The First Experimental Cryptanalysis of the Data Encryption Standard

Mitsuru Matsui

Computer & Information Systems Laboratory
Mitsubishi Electric Corporation
5-1-1, Ofuna, Kamakura, Kanagawa, 247, Japan
matsui@mmt.isl.melco.co.jp

Abstract. This paper describes an improved version of linear cryptanalysis and its application to the first successful computer experiment in breaking the full 16-round DES. The scenario is a known-plaintext attack based on two new linear approximate equations, each of which provides candidates for 13 secret key bits with negligible memory. Moreover, reliability of the key candidates is taken into consideration, which increases the success rate. As a result, the full 16-round DES is breakable with high success probability if 2^{43} random plaintexts and their ciphertexts are available. The author carried out the first experimental attack using twelve computers to confirm this: he finally reached all of the 56 secret key bits in fifty days, out of which forty days were spent for generating plaintexts and their ciphertexts and only ten days were spent for the actual key search.

1 Introduction

In the first paper on linear cryptanalysis [2], we introduced a new measure of linearity of S-boxes and extended it to the entire cipher structure of DES. The resultant linear approximate equations are effectively applicable to a known-plaintext attack, which proved that DES is breakable with negligible memory if 2^{47} random plaintexts and their ciphertexts are available. This is the first known-plaintext attack faster than an exhaustive key search, though the origin of linear cryptanalysis can be seen in several papers [4][5][6][7].

This paper studies an improved version of linear cryptanalysis and its application to the first successful computer experiment in breaking the full 16-round DES. We newly introduce two viewpoints; linear approximate equations based on the best $(n-2)$-round expression, and reliability of the key candidates derived from these equations. The former reduces the number of required plaintexts, whereas the latter increases the success rate of our attack.

In the 2^{47}-method, we established two linear approximate equations of 16-round DES using the best 15-round expression, where each equation includes one active S-box and hence recovers 7 secret key bits. This paper, however, begins with two new linear approximate equations derived from the best 14-round expression, where each equation has two active S-boxes and can recover 13 secret key bits. These equations give us, therefore, a total of 26 secret key

bits, and then the remaining $56 - 26 = 30$ secret key bits are within the reach of an exhaustive search.

Moreover, we treat not only one solution of each equation but also "candidates" for the 13 secret key bits, where each candidate has its ranking of reliability such that the i-th rank represents the i-th likely solution. The aim of this approach is to give a table that relates ranking of the 26 secret key bits to that of the 13 secret key bits. This table increases the success rate of our attack at the cost of computational complexity; that is to say, if the most likely 26 key bits turn out to be wrong, we can adopt the second likely 26 key bits and search for the remaining 30 key bits again. If they are not correct either, we can try the third likely 26 key bits.

We also prove that the effectiveness of this method can be measured by DES reduced to 8 rounds. This fact enables us to experimentally determine the relationship among the number of required plaintexts, the complexity and the success rate of our attack. As a result, DES is breakable with complexity 2^{43} and success rate 85% if 2^{43} known-plaintexts are available. For another example, success rate is 10% with complexity 2^{50} if 2^{38} known-plaintexts are available.

We carried out the first experimental attack of the full 16-round DES using twelve computers (HP9735/PA-RISC 99MHz) to confirm this scenario. The program, described in C and assembly languages consisting of a total of 1000 lines, was designed to solve two equations while generating 2^{43} random plaintexts and enciphering them. We finally reached all of the 56 secret key bits in fifty days, out of which forty days were spent for generating plaintexts and their ciphertexts and only ten days were spent for the actual key search.

