathbf{w}^{(0)} = T\mathbf{v}^{(0)}$.*

Now, we do not yet have any criterion for measuring an NF-LFSR's resistance to Anderson information leakage. And, as stated in Anderson's paper [0], random NF-LFSR's tend to leak a lot of information. Hence there is a non-dismissible chance of (12) and hence (10) representing a stream cipher which is not resistant to Anderson information leakage.

Remark 2. Anderson information leakage does not seem to be applicable to the nonlinear combiner model. Hence Anderson information leakage is probably not applicable to the FC model of [0]. But, once again, this is due to the use of *combiner* part of the FC model rather than from the three constraints of Section .

5 Explicit example of leaking NF-CA

We have constructed a small but concrete example to verify that it is possible for an NF-CA to satisfy all three of the constraints introduced in Section and still show Anderson information leakage.

5.1 CA and its relation to an LFSR

Consider the 90/150 CA represented by a matrix M of the form (6) with the diagonal entries given by

$$(c_0, c_1, \ldots, c_{22}) = (1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0).$$

For clarity, we have written down the explicit matrix M in Appendix . Characteristic polynomial $p(x)$ of M is

$$1 + x^2 + x^4 + x^7 + x^9 + x^{10} + x^{11} + x^{13} + x^{14} + x^{15} + x^{17} + x^{19} + x^{20} + x^{22} + x^{23}.$$

This is a primitive polynomial so that each of the 23 cells of the CA gives a sequence of period $(2^{23} - 1)$. Let L be the companion matrix of $p(x)$. To define T, we let $T_1 = (1, 0, 0, \ldots, 0)$ be its top row and recursively fix the i-th row T_i by setting

$$T_i = MT_{i-1} \quad \text{for } 1 < i \leq 23. \tag{13}$$

The actual matrix T may be found in Appendix .

Checking

$$TMT^{-1} = L \tag{14}$$

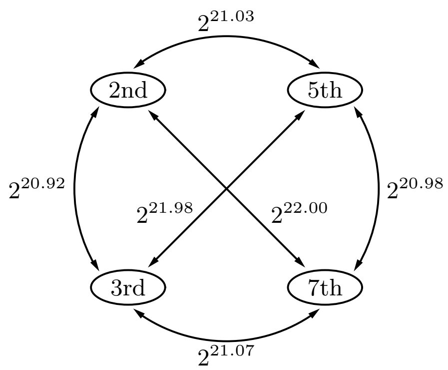
is easy. We remark that the such a T is not unique. Any invertible matrix obtained through the process (13) starting with an arbitrary nonzero vector will satisfy (14). For the above T, its inverse, T^{-1}, is given in Appendix .

5.2 Shifts between CA cells and the nonlinear filter

Let $m_i^{(t)}$ be the sequences generated by the i-th cell of the CA for $1 \leq i \leq 23$. Then the relative shifts between $m_1^{(t)}$ and $m_i^{(t)}$ are

0 **1988170** **8388605** 5964510 **4125305** 3763873 **6190462** 6778815 $\cdots$.

These have been calculated using a program implementing [0]. From this, one can check that the relative shifts between the four sequences obtained from the 2nd, 3rd, 5th, and 7th cells are quite close to 2^{21} or 2^{22}.



For example, the shift between $m_3^{(t)}$ and $m_2^{(t)}$ is

$$\begin{aligned} 1988170 - 8388605 &= -6400435 \\ &\equiv 1988172 \pmod{2^{23} - 1} \\ &\fallingdotseq 2^{20.92301\cdots}. \end{aligned}$$

Hence, if we apply the nonlinear filter given by

$$f = m_2 \oplus m_3 \oplus (m_5 \cdot m_7) \tag{15}$$

on the CA, Rule 2 of Section is satisfied. It is easily checked that f is a 1-resilient function, so we are not using a very bad filter.

That Rule 1 is also satisfied is easily checked by calculating

$$\log_2 23 = 4.52356 \cdots \geq 4.$$

To satisfy Rule 3, we just need to use less than 2^{21} bits from the NF-CA we shall create. This should not be a problem since we shall be using less than 20 key stream bits.

