REPORTS IN INFORMATICS

ISSN 0333-3590

Linear Known Plaintext Attack on DES

John Erik Mathiassen
and
Lars R. Knudsen

REPORT NO 274 June 2004

Department of Informatics
UNIVERSITY OF BERGEN
Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2004-274.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Linear Known Plaintext Attack on DES

John Erik Mathiassen
The Selmer Senter*
Institute for informatics
University of Bergen
Norway

Lars R. Knudsen
Department of Matematics
Technical University of Denmark
Denmark

1st June 2004

Abstract

This paper presents a known and a chosen plaintext attack on the DES both of success rate 80% using 2^{42} texts. The attacks improve on earlier results in that the total complexity is reduced by a factor of two both in the number of texts needed and in the number of computations required.

Keywords: Linear cryptanalysis, Cryptanalysis, Block cipher, DES, key recovery.

1 Introduction

The block cipher DES [1] is a Feistel cipher and was adopted as an US standard by NBS (now NIST) in 1977. It encrypts text of 64 bits using a (secret) 64 bits key, but where only 56 bits are used and the remaining 8 bits are discarded. An exhaustive search for the key would require at most2^{56} DES computations, but there has been some shortcut attacks involving less DES computations still finding the key.

In [2] the differential attack is introduced. The differential attack is a chosen plaintext attack which makes it possible to find the key in DES using 2^{47} chosen plaintexts. It was the first attack on DES which successfully can recover the secret key faster than by brute force.

The linear attack was first introduced on FEAL [3] and later adapted to DES [4, 5]. The attack is a known plaintext attack and applied to DES finds the key using some 2^{43} known plaintexts. To this date this is the best attack on DES in open literature. Despite the introduction of the new encryption standard AES, DES is still widely used. Other related articles not introduced in the rest of the text are [6, 7, 8].

2 The Linear Attack

In this section the general linear attack [4] is presented and applied to block ciphers. The ciphertext C is the encryption of the plaintext P - both of size n - by an encryption function $C = E_K(P) = E(P, K)$ using the key K of size n_K bits. Most

*This work was supported by the Norwegian Research Council.

block ciphers repeat a simple round function $C_{i+1} = G(C_i, K_i)$, and maps the ciphertext C_i after i rounds of encryption to C_{i+1} using a round key K_i.

Preparing a linear attack one starts by finding one round approximations for the cipher in question. An equation for such an approximation is:

$$(C_i \cdot \alpha_i) \oplus (C_{i+1} \cdot \beta_i) = (K_i \cdot \gamma_i)$$

where the "$\cdot$" denotes the dot product, and the Greek letters denote masks. The sum of some input bits plus the sum of some output bits is equal to the sum of some key bits with a probability p_i. In linear attacks one is often interested in $e_i = (p_i - 1/2)$ which is the imbalance. The size of the absolute value $|e_i|$ indicates the strength of an equation.

The single round approximations are concatenated to a r-round approximation

$$(P \cdot \alpha) \oplus (C \cdot \beta) = (K \cdot \gamma) \tag{1}$$

from the first round to the last round. The concatenations and the probability calculations assume that the rounds are independent, and it works for many ciphers. The calculation of the probability of success p is done using piling up Lemma

$$p = \frac{1}{2} + 2^{r-1} \prod_{i=1}^{r} (p_i - \frac{1}{2}) = \frac{1}{2} + 2^{r-1} \prod_{i=1}^{r} e_i$$

assuming independent rounds.

When using enough (P, C) pairs we have an indication on the value of $K \cdot \gamma \in \{0, 1\}$ with a high probability. Using the normal distribution function $\Phi(2\sqrt{N}|e|)$ we get that $N = (p - 1/2)^{-2} = e^{-2}$ is the expected number of (P, C)-pairs needed to give the value of $K \cdot \gamma$ with a probability of $97,72\%$. Half the possible keys $K \in \mathbf{K}$ will satisfy this value, and we have one bit information. Assume we use N text pairs in an attack. Let T be the number of times the left side of Equation 1 is 0, and let $U = T - N/2$. Then $K \cdot \gamma$ is guessed to be

$$K \cdot \gamma = \frac{sign(U) \cdot sign(e) + 1}{2}$$

which says that $K \cdot \gamma = 0$ if both $(T - N/2)$ and $(p - 1/2)$ have the same sign, and $K \cdot \gamma = 1$ if they have different sign, where $sign(x) = \frac{x}{|x|}$. The rest of the bits may be found by other similar equations, or by exhaustive key search.

