



# LUND UNIVERSITY

## Using Coding Techniques to Analyze Weak Feedback Polynomials

Hell, Martin

*Published in:*  
Proceedings

2010

[Link to publication](#)

*Citation for published version (APA):*

Hell, M. (2010). Using Coding Techniques to Analyze Weak Feedback Polynomials. In *Proceedings* (pp. 2523-2527)

*Total number of authors:*

1

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

# Using Coding Techniques to Analyze Weak Feedback Polynomials

Martin Hell

Department of Electrical and Information Technology  
Lund University, Sweden  
E-mail: martin.hell@eit.lth.se

**Abstract**—We consider a class of weak feedback polynomials for LFSRs in the nonlinear combiner. When feedback taps are located in small groups, a distinguishing attack can sometimes be improved considerably, compared to the common attack that uses low weight multiples. This class of weak polynomials was introduced in 2004 and the main property of the attack is that the noise variables are represented as vectors. We analyze the complexity of the attack using coding theory. We show that the groups of polynomials can be seen as generator polynomials of a convolutional code. Then, the problem of finding the attack complexity is equivalent to finding the minimum row distance of the corresponding generator matrix. A modified version of BEAST is used to search all encoders of memory up to 13. Moreover, we give a tight upper bound on the required size of the vectors in the attack.

## I. INTRODUCTION

One of the most common building blocks in stream ciphers is the Linear Feedback Shift Register (LFSR). Two important design ideas based on LFSRs are the nonlinear combiner and the nonlinear filter generator. As an example, the E<sub>0</sub> stream cipher used in Bluetooth is based on the linear combiner, but adds some extra memory for increased security. In this paper we focus on the nonlinear combiner. It is well known that LFSRs with feedback polynomials of low weight should be avoided in stream ciphers. An attack exploiting low weight feedback polynomials is the fast correlation attack, proposed by Meier and Staffelbach [1]. Such polynomials provide a low weight parity check equation which can be exploited together with a bias in the output function.

Due to the fast correlation attack, low weight feedback polynomials can be considered weak in the context of stream cipher design. Much research has been put into the fast correlation attack and several improvements have been found, resulting in the fact that this attack is one of the most important cryptanalytic attacks on stream ciphers. In 2004, Englund, Hell and Johansson introduced a new class of weak feedback polynomials [2]. These polynomials were not necessarily of low weight, but instead had their feedback taps located in small groups, possibly very far apart. Each group can be represented by a polynomial,  $g_i(x)$ , with the first tap in a group being the constant term in  $g_i(x)$ . In 2008, Hell and Brynielsson [3] analyzed this attack further and showed that the Walsh transform could be used to efficiently find the complexity and the required vector length in the attack. However, the complexity of the algorithm limited the results

to relatively modest vector lengths. An exhaustive search for all combinations of two groups,  $g_0(x)$  and  $g_1(x)$  with degree at most 8 was performed for all vector lengths up to 25. In this paper we consider the same problem, but from a coding theory point of view. We show that the polynomials  $g_i(x)$  can be seen as generator polynomials for a convolutional code. Doing this, the problem of finding the best attack complexity is equivalent to the problem of finding the minimum row distance of the corresponding generator matrix. If the generator matrix is noncatastrophic, this in turn is equivalent to finding the free distance of the code. There are several well known algorithms dedicated to the problem of finding free distances and the currently most efficient is the Bidirectional Efficient Algorithm for Searching code Trees (BEAST) [4]. It is designed to find the weight spectrum of a code. We slightly modify BEAST to work also with catastrophic generator matrices and use it to exhaustively search all combinations of two groups,  $g_0(x)$  and  $g_1(x)$  with degree at most 13. With our approach there is practically no limitation on the size of the vectors, as there was in previous work. Moreover, we theoretically derive the largest possible vector length required in an attack for a given degree of  $g_i(x)$ , and also show the exact number of combinations requiring this vector length.

The paper is organized as follows. In Section II we give some preliminaries. In Section III we look at the previous work that is relevant to our analysis. The relation to coding theory is given in Section IV and in Section V we present the row distance search using our modified variant of BEAST. The theoretical results are given in Section VI and our results are concluded in Section VII.

