Linear Cryptanalysis of the Fast Data Encipherment Algorithm

Kazuo Ohta[1] and Kazumaro Aoki[2]

[1] NTT Network Information Systems Laboratories
1-2356 Take, Yokosuka-shi, Kanagawa-ken, 238-03 Japan
[2] Department of Mathematics
School of Science and Engineering, Waseda University
3-4-1 Ookubo, Shinjuku-ku, Tokyo-to, 169 Japan

Abstract. This paper discusses the security of the Fast Data Encipherment Algorithm (FEAL) against Linear Cryptanalysis. It has been confirmed that the entire subkeys used in FEAL–8 can be derived with 2^{25} pairs of known plaintext and ciphertext with a success rate over 70% spending about 1 hour using a WS (SPARCstation 10 Model 30). This paper also evaluates the security of FEAL–N in comparison with that of the Data Encryption Standard (DES).

1 Introduction

This paper analyzes the applicability of Linear Cryptanalysis to the Fast Data Encipherment Algorithm (FEAL)[MSS88]. The structure of FEAL is similar to DES, except, for example, the permutation and the S-Boxes in F-function of DES are replaced by byte rotation and addition operation which are suitable for software implementation, and these differences are interesting from the viewpoint of cryptanalysis. In the Linear Cryptanalysis of FEAL, our main concerns in evaluating the security of FEAL considering the replacement of F-function and S-Boxes are: 1) how to find effective linear expressions, 2) an estimation of the success rate against the number of pairs of plaintexts and corresponding ciphertexts, N, and the approximate probability, p', and 3) an estimation of the memory size and the processing amount of the attack.

2 Linear Cryptanalysis

2.1 Notations and Preliminaries

The modified FEAL and its modified F-function [MY92] are analyzed here. The S-Boxes, S_0 and S_1, of FEAL are defined as $\mathrm{S}_i(x, y) = \mathrm{ROL2}(x + y + i \pmod{256})$, where ROL2 rotates its input two bits to the left.

We use the similar notations and defines the right most bit of each symbol as the 0-th bit, which is the lowest bit, as well as in the reference [M93-1]

2.2 Principle

Linear Cryptanalysis analyzes the probability that the following equation holds.

$$P[i_1, i_2, \ldots, i_a] \oplus C[j_1, j_2, \ldots, j_b] = S[k_1, k_2 \ldots, k_c], \tag{1}$$

where $i_1, i_2, \ldots, i_a, j_1, j_2, \ldots, j_b, k_1, k_2 \ldots, k_c$ are fixed bit locations defined by the linear expression, $(\Gamma P, \Gamma C, \Gamma K)$. The value of the right side of this equation depends only on the key values. We denote $S[k_1, k_2 \ldots, k_c]$ by S simply.

Two kinds of probability are defined in Linear Cryptanalysis: one is $p = \mathrm{Prob}_{P,K}\{E(P, K)(\Gamma C) \oplus P(\Gamma P) = K(\Gamma K)\}$, and the other is the absolute value of probability different from a half, $p' = |p - 1/2|$. Hereafter, p' will be used as the *probability* of the linear expression $(\Gamma P, \Gamma C, \Gamma K)$.

2.3 Implementation Techniques

Matsui proposed the following practical implementation, and captured the *effective text bits* among text information, P and C, which are essential to calculate Equation (2), and the *effective key bits* among key information, K_1 and K_n, which are essential to calculate Equation (2). Hereafter, t and k denote the number of effective text bits and the number of effective key bits, respectively.

$$\begin{aligned} P[i_1,\ldots,i_a] \oplus C[j_1,\ldots,j_b] \oplus F_1(P_L,K_1)[u_1,\ldots,u_d] \\ \oplus F_n(C_L,K_n)[v_1,\ldots,v_e] = S. \end{aligned} \qquad (2)$$

In the following procedure, we first count the text frequency on the effective text bits and then count the key frequency on the effective key bits.