2 Preliminaries

We follow the notations introduced in [2]. Since our scope is a known-plaintext attack using random plaintexts, we omit the initial permutation IP, the final permutation IP^{-1}, and PC-1. The right most bit of each symbol is referred as the 0-th (lowest) bit, whereas the traditional rule defines the left most bit as the first bit [1]. The following are used throughout this paper;

P	The 64-bit data after the IP; the plaintext.
C	The 64-bit data before the IP^{-1}; the ciphertext.
P_H, P_L	The upper and the lower 32-bit data of P, respectively.
C_H, C_L	The upper and the lower 32-bit data of C, respectively.
K	The 56-bit data after the PC-1; the secret key.
K_r	The r-th round 48-bit subkey.
$F_r(X_r, K_r)$	The r-th round F-function.
$A[i]$	The i-th bit of A, where A is any binary vector.
$A[i, j, .., k]$	$A[i] \oplus A[j] \oplus ... \oplus A[k]$.

3 Principle of the New Attack

The first purpose of linear cryptanalysis is to find the following linear approximate expression which holds with probability $p \neq 1/2$ for randomly given plaintext P, the corresponding ciphertext C and the fixed secret key K :

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

where $i_1, i_2, .., i_a, j_1, j_2, .., j_b$ and $k_1, k_2, .., k_c$ denote fixed bit locations.

Since both sides of equation (1) essentially represent one-bit information, the magnitude of $|p - 1/2|$ expresses the effectiveness. We will refer to the most effective linear approximate expression (i.e. $|p - 1/2|$ is maximal) as the best expression and its probability as the best probability, respectively. We have found the best expression and the best probability of DES, whose results are summarized in [2] for the number of rounds varying from $n = 3$ to $n = 20$. A practical algorithm for deriving these values is described in [3].

In the 2^{47}-method, we established two equations of 16-round DES using the best 15-round expression, which holds with probability $1/2 + 1.19 \times 2^{-22}$ [2]. Our new attack, however, starts with the following two best 14-round expressions, which hold with probability $1/2 - 1.19 \times 2^{-21}$:

$$\begin{aligned} & P_L[7, 18, 24] \oplus C_H[7, 18, 24, 29] \oplus C_L[15] \\ &= K_2[22] \oplus K_3[44] \oplus K_4[22] \oplus K_6[22] \oplus K_7[44] \oplus K_8[22] \oplus K_{10}[22] \oplus \\ &\quad K_{11}[44] \oplus K_{12}[22] \oplus K_{14}[22], \end{aligned} \tag{2}$$

$$\begin{aligned} & C_L[7, 18, 24] \oplus P_H[7, 18, 24, 29] \oplus P_L[15] \\ &= K_{13}[22] \oplus K_{12}[44] \oplus K_{11}[22] \oplus K_9[22] \oplus K_8[44] \oplus K_7[22] \oplus K_5[22] \oplus \\ &\quad K_4[44] \oplus K_3[22] \oplus K_1[22], \end{aligned} \tag{3}$$

where P, C and K denote the plaintext, the ciphertext and the secret key of DES reduced to 14 rounds, respectively.

Then applying equations (2) and (3) to fourteen consecutive F-functions from the 2nd round to the 15th round of 16-round DES, we have the following two equations that hold with probability $1/2 - 1.19 \times 2^{-21}$ for random plaintexts and their ciphertexts (figure 1 illustrates the detailed construction of equation (4)):

$$\begin{aligned} & P_H[7, 18, 24] \oplus F_1(P_L, K_1)[7, 18, 24] \oplus C_H[15] \oplus C_L[7, 18, 24, 29] \oplus \\ &\quad F_{16}(C_L, K_{16})[15] \\ &= K_3[22] \oplus K_4[44] \oplus K_5[22] \oplus K_7[22] \oplus K_8[44] \oplus K_9[22] \oplus K_{11}[22] \oplus \\ &\quad K_{12}[44] \oplus K_{13}[22] \oplus K_{15}[22], \end{aligned} \tag{4}$$

$$\begin{aligned} & C_H[7, 18, 24] \oplus F_{16}(C_L, K_{16})[7, 18, 24] \oplus P_H[15] \oplus P_L[7, 18, 24, 29] \oplus \\ &\quad F_1(P_L, K_1)[15] \\ &= K_{14}[22] \oplus K_{13}[44] \oplus K_{12}[22] \oplus K_{10}[22] \ \oplus K_9[44] \oplus K_8[22] \oplus K_6[22] \oplus \\ &\quad K_5[44] \oplus K_4[22] \oplus K_2[22]. \end{aligned} \tag{5}$$

