Improving the Search Algorithm for the Best Linear Expression

Kazuo Ohta, Shiho Moriai, and Kazumaro Aoki[⋆]

NTT Laboratories, 1–2356 Take, Yokosuka, Kanagawa, 238-03 Japan

Abstract. It is important to find the best linear expression to estimate the vulnerability of cryptosystems to Linear Cryptanalysis. This paper presents a method to improve Matsui's search algorithm which determines the best linear expression. This method is based on analyzing the dominant factor of search complexity. We introduce the *search pattern* in order to reduce unnecessary search candidates, and apply the proposed search algorithm to DES and FEAL. The n-round best linear expressions of DES are found as fast as Matsui's algorithm for $n \leq 32$. Those of FEAL are found much faster than his algorithm; the required time is decreased from over three months to about two and a half days. New results for FEAL are also described; we find the n-round best linear expressions ($n \leq 32$) with higher deviations than those derived from Biham's 4-round iterative linear approximations.

1 Introduction

Linear Cryptanalysis was proposed by Matsui [M93] and is known to be one of the most effective known plaintext attacks. In Linear Cryptanalysis, we find the following linear approximation (equation (1)) which holds with probability $p \neq 1/2$ for randomly given plaintext P and the corresponding ciphertext C, and then determine one key bit $K[k_1, k_2, \ldots, k_c]$ by the maximum likelihood method.

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

where $i_1, i_2, \ldots, i_a,\ j_1, j_2, \ldots, j_b$, and $k_1, k_2, \ldots, k_c$ denote fixed bit locations and $A[i_1, i_2, \ldots, i_a] = A[i_1] \oplus A[i_2] \oplus \cdots \oplus A[i_a]$ ($A[i]$ denotes the i-th bit of A). If the success rate is fixed, equation (2) holds [M93, *Lemma 2*],

$$N|p - 1/2|^2 = c \text{ (fixed)}, \tag{2}$$

where N denotes the number of the known plaintexts needed to attack the cryptosystem. We call $|p - 1/2|$ the *deviation*. Equation (2) shows that it is important to find the linear approximation with the highest deviation, the *best linear expression*, to estimate the smallest value of N[†].

At EUROCRYPT'94, two important topics were discussed in [B94] and [M94]; 1) the duality between Linear Cryptanalysis and Differential Cryptanalysis, and

[⋆] Affiliation during this work: Department of Mathematics, School of Science and Engineering, Waseda University.

[†] Here we don't consider the reduction in N possible with multiple approximations [KR94].

2) a search algorithm for determining the best linear expression (differential characteristic) of DES-like cryptosystems. On 1), it was clarified how to construct the global linear expressions (differential characteristics) from the local ones in the field of Linear Cryptanalysis (Differential Cryptanalysis). On 2), it should be noted that confirming that the "best linear expression" is the best is difficult, for example, Biham described that "We have exhaustively verified that this iterative characteristic is the best among all the characteristics with at most one active S box at each round, ... (Matsui claims that his linear expression is the best without any restriction.)" in his paper [B94], which implies that there might be more effective linear expressions than that used by Matsui [M93]. To ensure his characteristic is the best without any restriction, Matsui developed a search algorithm where he introduced the temporary value of the best probability, $\overline{BEST_n}$, obtained under the restriction of limiting the number of "active" S-boxes at each round, and then got the best probability, $BEST_n$, without the restriction.

Matsui's search algorithm was applied to DES, s^2DES, and LOKI successfully, whose f functions have similar structures to that of DES [TSM94]. In applying this search algorithm to the above cryptosystems, we can find the best linear expressions with ease, because the speed-up technique using $\overline{BEST_n}$ is effective in pruning off the unnecessary branches.

Unfortunately, for some cryptosystems, his search algorithm is not fast enough and it takes over 3 months to find the best linear expression of FEAL–8 on a workstation (about 30 MIPS), for example. The main reason is that the search bound of his algorithm depends on the cryptosystem, and is so loose for some cryptosystems that an enormous number of candidates are searched for.

If we impose restrictions on the type of linear expression, for example, those constructed using the iterative linear approximations with small number of rounds, the search problem becomes easy. An efficient method was proposed in [K92], where the idea is to find effective differential characteristics of iterative type in Differential Cryptanalysis. His idea is expected to be also applicable to Linear Cryptanalysis because of the duality of these cryptanalyses. His approach might satisfy some attackers, since it helps them find an effective linear approximation with a large probability, but it doesn't satisfy the designers of cryptosystems. Thus, it is important to develop a search algorithm for the best linear expression with less complexity in order to estimate the vulnerability of cryptosystems to Linear Cryptanalysis.

We extend Matsui's search algorithm by analyzing the dominant factor of its complexity carefully, and propose a method to reduce the number of candidates, in other words, not to expand unnecessary nodes in the search tree. We can eliminate the unnecessary candidates using the set of possible values of the deviations of the linear approximations of f functions of all rounds, what we call the *search pattern*, before determining the linear approximations themselves.