5.3 Equivalent NF-LFSR

We may recall from equations (10) and (11) that

$$f(\mathbf{v}^{(t)}) = (f \circ T^{-1})(\mathbf{w}^{(t)}).$$

And in terms of the state $(l_1, l_2, \ldots, l_{23})$ of the LFSR, the nonlinear filter f translates into the nonlinear filter

$$g = (f \circ T^{-1}) = (l_1 \oplus l_2 \oplus l_3) \oplus (l_2 \oplus l_4 \oplus l_5) \cdot (l_1 \oplus l_2 \oplus l_3 \oplus l_5 \oplus l_7).$$

We have used the 2nd, 3rd, 5th, and 7th rows of the explicitly calculated T^{-1} given in Appendix . Notice that the 7th bit from the LFSR is the rightmost bit used to find states of the 4 CA cells we have chosen to use for the NF-CA.

5.4 Information leakage

Since the 7th bit of LFSR remains in effect until we obtain 7 bits of the output stream, we trace the 7-bit output stream by running the obtained NF-LFSR on all possible 13-bit states of the leftmost part of the LFSR. Input count for each output is given in Table . The 7-bit output stream is represented in hexadecimal

Output	00	01	02	03	04	05	06	07
#	38	51	61	73	51	75	73	89
Output	08	**09**	0A	0B	0C	0D	0E	0F
#	65	**29**	67	63	69	69	87	63
Output	10	11	12	13	14	15	16	17
#	79	63	45	37	87	67	77	57
Output	18	19	1A	1B	1C	1D	1E	1F
#	49	73	43	59	73	85	59	71
Output	20	21	22	23	24	25	26	27
#	73	93	51	71	61	69	39	55
Output	28	29	2A	2B	2C	2D	2E	2F
#	79	83	77	49	75	43	57	49
Output	30	31	32	33	34	35	36	37
#	65	49	99	75	57	45	67	55
Output	38	39	3A	3B	3C	3D	3E	3F
#	63	71	69	85	39	59	53	73

Output	40	41	42	43	44	45	46	47
#	51	83	65	89	43	55	65	61
Output	48	49	4A	4B	4C	4D	4E	4F
#	77	53	87	71	53	33	71	67
Output	50	51	52	53	54	55	56	57
#	87	59	85	57	75	59	41	49
Output	58	59	5A	5B	5C	5D	5E	5F
#	65	101	59	63	61	69	39	55
Output	60	61	62	63	64	65	66	67
#	61	61	47	55	69	89	47	83
Output	68	69	6A	6B	6C	6D	6E	6F
#	67	59	57	41	91	79	73	45
Output	70	71	72	73	74	75	**76**	77
#	57	53	59	55	69	53	**103**	63
Output	78	79	7A	7B	7C	7D	7E	7F
#	47	43	53	81	51	75	73	89

Table 1. The number of possible input states for each 7-bit output

notation and # denotes the corresponding possible input state count. Leftmost bit of the 7 bits in hexadecimal notation(exclude the leading 0 from the 8 bits) is the first output bit.

In the ideal case, all the counts should be equal to (or, at least near) $2^6 = 64$. But as we see in Table , this is not the case. It is quite irregular. Number as big as 103 appears and number as small as 29 also appears. So this shows the potential of this structure leaking information.

For example, let us look at the following list of all 13-bit input states that give the 7-bit output stream `0x09 = 0001001`. As given by the table, there are 29 such states.

1101000000100	0011000000110	0110101110001	0110010001011
0011000000100	1011000000110	0110101110101	1110010001011
1011000000100	0111000000110	0110100011101	0110100101011
0111000000100	0110101110110	0110101110011	0110101110111
0110101110100	0110100011110	1101000001011	0110100011111
0110100011100	0000011111110	0011000001011	
0110100101010	0000011100001	1011000001011	
1101000000110	0000000110001	0111000001011	

We find that, with probability 28/29, only one of the bits 4, 5, and 6 is equal to 1. In particular, sum of the three bits is equal to 1 with probability 28/29. Anderson information leakage theory is applicable to this structure. Therefore, applying a nonlinear filter having good cryptographic properties to a CA and using cells of large relative shifts, as suggested in [0], does not necessarily prevent Anderson information leakage.

6 Checking the vulnerability of a given NF-LFSM

In this section, we shall give a general method for checking whether or not a given NF-LFSM allows reduction to a vulnerable NF-LFSR. Since CA and LFSR are subclasses of LFSM, our method applies even to NF-LFSR. That is, we can check whether the nonlinear filter to an NF-LFSR may be rewritten in a form that shows information leakage.