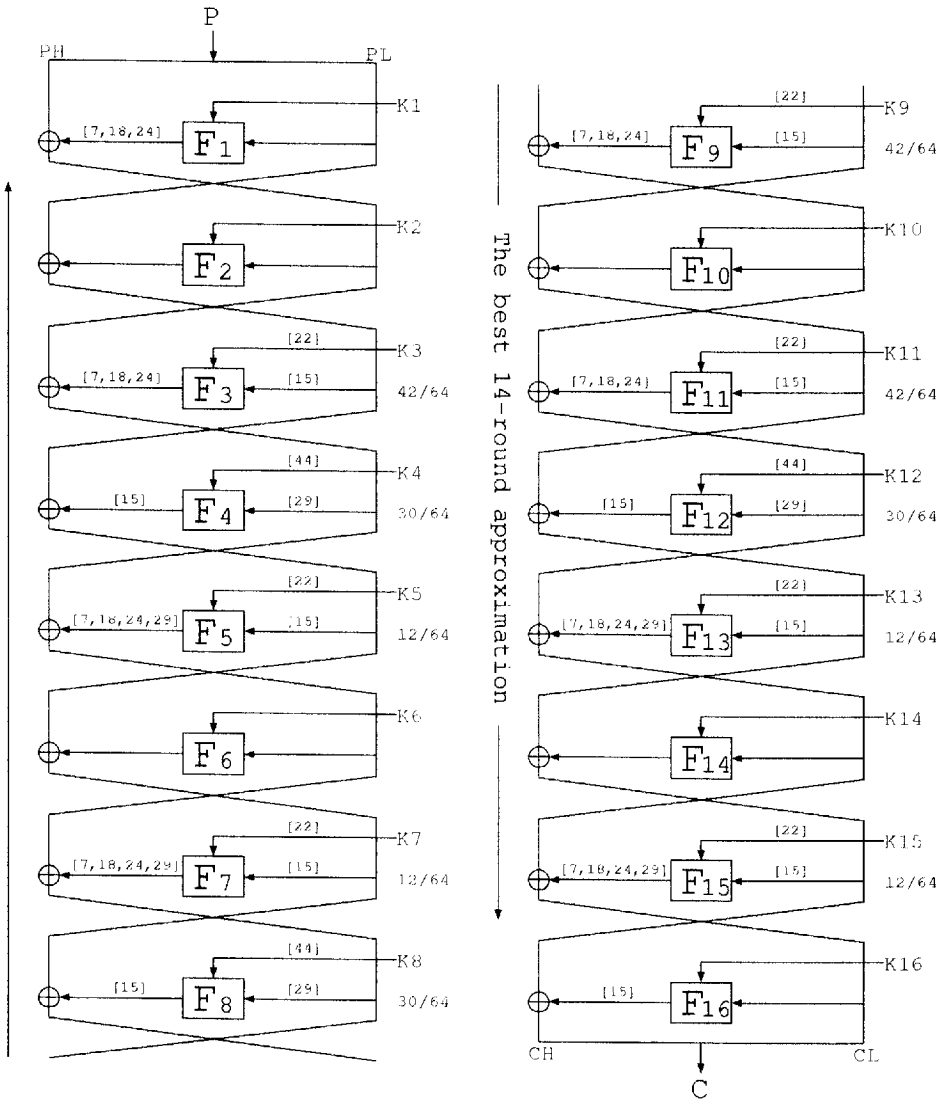

Fig. 1. New linear approximation of 16-round DES.