We apply the improved search algorithm to DES and FEAL, and compare its search complexity against that of Matsui's algorithm. The n-round best linear expressions of DES are found as fast as Matsui's algorithm for $n \leq 32$. Those of

FEAL are found much faster than his algorithm; the time required is decreased from over three months to about two and a half days.

In truth, the best linear expressions obtained by Matsui's algorithm might not be the best (it doesn't mean that we don't consider the effect of multiple approximations), and those obtained by ours might not, either. This is because both algorithms calculate the deviation of the linear approximation of the f function by the *Piling-up Lemma* [M93] which assumes the linear approximations of the "active" S-boxes in the f function hold independently. The subject of this paper is to reduce the search time of Matsui's algorithm.

New results for FEAL are also obtained; we find the best linear expressions of 7-round with the deviation of 1.15×2^{-8}[‡], those of 15-round with 1.48×2^{-20}, and those of 31-round with 1.99×2^{-41}, while Biham's equivalent values were 2^{-11}, 2^{-23}, and 2^{-47}, respectively [B94].

This paper is organized as follows. In chapter 2 we introduce Matsui's search algorithm [M94]. Next, in chapter 3, after considering the search bounds and complexity of his search algorithm, we give two problems. In chapter 4 we propose an improved search algorithm to solve these problems. In chapter 5 we apply the improved search algorithm to DES and FEAL, and show how the search time is reduced. We also show the best deviations of DES and FEAL obtained from the search.

2 Preliminary

2.1 Linear Approximation of *f* Function

This paper discusses the security of iterated cryptosystems, where f is the round function. Let I_i, O_i, and K_i be the input data, the output data, and the subkey data of the i-th round f function. We define ΓX as the masking value of data X, and $X[\Gamma X]$ as the even parity value of $X \cdot \Gamma X$, where $\cdot$ represents a bitwise AND operation. We call equation (3) the *linear approximation of the i-th round f function*. This linear approximation may be abbreviated to the pair of masking values, $(\Gamma O_i, \Gamma I_i)$.

$$I_i[\Gamma I_i] \oplus O_i[\Gamma O_i] = K_i[\Gamma K_i] \tag{3}$$

This paper uses the term, *deviation*, which means the absolute value of the difference of the probability from 1/2. The deviation of the linear approximation of the i-th round f function $(\Gamma O_i, \Gamma I_i)$, denoted as $p'_i(\Gamma O_i, \Gamma I_i)$, is defined as follows[§]. We may simply use p'_i for $p'_i(\Gamma O_i, \Gamma I_i)$.

$$p'_i(\Gamma O_i, \Gamma I_i) = \mid Prob\{I_i[\Gamma I_i] \oplus O_i[\Gamma O_i] = 0\} - 1/2 \mid \tag{4}$$

[‡] Ohta and Aoki showed 7-round linear approximations with the deviation of 1.15×2^{-8} in [OA94]. We have confirmed that they found some of the 7-round "best" linear expressions.

[§] Since in our search algorithm $p'_i(\Gamma O_i, \Gamma I_i)$ is calculated by the *Piling-up Lemma* using the deviations of the linear approximations of "active" S-boxes considering complexity, the value of $p'_i(\Gamma O_i, \Gamma I_i)$ might differ from that obtained by equation (4). This problem was also not solved in [M94]. See section 4.3.

2.2 Best Linear Expression

In this paper we use $BEST_n$ defined by equation (5), following the definition of $BEST_n^{LC}$ used in [TSM94]. We don't consider multiple approximations described in [N94]. We call $BEST_n$ the *n-round best deviation.* The n-round linear approximation with the n-round best deviation is called the *n-round best linear expression.*

$$BEST_n = \max_{\Gamma O_i=\Gamma O_{i-2}\oplus\Gamma I_{i-1}(3\leq i\leq n),\ \Gamma P\neq 0}\{2^{n-1}\prod_{i=1}^{n}(p'_i(\Gamma O_i,\Gamma I_i))\} \qquad (5)$$

2.3 Matsui's Search Algorithm for Best Linear Expression

This section introduces the search algorithm for the best linear expression proposed in [M94]. It is based on the mathematical inductive method and derives $BEST_n$, which is the n-round best linear deviation, from the knowledge of all i-round best linear deviations, $BEST_i$ $(1 \leq i \leq n-1)$.

$\overline{BEST_n}$ is the temporary value of $BEST_n$ during the search. We have to pay attention when setting the initial value of $\overline{BEST_n}$. The search program can determine $BEST_n$ as long as $\overline{BEST_n} \leq BEST_n$ holds, but the farther from $BEST_n$ the initial value of $\overline{BEST_n}$ is, the more complex the search becomes. How close the initial value of $\overline{BEST_n}$ is to $BEST_n$ decides the efficiency of the search. The framework of Matsui's algorithm consists of the following recursive procedures. Here the value of $[p'_1, p'_2, \ldots, p'_t]$ is defined as the following equation,

