Correlations in Stream Ciphers: A Systems Theory Point of View

S. D. Cardell, G. Maze, J. Rosenthal and U. Wagner

***Abstract*— Given a sequence of some autonomous behavior, this sequence can be computed as the output of a linear system. If one receives a highly noisy sequence correlated with such a linear sequence, the problem we study is how to obtain the input of the linear system. We explain known correlation attacks in this general setting and we show types of autonomous behaviors which should be avoided.**

I. INTRODUCTION

Symmetric key ciphers, also known as secret key ciphers, are an important class of cipher systems. They are characterized by the fact that the same key is used for encryption and decryption. There are two standard approaches for the design of the cipher, a *block cipher* or a *stream cipher*.

In stream ciphers, the block size to be encrypted normally equals the size of one character in the alphabet. If we use the binary alphabet $\{0,1\}$, the encryption is done bit by bit. Contrary to block ciphers, where the encryption is the same for each block, the encryption in stream ciphers is time-varying, i.e. the encryption varies from bit to bit. Further, every bit is encrypted separately and independently from the previous or following bits. Consequently, patterns in the plaintext are not recognizable in the ciphertext and error propagation is very limited. Many of the properties of stream ciphers make them suitable for use in telecommunication and low-level network encryption. They are normally much faster than block ciphers and do not cost more to implement in terms of hardware gates or memory, not hardware memory. If we can, having a secret key k of length l, generate a random looking sequence $r = (r_1, r_2, \ldots, r_N)$ of length N, called *keystream*, we can encrypt messages of length N. Encryption is done by just XORing r to the message m

$$c_i = m_i + r_i, \quad i = 0, 1, \ldots, N$$

and decryption by again XORing r to the ciphertext c

$$m_i = c_i + r_i, \quad i = 0, 1, \ldots, N.$$

In fact, this is a widely used approach for stream ciphers, called *additive stream ciphers*. A famous example of an additive stream cipher is the so-called *one-time pad* (OTP).

This work was supported in part by Swiss National Science Foundation under grant number 200020-126948. The work of S. D. Cardell was supported by a grant for research students from the Generalitat Valenciana with reference BFPI/2008/138 and by Spanish grant MTM2008-06674-C02-01.

G. Maze, J. Rosenthal and U. Wagner are with Institut für Mathematik, Universität Zürich, Winterthurerstrasse 190 , CH-8057, Switzerland Zürich gerard.maze@math.uzh.ch, rosenthal@math.uzh.ch, urs.wagner@math.uzh.ch

S. D. Cardell is with Departament d'Estadística i Investigació Operativa, Universitat d'Alacant, Campus de Sant Vicent del Raspeig, Ap. Correos 99, 03080, Alicante, Spain s.diaz@ua.es

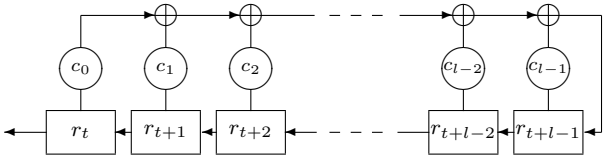

Fig. 1. An LFSR of length l

The rest of the paper is structured in the following way. In Section II we explain the concept of LFSR and its construction. In Section III we treat briefly the idea of correlation attack and fast correlation attack. In Section IV we show the construction of an autonomous system and we present a correlation problem which stands at the center of the theory. In Section V we present some conclusions.

II. LINEAR FEEDBACK SHIFT REGISTER

The task of generating a random looking keystream $r = (r_i)_{i\geq 0}$ from a short key has to be solved. Algorithms which generate keystreams from a secret key are called *keystream generators*. One way to do this is through some nonlinear recurrence relation

$$r_{t+l} = f(r_{t+l-1}, r_{t+l-2}, \ldots, r_t), \quad t = 0, 1, \ldots$$

having the initial state as the key. This recurrence sequence will act as a pseudo-random number generator. If the recurrence relation is linear, then it is possible to implement the recurrence relation with a *linear feedback shift register* (LFSR).

An LFSR of length l is a device made up by l registers, called *taps*. Each tap is able to hold one symbol at each round. These symbols are elements from a field $\mathbb{F}_q$, over which we have chosen to define the LFSR. In stream cipher applications we often have $q = 2$, the binary field, or some extension of the binary field $q = 2^r$, where r is the symbol size of the stream cipher. Thus, we will work in the finite field $\mathbb{F}_2$.