Algorithm 1 (Counter Technique)

[**Data Counting Phase**]

Step 1: Prepare 2^t counters $U_i (0 \leq i < 2^t)$, where i corresponds to each value on the t effective text bits of Equation (2).

Step 2: For each plaintext and the corresponding ciphertext, compute the value 'i' of **Step 1** and count up the counter U_i by one.

[**Key Counting Phase**]

Step 3: Prepare 2^k counters $T_j (0 \leq j < 2^k)$, where j corresponds to each value on the k effective text bits of Equation (2).

Step 4: For each 'j' of **Step 3**, let T_j be the sum of U_i's such that the left side of Equation (2), whose value can be uniquely determined by i and j, is equal to 0.

Step 5: Let T_{max} be the maximal value and T_{min} be the minimal value of all $T_{i,j}$'s.

If $|T_{max} - N/2| > |T_{min} - N/2|$, then adopt the key candidate corresponding to T_{max} and guess $S = 0$ when $p > 1/2$ or 1 when $p < 1/2$.

If $|T_{max} - N/2| < |T_{min} - N/2|$, then adopt the key candidate corresponding to T_{min} and guess $S = 1$ when $p > 1/2$ or 0 when $p < 1/2$.

[**Exhaustive Search Phase**]

Step 6: Derive the remaining key bits exhaustively.

The computational complexity of this procedure except **Step 6** is $O(N) + O(2^{t+k})$. The number of counters, U_i and T_j, required by this procedure is $2^t + 2^k$.

3 Linear Approximation of FEAL

3.1 What are the Problems

The essential differences between DES and FEAL are the structure of S-Boxes and that of F-function. More exactly, S-Boxes of DES are defined in a non-mathematical way using tables. S-Boxes of FEAL are defined mathematically using modular addition calculation with two bits left rotation. So it seems easier to find some property of S-Boxes of FEAL than that of DES. On the other hand, the eight S-Boxes in F-function of DES act in parallel more independently than four S-Boxes in F-function of FEAL which act sequentially, where the byte rotation is built in instead of the permutation of DES. Thus, it seems easier to find some semi-global property of F-function of DES than that of FEAL.

It is not clear how these differences influence Linear Cryptanalysis.

3.2 Linear Expressions of F-function

We get various linear expressions of S-Boxes approximating the addition operation with the bitwise consideration of carry propagation as was done in [CG91].

When $a + b = x$, for example, $a[i] \oplus b[i] = x[i]$ holds with probability of $2^{-(i+1)} (i \geq 0)$, $a[i, i-j] \oplus b[i] = x[i]$, $a[i] \oplus b[i, i-j] = x[i]$ and $a[i] \oplus b[i] = x[i, i-j]$ hold with probability of $2^{-(j+1)} (1 \leq j \leq i)$.

Note that $a[0] \oplus b[0] = x[0]$ always holds, since there is no carry at the least significant bit in the addition operation, and this gives 15 non-trivial linear expressions of F-function with probability of 1/2, which can be always extended to 3-round linear expressions. If $j = 1$, we can make many examples with probability of 1/4 ignoring the bit position of i. This gives many local linear expressions with probability of 1/4.

Here the concatenation rule of XOR and BRANCH operations inside the F-function is also applicable in the same way as that between F-functions described in Section **??**.

3.3 Linear Expressions of Reduced Round FEAL

We developed the following search algorithm to find effective 7-round linear expressions, where $(\varGamma Y_{4-r}, \varGamma X_{4-r}) = (\varGamma Y_{4+r}, \varGamma X_{4+r})$ holds for $r = 1, 2, 3$.

Algorithm 2 (Search Algorithm of 7-Round Linear Expression)

Step 1: Set $(\varGamma Y_4, \varGamma X_4) = (0, 0)$.

Step 2: Select a linear expression, $(\varGamma Y_2, \varGamma X_2)$, of F-function whose probability is $\frac{1}{2}$.

Step 3: Search $\varGamma Y_3$ where $(\varGamma Y_3, \varGamma X_3)$ has the probability of 2^{-2}, given $\varGamma X_3 = \varGamma Y_2$.