$$[p'_1, p'_2, \ldots, p'_t] = 2^{t-1}\prod_{i=1}^{t} p'_i. \qquad (6)$$

[Matsui's Search Algorithm] [M94]

```
Procedure Round-1:
  For each candidate for $\Gamma O_1$, do the following:
  ▷ Let $p'_1 = \max_{\Gamma I} p'_1(\Gamma O_1, \Gamma I)$.
  ▷ If $[p'_1, BEST_{n-1}] < \overline{BEST_n}$, then try another candidate for $\Gamma O_1$.
  ▷ Call Procedure Round-2.

  Let $BEST_n = \overline{BEST_n}$.
  Exit the program.

Procedure Round-2:
  For each candidate for $\Gamma O_2$ and $\Gamma I_2$, do the following:
  ▷ Let $p'_2 = p'_2(\Gamma O_2, \Gamma I_2)$.
  ▷ If $[p'_1, p'_2, BEST_{n-2}] < \overline{BEST_n}$, then try another candidate for $\Gamma O_2$ and $\Gamma I_2$.
  ▷ Call Procedure Round-3.

  Return to the upper procedure.

Procedure Round-i: $(3 \leq i \leq n-1)$
  For each candidate for $\Gamma I_i$, do the following:
    ▷ Let $\Gamma O_i = \Gamma O_{i-2} \oplus \Gamma I_{i-1}$.
    ▷ Let $p'_i = p'_i(\Gamma O_i, \Gamma I_i)$.
    ▷ If $[p'_1, p'_2, \ldots, p'_i, BEST_{n-i}] < \overline{BEST_n}$, then try another candidate for $\Gamma I_i$.
    ▷ Call Procedure Round-(i+1).
  Return to the upper procedure.

Procedure Round-n:
  Let $\Gamma O_n = \Gamma O_{n-2} \oplus \Gamma I_{n-1}$.
  Let $p'_n = \max_{\Gamma I} p'_n(\Gamma O_n, \Gamma I)$.
  If $[p'_1, p'_2, \ldots, p'_n] > \overline{BEST_n}$, then $\overline{BEST_n} = [p'_1, p'_2, \ldots, p'_n]$.
  Return to the upper procedure.
```

3 Consideration of Matsui's Search Algorithm

3.1 Search Bound

Consider the search bound of Matsui's search algorithm that determines the range of p'_i, which is the deviation of the linear approximation of each round. In his search algorithm, the linear approximations whose deviation satisfies equation (7) become search candidates at the i-th round $(1 \le i \le n)$.

$$[p'_1, p'_2, \ldots, p'_i, BEST_{n-i}] \ge \overline{BEST_n} \tag{7}$$

Equation (7) is transformed into equation (8) using equation (6).

$$p'_1 \times p'_2 \times \cdots \times p'_i \ge \frac{\overline{BEST_n}}{2^i \times BEST_{n-i}} \tag{8}$$

The above equation shows that the product of $p'_1, p'_2, \ldots, p'_i$ depends on the ratio of $\overline{BEST_n}$ to $BEST_{n-i}$. There is a possibility that the value of $\overline{BEST_n}/BEST_{n-i}$ is too small for some cryptosystems, even if the initial value of $\overline{BEST_n}$ can be set close to $BEST_n$. If the value of the right side of inequality (8) is small, an excessive number of candidates of the linear approximations with deviation p'_i are searched for. This is because the number of the linear approximations of the f function with deviation $\ge p'$ increases, as the value of p' decreases, as Table 1 shows.

p'	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	$\cdots$
DES	1	13	195	3803	40035	371507	$\cdots$
FEAL	16	1808	98576	3453200	774484304	1215648016	$\cdots$

Table 1. Number of linear approximations of f function with deviation $\ge p'$

3.2 Complexity

This section discusses two subjects on the search complexity of Matsui's search algorithm. One is the dominant factor of search complexity, and the other is the relation between search complexity and round number.

First, we show that the complexity of the search for the n-round best linear expression is dominated by the number of candidates in *Procedures Round-1* and *Round-2*. In *Procedure Round-1*, all the $\mathit{\Gamma O}_1$s that satisfy equation (9) are search candidates, and in the *Procedure Round-2*, all the $(\mathit{\Gamma O}_2, \mathit{\Gamma I}_2)$s that satisfy equation (10) are search candidates.

$$p'_1 = \max_{\mathit{\Gamma I}} p'_1(\mathit{\Gamma O}_1, \mathit{\Gamma I}) \geq \frac{\overline{BEST_n}}{2 \times BEST_{n-1}} \tag{9}$$

$$p'_2 = p'_2(\mathit{\Gamma O}_2, \mathit{\Gamma I}_2) \geq \frac{\overline{BEST_n}}{2^2 \times p'_1 \times BEST_{n-2}} \tag{10}$$

In *Procedure Round-i* ($3 \leq i \leq n$), $\mathit{\Gamma O}_i$ is fixed by the equation, $\mathit{\Gamma O}_i = \mathit{\Gamma O}_{i-2} \oplus \mathit{\Gamma I}_{i-1}$, and only the $\mathit{\Gamma I}_i$s that satisfy equation (7) are search candidates. The number of candidates heuristically increases from *Procedure Round-1* to *Procedure Round-2* but then decreases in subsequent procedures. Thus the complexity of the search can be estimated from the number of the candidates in the first two procedures.