Each tap has a feedback coefficient $c_0, c_1, \ldots, c_{l-1} \in \mathbb{F}_2$ which decides whether the bit in the according tap is used or not in the calculation of the feedback sum. We call the taps with feedback coefficient equal to one *feedback taps*. In each round, the elements in the registers are shifted to the left. The element in the leftmost tap is the output of the LFSR in that round. The rightmost tap receives the mod 2 sum of the bits in the feedback taps of the previous round, see Fig.1.

ISBN 978-963-311-370-7

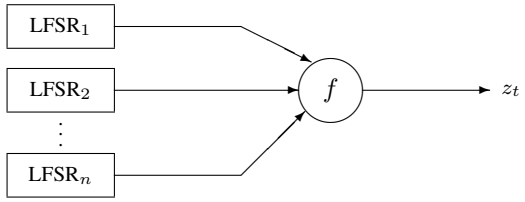

Fig. 2. The output of n LFSRs are combinated via the nonlinear Boolean function

a_t	b_t	c_t	z_t
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

TABLE I
EXAMPLE OF GEFFE GENERATOR

We assume $c_0 = 1$, otherwise one bit depends only on the $l-1$ preceding bits. Hence the LFSR would have length $l-1$. As we do not need more registers than necessary to make the feedback connection work, we also assume $c_l = 1$.

Let $r = (r_i)_{i\geq 1}$ be an output sequence of the LFSR, then r fulfills the linear recurrence relation

$$c_0 r_t + c_1 r_{t+1} + \cdots + c_{l-1} r_{t+l-1} + c_l r_{t+l} = 0, \quad t \geq 0.$$

The polynomial

$$f(x) = c_0 + c_1 x + \cdots + c_{l-1} x^{l-1} + c_l x^l$$

in $\mathbb{F}_2[x]$ is called the *feedback polynomial* of the LFSR. This polynomial gives a compact description of the LFSR. The sequence r is completely determinated by its l initial bits $R_0 = (r_0, r_1, \ldots, r_{l-1})$, called *initial state*. There exist 2^l different initial states, so the LFSR can generate 2^l different sequences. Note that one of the sequences is the zero sequence. We want to hide long messages, so we need the sequences to be as long as possible. The period τ of an LFSR with a primitive polynomial $f(x) \in \mathbb{F}_2[x]$ of degree l as the feedback polynomial and with a non-zero initial state is $\tau = 2^l - 1$, the maximum possible period [1].

Definition 1: Given a sequence $s = (s_t)_{t=0}^N$, we define the *linear complexity* of s, denoted by $\mathcal{L}(s)$, as the the length of the shortest LFSR that generates the sequence.

The linear complexity can be determinated with the Berlekamp-Massey algorithm [2], which efficiently computes the feedback polynomial of the LFSR given at least $2\mathcal{L}(s)$ of the output bits. Because of this, an LFSR is not a good keystream generator. The sequences generated have good statistical properties, desirable for keystream generator construction, but we need to destroy the linearity, i.e. increase the linear complexity, before the sequence can be used. One classical approach is to use several binary LFSRs and combine the output from each of them using a non-linear Boolean function as pictured in Fig. 2.

Example 1: The Geffe generator is a well known nonlinear stream cipher. It consists of three LFSRs with different periods. In each step, the output of the generator equals the output of the first or the second LFSR, depending on the output of the third LFSR. Let a_t, b_t and c_t be the outputs of the different LFSRs at time t. The output z_t of the generator is defined by the following rule

$$z_t = \begin{cases} a_t, & \text{if} \quad c_t = 0, \\ b_t, & \text{if} \quad c_t = 1. \end{cases}$$

In this case the nonlinear Boolean function used to destroy the linearity is $f : \mathbb{F}_2^3 \to \mathbb{F}_2$, $f(x_1, x_2, x_3) = x_1(1 + x_3) + x_2 x_3$.

III. CORRELATION ATTACKS

A. Idea

Correlation attacks are a class of plaintext attacks for breaking stream ciphers whose keystream is generated by combining the output of several LFSRs using a Boolean function. Correlation attacks exploit a statistical weakness that arises from a poor choice of the Boolean function.

As an illustration consider Example 1. There are eight possible values for the outputs of the three registers and the value of the output of the combining function for each of them is shown in table I. Let us consider the output of the first register. One notices that the sequence z agrees in six of eight bits with the sequence a, i.e., $f(a_t, b_t, c_t) = a_t$ in 75% of the cases. This generator is a good candidate to be attacked by a correlation attack.