A more efficient method is described by Matsui in [5]. Using this method we may shorten our approximation by two rounds, in addition to find more key bits using one equation. A way to do it is to find a good approximation

$$(C_1 \cdot \alpha) \oplus (C_{r-1} \cdot \beta) = (K \cdot \gamma)$$

from the second round to the second last round having probability $p = \frac{1}{2} + e$, where $|e| > 0$. Notice that $C_1 = G(P, K_1)$ and $C_{r-1} = G^{-1}(C, K_r)$, but this require us to know or guess K_1 and K_r. This is not efficient if the round keys K_i are close to n bits, which is the block size. Most ciphers use S-boxes, which are typically mappings of a small number of bits, e.g. 4 or 8. This allows us to guess the key bits input to the S-boxes involved in the equation

$$(G(P, K_1^*) \cdot \alpha) \oplus (G^{-1}(C, K_r^*) \cdot \beta) = (K \cdot \gamma) \tag{2}$$

where '*' indicates that K_i^* is the effective key bits of K_i involved in $G(X, K_i) \cdot \alpha$. If the S-box size is s bits, and there is only one S-box involved in the first and the

last round, one needs to guess $2s$ key bits. This is the case in Matsui's attack where the S-box size is $s = 6$ and one needs to guess $2s = 12$ key bits in each equation.

The Equation 2 will have probability of p of being correct if the correct sub key $K^*_{1,r} = (K^*_1, K^*_r)$ is used. If the imbalance $e = p - 1/2 > 0$ is greater than 0 the Equation 2 will follow this imbalance. Using N plaintext/ciphertext pairs $T^{K^{g*}_{1,r}}$ counts how many times the left side of Equation 2 is equal to 0 using the key $K^{g*}_{1,r}$ in the equation. We then have one counter for all possible sub-key guess $K^{g*}_{1,r} \in \mathbf{K}^*_{1,r}$. Let the bias of the counter be $U^{K^*_{1,r}} = T^{K^*_{1,r}} - \frac{N}{2}$, then hopefully the bias of the counter for the correct sub-key $U^{K^*_{1,r}}$ is greater that the imbalance for all the wrong guesses $K^{w*}_{1,r} \in \mathbf{K}^*_{1,r} \backslash K^*_{1,r}$. That is $\left|U^{K^*_{1,r}}\right| > \left|U^{K^{w*}_{1,r}}\right|$. The sub-key $K^*_{1,r}$ according to $\left|U^{K^*_{1,r}}\right| > \left|U^{K^{g*}_{1,r}}\right|$ for all possible sub-keys $K^{g*}_{1,r}$ is guessed to be the correct one.

Then we perform an exhaustive search on the rest of the bits, but during the exhaustive search we only try the keys where

$$K \cdot \gamma = \frac{sign(U^{K^*_{1,r}}) \cdot sign(e) + 1}{2}$$

which will be the case for half the keys, and we have $2s + 1$ key bits from our linear equation.

If the key shows to be wrong after the exhaustive search for the rest of the key bits, one is able to select the key corresponding to the second best value of $U^{K^{g*}_{1,r}}$, and continue until the correct key is found.

It is harder to predict the accurate success rate of this method, because it is not enough to calculate the probability that $T > \frac{N}{2} \Leftrightarrow U > 0$ given that $K \cdot \gamma = 0$ and $e > 0$ which by the normal distribution function is approximated to be $97,72\%$ using $N = e^{-2}$ (P, C)-pairs. The approximation of the success rate works fine in most cases, but there are at least two reasons why this simple approximation using the normal distribution is not accurate. For the first the counter for the correct sub key does not compete with one threashold $U^{K^*_{1,r}} > 0$, but all the other counters $\left|U^{K^*_{1,r}}\right| > \left|U^{K^{w*}_{1,r}}\right|$, for all wrong keys $K^{w*}_{1,r} = K^*_{1,r} \oplus w$ $w \neq 0$, where w is of same size as the keys. It is usually satisfying to include a constant $1 < c < 8$ to the prediction of the number of texts needed $N = c \cdot e^{-2}$, and we need more texts to keep the success rate at $97,72\%$.

3 Previous improvements

In the past ten years there has been some improvements of Matsui's attack and some additional techniques have been introduced some of which are presented here.

Multiple linear approximations [9]

Using multiple linear equations [9] is more powerful than a single equation. The equations should have the same key bits involved and the same S-boxes involved in the outer rounds. The equations used should also have a bias e approximately the same and greater than 0.