The search becomes more complex as the number of the candidates in the first two procedures increases; in other words, as the ratio of $\overline{BEST_n}$ to $BEST_{n-1}$ and that of $\overline{BEST_n}$ to $BEST_{n-2}$ decrease, according to inequalities (9) and (10).

Next, we show the relation between search complexity and the number of rounds. We define $\mathcal{C}_n$ as the estimated value of the *incremental* search complexity for the n-round best linear expression under the desirable condition that we know all $BEST_r$ ($r \leq n$) values, *i.e.* the least search complexity. As mentioned above, the search complexity can be estimated from the number of candidates in the first two procedures. Thus $\mathcal{C}_n$ is obtained using the data in Table 1 and calculating all the $\mathit{\Gamma O}_1$s and $(\mathit{\Gamma O}_2, \mathit{\Gamma I}_2)$s that satisfy both inequalities (9) and (10).

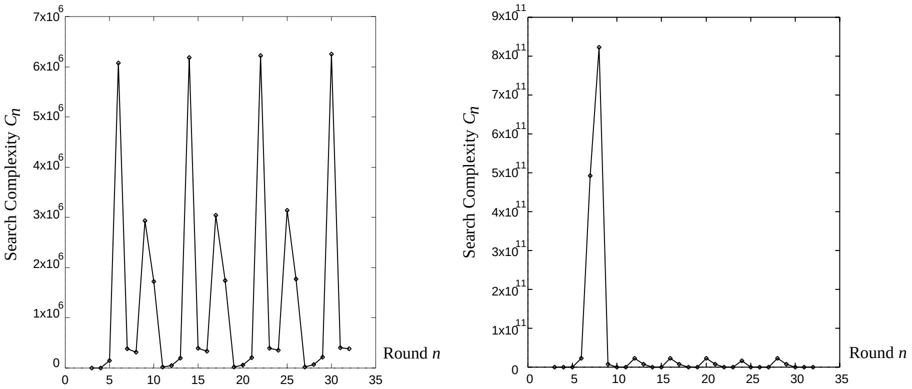
The relations between $\mathcal{C}_n$ and n for DES and FEAL are shown in Figure 1. Figure 1 shows that $\mathcal{C}_n$ of FEAL is much more than that of DES. (Note the scale of the vertical axis.) It is also clear that $\mathcal{C}_n$ isn't related to n.

3.3 Problems

This section describes two problems of Matsui's search algorithm. We begin by introducing the term, *search pattern.* We define the search pattern as the set of n deviations $(q'_1, q'_2, \ldots, q'_n)$ that satisfies $p'_1 = q'_1$, $p'_2 = q'_2$, $\ldots$, and $p'_n = q'_n$ when we search for the n-round linear approximations.

Problem 1. Duplicate Candidates

The n-round linear approximation whose linear approximation of the i-th round f function $(\mathit{\Gamma O}_i, \mathit{\Gamma I}_i)$ is exchanged for that of the $(n - i + 1)$-th round



(1) DES (2) FEAL

Fig. 1. Relation between $\mathcal{C}_n$ and n

f function $(\mathit{\Gamma O}_{n-i+1},\ \mathit{\Gamma I}_{n-i+1})$ for all i $(1 \leq i \leq n)$ has the same meaning as the original one, since we can exchange the roles of P and C, and also K_i and K_{n+1-i} in equation (1). In Matsui's search algorithm, when search pattern $(q'_1, q'_2, \ldots, q'_n)$ satisfies equation (7), so does its inverse pattern $(q'_n,\ q'_{n-1},\ \ldots,\ q'_1)$. Two linear approximations with the same meaning from the viewpoint of Linear Cryptanalysis are searched for in his search algorithm, and one of them is unnecessary.

Problem 2. Nonexistent Candidates

In Matsui's search algorithm each search pattern $(q'_1, q'_2, \ldots, q'_n)$ satisfies all the following inequalities, which are derived from equation (7).

$$\begin{aligned}
[\,q'_1, q'_2,\ \ldots\ldots, q'_n] &\leq \overline{BEST_n} \\
[\,q'_2, q'_3,\ \ldots\ldots, q'_n] &\leq BEST_{n-1} \\
[\,q'_3, q'_4,\ \ldots\ldots, q'_n] &\leq BEST_{n-2} \\
&\vdots \\
[\,q'_n] &\leq BEST_1
\end{aligned} \tag{11}$$

These inequalities are not guaranteed to yield the conditions that *all* r-round $(1 \leq r < n)$ linear approximations contained in the n-round linear approximation have the deviation less than or equal to $BEST_r$. In other words, there is a possibility that the linear approximation whose search pattern satisfies inequality (12) is searched for unnecessarily.

$$[q'_i, q'_{i+1}, \ldots, q'_{i+r-1}] > BEST_r \quad (1 \leq i \leq n,\ i + r - 1 < n) \tag{12}$$

From the definition of $BEST_r$ (equation (5)), the linear approximations that satisfy inequality (12) don't exist, and need not be searched for.

4 Discussion

4.1 Solving the Problems

The problems explained in section 3.3 are solved as follows. Note that the problems concern only the deviations of the linear approximations of the f function, and are independent of the choice of $(\Gamma O_i, \Gamma I_i)$.