Let us be given an NF-LFSM defined by a matrix M of size n and a nonlinear filter f. Denote by L the companion matrix of M and write

$$\mathcal{Z}(L) = \{Z \in \mathrm{GL}(n) \mid ZL = LZ\}$$

for the centralizer of L.

For a given companion matrix L, it is easy to write down $\mathcal{Z}(L)$ more explicitly. The following lemma may be proved through a straightforward application of $ZL = LZ$.

Lemma 1. *For the companion matrix L given by* (3)*, the centralizer $\mathcal{Z}(L)$ consists of elements $Z = (z_{i,j})_{i,j=0}^{n-1} \in \mathrm{GL}(n)$ satisfying*

$$\begin{aligned}
z_{i+1,0} &= a_0 z_{i,n-1}, \\
z_{i+1,1} &= a_1 z_{i,n-1} \oplus z_{i,0}, \\
z_{i+1,2} &= a_2 z_{i,n-1} \oplus z_{i,1}, \\
&\vdots \\
z_{i+1,n-1} &= a_{n-1} z_{i,n-1} \oplus z_{i,n-2}
\end{aligned}$$

for all $0 \leq i < n-1$.

The important implication of this lemma is that every entry of a $Z \in \mathcal{Z}(L)$ may be written as a linear sum of the terms belonging to its first row in a uniform way. For example,

$$\mathcal{Z}\Bigg(\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & b & c \end{pmatrix} \Bigg) = \left\{ \begin{pmatrix} x & y & z \\ az & x \oplus bz & y \oplus cz \\ ay \oplus acz & az \oplus by \oplus bcz & x \oplus bz \oplus cy \oplus cz \end{pmatrix} \in \mathrm{GL}(3) \right\}.$$

Now, fix any matrix $\bar{T}$ satisfying $\bar{T}M\bar{T}^{-1} = L$. It is an easy exercise in linear algebra to show that the set of all T satisfying (7) is given by $\mathcal{Z}(L)\bar{T} := \{Z\bar{T} \mid Z \in \mathcal{Z}(L)\}$. And since $Z \in \mathcal{Z}(L)$ if and only if $Z^{-1} \in \mathcal{Z}(L)$, we have the following proposition.

Proposition 3. *The set of all T^{-1} satisfying* (7) *is given by*

$$\bar{T}^{-1}\mathcal{Z}(L) := \{\bar{T}^{-1}Z \mid Z \in \mathcal{Z}(L)\}.$$

We are now ready to give a general method for checking whether or not a given NF-LFSM allows reduction to a vulnerable NF-LFSR. If a nonlinear filter applied to an LFSR uses a small number of variables and if those variables correspond to LFSR cells that are close to each other, then such an NF-LFSR is vulnerable to Anderson information leakage. Otherwise, the NF-LFSR is highly immune to information leakage. Hence, for a given NF-LFSM, it suffices to check the possibility of finding a T, for which the filter $g = f \circ T^{-1}$ given by Proposition uses variables from a small clustered set.

Decide on a (small) number $s < n$. If it is possible to choose T so that all variables used by g falls within some s consecutive LFSR cells, we shall conclude that there is a high probability that the NF-LFSM yields to Anderson information leakage. Otherwise we shall presume that the NF-LFSM does not leak information.

Procedure for checking vulnerability.

1. From the matrix M of size n, defining the LFSM, calculate its characteristic polynomial and the associated LFSR L.
2. Fix any matrix $\bar{T}$ satisfying $\bar{T}M\bar{T}^{-1} = L$. Using the idea of (13) is one way to do this.
3. Write $\mathcal{Z}(L)$ in the form given by Lemma , so that all entries of lower rows are expressed as linear combinations of the first row terms. We shall denote the first row terms by $x_0, \ldots, x_{n-1}$.
4. Recalling Proposition , multiply $\bar{T}^{-1}$ to the general element of $\mathcal{Z}(L)$ obtained in the previous step to express the general T^{-1}. All entries of the general T^{-1} will again be linear combinations of x_j.

$\star$. Let r be the number of rows used by f. Note that from the general expression of T^{-1}, which is an $n \times n$ array of linear sums over x_j, only the r rows that that correspond to variables used by the nonlinear filter f will have any significance.