## II. PRELIMINARIES

Consider the fast correlation attack [1]. The ideas behind this attack, originally given as a key recovery attack on a nonlinear combiner, can easily be turned into a distinguishing attack on the same nonlinear combiner. The nonlinear combiner uses a set of  $T$  LFSRs, preferably with primitive feedback polynomials, and a nonlinear Boolean output function. We denote the  $i$ th LFSR by  $R_i$  and its size by  $L_i$ . The output of  $R_i$  at time  $t$  is denoted  $x_i(t)$ .

Let  $z(t)$  be the keystream bit at time  $t$ . The correlation attack [5] relies on the fact that there is always a subset of the





Assuming that we start and end in the all-zero state, a possible interpretation for noncatastrophic encoding matrices in coding theory could be that  $\hat{j} + 1$  is the smallest length of an information sequence such that there are two codewords with Hamming distance  $d_{\text{free}}$ .

The vector length  $N = j + 1$  when  $j$  is chosen to fulfill (25) will be denoted  $N_{\text{opt}}$  reflecting the fact that it is the optimal choice for the vector length in the attack.

## V. COMPUTING THE ROW DISTANCE

In this section we discuss algorithms that can be used to find  $d_j^r$  in general, and, more importantly in our attack,  $d_j^r$ . The most straight forward algorithm is to recursively search the tree resulting from the state transitions in the convolutional encoder. By leaving the all-zero state, we search for the path leading back to the all-zero state with smallest Hamming weight. This weight is equivalent to the smallest Hamming weight of all codewords. A recursive search will first compute  $d_0^r$ . For each leaf in the tree we check if the accumulated Hamming weight is larger than or equal to  $d_0^r$ . If this is the case, that path is abandoned. Since the corresponding generator matrix might be catastrophic, it is possible to end up in an infinite loop and never return to the all-zero state. By saving the states corresponding to the visited nodes as long as the output weight is zero, we can easily check if we are in such a loop.

A more efficient algorithm is BEAST [4]. It was proposed in 2004, as an efficient algorithm to compute the weight spectra for convolutional codes. The minimum nonzero value of the weight spectrum corresponds to the free distance of the code, assuming a noncatastrophic generator matrix. The algorithm is similar to the straight forward algorithm given above, but additionally takes advantage of the fact that we know which state to return to, i.e., the all-zero state. Assume that we want to find the number of codewords of weight  $\alpha$ . Then the end state of all forward paths in the tree with Hamming weight  $\alpha/2$  are saved in a list. A similar list is constructed for all backward paths of weight  $\alpha/2$  and then the two lists are combined. Naturally, for a rate  $1/c$  code, we have to compute  $c$  forward lists and  $c$  backward lists and combine the lists that will result in codewords of Hamming weight  $\alpha$ . The BEAST algorithm is currently the fastest known algorithm to find  $d_{\text{free}}$  of a convolutional code. In [4], BEAST is described for noncatastrophic encoding matrices and a similar change as described above to detect loops of weight zero has to be implemented in order to apply it to our situation. Also, the algorithm has been modified to save  $\hat{j}$  when  $d_j^r$  has been computed.

Using our slightly modified version of BEAST, we have computed the minimum row distance  $d_j^r$  and  $\hat{j}$  for all possible combinations of two polynomials of degrees  $k_0$  and  $k_1$ ,  $1 \leq k_0, k_1 \leq 13$ .

We examine two main problems related to the current attack:

- What is the number of samples needed in the distinguisher?

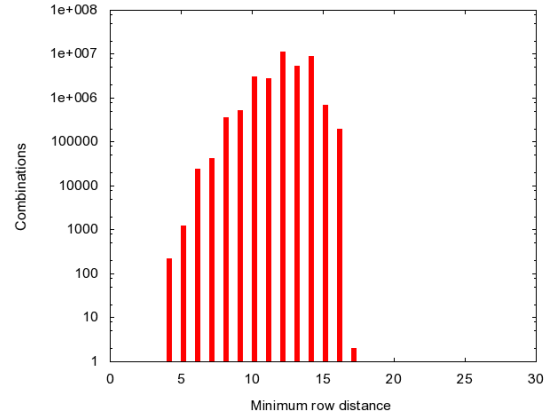


Fig. 1. The number of polynomial combinations with given  $d_j^r$ .