First we choose the search patterns $(q'_1,\, q'_2,\, \ldots,\, q'_n)$ that satisfy the following two conditions, and secondly decide $(\Gamma O_i, \Gamma I_i)$ for each round using the chosen search patterns. In order to cover all possible search patterns, we make use of the idea of Matsui's search algorithm and list all the $(q'_1, q'_2, \ldots, q'_n)$s that satisfy inequality (7). We can guarantee that the set of the chosen search patterns contains all the candidates needed to be searched for, since Matsui's search algorithm doesn't miss possible candidates.

Condition 1 (Deletion of Duplicate Candidates)
The search pattern $(q'_1,\, q'_2,\, \ldots,\, q'_n)$ must satisfy the following condition,

$$\mathcal{C}(q'_1, q'_2, \ldots, q'_n) \leq \mathcal{C}(q'_n, q'_{n-1}, \ldots, q'_1)$$

where $\mathcal{C}(q'_1, q'_2, \ldots, q'_n)$ that satisfies $p'_1 = q'_1$, $p'_2 = q'_2$, $\ldots$, and $p'_n = q'_n$ is the complexity of the search for the n-round linear approximations.

As shown in section 3.2, $\mathcal{C}(q'_1, q'_2, \ldots, q'_n)$ can be estimated well by the number of candidates in the first two procedures. Thus we might compare $\mathcal{C}(q'_1, q'_2)$ with $\mathcal{C}(q'_n, q'_{n-1})$ instead of the inequality above, where $\mathcal{C}(a, b)$ denotes the number of the linear approximations with $p'_1 = a$ and $p'_2 = b$. $\mathcal{C}(a, b)$ is also calculated using the data in Table 1.

Consider the following example; when we search for the 7-round best linear expression of FEAL, there are such search patterns as (2^{-3}, 2^{-3}, 2^{-2}, 2^{-2}, 2^{-1}, 2^{-1}, 2^{-1}) and its inverse pattern (2^{-1}, 2^{-1}, 2^{-1}, 2^{-2}, 2^{-2}, 2^{-3}, 2^{-3}). We choose the latter search pattern, because $\mathcal{C}(2^{-3}, 2^{-3})$ is 9,364,045,824, and $\mathcal{C}(2^{-1}, 2^{-1})$ is only 256.

Condition 2 (Deletion of Nonexistent Candidates)
For all i and r $(1 \leq i \leq n\ ,\ i + r - 1 < n)$, the search pattern $(q'_1,\, q'_2,\, \ldots,\, q'_n)$ must also satisfy the following condition.

$$[q'_i, q'_{i+1}, \ldots, q'_{i+r-1}] \leq BEST_r$$

4.2 Improved Search Algorithm

The improved search algorithm is shown below. It determines $BEST_n$ and the best linear expression *i.e.* the set of n linear approximations of the f function $(\Gamma O_i, \Gamma I_i)$ $(1 \leq i \leq n)$. It consists of four routines, *Procedure Main*, *MakeList-round-i*, *PickandChoose* and *Search*.

In *Procedure MakeList-round-i*, the list of all the possible search patterns is made. In *Procedure PickandChoose*, the search pattern $Pattern_j(p'^j_1,\, p'^j_2,\, \ldots,$

$p_n'^j$) which satisfies *Conditions 1* and *2* is chosen from the list made in *Procedure MakeList-round-i*, and then *Procedure Search* is called with $Pattern_j$ as a parameter. In *Procedure Search*, the linear approximations of the f function whose deviation of the i-th round p_i' equals $p_i'^j$ are searched for. The difference of *Procedure Search* from Matsui's search algorithm is that the deviation of the linear approximation of the f function is decided using $Pattern_j(p_1'^j,\ p_2'^j,\ \ldots,\ p_n'^j)$ chosen in *Procedure PickandChoose*, not equation (7).