5. Remove all rows not corresponding to variables used by f.
⋆. Now, suppose that for some explicit nontrivial values of the variables x_j, all the remaining entries evaluate to zero, except for those contained in the first s columns. Then $g = f \circ T^{-1}$ would used only s variables for the T^{-1} evaluated at the explicit values.
6. Temporarily remove the first s columns from the remaining array of T^{-1} entries.
7. Check whether setting all remaining entries to zero yields a nontrivial solution.
8. If a nontrivial solution is found, conclude that the NF-LFSM allows reduction to a vulnerable NF-LFSR.
9. Otherwise, bring back the array of linear sums obtained after Step 5.
10. Unless we've tried all possible consecutive s columns, (temporarily) remove the next set of s consecutive columns and go back to Step 7.
11. If no nontrivial solution is found, conclude that the NF-LFSM resists Anderson information leakage.

Notice that at Step 7, we have a set of $r \times (n-s)$ equations in n variables. For most interesting values of r and s, the number of equations would be larger than the number of variables. But our (small number of) testings show that nontrivial solutions do occur from time to time.

We close this section by adding that the complexity of this process can easily be seen to be of polynomial order in n.

7 Conclusion

We have seen that an LFSM (or a CA) is intimately connected to an LFSR by the simple relation (9). This structure allows one to realize an LFSM using an LFSR and a linear transformation. Since an LFSR is much simpler than an LFSM, this will have implications on the security of any system that (iteratively) uses an LFSM as one of its building blocks, and has used it assuming that it is more complex than an LFSR.

An example of such an attempt has been examined in this paper. Using a CA in place of an LFSR in an attempt to remove Anderson information leakage from a nonlinear filter model has failed.

We have also given a general method for checking whether or not a given NF-LFSM allows reduction to an NF-LFSR which is vulnerable to information leakage.

References

1. Ross Anderson, Searching for the optimum correlation attack. *Proceedings of FSE*, LNCS 1008, pp. 137–143, Springer-Verlag, 1995.
2. Jovan Dj. Golić, On the security of nonlinear filter generators. *Fast Software Encryption*, LNCS 1039, pp. 173–188, Springer-Verlag, 1996.

3. Jovan Dj. Golic, Correlation via linear sequential circuit approximation of combiners with memory. *Advances in Cryptology - EUROCRYPT'92,* LNCS 658, pp. 113-123, Springer-Verlag.
4. Jovan Dj. Golic, Andrew Clark, and Ed Dawson, Generalized inversion attack on nonlinear filter generators. *IEEE Transations on Computers*, vol. 49 (10), pp. 1100–1109, October, 2000.
5. Thomas W. Hungerford, *Algebra.* GTM 73, Springer-Verlag, 1980.
6. Sangjin Lee, Seongtaek Chee, Sangjoon Park, and Sungmo Park, Conditional correlation attack on nonlinear filter generators. *Advances in Cryptology - ASIACRYPT'96,* LNCS 1163, pp. 360–367, Springer-Verlag, 1996.
7. R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications.* Cambridge University Press, 1994.
8. Miodrag Mihaljević, Yuliang Zheng, and Hideki Imai, A Family of Fast Dedicated One-Way Hash Functions Based on Linear Cellular Automata over GF(q), *IEICE Trans. Fundamentals*, vol.E82-A(a), pp. 1–8, January 1999.
9. W. Meier and O. Staffelbach, Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, vol.1, pp. 159–176, 1989.
10. Palash Sarkar, The filter-combiner model for memoryless synchronous stream ciphers. *Advances in Cryptology - CRYPTO 2002,* LNCS 2442, pp. 533–548, Springer-Verlag, 2002.
11. Palash Sarkar, Computing shifts in 90/150 cellular automata sequences. *Finite Fields and their Applications*, vol. 9 (2), pp. 175–186, April 2003.
12. Palash Sarkar, Brief History of Cellular Automata, *ACM Computing Serveys*, vol. 32 (1), pp. 80–107, March 2000.
13. Palash Sarkar, Hiji-bij-bij: A New Stream Cipher with a Self-Synchronizing Mode of Operation, ICAR e-print 2003-014, 2003. `http://eprint.iacr.org`

A Explanation for Fact 1

Let us denote by E, a vector space over a field $\mathbf{K}$. We have gathered some well-known facts from linear algebra in the following theorem. Basic definitions and proofs may be found in standard textbooks (for example, [0]).