We have q equations

$$(G(P, K_1) \cdot \alpha_i) \oplus (G^{-1}(C, K_r) \cdot \beta_i) = (K \cdot \gamma) \tag{3}$$

where all the q mask pairs (α_i, β_i) are different. Otherwise we have some equal equation. Each equation have the probability $p_i = \frac{1}{2} + e_i$, and the counters for each possible sub-key $K_{1,r}^{g*}$ should be $U^{K_{1,r}^{g*}} = \sum a_i U_i^{K_{1,r}^{g*}}$, where the weights a_i are calculated

$$a_i = \frac{e_i}{\sum_{j=1}^{q} e_i}$$

and the $U^{K_{1,r}^*}$ with the greatest absolute value $\left|U^{K_{1,r}^*}\right| > \left|U^{K_{1,r}^{g*}}\right|$ is guessed as the key value and $K \cdot \gamma = \frac{sign(U^{K_{1,r}^*}) \cdot sign(e)+1}{2}$ is guessed to be the right side of Equation 3. One advantage of using multiple equations is that the variance is reduced, so we need less plaintexts in order to keep the same success rate. Having q equations with identical bias e an expected success rate of $97,72\%$ requires $N_M = c_M \cdot q^{-1} \cdot e^{-2} = N/q$ (P,C) pairs, where N_M is the number pairs in the multiple case and N is the required number in the single equation case. The noise factor c_M is somewhat more complicated, but is due to the weighting of the counters a weighted average over the constants c_i for the different equations.

Optimal key ranking [10]

Due to the symmetry of Feistel-ciphers it is possible to exchange the first and the last round masks, and have an equation involving different key bits. In Matsui's attack these two sets of bits are disjunct. The challenge here is to combine these two ranked key tables in order to find the correct combination in an optimal way. An optimal way of combining these two is described in [10].

The Equation 2 may be used in combination with

$$(G(P, K_1^*) \cdot \beta) \oplus (G^{-1}(C, K_r^*) \cdot \alpha) = (K \cdot \gamma') \tag{4}$$

where γ' is different from γ in Equation (2). Last section we saw that if $\gamma = \gamma'$, and if α and β involve the same S-box we have multiple equations, and the number of texts required for success is reduced. Notice also since different S-boxes are involved in Equation (2) and (4) one may have a disjoint set of key bits involved, and one gets two independent key ranking tables. Junod *et al* proved that to get an optimal ranking one should sort the sub-key candidates by a decreasing sum of squares of the biases. That is, for each equation one has a counter T_1 and T_2, and each possible key K_i^{g*} has its own counter $T_i^{K_{1,r}^{g*}}$. The two tables are then sorted by decreasing values $(T_i^{K_i^{g*}} - 1/2)^2$, and the sub-key candidate K^{g*} with the highest sum $U^{K^{g*}} = (T_1^{K_1^{g*}} - 1/2)^2 + (T_2^{K_2^{h*}} - 1/2)^2$, where $K^{g*} = (K_1^{g*}, K_2^{h*})$, is ranked the most likely key. This is optimal, and slightly better that previously used methods: $U^k = R_1(U^{k_1}) \cdot R_2(U^{k_2})$, where $R_i(U^{k_i})$ is the ranking of the key k_i in the table of sorted $\left|U_1^{K_{1,r}^{g*}}\right| = \left|T_1^{K_{1,r}^{g*}} - N/2\right|$.

Some other improvements on the linear attack

[11] introduced non-linear approximations in linear cryptanalysis. In the outer rounds the linear approximations are replaced by non-linear, and there was an improvements in a five rounds attack on DES recovering one bit. There was no significant improvements on the full rounds attack. [12] introduced an attack using a quadratic relation in S-box no. 5 to improve the linear attack. They use Matsui's

original linear relation in addition to a quadratic relation to increase the effectiveness of the attack by reducing the number of texts required by a factor 25/34.

There is also a chosen plaintext attack presented in [13] where the success rate is good, but the exhaustive search phase requires 2^{44} computations to succeed. In the next section we show how to reduce this complexity by a factor 4 by using the full potential of the effective key bits in the equation used.

Another efficient attack using both differential and linear techniques is presented in [14, 15]. It seems quite effective on reduced rounds DES, but is not so effective on the full DES. We are not going into details of these improvements of the linear attack, since these methods are not used in this article.

4 Improvements in the Linear Attack on DES

In this chapter we present what we believe are the two most effective attacks on DES due to complexity. First one we improve the performance of the chosen plaintext attack presented in [13]. Secondly, we present a known plaintext attack which uses Matsui's equations combined with some new equations.

