CRYPTANALYSTS REPRESENTATION OF NONLINEARLY FILTERED ML-SEQUENCES

T.Siegenthaler
Institute for Communication Technology
Federal Institute of Technology
8092 Zurich, Switzerland

Abstract

A running key generator consisting of a maximum-length (ML) linear feedback shift register (LFSR) and some nonlinear feedforward state filter function is investigated. It is shown how a cryptanalyst can find an equivalent system in a ciphertext-only attack. The analysis uses a Walsh orthogonal expansion of the state filter function and its relation to the crosscorrelation function (CCF) between the ML-sequence and the produced running key sequence.

I Introduction

Nonlinearly filtered ML-sequences are frequently used as the running key sequence in stream ciphers. It will be shown how, under some assumptions, a cryptanalyst can find a system which generates the same running key sequence. The following system will be investigated:

Fig. 1. A nonlinearly state filtered ML-sequence

In Fig. 1 it is assumed that the stages i_j with $1 \leq i_1 \leq i_2 \leq \dots \leq i_n \leq L$ of the LFSR of length L are tapped as inputs to the nonlinear function f . The described method, however, applies to any choice $1 \leq i_1 \leq i_2 \leq \dots \leq i_n \leq 2^L - 1$ and is useful if $n < L$. The binary noise source (BNS) models the generation of the plaintext to be enciphered. We shall require that $\Pr(N_k=1) \neq 1/2$ and the analysis will make use of the resulting redundancy in the plaintext. It is assumed that the task for the

cryptanalyst is to find an equivalent system which generates the full period of the sequence (z_k) , given only the primitive connection polynomial of the LFSR and some portion of (z_k) or of (y_k) . The number n , and thus the positions $i_1, i_2, \dots, i_n$ of the taps, as well as the function f and the initial state of the LFSR are not known to the cryptanalyst. The described system could of course be simulated by a binary counter, counting through 2^L-1 states and an appropriate function of L variables. In a worst case situation, the cryptanalyst could do nothing better than this. A necessary but not sufficient condition for this worst case is $n=L$. Alternatively the above system could be simulated by the shortest LFSR generating the sequence (z_k) , but this seems not be possible if only a portion of (y_k) instead of (z_k) is known. However, if the cryptanalyst knows the connection polynomial used in the original system given in Fig. 1 (if not he could try all possible primitive polynomials of degree L), he can try to determine an equivalent system as given in Fig. 2.

Fig. 2. Cryptanalyst's equivalent system with $1 \leq m \leq n$.

The m LFSR's in Fig. 2 all have the same known primitive connection polynomial but different initial states. It can be easily seen that such an equivalent system always exists for an m with $1 \leq m \leq n$. If f in the original system is not the constant function $z_k = 0$ (or 1) for all inputs, then obviously $m \geq 1$. But for every f the above equivalent has at most $m=n$ because one could always choose $m=n$ and $d_1=i_1, d_2=i_2, \dots, d_n=i_n$ and $g=f$. However, in certain cases an equivalent system with $m < n$ exists. The cryptanalyst generally finds the equivalent in two steps. He determines first m initial states $s^1, s^2, \dots, s^m$ for the LFSR's which then generate the sequences $(s_{k-d_1}), (s_{k-d_2}), \dots, (s_{k-d_m}), 1 \leq m \leq n$, and afterwards he determines the function g from the known inputs $s_{k-d_i}, i=1, 2, \dots, m$, and the observed y_k . The sequences generated by the given primitive polynomial form a vectorspace V^L of dimension L over $GF(2)$. The sequences $(s_{k-i_1}), (s_{k-i_2}), \dots, (s_{k-i_n})$ and its linear combinations form a subspace V^n of V^L . It will be shown that the sequences $(s_{k-d_1}), (s_{k-d_2}), \dots, (s_{k-d_m})$ are elements of V^n and can be found by inspection of the CCF between (s_k) and (y_k) .