Suppose we intercept the ciphertext c and we know n bits of the plaintext m. It is possible to obtain n bits of the keystream just XORing the known bits of m with the corresponding bits of c. Now we may begin a brute force search of the space of possible keys (initial values) for the first LFSR. For any initial state, we compute the n corresponding bits of a and compare these to the n recovered bits of the keystream. We have established that there is a correlation of 75% between the output of the first LFSR a and the keystream. Then we know that if we have correctly guessed the key, approximately $\frac{3n}{4}$ bits will match. If we have guessed incorrectly, we expect $\frac{n}{2}$ to match. Thus we may recover the key for the first LFSR independently of the keys of the other LFSRs.

At this point we have reduced the problem of brute forcing a system of three LFSRs to the problem of brute forcing a single LFSR and then a system of two LFSRs. Observe in the table that b also agrees with the keystream in 75% of the cases, so we can do the same with sequence b.

Geffe generator is a very efficient keystream generator, but because of the correlation attacks it is not used any more.

This sort of correlation attacks was proposed by Siegenthaler [3] [4]. They suppose a significant reduction in complexity compared to the brute force approach. Suppose we have n LFSRs of length $l_1, l_2, \ldots, l_n$. In order to test every

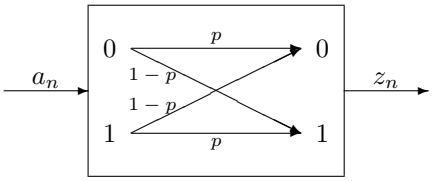

Fig. 3. BSC with crossover probability $1-p$

possible combination of initial states,

$$\prod_{i=1}^{n}(2^{l_i}-1)$$

different combinations have to be tested. Suppose that we have only one LFSR, the first one, concerned with the correlation attack, then the complexity is reduced by a factor 2^{l_1}

$$2^{l_1}+\prod_{i=2}^{n}(2^{l_i}-1).$$

B. Fast correlation attacks

In [5], [6] Meier and Staffelbach proposed the first fast correlation attack. Let the output sequence $z=(z_i)_{i=0}^{N}$ of a running key generator be correlated to an LFSR-sequence $a=(a_i)_{i=0}^{N}$ of length l with correlation probability p higher than 0.5. They presented two algorithms, A and B, to determine the initial digits of a. These algorithms were only efficient when the number t of feedback taps was small, for example $t\leq 10$ when $0.5\leq p\leq 0.75$. The idea was to view the sequence z as a perturbation of the sequence a by a binary asymmetrix memoryless noise source, with $\Pr(z_n=a_n)=p$. Every digit a_n satisfies several linear relations derived from the basic feedback relation, all of them involving t other digits of a. By substituting the corresponding digits of z in these relations, they obtained equations for each digit z_n which may hold or not. To test if $a_n=z_n$ they counted the number of all equations which hold for z_n. The more of these equations hold, the higher is the probability for z_n to agree with a_n.

From the coding theory point of view they used a linear block code model, see [7]. Let us consider the sequence a to be a codeword of a linear $[N,l]$-block code. Then z is the received word, after the transmission over a BSC with crossover probability $1-p$, see Fig. 3. If one correctly decode z then can know the initial state of the LFSR. In [5], [6] the algorithm A can be seen as an one-step decoding algorithm and algorithm B is an iterative algorithm very similar to the belief propagation algorithm known from LDPC codes [8], [9].

Other fast correlations attacks and improvements have been proposed after Meier and Staffelbach, see [10], [11], [12], [13].

IV. AUTONOMOUS SYSTEMS

A. Construction

An alternative way to describe an LFSR is by means of its state transition matrix. Let again

$$f(x)=c_0+c_1x+c_2x^2+\cdots+c_{l-1}x^{l-1}+c_lx^l$$

be the feedback polynomial of an LFSR. The according state transition matrix M is defined by

$$M=\begin{pmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&&\vdots\\0&0&0&\cdots&1\\c_0&c_1&c_2&\cdots&c_{l-1}\end{pmatrix}.$$

Let R_0 be the initial state of the LFSR. The consecutive states R_i, $i\geq 0$, can be computed by

$$R_i=G^iR_0.$$

Thus an LFSR can be studied as a linear system. Recover the initial state comes down to solving the linear system.

We consider the generalization of the LFSRs in autonomous systems.