Chosen plaintext linear attack

This attack uses what is called pseudo keys to be able to reduce the number of approximated rounds by one. The probability of success increases and one finds more key bits. We will start by explaining some simpler techniques to explain the more complicated case. Instead of having a 14 round approximation from the 2nd to the (r-1)st round, we use a 13 round approximation from the 3rd to the (r-1)st round:

Original approximation							Our Approximation					
2:	-	-	-									
3:	A	$\leftarrow$	D	$\searrow$			-	-	-			
4:	D	$\leftarrow$	A$\oplus$B	$\searrow$			A	$\leftarrow$	D			
5:	B	$\leftarrow$	D	$\searrow$			D	$\leftarrow$	A$\oplus$B			
6:	-	-	-	$\searrow$			B	$\leftarrow$	D			
7:	B	$\leftarrow$	D	$\searrow$			-	-	-			
8:	D	$\leftarrow$	A$\oplus$B	$\searrow$			B	$\leftarrow$	D			
9:	A	$\leftarrow$	D	$\searrow$			D	$\leftarrow$	A$\oplus$B			
10:	-	-	-	$\searrow$			A	$\leftarrow$	D			
11:	A	$\leftarrow$	D	$\searrow$			-	-	-			
12:	D	$\leftarrow$	A$\oplus$B	$\searrow$			A	$\leftarrow$	D			
13:	B	$\leftarrow$	D	$\searrow$			D	$\leftarrow$	A$\oplus$B			
14:	-	-	-	$\searrow$			B	$\leftarrow$	D			
15:	B	$\leftarrow$	D				-	-	-			

The probability of the approximations are:

Figure 1: Detailed description of the first round trick. The plaintext is divided in two halves $P = (P^L, P^R)$. The figure illustrates in detail in which order and how the different operations influence the input of S-box 5 in the second round.

Approximation	S-boxes	Probability	Masks in hex
A $\leftarrow$ D	S_5	$\frac{1}{2} + \frac{10}{64}$	$E_x \leftarrow 10_x$
B $\leftarrow$ D	S_5	$\frac{1}{2} - \frac{20}{64}$	$F_x \leftarrow 10_x$
B $\leftarrow$ D'	S_5	$\frac{1}{2} - \frac{12}{64}$	$F_x \leftarrow 22_x$
A $\leftarrow$ D'	S_5	$\frac{1}{2} - \frac{16}{64}$	$E_x \leftarrow 22_x$
D $\leftarrow$ A$\oplus$B	S_1	$\frac{1}{2} - \frac{2}{64}$	$4_x \leftarrow 04_x$

The full equation where one guesses a pseudo-key in the second round is:

$$(P^R \cdot A) \oplus (F_5(P^L, F_6(P^R, K_1^*) \oplus K_2^{'*}) \cdot A) \oplus (C^L \cdot B) \oplus (F_5(C^R, K_{16}^*) \cdot B) = K \cdot \gamma \quad (5)$$

where $K_2^{'*}$ is a pseudo key, and K_1^* and K_{16}^* are the involved key bits from the round key in round 1 and 16 respectively. The subindex i in F_i refers to which S-box is involved in the approximation. That is, K_{16}^* in $F_5(C^R, K_{16}^*)$ refers to the key bits from K_{16} which are involved in S-box 5 in round 16. We will explain how to interpret this equation. In principle one approximates the sum of some output bits from S-box 5 in the second round plus some active key bits ($K \cdot \gamma$) and the sum of some of the output bits from S-box 5 in the last round. One knows the plaintext and the corresponding ciphertext, but not the key K nor the round keys. To get the value of $F_1(C^R, K_{16}^*) \cdot B$ one needs to know the input to the S-box 5. The text part C^R is known, but the 6 key bits K_{16}^* are unknown, so they are guessed. This is simple and has been done before, the problem is to find the input to the S-box 5 in the second round. It is called the pseudo key trick, and is shown by Figure 1.

The six input bits come from six different S-boxes in the first round $\{S_3,\ S_1,\ S_2,\ S_6,\ S_4,\ S_8\}$, presented according to the actual permuted order. To guess all the unknown key bits input to all these functions will involve too many effective key and text bits. The unknown input to the S-box 5 in the second round is denoted $x_2 = y_1(P^R, K_1) \oplus K_2^* \oplus P^L$ where $y_1 = (s_3, s_1, s_2, s_6, s_4, s_8)$ is the output bits from the S-boxes which give input to S-box 5 in the second round. If the input to these S-boxes involved in y_1 is fixed, one has a fixed, but unknown value y_1. The fixed value of $K_2^{'*} = K_2^* \oplus y_1$ could be guessed instead. In that case the equation is

$$(P^L \cdot A) \oplus (F_5(P^L, K_2^{'*}) \cdot A) \oplus (C^L \cdot B) \oplus (F_5(C^R, K_{16}^*) \cdot B) = K \cdot \gamma, \quad (6)$$

which involves guessing only twelve key bits, and has only 13 effective text bits. The problem of keeping all the input bits to six S-boxes fixed is, that there are only 36 bits left to vary, and one does not have enough plaintexts to attack the DES. Of all these six S-boxes the sixth has a special property: Varying all six bits of this S-box does not influence the neighboring S-boxes. The other S-boxes have the property that varying all the bits will also vary the input to a neighboring S-box because of the expansion function E. We then redefine $y_1 = (s_3, s_1, s_2, 0, s_4, s_8)$ such that it is still fixed and $K_2^{'*} = K_2^* \oplus y_1$ is still fixed, but $x_2 = y_1(P^R, K_1) \oplus K_2^* \oplus F_6(P^R, K_1^*) \oplus P^L = K_2^{'*} \oplus F_6(P^R, K_1^*) \oplus P^L$, where $F_6(P^R, K_1^*) = (0, 0, 0, s_6, 0, 0)$ and P^L is varying. This explains the Equation (5).