Step 4: Search $\varGamma X'_2$ where $(\varGamma X_2, \varGamma X'_2)$ has the probability of greater than or equal to 2^{-3}.

Step 5: Put $(\varGamma Y_1, \varGamma X_1) = (\varGamma Y_3 \oplus \varGamma X_2, \varGamma X_3 \oplus \varGamma X'_2)$. Check whether its probability is greater than or equal to 2^{-4} exhaustively, if $(\varGamma Y_3, \varGamma X_3)$ activates the same S-Boxes of F-function as $(\varGamma X_2, \varGamma X'_2)$.

We have found the following eight pairs $(\varGamma X_2, \varGamma X'_2)$ using the above algorithm and sixteen 7-round linear expressions with probability of greater than 2^{-9}. This is one of examples with probability of 1.764×2^{-9}, which is effective in our implementation described in Section 5.

$\varGamma X_2$	$\varGamma X'_2$
00000100	10105050
00000100	18185858
00000100	10107878
00000100	18187070
01000000	50101010
01000000	58181818
01000000	70101818
01000000	78181010

r	$\varGamma Y_r$	$\varGamma X_r$	p'_r	
P	1D000400	50101010		
1	1D000400	54111010	85×2^{-10}	1.764×2^{-9}
2	04010000	01000000	2^{-1}	
3	1C000400	04010000	2^{-2}	
4	00000000	00000000	2^{-1}	
5	1C000400	04010000	2^{-2}	
6	04010000	01000000	2^{-1}	
7	1D000400	54111010	85×2^{-10}	
C	1D000400	50101010		

Note that the middle 5-round part of the above expression also has the probability of 1/8, while Biham described a 5-round linear expression with probability of 1/32 in [B94].

4 Discussion

4.1 Attack Strategy

We will discuss the attack strategy against FEAL–8 in this section: that is, 1) which is a better technique, 2-Round Elimination or 1-Round Elimination, and 2) which is the better expression for the 7-round case, Biham's or ours, as estimated from the memory size and processing amount.

Since the approximate probability of a linear expression for 6-round is larger than that for 7-round and $N = c \times p'^{-2}$ holds, the 2-Round Elimination strategy is better than 1-Round Elimination from the standpoint of the required number of pairs for attack. However, 2-Round Elimination is infeasible, since the number of effective text bits, t, and the number of effective key bits, k, satisfy $t, k = 24 \sim 30$ roughly, and the processing amount is $O(2^{42\sim 48})$ in 2-Round Elimination where we assume $N = 2^{18}$.

Let us estimate t and k for Biham's linear expression and those for our expression for 7-round, assuming the 1-Round Elimination Technique. An attack using our linear expression requires more running time than that using Biham's, since $t, k = 20 \sim 24$ holds roughly, and the processing amount is $O(2^{36\sim 40})$. An attack of 1-Round Elimination using Biham's 7-round expression is practical, since $t, k = 12 \sim 15$ and $N = c \times 2^{22}$ holds, and the processing amount of **Algorithm 1** is $O(2^{24\sim 30})$. The number of counters, U_i and T_j, required by **Algorithm 1** is $2^{12\sim 15}$, which is acceptable.

We decided to adopt the **1-Round Elimination Technique** using Biham's linear expression for the first phase of an attack against FEAL–8, that requires us to analyze the following equation:

$$\begin{aligned} P_L[16, 23, 25, 26, 31] \oplus C_H[31] \oplus C_L[16, 23, 25, 26] \\ \oplus F_8(C_H \oplus C_L, K_8)[23, 25, 31] = S. \end{aligned} \tag{3}$$

Our linear expression is effective for the later phases of an attack to derive subkeys other than those derived from the above equation.

4.2 Comparison with DES

The best expression can be obtained by an exhaustive search algorithm against DES [M93-1, M94-2]. It is an interesting fact that the number of active S-Boxes, which are approximated with a certain masking value, at each round of F-function is at most one in the best expression of 16-round DES. Since the bit length of data input to each S-Box is 6, the number of effective key bits is 12 in Equation (2).