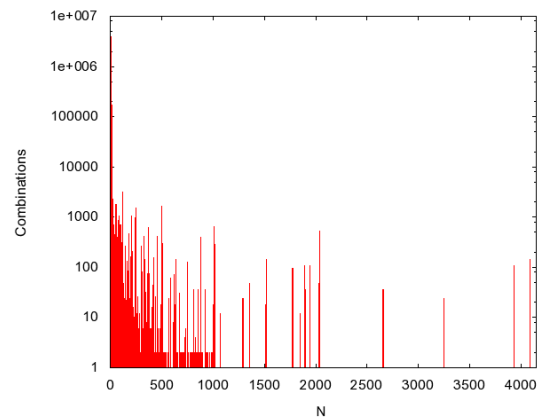


Fig. 2. The number of polynomial combinations requiring a given vector length  $N$ .

- Which vector length is needed in order to get the most efficient attack in terms of number of samples needed in the distinguisher?

Our new analysis allows us to answer the first question, not only for vector lengths  $N < 30$  as previously done [3], but for the *optimal* vector length. Furthermore, the second question can now be answered with the *optimal* vector length, not the optimal with the restriction  $N < 30$ .

Fig. 1 is a histogram that shows the number of polynomial combinations that results in a given  $d_j^r$ .

Since  $d_j^r$  immediately tells us the number of samples needed in the distinguisher, this graph gives a rough idea on what performance we can expect from our attack. Note that the largest free distance for a convolutional code with memory 13 is known to be 16. The two encoders in Fig. 1 with  $d_j^r = 17$  both corresponds to catastrophic encoding matrices.

Fig. 2 is a histogram showing the number of polynomial combinations that require a given vector length  $N = \hat{j} + 1$  in order to mount the best attack.

We see that even though most combinations require a modest value of  $N$ , some combinations require a very high

value, the largest being  $N = 4084$ . As with the free distance of a convolutional code, it is difficult to predict the efficiency of the attack, as well as the required vector length, by just looking at two arbitrary polynomials. However, for very large vector length it is possible to derive theoretical results.

## VI. THEORETICAL RESULTS

In this section we give a tight bound on the vector length given the maximum degree of the involved polynomials  $g_i(x)$ . We give expressions for the number of polynomial combinations with maximum vector length and also characterize the form of these polynomials. Further, we show that for *any* degree  $k > 3$  on the involved polynomials, there will *always* be combinations of weight  $> 6$  that result in  $d_j^r = 6$ , i.e., a very efficient attack compared to the basic attack.

In the sequel, the following property will be used.

*Property 1:* In a rate 1/2 encoder, if  $g_1(x) = (1+x)g_0(x)$ , then the output sequence generated by  $g_1(x)$  in the encoder is the derivative of the sequence generated by  $g_0(x)$ ,

$$v_i^{(1)} = v_i^{(0)} \oplus v_{i-1}^{(0)}. \quad (26)$$

Note that  $v_{-1}^{(0)} = 0$  since we start in the all-zero state.

Now, assume that the polynomial  $g_0(x)$  is primitive, and  $g_1(x) = (1+x)g_0(x)$ . Thus, the degree of  $g_1(x) = k$ , which is also the memory order of the encoder, and the degree of  $g_0(x) = k-1$ . Let  $k > 3$  and the input sequence to the encoder be the  $m$ -sequence with characteristic polynomial being the reciprocal of  $g_0(x)$ , followed by two trailing zeros. Moreover, the sequence is chosen such that it ends with  $k-2$  zeros. This necessarily means that it starts with a one. If the encoder starts in the all-zero state,  $v_0^{(0)} = 1$ . Then  $v_1^{(0)} = 0, \dots, v_{2^{k-1}-2}^{(0)} = 0$  because of the chosen input sequence. Since the sequence ended with  $k-2$  zeros, we have to input 2 extra zeros in order to force the encoder back to the all-zero state. Thus,  $v_{2^{k-1}-1}^{(0)} = 1$  and  $v_{2^k-1}^{(0)} = 0$  since the last memory cell is not used by  $g_0(x)$ . Combining this with Property 1 we get

$$\begin{aligned} v^{(0)} &= \underbrace{100000\dots 00}_{2^{k-1}-2}10 \\ v^{(1)} &= \underbrace{110000\dots 00}_{2^{k-1}-3}11. \end{aligned}$$