Table 1: Comparison between Matsui's attack, the chosen plaintext attack from [13] and the new attack using 100 simulations on 8 rounds of DES. It shows a reduction of a factor 4 in the exhaustive search computation compared to the previous attack, and a reduction of a factor 2 in the number of needed plaintexts in comparison to Matsui's attack.

	Matsui's attack		previous attack		new attack	
plaintexts	2^{42}	2^{43}	2^{41}	2^{42}	2^{41}	2^{42}
Success rate	30%	85%	32%	86%	31%	90%
Ex.key search	2^{42}	2^{43}	2^{44}	2^{44}	2^{42}	2^{42}

Table 2: Results of chosen plaintext attack on the DES using 2^42 plaintexts. These results is data from 10 attacks on the DES. It shows that the simulations on 8 rounds of DES work as expected. The tests were run on a single 2 GHz Pentium 4 in less than two weeks. The exhaustive key search with complexity 2^{41} will take another two weeks on a single PC.

Ex. search	Success rate
2^{41}	80%
2^{43}	90%
2^{45}	100%

The probability $p_1 = 1/2 + e_1 = 1/2 + 16 * 5^6/2^{38}$ of this equation is slightly higher biased than that of Matsui. The predicted number of plaintexts needed is:

$$N = c * (p_1 - \frac{1}{2})^{-2} = c * e_1^{-2} = c * 1.10 * 2^{40},$$

where c is a constant between 1 and 8. The constant c is dependent on a correlation between the equation for correct key and the equation applying the wrong ones. The prediction of the original attack is $N = c_M * 2.81 * 2^{40}$, where the constant c_M is different from c. In the attack one uses multiple equations [9] having exactly the same key bits involved, and thus reducing the number of texts needed. The approximation

$$(P^L \cdot A) \oplus (F_5(P^R, F_6(P^R, K_1^*) \oplus K_2'^*) \cdot A) \oplus (C^L \cdot B) \oplus (F_5(C^R, K_{16}^*) \cdot B) = K \cdot \gamma \quad (7)$$

involves exactly the same key bits as Equation (12) and have the same probability $p_2 = p_1$. The predicted number of texts needed is then

$$N = c * (e_1^2 + e_2^2)^{-1} = c * 1.10 * 2^{39},$$

which is a reduction by a factor 2 compared to only using a single equation. This looks like an improvement by a factor of 5.12, but noise from the wrong keys are greater in this attack, since one guesses 15 key bits instead of 12 key bits, and because of the first round trick.

The results in the Tables 1 and 2 show an overall factor 2 improvement over Matsui's attack including the exhaustive search for the remaining key bits. The reason why one does not see a greater improvement is that the wrong keys have a high correlation with the correct key, which leads to a greater value of the constant c.

Known plaintext linear attack

Here we first present some new equation for use in the linear attack. Then this equation is combined with Matsui's original equation in order to have an improved attack.

The attack uses Matsui's equations:

$$(P^L \cdot A) \oplus (F_5(P^R, K_1^*) \cdot A) \oplus (C^R \cdot B) \oplus (C^L \cdot D) \oplus (F_1(C^R, K_{16}^*) \cdot D) = K \cdot \gamma_1 \quad (8)$$

and

$$(C^L \cdot A) \oplus (F_5(C^R, K_{16}^*) \cdot A) \oplus (P^R \cdot B) \oplus (P^L \cdot D) \oplus (F_1(P^R, K_1^*) \cdot D) = K \cdot \gamma_2, \quad (9)$$

where the index of F_i indicates that S-box i is involved in the function. Equations (8) and (9) have the same probability since the same approximations are included, however in different rounds. The probability of the equations is $\frac{1}{2} + 10 \cdot \frac{5^6}{2^{38}}$. Notice also that the right side of the equations involve different key bits.