On the other hand, since the number of active S-Boxes of Equation (2) against FEAL is four if the Biham's linear expression is used, and the byte rotation is built in implicitly between S-Boxes, the number of effective key bits and effective text bits seems to be $24 \sim 30$, which is larger than is true with DES. Thus 2-Round Elimination is infeasible in FEAL, which it is efficient in DES. Unfortunately, since we don't have any practical search algorithm to obtain the best expression of FEAL, there might be a better linear expression than Biham's.

How about the effective key and text bits? The closer from the right side an input bit is to a bit position related to a reference point output by the eighth F-function, the more strongly the value of the input bit determines the value of the XOR operation performed on the reference points, $F_8(C_H \oplus C_L, K_8)[23, 25, 31]$ in Equation (2). Therefore, the *effective key bits* should be subdivided into *explored key bits* and *detected key bits* for an attack against FEAL. Note that since each key bit input to an S-Box of DES influences all output bits more equally than that of FEAL, detected key bits are identical to explored key bits in DES. As a result, the treatment of effective key bits is simpler in an attack against DES than against FEAL. The similar discussion is valid in the treatment of effective text bits, which provides the number of counter U_i. Thus there are various strategies to reduce the number of effective (explored/detected) key/text bits in an attack against FEAL.

Concerning the parameter, N, of FEAL–N, where N means the iteration number of F-function, it seems that while FEAL–32 is as secure as DES against Differential Cryptanalysis, FEAL–16 is as secure as DES from the standpoint of Linear Cryptanalysis, since the number of key bits which are explored by the attack is 12 with the 14-round linear expression with probability of 1.192×2^{-21} in DES, while it is $12 \sim 15$ with the 15-round linear expression with probability of 2^{-23} in FEAL assuming the Biham's iterative 4-round expression is applied to the 15-round case.

5 Experimentation Results

The following information was described in [OA94]

(1) A table relating the success rate, the number of pairs, and the effective (explored/ detected) key bits needed to solve Equation (3) using the 1-Round Elimination Technique.
(2) How to derive the remaining values of all subkeys .
(3) How to improve the success rate.

Note that there might also be more efficient strategy than ours to decrease the number of effective text and key bits.

6 Concluding Remarks

It has been confirmed that the entire subkeys used in FEAL–8 can be derived from 2^{25} known plaintexts with a success rate over 70% spending about 1 hour, from 2^{26} known plaintexts with a success rate about 100% spending a little over 1 hour using a WS (SPARCstation 10 Model 30).

It seems that while FEAL–32 is as secure as DES against Differential Cryptanalysis, FEAL–16 is as secure as DES from the standpoint of Linear Cryptanalysis if we restrict ourselves to Matsui's implementation technique using Biham's linear expression. There are several open problems:

(1) Search algorithm to obtain the best expression of FEAL,
(2) More efficient technique than **Algorithm 1**, and
(3) More efficient strategy for reducing the numbers of effective text bits and effective key bits in an attack against FEAL–8.

References

[B94] E. Biham, "On Matsui's Linear Cryptanalysis," EUROCRYPT'94

[CG91] A. Tardy-Corfdir and H. Gilbert, "A known plaintext attack of FEAL–4 and FEAL–6," CRYPTO'91

[M93-1] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," EUROCRYPT'93

[M94-2] M. Matsui, "On Correlation between the Order of S-Boxes and the strength of DES," EUROCRYPT'94

[MY92] M. Matsui and A. Yamagishi, "A New Method for Known Plaintext Attack of FEAL Cipher," EUROCRYPT'92

[MSS88] S. Miyaguchi, A. Shiraishi and A. Shimizu, "Fast Data Encipherment algorithm FEAL–8," Review of Electrical Communication Laboratories, Vol. 36, No. 4, 1988

[OA94] K.Ohta and K.Aoki, "Linear Cryptanalysis of the Fast Data Encipherment Algorithm," Manuscript