Let $\sigma:(\mathbb{F}_2^n)^{\mathbb{Z}}\to(\mathbb{F}_2^n)^{\mathbb{Z}}$, $y_t\to y_{t+1}$ be the shift operator. Given an $n\times n$ polynomial matrix $Q\in\mathbb{F}_2[z]^{n\times n}$, Q defines an autonomous behavior in the sense of Willems [14]:

$$\mathcal{B}=\ker Q(\sigma)\subset(\mathbb{F}^n)^{\mathbb{Z}}$$

Let $U(z)$ be an $n\times n$ unimodular matrix and

$$P(z)=U(z)Q(z).$$

Then $P(\sigma)=Q(\sigma)$, i.e., the behaviors are the same. There exists an $n\times n$ unimodular matrix such that $P(z)$ has row degrees $\delta_1\geq\delta_2\geq\ldots\geq\delta_n$. Let $r=\sum_{i=1}^{n}\delta_i$ and consider the $n\times r$ basis matrix

$$X(z)=\begin{pmatrix}1,z,z^2,\ldots,z^{\delta_1-1}&&\\&\ddots&\\&&1,z,z^2,\ldots,z^{\delta_n-1}\end{pmatrix}.$$

Theorem 1: Given a polynomial matrix $P(z)$ and a basis matrix $X(z)$ as above, then there exist an $r\times r$ matrix A and an $n\times r$ matrix C such that

$$\ker(X(z)|P(z))=\operatorname{im}\begin{pmatrix}zI_r-A\\C\end{pmatrix}.$$

The autonomous behavior can be described through the system:

$$x_{t+1}=Ax_t,\quad y_t=Cx_t$$

Moreover, the matrices A and C can be computed 'by Inspection' [15].

Also, if $P(z) = \sum_{i=0}^{\delta} P_i z^i$, with $P_\delta = I_n$ and $\delta_1 = \delta_2 = \cdots = \delta_n = \delta$, then

$$\ker(X(z)|P(z)) = \text{im} \left(\begin{array}{cccc|c} zI & 0 & \cdots & 0 & P_0 \\ I & zI & \cdots & 0 & P_1 \\ 0 & I & \cdots & 0 & P_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & zI & P_{\delta-2} \\ 0 & 0 & \cdots & I & P_{\delta-1} + zI \\ \hline 0 & 0 & \cdots & 0 & I \end{array} \right).$$

Example 2: Let P be

$$P(z) = \begin{pmatrix} z^2+1 & z \\ 1 & z^2+1 \end{pmatrix}$$

and

$$X(z) = \begin{pmatrix} 1 & 0 & z & 0 \\ 0 & 1 & 0 & z \end{pmatrix}.$$

Observe that we can permute the columns of $X(z)$ [15]. Then

$$\ker(X(z)|P(z)) = \text{im} \left(\begin{array}{cc|cc} z & 0 & 1 & 0 \\ 0 & z & 1 & 1 \\ 1 & 0 & z & 1 \\ 0 & 1 & 0 & z \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

and

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Notice that in this case the output sequence is a vector of vectors. Let $R_0 = (0001)^T$ be the initial state, then the output sequence is

$$y = \begin{array}{l} ((01),(10),(01),(01),(11),(11),(00),(10), \\ (00),(11),(10),(10),(11),(01),(00),...)^T \end{array}$$

Let $p(z) = \det P(z)$, by [16] we can deduce also that $p(z) = \det(zI - A)$, i.e., it is the characteristic polynomial of the matrix A. This polynomial behaves like the feedback polynomial in the LFSR case. We want to hide very large messages, so we need the period of A to be as large as possible.

Theorem 2: Let $P(z) = \sum_{i=0}^{\delta} P_i z^i$, $P_\delta = I_n$, be an $n \times n$ polynomial matrix. If $p(z) = \det P(z)$ is a primitive polynomial then the matrix A computed as in Theorem 1 has maximal period $2^{\delta n} - 1$.

B. Correlation problem

Let $P(z)$ be a polynomial matrix described as in subsection IV-A and let $y = (y_i)_{i=0}^N$ be a sequence in some behavior $\ker P(z)$. This sequence can be also computed as

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix} = \Omega x_0$$

where

$$\Omega = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^N \end{pmatrix}$$

and A, C are computed as in theorem 1, and x_0 is the initial state.

Assume one receives the noisy sequence $\tilde{y} = (\tilde{y}_i)_{i=0}^N$, instead of the sequence y. The problem now is to recover the initial state x_0 (the key).

Let ϵ be the correlation between y and $\tilde{y}$. If $\epsilon = \frac{1}{2} + \beta > 0.5$ it can be shown [12] that the observation of

$$N \approx \frac{1}{\beta^2}$$

bits of the sequence $\tilde{y}$ allows the recovery of the initial state. In this case if the the parity check matrix of the code generated by Ω is sparse we can obtain the initial state just considering the problem as an LDPC decoding problem.