Theorem 1. *Let $\phi : E \to E$ be a linear transformation. Then the following statements hold.*

1. *There exists monic polynomials of positive degree $p_1, \ldots, p_t \in \mathbf{K}[x]$ and ϕ-cyclic subspaces $E_1, \ldots, E_t$ of E such that $E = E_1 \oplus \cdots \oplus E_t$ and $p_1 | p_2 | \cdots | p_t$.*
2. *The sequence $p_1, \ldots, p_t$ is uniquely determined by E and ϕ. (This is called the* invariant factors *of ϕ.)*
3. *If E is a ϕ-cyclic space and ϕ has minimal polynomial $p(x)$ of degree r, then $\dim_K E = r$ and there exists an ordered basis of E relative to which the matrix of ϕ is the* companion matrix *of $p(x)$.*
4. *The characteristic polynomial of ϕ is the product of its invariant factors.*

From these statements, we may easily obtain the following corollary, stated in terms of matrices.

Corollary 1. *Let A be an $n \times n$ matrix with entries in the field $\mathbf{K}$. If the characteristic polynomial of A is irreducible, then the matrix A is* similar *to the companion matrix of the characteristic polynomial.*

To make it easier for those without a mathematical background and to make everything explicit, we shall explain this corollary, giving out some basic definitions.

The companion matrix of a monic polynomial $p(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + x^n \in \mathbf{K}[x]$ is usually defined to be the matrix

$$\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & 0 & -a_2 \\ \vdots & & & & \vdots & \vdots \\ \vdots & \vdots & & & 0 & \vdots \\ 0 & 0 & \cdots & 0 & 1 & -a_{n-1} \end{pmatrix}.$$

We can see that the form given by (3) is the transpose of this one, if we take into consideration the fact that we were dealing with the binary field there.

Now, let $p(x)$ be the characteristic polynomial of a square matrix A and let B be the companion matrix of $p(x)$. Corollary states that if $p(x)$ is irreducible, then there exists some invertible matrix C satisfying

$$CAC^{-1} = B.$$

Notice that we may take the transpose of both sides to obtain

$$(C^T)^{-1} A^T C^T = B^T.$$

It is clear that the characteristic polynomial of A is equal to that of A^T. Hence, Fact 1 follows.

B Matrix M defining the CA

```
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
```

C Matrix $\boldsymbol{T}$

```
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0
1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 1 0 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 1 1 0 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
1 0 0 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 0 0 0 0 0
1 1 0 1 1 0 0 1 0 0 0 1 0 1 1 0 1 0 1 0 0 0 0
0 0 0 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 0 0 0
0 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 0 1 0 0
0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 1 0 1 1 0
1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 1 1
```

D Matrix $\boldsymbol{T}^{-1}$

```
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0
1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0
1 1 0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0
0 0 1 1 0 1 0 0 1 0 1 0 0 0 1 1 1 1 0 0 0 0 0
1 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0
0 1 0 0 1 0 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0
1 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 0
1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 1 0
1 0 1 0 0 0 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1
```

E Remarks on further developments

Here, we have gathered some remarks on what other implications the basic idea of this paper might have.

Remark 3. The arguments of this paper need not be constrained to the binary field. An LFSM which uses cells representing elements of any finite field can be realized using an LFSR over the same finite field.

Remark 4. One can deduce from Fact 1 that any two square matrices with a common irreducible characteristic polynomial are related by an invertible matrix. So even though we have focused on the relationship between an LFSM and an LFSR, the arguments of this paper can be applied in realizing an LFSM using a different (and maybe simpler) LFSM.

Remark 5. Those familiar with linear algebra will know that it is easy to extend the idea of this paper to the case when the characteristic polynomial of an LFSM is not irreducible. In such a case, the resulting realization will use several LFSR's whose sizes add up to the size of the original LFSM.

Remark 6. One can view the idea of this paper from a different direction and use it in realizing an LFSR with an LFSM. In this case, we can exploit our freedom over the choice of transition matrixes. So, for example, applying it to an NF-LFSR, one can turn any nonlinear filter of high resiliency into a filter having correlation of degree one.

Remark 7. In a way, Anderson information leakage is fundamentally due to the fact that inputs to different variables of the filter is supplied by sequences that are just small shifts of each other. This paper shows that as long as LFSM's are used, this is unavoidable. So it might be a good idea to look at *nonlinear* feedback shift registers now.