The first stage of our attack is to solve these equations to derive some of the 56 secret key bits. Now let us consider how much memory is required to solve them and how many secret key bits can be derived from them. For this purpose, we here define "effective text bits" and "effective key bits" of equation (4) or (5) as the text bits and the key bits which affect the left side of each equation, respectively. If an XORed value of several text/key bits affects the left side, we count as one effective text/key bit. Then the following can be easily seen:

- The effective text bits of equation (4) (13 bits):
  $P_L[11], P_L[12], P_L[13], P_L[14], P_L[15], P_L[16], C_L[0], C_L[27], C_L[28],$
  $C_L[29], C_L[30], C_L[31], P_H[7, 18, 24] \oplus C_H[15] \oplus C_L[7, 18, 24, 29]$.
- The effective key bits of equation (4) (12 bits):
  $K_1[18], K_1[19], K_1[20], K_1[21], K_1[22], K_1[23],$
  $K_{16}[42], K_{16}[43], K_{16}[44], K_{16}[45], K_{16}[46], K_{16}[47]$.

- The effective text bits of equation (5) (13 bits):
  $C_L[11], C_L[12], C_L[13], C_L[14], C_L[15], C_L[16], P_L[0], P_L[27], P_L[28],$
  $P_L[29], P_L[30], P_L[31], C_H[7, 18, 24] \oplus P_H[15] \oplus P_L[7, 18, 24, 29]$.
- The effective key bits of equation (5) (12 bits):
  $K_{16}[18], K_{16}[19], K_{16}[20], K_{16}[21], K_{16}[22], K_{16}[23],$
  $K_1[42], K_1[43], K_1[44], K_1[45], K_1[46], K_1[47]$.

Note that $P_H[7, 18, 24] \oplus C_H[15] \oplus C_L[7, 18, 24, 29]$ and $C_H[7, 18, 24] \oplus P_H[15] \oplus P_L[7, 18, 24, 29]$ represent one-bit information. This observation shows that 13 secret key bits — 12 effective key bits and one bit of the right side — can be derived from each equation using just 13 text bits. We hence obtain a total of 26 secret key bits — they are not duplicated — from equations (4) and (5) using information on 26 text bits.

Let us next consider how to solve these equations. If one substitutes an incorrect key value for K_1 or K_{16} in equation (4) or (5), the probability that the left side equals zero is expected to be closer to 1/2 (not necessarily 1/2). This leads us to maximum likelihood method in regard to key candidates; for each key candidate, we count the number of times that the left side of the equation equals zero. Then, the resultant counter value must reflect the reliability of the corresponding key candidate. We have implemented this scenario as follows:

Algorithm for breaking 16-round DES (I)

Data Counting Phase

Step 1 Prepare 2^{13} counters TA_{t_A} $(0 \leq t_A < 2^{13})$ and initialize them by zeros, where t_A corresponds to each value on 13 effective text bits of equation (4).

Step 2 For each plaintext P and the corresponding ciphertext C, compute the value 't_A' of **Step 1**, and count up the TA_{t_A} by one.

Key Counting Phase

Step 3 Prepare 2^{12} counters KA_{k_A} $(0 \leq k_A < 2^{12})$ and initialize them by zeros, where k_A corresponds to each value on 12 effective key bits of equation (4).

Step 4 For each k_A of **Step 3**, let KA_{k_A} be the sum of TA_{t_A}'s such that the left side of equation (4), whose value can be uniquely determined by t_A and k_A, is equal to zero.

Step 5 Rearrange KA_{k_A} in order of magnitude of $|KA_{k_A} - N/2|$ and rename them $\overline{KA}_{l_A}$ $(0 \leq l_A < 2^{12})$. Then, for each l_A,

- If $(\overline{KA}_{l_A} - N/2) \leq 0$, guess that the right side of equation (4) is 0.
  If $(\overline{KA}_{l_A} - N/2) > 0$, guess that the right side of equation (4) is 1.