[Our Search Algorithm] (for FEAL[¶])

```
Procedure Main:
    Let $\overline{BEST_n} = 2 \times BEST_{n-1}$, and $BEST_n = 1$.
    Do
      ▷ Let $\overline{BEST_n} = 2^{-1} \times \overline{BEST_n}$.
      ▷ Call Procedure MakeList-round-1.
      ▷ Call Procedure PickandChoose.
    while $BEST_n \neq \overline{BEST_n}$.
    Exit the program.

Procedure MakeList-round-i: $(1 \leq i \leq n)$
    For each candidate for $p_i'$, do the following:
      ▷ If $[p_1', p_2', \ldots, p_i', BEST_{n-i}] < \overline{BEST_n}$, then try another candidate for $p_i'$.
      ▷ If $i < n$, then call Procedure MakeList-round-(i+1),
        else if $i = n$, then add $(p_1', p_2', \ldots, p_n')$ satisfying $[p_1', p_2', \ldots, p_n'] = \overline{BEST_n}$
        to the list of the search patterns.
    Return to the upper procedure.

Procedure PickandChoose:
    For each candidate for $(p_1', p_2', \ldots, p_n')$, do the following:
      ▷ If $^{\exists}i, ^{\exists}r\ (1 \leq i \leq n, i + r - 1 < n)$ satisfying $[p_i', p_{i+1}', \ldots, p_{i+r-1}'] > BEST_r$,
        then try another candidate for $(p_1', p_2', \ldots, p_n')$.
      ▷ If $\mathcal{C}(p_1', p_2') > \mathcal{C}(p_n', p_{n-1}')$, then try another candidate for $(p_1', p_2', \ldots, p_n')$.
      ▷ Let $Pattern_j(p_1'^j, p_2'^j, \ldots, p_n'^j) = (p_1', p_2', \ldots, p_n')$.
      ▷ Call Procedure Search $(Pattern_j)$.
    Return to the upper procedure.

Procedure Search $(Pattern_k)$:
    Procedure Round-h: $(h = 1, 2)$
       For each $\mathit{\Gamma O}$ and $\mathit{\Gamma I}$ s.t. $p_h'(\mathit{\Gamma O}, \mathit{\Gamma I}) = p_h'^k$,
      ▷ Let $\mathit{\Gamma O}_h = \mathit{\Gamma O}$, and $\mathit{\Gamma I}_h = \mathit{\Gamma I}$.
      ▷ Call Procedure Round-(h+1).
       Return to the upper procedure.

    Procedure Round-i: $(3 \leq i < n)$
       Let $\mathit{\Gamma O}_i = \mathit{\Gamma O}_{i-2} \oplus \mathit{\Gamma I}_{i-1}$.
```

[¶] This search algorithm cannot always find the best linear expression of DES faster than Matsui's algorithm. In the case of DES, the technique of improving the value of the right side of inequality (8) using the search patterns chosen in *Procedure PickandChoose* speeds up his algorithm.

```
    For each $\Gamma I$ s.t. $p'_i(\Gamma O_i, \Gamma I) = p'^k_i$,
  ▷ Let $\Gamma I_i = \Gamma I$.
  ▷ Call Procedure Round-(i+1).
    Return to the upper procedure.

Procedure Round-n:
    Let $\Gamma O_n = \Gamma O_{n-2} \oplus \Gamma I_{n-1}$.
    Let $\Gamma I_n = \Gamma I$ and $BEST_n = \overline{BEST_n}$ if $p'_n(\Gamma O_n, \Gamma I) = p'^k_n$.
    Return to the upper procedure.
```

4.3 Revision of Best Deviation

Because both of Matsui's search algorithm and ours calculate the deviation of the linear approximation of the f function by the *Piling-up Lemma* for the sake of complexity, it might differ from the true one obtained from equation (4). We revise $BEST_n$ determined by the search program as follows; for all the best linear expressions obtained from the search, we calculate the true deviations of all rounds of linear approximations and determine the best deviations.

However, we might miss the *truly* best linear expression. Because the deviation of linear approximation of each round can be increased or decreased by revision, the truly best linear expression might come from the linear approximation with smaller deviation than $BEST_n$[‖]. It is difficult to search all possible linear expressions for the truly best deviation because of too much complexity. The method to compute the true deviation of the linear approximation of the f function with less complexity will solve this problem. (See Appendix.)

5 Experimental Results

5.1 Search Time

We applied our search algorithm to DES and FEAL using a SPARCstation 10 (SuperSPARC/36MHz, 31MIPS). It takes the times shown in *Tables 2* and *3*, respectively, to find the n-round ($4 \leq n \leq 8$[**]) best linear expression when we set the initial value of $\overline{BEST_n}$ equal to $BEST_n$. Note that the data with "$*$" are theoretical estimations from the number of candidates.

The best linear expressions of DES are found with ease by Matsui's algorithm. One of the reasons is that DES doesn't have so many linear approximations of the f functions with large deviations, as *Table 1* shows. Another reason is that the technique that reduces candidates by limiting the number of "active" S-boxes in the f function is effective for DES [M93]. On the other hand, Matsui's algorithm takes too much time to search for those of FEAL. This is because FEAL has many more candidates than DES, and the technique of limiting the number of

[‖] The 7- and 15-round linear expressions of FEAL with the best deviation we have found are in the linear approximations with the deviation of $2^{-1} \times BEST_n$ [MAO96].

[**] The search times of the other n-round ($9 \leq n \leq 32$) best expressions can be estimated from *Figure 1*.

Round	4	5	6	7	8
Our Algorithm	0.2 sec	1.2 sec	0.2 sec	0.2 sec	0.3 sec
Matsui's Algorithm	0.2 sec	1.2 sec	19.2 sec	1.3 sec	0.9 sec

Table 2. Search Time (DES)

Round	4	5	6	7	8
Our Algorithm	0.0 sec	0.4 sec	3.8 min	4.2 hr	2.3 day
Matsui's Algorithm	0.3 sec	23.1 sec	39.0 hr	*34 day	*58 day

Table 3. Search Time (FEAL)

"active" S-boxes is not effective for FEAL, since the linear approximation of each round in the n-round ($n \leq 32$) best linear expressions of DES has no "active" S-box or only one, but almost all S-boxes are "active" in the case of FEAL.