We use also the following two equations:

$$(P^L \cdot B) \oplus (F_5(P^R, K_1^*) \cdot B) \oplus (C^R \cdot A) \oplus (C^L \cdot D) \oplus (F_1(C^R, K_{16}^*) \cdot D) = K \cdot \gamma_1 \quad (10)$$

and

$$(C^L \cdot B) \oplus (F_5(C^R, K_{16}^*) \cdot B) \oplus (P^R \cdot A) \oplus (P^L \cdot D) \oplus (F_1(P^R, K_1^*) \cdot D) = K \cdot \gamma_2, \quad (11)$$

where the right side of Equations (10) and (11) are equal to those of Equations (8) and (9), respectively. The probability of these equations is $\frac{1}{2} + 5 \cdot \frac{5^6}{2^{38}}$. The Equations (8) and (10) also involve the same sub key bits in the functions. The same applies to the other two. We use several such pairs of equations where the involved key bits are exactly the same, and call them twin equations. One might calculate a new probability of success for the twin by the formula $p = \frac{1}{2} + \sqrt{(p_1 - \frac{1}{2})^2 + (p_2 - \frac{1}{2})^2}$, so the imbalance is expected to be $e = \sqrt{e_1^2 + e_2^2}$. The predicted number of text pairs needed to succeed with $97,72\%$ is $N = c \cdot e^{-2} = c/(e_1^2 + e_2^2)$. The number of text pairs required to get $97,72\%$ success using one of the two twin pairs is

$$N_1 \quad = \quad N_2 = \frac{c_1 \cdot 2^{76}}{5^{12} \cdot (5^2 + 10^2)} = \frac{c_1 \cdot 2^{76}}{5^{15}},$$

so the gain compared to using a single equation is $R = 2^{76} \cdot 5^{14}/2^{74} \cdot 5^{15} = 4/5$. Using both twin pairs is equivalent to what Matsui does when he use two single equations, which is a way to double the numbers of bits found. But we may use other equations to find more key bits. We found another set of equations which can be used in advantage to get more key bits.

These two approximations are twin pairs. The approximations involve S-box 5 in the first round and S-box 3 and S-box 4 in the last round. The equation of the first approximation is:

$$(P^L \cdot A) \oplus (F_5(P^R, K_1^*) \cdot A) \oplus (C^R \cdot B) \oplus (C^L \cdot D') \oplus (F_{3,4}(C^R, K_{16}^*) \cdot D') = K \cdot \gamma_3 \quad (12)$$

and the second approximation is:

$$(P^L \cdot B) \oplus (F_5(P^R, K_1^*) \cdot B) \oplus (C^R \cdot A) \oplus (C^L \cdot D') \oplus (F_{3,4}(C^R, K_{16}^*) \cdot D') = K \cdot \gamma_3, \quad (13)$$

where γ_3 is identical and the involved S-boxes are the same in both equations. A fourth twin pair having the same approximation is made by interchanging the role of the plaintext and ciphertext bits in the Equations (12) and (13) is

$$(C^L \cdot A) \oplus (F_5(C^R, K_{16}^*) \cdot A) \oplus (P^R \cdot B) \oplus (P^L \cdot D') \oplus (F_{3,4}(P^R, K_1^*) \cdot D') = K \cdot \gamma_4 \quad (14)$$

and the second approximation is:

$$(C^L \cdot B) \oplus (F_5(C^R, K_{16}^*) \cdot B) \oplus (P^R \cdot A) \oplus (P^L \cdot D') \oplus (F_{3,4}(P^R, K_1^*) \cdot D') = K \cdot \gamma_4 \quad (15)$$

and the probability of the Equations (12) and (14) is $\frac{1}{2} + 6 \cdot \frac{5^6}{2^{38}}$ and the Equations (12) and (14) has the probability $\frac{1}{2} + 8 \cdot \frac{5^6}{2^{38}}$. The number of text pairs required to get $97,72\%$ success using the these one of these two twin pairs is

$$N_3 \quad = \quad N_4 = \frac{c_2 \cdot 2^{76}}{5^{12} \cdot (6^2 + 8^2)} = \frac{c_2 \cdot 2^{76}}{5^{14}},$$

which is the same as for Equations (8) and (9). Using all these twin pairs together is useful, and gives us as many as 38 distinct key bits in case of a correct guess, that is we have 10 overlapping bits. The right side of the equations also give us 4 bits information, which leaves 14 remaining key bits to an exhaustive search.

It is important to know that the new equations include one S-box in the first (or last*) round and two S-boxes in the last (or first*) round. As more unknown key bits is guessed more noise is added, so c_2 is expected to be greater than c_1. This should also be considered by using a weight $w = c_1/c_2$ to cancel this difference to give an optimal key ranking. Then the exact values of c_1 and c_2 is needed, and could possibly be calculated from the correlation between the functions for the correct and the right keys, but seems quite involved, so we leave it for future research.