II Walsh orthogonal expansion of the state filter function

To show that the sequences used in the equivalent system are in fact elements of the subspace V^n mentioned in chapter I and can be found by inspection of a crosscorrelation function, a Walsh orthogonal expansion of the state filter function will be applied. In this section, the Walsh-Transform is reviewed. Let f be any real-valued function of binary $(0,1)$ variables $x_1, x_2, \dots, x_n$. The domain of f is then the set of 2^n binary vectors $\underline{x} = [x_1, x_2, \dots, x_n]$ in the vectorspace Z^n . Since f is completely specified by its value at each of these 2^n points in Z^n , we may consider f as a member of a 2^n -dimensional vectorspace Q . For any fixed $\underline{w} = [w_1, w_2, \dots, w_n]$ in Z^n , we can write the corresponding Walsh function $W_{\underline{w}}(\underline{x}) \in Q$ as

$$W_{\underline{w}}(\underline{x}) = (-1)^{\underline{x} \cdot \underline{w}} = (-1)^{x_1 w_1 \oplus x_2 w_2 \oplus \dots \oplus x_n w_n} \quad (1)$$

There are exactly 2^n different Walsh-Functions $W_{\underline{w}}(\underline{x})$ corresponding to the 2^n vectors $\underline{w}$ in Z^n . These functions form a complete orthogonal basis for Q [2].

We now write $f(\underline{x})$ to denote $f(x_1, x_2, \dots, x_n)$. The function f can be expanded in a Walsh function expansion as

$$f(\underline{x}) = \sum_{\underline{w}} F(\underline{w}) W_{\underline{w}}(\underline{x}) \quad (2)$$

where

$$F(\underline{w}) = 2^{-n} \sum_{\underline{x}} f(\underline{x}) W_{\underline{w}}(\underline{x}) \quad (3)$$

is the Walsh-Spectrum [2] of f . The following properties will be used later on.

Property 1: (initial value)

$$\sum_{\underline{w}} F(\underline{w}) = f(\underline{0}) \quad (4)$$

Proof: $F(\underline{w}) = 2^{-n} \sum_{\underline{x}} f(\underline{x}) W_{\underline{w}}(\underline{x})$. Note, however, that $W_{\underline{w}}(\underline{0}) = 1$ for all $\underline{w}$.

Hence, summing on $\underline{w}$, $\sum_{\underline{w}} F(\underline{w}) = 2^{-n} \sum_{\underline{x}} f(\underline{x}) \sum_{\underline{w}} W_{\underline{w}}(\underline{x}) W_{\underline{w}}(\underline{0}) = f(\underline{0})$ from the orthogonality of Walsh-Functions.

Property 2: (idle variables)

If the variable x_j in $f(x_1, x_2, \dots, x_n)$ is idle, then

$$F(\underline{w}) = 0 \quad \text{for } \underline{w} \in (\underline{w}: w_j = 1). \quad (5)$$

Proof (Titsworth [1]):

$$\begin{aligned} F(\underline{w}) &= 2^{-n} \sum_{\underline{x}} f(\underline{x}) (-1)^{\underline{x} \cdot \underline{w}} \\ &= 2^{-n} \sum_{\underline{x}: x_j=0} f(\underline{x}) \prod_{k \neq j} (-1)^{x_k \cdot w_k} \left[\sum_{x_j=0}^1 (-1)^{x_j} \right] \end{aligned}$$

but the last bracket vanishes for every choice of x_k , all $k \neq j$.

Property 3: (degenerate functions)

It has been shown by Lechner [3] that if the smallest subspace of Z^n which contains the set $S = \{w: F(w) \neq 0\}$ has dimension $m \leq n$ then it is guaranteed that a function $g(y)$ with $g(y) = g(xH^t) = f(x)$ exists, where H^t is the transpose of

$$H = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

and where any m linear independent elements of S can be chosen as the row vectors w_i , $i=1,2,\dots,m$. For $m < n$, $f(x)$ is called a degenerate function and $y_1, y_2, \dots, y_m$ are called essential variables.