5.2 Best Deviations of DES and FEAL

Figure 2 shows the n-round best deviations of DES and FEAL ($1 \leq n \leq 32$). This is based on the data in *Table 4*, which are the revised data as explained in section 4.3. Note that each best deviation of DES is the same as $BEST_n$ determined by the search program, because the linear approximation of each round of the best linear expressions of DES has no "active" S-box or only one, and the deviation calculated by the *Piling-up Lemma* is the same as the true deviation. *Figure 2* also shows that all the n-round ($n \leq 32$) best deviations of FEAL we found are higher than those derived from Biham's 4-round iterative linear approximations [B94].

All the n-round ($10 \leq n \leq 32$) best linear expressions of FEAL that we obtained are based on the following type of 8-round iterative linear approximations. The following iterative linear approximation is one of them, and there are other variations of this type. The best linear expressions of DES are also based on a similar 8-round iterative linear approximation "`-ACD-DCA`" [M93]. Note that numbers in brackets are the true deviations obtained from the definition of p'_i (equation (4)). The difference occurs because p'_i is calculated by the *Piling-up Lemma* in the search algorithm (See section 4.3).

i	ΓO_i	ΓI_i	p'_i	(true deviation)	
1	00000000	00000000	2^{-1}	(1.00×2^{-1})	2^{-13} (1.64×2^{-12})
2	02a40104	81010100	2^{-3}	(1.00×2^{-3})	
3	81010100	00600000	2^{-4}	(1.81×2^{-4})	
4	02c40104	81010100	2^{-2}	(1.00×2^{-2})	
5	00000000	00000000	2^{-1}	(1.00×2^{-1})	
6	02c40104	81010100	2^{-2}	(1.00×2^{-2})	
7	81010100	00600000	2^{-4}	(1.81×2^{-4})	
8	02a40104	81010100	2^{-3}	(1.00×2^{-3})	

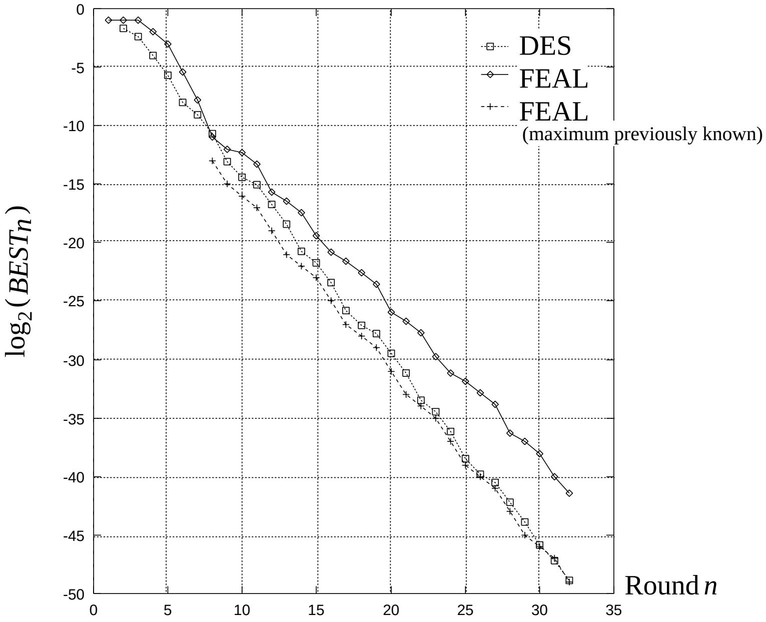


Fig. 2. The Best Deviation

6 Conclusion

We have improved Matsui's search algorithm by considering the dominant factor of search complexity. To discard many unnecessary search candidates before searching, we introduced the *search patterns*, which are preselected set of deviations of linear approximations of each round. Now we have two alternatives for finding the best linear expression. Matsui's search algorithm is easy to implement and works well for some cryptosystems. Our algorithm is equally effective and is suitable for those cryptosystems which cannot be readily searched by his algorithm.

Our search algorithm was applied to DES and FEAL. The n-round best linear expressions of DES we found as fast as Matsui's algorithm for $n \leq 32$. Those of FEAL we found much faster than his algorithm; the time required is decreased from over three months to about two and a half days. We have found the n-round best linear expressions of FEAL ($n \leq 32$) with higher deviations than those derived from Biham's iterative linear expressions.

Our idea is expected to be also effective in searching the differential characteristics in Differential Cryptanalysis.