Each Equation (8), (10), (9), (11), (12), (13), (14), (15) have its own counter T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8 which counts how many times the right side of each of the equation is 0. Each of the twin pairs (multiple) Equations ((8),(10)), ((9),(11)), ((12),(13)) and ((14),(15)) have their own counter T_{M1}, T_{M2}, T_{M3} and T_{M4}. Each T_{Mi} is a weighted sum of the counters for each of the twin pairs (multiple equations) $(1,2)$, $(3,4)$, $(5,6)$ and $(7,8)$

$$T_{Mi} = a_{2i-1} T_{2i-1} + a_{2i} T_{2i}$$

where the weight is calculated as in [9] for odd indexes

$$a_{2i-1} = \frac{e_{2i-1}}{e_{2i-1} + e_{2i}}$$

and for even indexes

$$a_{2i} = \frac{e_{2i}}{e_{2i-1} + e_{2i}}$$

Now we have individually weighted the set of equations that have exactly the same key bits involved. It remains to use a weighted ranking between the tables involving different key bits. The key bits in the different tables in our attack is not totally disjoint, but that is taken care of during the key ranking part. The ranking due to [10] is

$$T = \sum_{i=1}^{4} (b_i T_{Mi})^2$$

where

$$b_i = \sqrt{e_{2i-1}^2 + e_{2i}^2}$$

If we calculate the squared factor between the different weights being used in our attack we get

$$\left(\frac{b_1}{b_3}\right)^2 = \left(\frac{b_2}{b_4}\right)^2 = \left(\frac{\sqrt{10^2+5^2}}{\sqrt{6^2+8^2}}\right)^2 = \frac{5}{4}$$

and since we may change the weights by the same fraction without affecting the result we use $b_1 = b_2 = 5$ and $b_3 = b_4 = 4$ which shows that Matsui's equation using multiple equations is slightly stronger by a factor $5/4$. The combination of the different key tables is then

$$T = 5T_{M1} + 5T_{M2} + 4T_{M3} + 4T_{M4}$$

and the key ranking is done by sorting the values of T in decreasing order, and we get an optimal Neyman-Pearson ranking procedure.

If we look at Table 3 presenting results of 100 attacks we make the following observations

- The results in this table is included to compare with the results from the paper [10]. They want to point out the improvements of better ranking methods. Using the same amount of text data the improvements in the exhaustive part is non-trivial, and the improvement factors for the different parameters range from $1,11$ to $2,55$. Where we present a factor 2 improvement it should not be mixed with their factor $2,55$.
- The average factor of $2^{8,88}$ is not a good statistical measure, because the number are highly influenced by few high single event like C_{max}, which is a factor $2^{6,13}$ improvement. This could be due to few deviations, but the results could indicate a more spread complexity.
- C_{min} on 2^{14} is hardly a coincidence since we have 22 of the keys ranked at first place. In 50% of the attacks the correct key is ranked among the 11 (complexity $2^{17,46}$) best which C_{med} indicates, and this is an improvement of a factor $2^{19,38}$.
- We also use the ranking technique from [10], and therefore compare our results with the attack on DES with optimal ranking.

The results in Table 5 shows that a weighting of the different tables according to the [10] will give better results in the cases of using 2^{43} and 2^{42} known plaintexts, but is did not look better than not using weights if we used 2^{41} texts. This indicates that the optimal weighting must be adjusted to the number of texts used.

5 Conclusion

This paper presents what we believe is the best shortcut attacks on DES due to complexity. We present an chosen plaintext attack having an total improvement of a factor 2 over the original attack. The other attack we present is a known plaintext attack, which have a factor 2 improvement. If we compare and use the same measures as [10] we get an improvement factor of approximately 2^{20} in the exhaustive search phase. This way of measuring give us a huge advantage, and should not be considered as the complexity measure.

Table 3: Comparison between Matsui's attack using the optimal key ranking technique from [10] and our own attack also using this ranking. We do not think the table give a correct picture of the complexity, but the table is there just to compare the difference in improvement from [10]. In the 100 attacks giving the results in the table we simulate the use of 2^{43} text pairs by using the same simulation methods as Matsui. The argument that this works is due to the use of exactly the same approximation in the 8 round attack as an 16 round attack, and this is well founded by experiments.

	Matsui	Matsui/optimal ranking	Our/optimal ranking	Improvement factor
$\log_2 \mu_C$	41.4144	40.8723	31.9902	$2^{8,88}$
$\log_2 C_{85\%}$	40.7503	40.6022	27.4426	$2^{13,16}$
$\log_2 C_{min}$	32.1699	31.3219	14.0000	$2^{17,3219}$
$\log_2 C_{med}$	38.1267	36.7748	17.4594	$2^{19,38}$
$\log_2 C_{max}$	45.4059	44.6236	38.4936	$2^{6,13}$

Table 4: Results of the known plaintext attack comparing with Matsui's success rate 85% using 2^{43} texts with exhaustive search complexity of $2^{40.7503}$. This is a more fair comparison, which will give us a factor 2 improvement compared to [4, 5].