III Crosscorrelation function

In this section, the CCF $C(d)$ of the sequences (s_k) and (z_k) will be computed. By definition,

$$C(d) = (1/p) \sum_{k=0}^{p-1} (-1)^{z_k} (-1)^{s_{k-d}}, \quad \text{for } d=0,1,\dots,p-1, \quad (6)$$

where $p=2^{L-1}$ denotes the period of the ML-sequence (s_k) . First, we consider the general case, where $n=L$ and $i_1=1, i_2=2, \dots, i_L=L$ (see Fig.1). The L sequences $(s_{k-1}), (s_{k-2}), \dots, (s_{k-L})$ are linearly independent and form a basis for the vector space V^L . This follows from the assumption that a primitive and hence irreducible connection polynomial is used. The sequence (s_{k-d}) can be expressed as a linear combination of the above basis for every $d \in (0,1,\dots,p-1)$. Therefore, we can express s_{k-d} using the binary vectors $\underline{v}(d) = [v_1(d), v_2(d), \dots, v_L(d)]$ and $\underline{x}_k = [s_{k-1}, s_{k-2}, \dots, s_{k-L}]$ as $s_{k-d} = \underline{v}(d) \underline{x}_k$:

$$s_{k-d} = v_1(d)s_{k-1} \oplus v_2(d)s_{k-2} \oplus \dots \oplus v_L(d)s_{k-L}. \quad (7)$$

To simplify the notation, we will sometimes write

$$f^*(\underline{x}) = (-1)^{f(\underline{x})}$$

to denote the $(1,-1)$ function corresponding to $f(\underline{x})$. $F^*(\underline{w})$ denotes the Walsh-Transform of $f^*(\underline{x})$. From (2), (6) and (7) we get

$$\begin{aligned} C(d) &= (1/p) \sum_{k=0}^{p-1} \sum_{\underline{w}} F^*(\underline{w}) (-1)^{\underline{w} \cdot \underline{x}_k} (-1)^{\underline{v}(d) \cdot \underline{x}_k} = \\ &= (1/p) \sum_{\underline{w}} F^*(\underline{w}) \sum_{k=0}^{p-1} (-1)^{\underline{x}_k \cdot (\underline{w} \oplus \underline{v}(d))}. \end{aligned} \quad (8)$$

The exponent $\underline{x}_k \cdot (\underline{w} \oplus \underline{v}(d))$ for $k = 0,1,\dots,p-1$ corresponds, for some fixed binary vector $\underline{w} \oplus \underline{v}(d) \neq 0$, to a linear combination of ML-sequences and

therefore is itself a ML-sequence. Hence, the last sum in (8) is equal to -1 unless $\underline{w} \oplus \underline{v}(d) = 0$ and we obtain

$$C(d) = (1/p) [p F^*(\underline{v}(d)) - \sum_{\underline{w}: \underline{w} \neq \underline{v}(d)} F^*(\underline{w})] . \quad (9)$$

Using property 1 from sec. II, we get

$$C(d) = (1 + 1/p) F^*(\underline{v}(d)) - (1/p) f^*(0) , \quad (10)$$

where F^* and f^* are functions of L variables. Expression (10) gives, for an arbitrary state filter function f , the value of the CCF between the ML-sequence (s_k) and the running key sequence (z_k) . In the sequel, the case $n < L$ is of interest. We consider $C(d)$ as given in (6) at some position d . Assume that some variable s_{k-i} , for a fixed i in (7), belongs to the $L-n$ idle variables. Then we have from property 2 of sec. II:

$$F^*(\underline{v}(d)) = 0 , \quad \text{for } \underline{v}(d) \in (\underline{v}((d): v_i(d) = 1) . \quad (11)$$

Therefore, we have from (10) for all d such that $\underline{v}(d)$ contains at least one component $v_i(d) = 1$, for which the corresponding variable x_i is idle,

$$C(d) = -(1/p) f^*(0) = \text{constant} . \quad (12)$$

Hence, the only d 's where $C(d)$ may be different from the constant given in (12) are those where all components $v_i(d)$, $i \in (1, 2, \dots, L)$, in $\underline{v}(d)$ are zero for all $i \notin (i_1, i_2, \dots, i_n)$. Note that the only non-idle variables are s_{k-i_1} , s_{k-i_2} , ..., s_{k-i_n} . Therefore, we have finally our desired result:

$$C(d) = \begin{cases} -(1/p) f^*(0) & , d \notin M \\ (1 + 1/p) F^*(\underline{v}(d)) - (1/p) f^*(0) & , d \in M \text{ and } \underline{v}(d) \neq 0, \end{cases} \quad (13)$$

where M denotes the set of all d such that the sequence (s_{k-d}) can be obtained as a linear combination of the sequences (s_{k-i_1}) , (s_{k-i_2}) , ... (s_{k-i_n}) . Note that f^* and F^* in (13) are functions of the n non-idle variables only and that $\underline{v}(d)$ can be considered as the corresponding vector of n components only.