In other cases, the problem is different. One can try the brute force

$$\min_{x_0 \in \mathbb{F}_2^r} \omega(\Omega x_0 - \tilde{y})$$

where $\omega(x)$ is the Hamming weight of x. Since the columns of Ω are a basis of the space of the possible sequences it is not necessary to compute the product Ωx_0 for all $x_0 \in \mathbb{F}_2^r$. The problem can be reduced to

$$\min_{x_0 \in \mathbb{F}_2^r} \omega \left(\sum_{i \in \text{ supp}(x_0)} \Omega_i - \tilde{y} \right) \tag{1}$$

where $\text{supp}(x)$ are the components of x which are not zero and Ω_i is the i-th column of Ω.

If the sequence is highly noisy the situation is different. It is possible to find an initial state x_k which minimizes (1), which is not the desired state.

If we knew, for example, that the correlation $\epsilon < 0.5$ is small, then we can add one to all the bits of the sequence and the resultant sequence

$$\hat{y} = \tilde{y} + \mathbf{1}$$

is supposed to be correlated with y with correlation $1 - \epsilon > 0.5$. Then we can try to solve (1) with the new sequence $\hat{y}$.

V. CONCLUSIONS

Assume we have a sequence in some behavior, but we receive a noisy sequence, maybe because this sequence has been modified by a Boolean function. We want to recover the initial state of the system which has generated the sequence. Depending on how high is the correlation between the two sequences we can regard the problem as a decoding problem or not. If the matrix Ω of the autonomous systems generates a code whose parity check matrix is not sparse the only way to solve the problem is using brute force. However the problem is not as hard as it could be, because the columns of the matrix are a basis of the space of the possible sequences. Then we have to compute all the linear combinations of

columns of Ω, and check which has the corresponding correlation with the given sequence.

REFERENCES

[1] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. New York, NY: Cambridge University Press, 1997.

[2] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, pp. 122–127, 1969.

[3] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Transactions on Information Theory*, vol. 30, no. 5, pp. 776–780, 1984.

[4] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Transactions on Computers*, vol. 34, no. 1, pp. 81–85, 1985.

[5] W. Meier and O. Staffelbach, "Fast correlation attacks on stream ciphers," in *Advances in Cryptology – EUROCRYPT'88*, ser. Lecture Notes in Computer Science, C. G. Günter, Ed. Berlin: Springer-Verlag, 1988, vol. 330, pp. 301–314.

[6] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.

[7] U. Wagner, "Detection and exploitation of small correlations in stream ciphers," Diploma thesis, Institut für Mathematics, Universität Zürich, Zürich, Switzerland, 2008.

[8] R. G. Gallager, *Low-Density Parity-Check Codes*, ser. Research Monograph Series. Cambridge, MA: MIT Press, 1963, no. 21.

[9] W. E. Ryan, "An introduction to LDPC codes," in *CRC Handbook for Coding and Signal Processing for Recoding Systems*, B. Vasic, Ed. CRC Press, 2004.

[10] V. Chepyzhov and B. Smeets, "On a fast correlation attack on certain stream ciphers," in *Advances in Cryptology – EUROCRYPT '91*, ser. Lecture Notes in Computer Science, D. W. Davies, Ed. Berlin: Springer-Verlag, 1991, vol. 547, pp. 68–70.

[11] T. Johansson and F. Jönsson, "Improved fast correlation attacks on stream ciphers via convolutional codes," in *Advances in Cryptology – EUROCRYPT'99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Berlin: Springer-Verlag, 1999, vol. 1592, pp. 347–362.

[12] P. Ekdhal, "On LFSR based stream ciphers. Analysis and design," Ph.D. dissertation, Department of Information Technology, Lund University, Lund, Sweden, Oct. 2003.

[13] H. Molland, J. E. Mathiassen, and T. Helleseth, "Improved fast correlation attack using low rate codes," in *Cryptography and Coding 2003*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed. New York, NY: Springer-Verlag, 2003, vol. 2898, pp. 67–81.

[14] J. C. Willems, "Models for dynamics," in *Dynamics Reported*, U. Kirchgraber and H. O. Walther, Eds. John Willey & Sons Ltd. and B.G. Teubner, 1989, vol. 2, pp. 171–269.

[15] J. Rosenthal and J. Schumacher, "Realization by inspection," *IEEE Transactions on Automatic Control*, vol. 42, no. 9, pp. 1257–1263, 1997.

[16] M. Fiedler, "A note on companion matrices," *Linear Algebra and its Applications*, vol. 372, pp. 325–331, 2003.