Number	**DES**	**FEAL**		
of Round	Best Deviation	Best Deviation	$BEST_n$	Biham's Results
n	$(= BEST_n)$			[B94]
2	1.25×2^{-2}	1.00×2^{-1}	2^{-1}	–
3	1.56×2^{-3}	1.00×2^{-1}	2^{-1}	–
4	1.95×2^{-5}	1.00×2^{-2}	2^{-2}	–
5	1.22×2^{-6}	1.00×2^{-3}	2^{-3}	–
6	1.95×2^{-9}	1.52×2^{-6}	2^{-6}	–
7	1.95×2^{-10}	1.15×2^{-8}	2^{-8}	–
8	1.22×2^{-11}	1.00×2^{-11}	2^{-11}	2^{-13}
9	1.91×2^{-14}	1.00×2^{-12}	2^{-12}	2^{-15}
10	1.53×2^{-15}	1.64×2^{-14}	2^{-14}	2^{-16}
11	1.91×2^{-16}	1.64×2^{-14}	2^{-15}	2^{-17}
12	1.19×2^{-17}	1.24×2^{-16}	2^{-18}	2^{-19}
13	1.49×2^{-19}	1.48×2^{-17}	2^{-19}	2^{-21}
14	1.19×2^{-21}	1.48×2^{-18}	2^{-20}	2^{-22}
15	1.19×2^{-22}	1.48×2^{-20}	2^{-21}	2^{-23}
16	1.49×2^{-24}	1.13×2^{-21}	2^{-24}	2^{-25}
17	1.16×2^{-26}	1.34×2^{-22}	2^{-25}	2^{-27}
18	1.86×2^{-28}	1.34×2^{-23}	2^{-26}	2^{-28}
19	1.16×2^{-28}	1.34×2^{-24}	2^{-27}	2^{-29}
20	1.46×2^{-30}	1.02×2^{-26}	2^{-30}	2^{-31}
21	1.82×2^{-32}	1.21×2^{-27}	2^{-31}	2^{-33}
22	1.46×2^{-34}	1.21×2^{-28}	2^{-32}	2^{-34}
23	1.46×2^{-35}	1.21×2^{-30}	2^{-34}	2^{-35}
24	1.82×2^{-37}	1.85×2^{-32}	2^{-36}	2^{-37}
25	1.42×2^{-39}	1.10×2^{-32}	2^{-37}	2^{-39}
26	1.14×2^{-39}	1.10×2^{-33}	2^{-38}	2^{-40}
27	1.42×2^{-41}	1.10×2^{-34}	2^{-39}	2^{-41}
28	1.78×2^{-43}	1.67×2^{-37}	2^{-42}	2^{-43}
29	1.11×2^{-44}	1.99×2^{-38}	2^{-43}	2^{-45}
30	1.12×2^{-46}	1.99×2^{-39}	2^{-44}	2^{-46}
31	1.78×2^{-48}	1.99×2^{-41}	2^{-46}	2^{-47}
32	1.11×2^{-49}	1.51×2^{-42}	2^{-48}	2^{-49}

Table 4. The Best Deviation

Acknowledgement

We are grateful to Dr. Mitsuru Matsui for providing his software implementation of his DES search algorithm and helpful discussions, and to Toshio Tokita for showing details of experiments made to search for the best linear expressions of DES, s^2DES and LOKI.

References

[B94] E. Biham: "On Matsui's Linear Cryptanalysis (extended abstract)," Preproceedings of EUROCRYPT'94, 1994

[K92] L. R. Knudsen: "Iterative Characteristics of DES and s^2-DES," Advances in Cryptology – EUROCRYPT'92, Springer-Verlag 658, 1993

[KR94] B. S. Kaliski Jr. and M. J. B. Robshaw: "Linear Cryptanalysis Using Multiple Approximations," Advances in Cryptology – CRYPTO'94, Springer-Verlag 839, 1994

[N94] K. Nyberg: "Linear Approximation of Block Ciphers," Preproceedings of EUROCRYPT'94, 1994

[M93] M. Matsui: "Linear Cryptanalysis Method for DES Cipher," Advances in Cryptology – EUROCRYPT'93, Springer-Verlag 765, 1994

[M94] M. Matsui: "On Correlation between the order of S-Boxes and the Strength of DES (extended abstract)," Preproceedings of EUROCRYPT'94, 1994

[MAO96] S. Moriai, K. Aoki and K. Ohta: "The Best Linear Expression Search of FEAL," IEICE Trans. Fundamentals, Vol. E79-A, No. 1, 1996 (to appear)

[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," Advances in Cryptology – CRYPTO'94, Springer-Verlag 839, 1994

[TSM94] T. Tokita, T. Sorimachi and M. Matsui: "Linear cryptanalysis of LOKI and s^{2}DES (extended abstract)," Preproceedings of ASIACRYPT'94, 1994

Appendix Open Problem

When more than two S-boxes are approximated in the f function, the *Piling-up Lemma* can't always calculate the true deviation of the f function. *Table 5* gives examples of the linear approximations of the f functions of DES and FEAL whose deviations calculated by the *Piling-up Lemma* differ from the true values.

We can't currently compute the true deviation of the f function except using equation (4) with exhaustive search. How to compute the true deviation of the linear approximation of the f function with less complexity is an open problem.

	Linear approximation of f function $(\Gamma O_i, \Gamma I_i)$	$p'_i(\Gamma O_i, \Gamma I_i)$ calculated by	
		Piling-up Lemma	Definition (Equation(4))
DES	(00140000, a10400c0)	0	1.25×2^{-3}
FEAL	(81010100, 00600000)	2^{-4}	1.81×2^{-4}

Table 5. Examples of Linear approximations of f functions whose deviations calculated by the *Piling-up Lemma* differ from the true values

This article was processed using the LaTeX macro package with LLNCS style