The following facts follow from (13):

- 1) A peak in the CCF (value different from $-(1/p)f^*(0)$) occurs at d if and only if d is such that (s_{k-d}) can be obtained as some linear combination of tap-sequences (s_{k-i_1}) , (s_{k-i_2}) , ..., (s_{k-i_n}) and $F^*(\underline{v}(d)) \neq 0$.
- 2) The magnitude of a peak depends only on F^* (or equivalently on f^*), not on the tap positions or the chosen primitive connection polynomial.
- 3) The positions of peaks in the CCF depend only on the chosen connection polynomial and the tap positions $i_1, i_2, \dots, i_n$.
- 4) The number of peaks in the CCF is the number of $\underline{w} \neq 0$ such that $F^*(\underline{w}) \neq 0$, and thus is at most $2^n - 1$.

Example 1: ($L=7, n=3$)

Consider the state filter function $f(\underline{x}) = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$, where addition and multiplication is in $GF(2)$.

w_1 x_1	w_2 x_2	w_3 x_3	$f(\underline{x})$	$F(\underline{w})$	$f^*(\underline{x})$	$F^*(\underline{w})$
0	0	0	0	1/2	1	0
0	0	1	0	-1/4	1	1/2
0	1	0	0	-1/4	1	1/2
0	1	1	1	0	-1	0
1	0	0	0	-1/4	1	1/2
1	0	1	1	0	-1	0
1	1	0	1	0	-1	0
1	1	1	1	1/4	-1	-1/2

$$C(d_1)=C(d_2)=C(d_3)= (1 + 1/127)(1/2) - 1/127 = 63/127 \approx 1/2$$

$$C(d_4)=(1 + 1/127)(-1/2) - 1/127 = -65/127 \approx -1/2$$

$$C(d_i)=-1/127, \text{ for } i=5,6,\dots,127 \approx 0$$

$$\underline{v}(d_1) = [001] ; \underline{v}(d_2) = [010] ; \underline{v}(d_3) = [100] ; \underline{v}(d_4) = [111]$$

$$\text{and } (s_{k-d_j}) = (s_{k-i_j}) \text{ for } j=1,2,3 \text{ and } (s_{k-d_4}) = (s_{k-i_1} \oplus s_{k-i_2} \oplus s_{k-i_3}).$$

IV Cryptanalyst's equivalent

From the inverse Walsh-Transform (3), it follows that the function $f(\underline{x})$ is uniquely determined by its nonzero Walsh coefficients $F(\underline{w})$. These coefficients can be recognized as peaks in the CCF at positions $d_1, d_2, \dots$ as pointed out in chapter III. Up to now we have assumed that the CCF is computed as given in expression (6), where the sum is over the full period p . Moreover, the influence of the plaintext has not been considered. In practice, the CCF will be corrupted by the plaintext source as shown in Fig. 1; moreover, the cryptanalyst may usually use only a small portion of ciphertext instead of a full period. In this case, the values of the peaks are only estimates of the nonzero coefficients in $F(\underline{w})$. Therefore, the cryptanalyst proceeds in two steps. For the sequences $(s_{k-d_1}), (s_{k-d_2}), \dots \in V^n$ corresponding to the positions $d_1, d_2, \dots$ of the observed peaks, an arbitrary basis $(s_{k-d_1}), (s_{k-d_2}), \dots, (s_{k-d_m})$ of dimension $m \leq n$ is chosen, where $s_{k-d_1}, s_{k-d_2}, \dots, s_{k-d_m}$ correspond to the essential variables mentioned in sec. II, property 3. The above sequences are generated by m LFSR's with appropriate initial loadings $s^1, s^2, \dots, s^m$. It then remains to determine the function $g: (s_{k-d_1}, s_{k-d_2}, \dots, s_{k-d_m}) \rightarrow z_k$. This is an easy task because the inputs $s_{k-d_1}, s_{k-d_2}, \dots, s_{k-d_m}$ are known and the corresponding noisy output y_k can also be observed for $k = 1, 2, \dots$ as ciphertext from the original system. If the same input vector $(s_{k-d_1}, s_{k-d_2}, \dots, s_{k-d_m})$ occurred a sufficient number of times, the corresponding mapping for this input-output pair could be estimated by a simple majority decision.