From this it follows that the minimum row distance in this case is 6. The optimal vector length is the number of rows in the generator matrix (17). We know that the number of columns is  $N+k$  and this equals the number of output symbols,  $2^{k-1}+1$ . Hence,

$$N = 2^{k-1} - (k-1). \quad (27)$$

Since  $g_0(x)$  is primitive, this is a tight upper bound on the vector length for polynomials of degree  $\leq k$ . (Note that this analysis holds also for  $k=3$ , with  $g_0(x) = 1+x+x^2$  and  $g_1(x) = 1+x^3$ , but the total weight is here 5 meaning that the basic attack will be better.) Based on the analysis we give the following proposition.

*Proposition 1:* If  $g_0(x)$  is primitive and  $g_1(x) = (1+x)g_0(x)$ . Then the optimal vector length in our attack is given

by  $N_{opt} = 2^{k-1} - (k-1)$ . The minimum row distance of the corresponding encoder is  $d_{N_{opt}-1}^r = 6$ . Moreover, the number of samples needed in the attack is in the order of  $\varepsilon^{-12}$ .

For any given degree  $k > 3$  of  $g_1(x)$ , the number of polynomial combinations with this property is given by the number of primitive polynomials of degree  $k-1$ , i.e.,

$$\frac{\varphi(2^{k-1}-1)}{k-1}, \quad (28)$$

where  $\varphi(\cdot)$  is the totient function.

This analysis explains the largest vector length  $N = 4084$  in Fig. 2. With  $k = 13$  we have  $\varphi(4095)/12 = 144$  polynomial combinations that require a vector length  $N_{opt} = 2^{12} - 12 = 4084$ . There are no two polynomials with degree  $\leq k$  requiring a larger vector length for the best attack.

## VII. CONCLUSION

We have shown a relation between a distinguishing attack based on weak polynomials and coding theory. The relation allows the use of efficient algorithms for weight spectra to be used to compute the complexity of the attack. Moreover, the optimal vector length used in the attack can be found very efficiently and we show that for any degree  $k$  of the polynomials  $g_i(x)$ , there are always very efficient attacks. It is crucial that these polynomials are avoided in nonlinear combiners.

## REFERENCES

- [1] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.
- [2] H. Englund, M. Hell, and T. Johansson, "Correlation attacks using a new class of weak feedback polynomials." in *Fast Software Encryption 2004*, ser. Lecture Notes in Computer Science, B. Roy and W. Meier, Eds., vol. 3017. Springer-Verlag, 2004, pp. 127–142.
- [3] M. Hell and L. Brynielsson, "Another look at weak feedback polynomials in the nonlinear combiner," in *International Symposium on Information Theory—ISIT 2009*. IEEE, 2009.
- [4] I. Bocharova, M. Handlery, R. Johannesson, and B. Kudryashov, "A BEAST for prowling in trees," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1295–1302, June 2004.
- [5] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Transactions on Computers*, vol. 34, pp. 81–85, 1985.
- [6] —, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Transactions on Information Theory*, vol. 30, pp. 776–780, 1984.
- [7] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Wiley series in Telecommunication. Wiley, 1991.
- [8] T. Baignères, P. Junod, and S. Vaudenay, "How far can we go beyond linear cryptanalysis?" in *Advances in Cryptology—ASIACRYPT 2004*, ser. Lecture Notes in Computer Science, vol. 3329. Springer-Verlag, 2004, pp. 432–450.
- [9] M. Hell, T. Johansson, and L. Brynielsson, "An overview of distinguishing attacks on stream ciphers," *Cryptography and Communications*, 2008.
- [10] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology—EUROCRYPT'93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed., vol. 765. Springer-Verlag, 1994, pp. 386–397.
- [11] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, ser. IEEE Series on Digital and Mobile Communication. IEEE Press, 1999.
- [12] S. Lin and D. J. Costello, *Error Control Coding, Second Edition*. Prentice-Hall, Inc., 2004.