At this stage, the key candidate corresponding to $\overline{KA}_{l_A}$ represents the l_A-th likely 13 secret key bits. The total size of required counters is $2^{13} + 2^{12}$, and the computational complexity, which depends on **Step 2** only, is $O(N)$. Note that **Step 2** is parallelizable.

Equation (5) can be also solved in the same manner, in which case we will use the notations TB_{t_B}, KB_{k_B} and $\overline{KB}_{l_B}$ instead of TA_{t_A}, KA_{k_A} and $\overline{KA}_{l_A}$. Our algorithm recovers, therefore, a total of 26 secret key bits, whose bit locations (after the PC–1) are as follows:

$K[0], K[1], K[3], K[4], K[8], K[9], K[14], K[15], K[18], K[19], K[24], K[25], K[31],$
$K[32], K[38], K[39], K[41], K[42], K[44], K[45], K[50], K[51], K[54], K[55],$
$K[5] \oplus K[13] \oplus K[17] \oplus K[20] \oplus K[46],$
$K[2] \oplus K[7] \oplus K[11] \oplus K[22] \oplus K[26] \oplus K[37] \oplus K[52].$

The next stage of our attack is to derive the remaining $56 - 26 = 30$ secret key bits. Our aim is to increase the success rate by repeating the search in order of reliability of 26 secret key bits. In other words, we want to make the following algorithm work effectively:

Algorithm for breaking 16-round DES (II)

Exhaustive Search Phase

Step 6 Let W_m $(m = 0, 1, 2, ...)$ be a series of candidates for the 26 secret key bits arranged in order of their reliability.

Step 7 For each W_m, search for the remaining 30 secret key bits until the correct value is found.

Now we have to describe W_m explicitly by l_A and l_B. Since the most likely candidate for the 26 key bits clearly corresponds to $\overline{KA}_0$ and $\overline{KB}_0$, we should consider this combination at first, which will be referred to as $W_0 = (\overline{KA}_0, \overline{KB}_0)$. The second likely candidates are obviously $W_1 = (\overline{KA}_0, \overline{KB}_1)$ and $W_2 = (\overline{KA}_1, \overline{KB}_0)$ with the same reliability. Then, are the next likely ones $W_3 = (\overline{KA}_0, \overline{KB}_2)$ and $W_4 = (\overline{KA}_2, \overline{KB}_0)$, or $W_3 = (\overline{KA}_1, \overline{KB}_1)$? How many candidates are needed to finish **Step 7** in reasonable time ? In the next chapter we will give a practical solution of these problems.

4 Success Rate and Complexity

We relate the problems to DES reduced to 8 rounds, which will be referred to as "8-round DES" below. Now consider the following two equations of 8-round DES derived from the best 6-round expression which holds with probability $1/2 - 1.95 \times 2^{-9}$:

$$P_H[7, 18, 24] \oplus F_1(P_L, K_1)[7, 18, 24] \oplus C_H[15] \oplus C_L[7, 18, 24, 29] \oplus F_8(C_L, K_8)[15] = K_3[22] \oplus K_4[44] \oplus K_5[22] \oplus K_7[22], \quad (6)$$

$$C_H[7,18,24] \oplus F_8(C_L, K_8)[7,18,24] \oplus P_H[15] \oplus P_L[7,18,24,29] \oplus$$
$$F_1(P_L, K_1)[15] = K_6[22] \oplus K_5[44] \oplus K_4[22] \oplus K_2[22]. \quad (7)$$

Note that the left side of each equation is essentially the same as equation (4) or (5), respectively. We make use of this fact to evaluate the efficiency of our attack. The following lemma, which is an extension of lemma 4 in [2], relates the full 16-round DES to 8-round DES:

Lemma 1. *Let N be the number of given random plaintexts and p be the probability that the following equation holds:*

$$P[i_1, i_2, ..., i_a] \oplus C[j_1, j_2, ..., j_b] \oplus F_1(P, K_1)[u_1, u_2, ..., u_d] \oplus$$
$$F_n(C, K_n)[v_1, v_2, ..., v_e] = K[k_1, k_2, ..., k_c]. \quad (8)$$

Assuming $|p-1/2|$ is sufficiently small, the probability that the l-th likely solution of equation (8) agrees with the real key depends on l, $u_1, u_2, ..., u_d$, $v_1, v_2, ..., v_e$, and $\sqrt{N}|p-1/2|$ only.

This lemma tells us that the success rate of our attack on 8-round DES with N_8 plaintexts is the same as that on 16-round DES with N_{16} plaintexts as long as the following relation holds:

$$\sqrt{N_8}|1.95 \times 2^{-9}| = \sqrt{N_{16}}|1.19 \times 2^{-21}|. \quad (9)$$

This is equivalent to

$$1.49 \times 2^{-26} \times N_{16} = N_8, \quad (10)$$

and hence 2^{43} plaintexts on 16-round DES, for instance, correspond to 1.49×2^{17} plaintexts on 8-round DES.

Note: According to the common definition of 8-round DES, which adopts eight F-functions from the first to the eighth round of 16-round DES, equations (6) and (7) yield only 23 secret key bits because three of 26 bits are duplicated. To avoid this difference from the case of 16-round DES, this paper treats the 8-round DES whose key schedule part is modified so that no secret key bit is duplicated. Our computer experiments on 8-round DES below were carried out under this condition.

We made computer experiments in solving equation (6) 100,000 times to estimate the behavior of solutions of equation (4). Figure 2 illustrates the results interpreted as the case of 16-round DES, where the ordinate (y axis) shows the probability that the ranking of a solution of equation (4) is not greater than the value of the abscissa (x axis); for example, when we solve equation (4) with 2^{43} known plaintexts, the probability that the secret key agrees with one of $\overline{KA}_{l_A}$ ($0 \leq l_A < 100$) is expected to be 86%. The lowest curve represents the case where we select a key candidate randomly: namely, $y = x/2^{13}$.
Figure 3 summarizes our attack on the full 16-round DES, where the reliability of $W_m = (\overline{KA}_{l_A}, \overline{KB}_{l_B})$ has been determined in order of the magnitude of

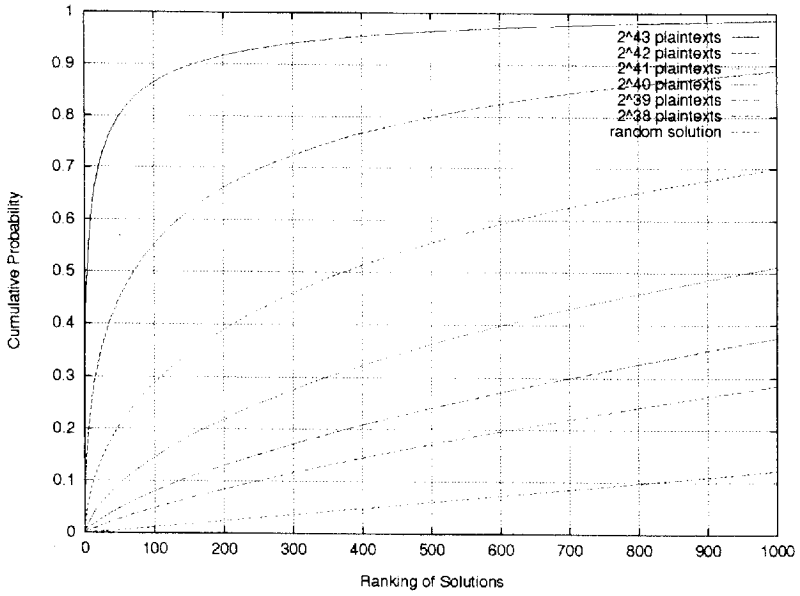

Fig. 2. Expected ranking of solutions of equation (4).