Example 2: The following system is cryptanalyzed: $L=13$

Fig. 3. The original system

The cryptanalyst knows only the connection polynomial and observes only 800 digits of ciphertext. However, from the shift-register states corresponding to the four observed peaks in the CCF (see Fig. 4), three linearly independent initial states s^1, s^2, s^3 can be determined, corresponding to the positions of (any) three of these peaks.

Fig.4. CCF for the system of Fig. 3 with protocol of 10 strongest values together with the corresponding states of cryptanalyst's LFSR.

It turns out that, for this example, the equivalent function g determined by the procedure described above is $z = y_1 \oplus y_2 \oplus y_1 y_3 \oplus y_1 y_2 \oplus y_2 y_3$; hence, $m=3$ and the representation is:

Fig. 5. Cryptanalyst's equivalent system

In fact, it can be seen from the Walsh-Transform of $f(\underline{x})$ that the cryptanalyst implicitly used the linear transformation

$$H = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 1010 \\ 1001 \\ 0110 \end{bmatrix} \text{ where } \underline{y} = [y_1 y_2 y_3] = [x_1 x_2 x_3 x_4] H^t,$$

even if he did not do this explicitly.

V Conclusions

It has been shown how a cryptanalyst can construct an equivalent system for a certain type of running key generator. The method uses properties of the CCF between the driving ML-sequence and the produced running key sequence. The Walsh-Transform of the state filter function played a crucial role in the computation and interpretation of this CCF. The advantage of the described method is that an equivalent system can be found in a ciphertext-only attack. To determine the equivalent function g even in a known plaintext attack (where $\Pr(N_k=1) = 0$), at least 2^m ciphertext digits are required. However, generally one needs much more ciphertext to be able to recognize enough CCF peaks and to determine g in a ciphertext-only attack when $\Pr(N_k=1) \neq 0$. The described method is not applicable if the length L of the LFSR is too large. We assume for a best case estimate (for the cryptanalyst) that $f(\underline{x})$ generates the maximum number of $\approx 2^n$ peaks in the CCF and moreover that these peaks are equally distributed over the period of the CCF. To find the equivalent at most n linearly independent sequences corresponding to n peaks must be found. Therefore, about $n2^{L-n}$ points of the CCF must be considered. If we assume that the feasible number of CCF points which can be tested is $2^{k_{\max}}$ we obtain:

$$L_{\max} \approx k_{\max} + n - \log_2(n). \text{ For } k_{\max} = 40 \text{ and } n=15, L_{\max} \approx 50.$$

Acknowledgement

The author is grateful to Prof. Dr. J. Massey and Dr. P. Schöbi of the Institute for Signal and Information Processing, ETH, Zurich, for valuable discussions and suggestions concerning this work and to A. Clivio from the Mathematics Department, ETH, Zurich, for helpful discussions concerning Ref. [3]. Also thanks go to Prof. Dr. P. Leuthold for his interest and the support of this work.

References

- [1] R.C.Titsworth, "Correlation Properties of Cyclic Sequences", Thesis, California Inst. of Technology, Pasadena, California, 1962.
- [2] M.G. Karpovsky, "Finite Orthogonal Series in the Design of Digital Devices", New York and Jerusalem, Wiley and IVP, 1976.
- [3] R.J. Lechner, "Harmonic Analysis of Switching Functions", in "Recent Developments in Switching Theory", edited by A.Mukhopadhyay, Academic Press, New York and London, 1971.