Data complexity	2^43	2^42	2^41
Time complexity	$2^{27.44}$	$2^{41.31}$	$2^{47.23}$
Success probability	85%	85%	85%

Table 5: Results of a simulation of 100 known plaintext attacks on 8 round of DES. The column marked with a "*" are the one where the different key tables are individually weighted. In the other columns the tables are not individually weighted.

Ex.search	2^43 texts	2^43 texts*	2^42 texts	2^42 texts*	2^41 texts	2^41 texts*
2^{38}	99%	99%	61%	66%	16%	16%
2^{39}	99%	100%	62%	71%	20%	17%
2^{40}	100%	100%	70%	80%	24%	22%
2^{41}	100%	100%	74%	84%	31%	28%
2^{42}	100%	100%	80%	92%	41%	37%
2^{43}	100%	100%	83%	93%	49%	44%
2^{44}	100%	100%	88%	96%	56%	53%

It was also pointed out by Matsui in [5] that the probability of the linear equation only depends on the value w where $K^*_{1,16} \oplus w$ is inserted into the equation and $K^*_{1,16}$ is the correct sub-key. This fact may give us a more accurate method to predict the success rate of linear attacks. It might be used in a method to increase the success rate, and also remove the noise factor c.

Different weighting of tables representing disjunct sub-keys gives different results dependent of how many text pairs we use. This shows the need for a more accurate weighting, or taking more factors into the consideration when calculating the weights. The fact that different equations have different noise factors c_i indicates that we should take this into consideration when calculation the weights.

References

[1] National Bureau of Standards, "Data encryption standard," Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.

[2] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[3] M. Matsui and A. Yamagishi, "A new method for known plaintext attack of FEAL cipher," in *Advances in Cryptology - EUROCRYPT'92, LNCS 658*, R. Rueppel, Ed. 1992, pp. 81–91, Springer Verlag.

[4] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology - EUROCRYPT'93, LNCS 765*, T. Helleseth, Ed. 1993, pp. 386–397, Springer Verlag.

[5] M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard," in *Advances in Cryptology - CRYPTO'94, LNCS 839*, Y.G. Desmedt, Ed. 1994, pp. 1–11, Springer Verlag.

[6] S. Vaudenay, "An experiment on DES - statistical cryptanalysis," in *Proceedings of the 3rd ACM Conferences on Computer Security, New Delhi, India.* 1995, pp. 139–147, ACM Press.

[7] M. Matsui, "On correlation between the order of S-boxes and the strength of DES," in *Advances in Cryptology - EUROCRYPT'94, LNCS 950*, A. De Santis, Ed. 1995, Springer Verlag.

[8] M. Matsui, "New structure of block ciphers with provable security against differential and linear cryptanalysis," in *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, D. Gollman, Ed. 1996, pp. 205–218, Springer Verlag.

[9] B.S. Kaliski and M.J.B. Robshaw, "Linear cryptanalysis using multiple approximations," in *Advances in Cryptology: CRYPTO'94, LNCS 839*, Y. Desmedt, Ed. 1994, pp. 26–39, Springer Verlag.

[10] P. Junod and S. Vaudenay, "Optimal key ranking procedures in a statistical cryptanalysis," in *Fast Software Encryption: FSE 2003, LNCS 2887*, T. Johansson, Ed. 2003, pp. 235–246, Springer Verlag.

[11] L.R. Knudsen and M.P.J. Robshaw, "Non-linear approximations in linear cryptanalysis," in *Advances in Cryptology: EUROCRYPT'96, LNCS 1070*, U. Maurer, Ed. 1996, pp. 224–236, Springer Verlag.

[12] T. Shimoyama and T. Kaneko, "Quadratic relation of s-box and its application to the linear attack of full round DES," in *Advances in Cryptology: CRYPTO'98, LNCS 1462*, H. Krawczyk, Ed. 1998, pp. 200–211, Springer Verlag.

[13] L. R. Knudsen and J. E. Mathiassen, "A chosen-plaintext linear attack on DES," in *Fast Software Encryption: FSE 2000, LNCS 1978*, B. Schneier, Ed. 2000, pp. 262–272, Springer Verlag.

[14] M.E. Hellman and S.K. Langford, "Differential–linear cryptanalysis," in *Advances in Cryptology: CRYPTO'94, LNCS 839*, Y. Desmedt, Ed. 1994, pp. 26–39, Springer Verlag.

[15] E. Biham, O. Dunkelmann, and N. Keller, "Enhancing differential-linear cryptanalysis," in *Advances in Cryptology: ASIACRYPT 2002, LNCS 2501*, Y. Zheng, Ed. 2002, pp. 254–266, Springer Verlag.