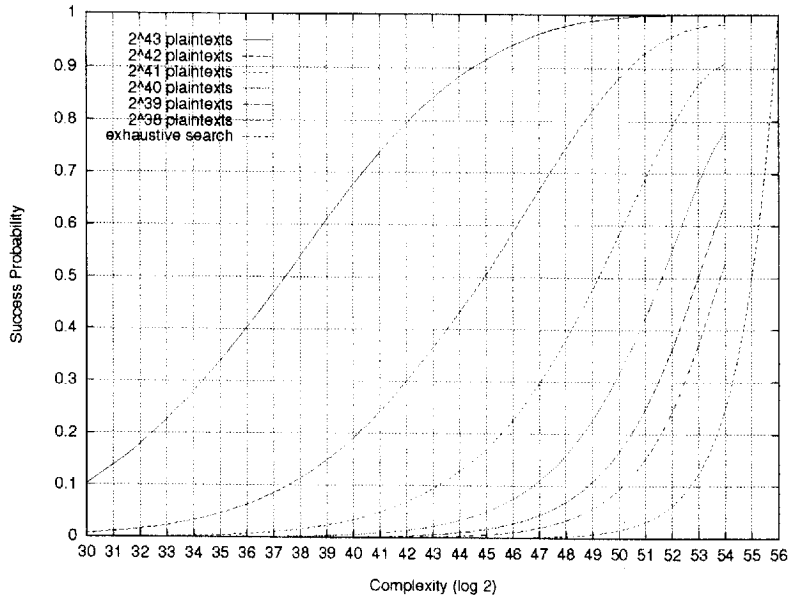

Fig. 3. Expected success rate and complexity of our attack on 16-round DES.

$(l_A+1)\times(l_B+1)$, which is the formula derived experimentally from the case of the 8-round DES. The abscissa and ordinate denote the computational complexity and the success probability, respectively. This figure tells us that when we attack the full 16-round DES with 2^{43} plaintexts, the probability that the secret key can be found within $m = 2^{13}$ (i.e. complexity $2^{30} \times 2^{13} = 2^{43}$), is expected to be 85%. For another example, the success probability is expected to be 10% with complexity 2^{50} if 2^{38} known plaintexts are available. The lowest curve represents the trivial case where we search for the 56 secret key bits exhaustively: $y = 2^{x-56}$.

5 The Computer Experiment

We made the first computer experiment in breaking the full 16-round DES on the basis of the above scenario. The program, implemented by software only, was described in C and assembly languages consisting of a total of 1000 lines. It occupies 1Mbyte in running. The main flow of the program is as follows (we use C-like notations):

```
for( i=0; i<2^43; i++ ){ /* parallelizable */
     P = Generate_Random_Plaintext();
     C = Encipher_Plaintext( P ); /* using the secret key K */
     TA[ 13bit_address_pointed_by_P_and_C ]++; /* Step 2 */
     TB[ 13bit_address_pointed_by_P_and_C ]++; /* Step 2 */
}

for( k=0; k<2^12; k++ ){ /* each value on effective key bits */
for( t=0; t<2^13; t++ ){ /* each value on effective text bits */
     if( Left_Side_of_Equation_4( t, k ) == 0 )
          KA[ k ] += TA[ t ]; /* Step 4 */
     if( Left_Side_of_Equation_5( t, k ) == 0 )
          KB[ k ] += TB[ t ]; /* Step 4 */
}
}

Rearrange_Counters( KA, KA-bar ); /* Step 5 */
Rearrange_Counters( KB, KB-bar ); /* Step 5 */

for( m=0; m<2^24; m++ ){ /* parallelizable */
     K26 = Derive_m_th_Likely_26bits( m, KA-bar, KB-bar ); /* Step 6 */
     Return_Value = Search_Remaining_30bits( K26 ); /* Step 7 */
     if( Return_Value == FOUND ) exit( SUCCESS );
}

exit( FAILURE ); /* theoretically possible
                    but practically unreachable */
```

We used a sequence $\{g^0, g^1, g^2, g^3, ...\}$ for `Generate_Random_Plaintext()` routine, where g is a generator of cyclic group $GF(2^{64})^{\times}$, which is convenient for our purpose, parallel computing. `Encipher_Plaintext()` is a routine for enciphering plaintexts under a fixed key, which runs at the rate of 19Mbit/sec. On the other hand, `Search_Remaining_30bits()` is also an enciphering routine but encodes a fixed plaintext under given keys, which runs at the rate of 9Mbit/sec. `Rearrange_Counters()` is a sorting routine for a 2^{12}-dimensional array. `Derive_m_th_Likely_26bits()` can be also easily implemented using the $(l_A + 1) \times (l_B + 1)$ rule.

Calculations of both the first and last loops were carried out in parallel by 12 computers. It took 40 days to finish the first loop, where almost all time was spent for `Encipher_Plaintext()` routine. The middle loop and the sorting routine were easily executed. The last loop took 10 days and finally resulted in all of the 56 secret key bits.

6 Concluding Remarks

We have described an improvement of linear cryptanalysis and presented the first successful experiment in breaking the full 16-round DES. The topics below are remarks and possible further improvements.

- The author does not know whether **Step 1** $\sim$ **Step 5** give the best way for solving equations (4) and (5). It should be noted that we have not made use of all information available from these equations: to be concrete, when we substitute $K_1'(\neq K_1)$ and $K_{16}'(\neq K_{16})$ for K_1 and K_{16} in the left side of equation (4), the probability that the equation holds depends on only $K_1 \oplus K_1'$ and $K_{16} \oplus K_{16}'$. This fact obviously indicates more than what we have realized in this paper. Therefore if this property could be used effectively, the reliability of the solution might be improved.
- In this paper, we have solved two equations to obtain 26 key bits and then searched for the remaining 30 key bits. However, it is also possible to solve more equations to have more key bits before the search procedure (**Steps 6** and **7**). For example, there are two second best expressions that hold with probability $1/2 - 1.49 \times 2^{-21}$. Although the reliability of these solutions is lower, this loss might be recoverable by repeating the search procedure, because the number of the remaining key bits is then smaller.
- The results on figure 2 and figure 3 have been derived experimentally. If we succeed in tracing curves in figure 2 with simple functions, figure 3 can be also formalized and then a new combination rule will give more effective results instead of the $(l_A + 1) \times (l_B + 1)$ rule.

More detailed discussion including experimental data, which we have omitted due to lack of space, will appear in the full paper.

References

1. National Bureau of Standards: Data Encryption Standard. U.S. Department of Commerce, Federal Information Processing Standards **46** (1977)
2. Matsui, M.: Linear Cryptanalysis Method for DES cipher. Advances in Cryptology - Eurocrypt'93, Lecture Notes in Computer Science, Springer-Verlag **765** (1993) 386–397
3. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. Pre-proceedings of Eurocrypt'94 (1994) 375 387
4. Hellman, M., Merkle, R., Schroeppel, R., Washington, L., Diffie, W., Pohlig, S., Schweitzer, P.: Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard. Information Systems Laboratory, Stanford University **76-042** (1976)
5. Shamir, A.: On the security of DES. Advances in Cryptology – Crypto'85, Lecture Notes in Computer Science, Springer-Verlag **218** (1985) 280 281
6. Davies, D., Murphy, S.: Pairs and triplets of DES s-boxes. (preprint)
7. Rueppel, R.A.,: Analysis and design of stream ciphers. Springer Verlag (1986)