On the Complexity of Matsui's Attack

Pascal Junod

Security and Cryptography Laboratory
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
pascal.junod@epfl.ch

Abstract. Linear cryptanalysis remains the most powerful attack against DES at this time. Given 2^{43} known plaintext-ciphertext pairs, Matsui expected a complexity of less than 2^{43} DES evaluations in 85 % of the cases for recovering the key. In this paper, we present a theoretical and experimental complexity analysis of this attack, which has been simulated 21 times using the idle time of several computers. The experimental results suggest a complexity upper-bounded by 2^{41} DES evaluations in 85 % of the case, while more than the half of the experiments needed less than 2^{39} DES evaluations. In addition, we give a detailed theoretical analysis of the attack complexity.

Keywords: linear cryptanalysis, DES

1 Introduction

Linear cryptanalysis against DES [10] has been introduced by Matsui [6, 7] and remains at this time the most powerful attack against this cipher. A single experimental implementation [7] has been carried out. During this attempt, Matsui managed to break a DES key in about 50 days on 12 powerful computers, the plaintext-ciphertext pairs generation lasting 40 days and the exhaustive search for the remaining unknown bits taking the last 10 days. It was noticed that the second phase performed faster than one could expect theoretically.

Although several authors have studied, generalized and applied the linear cryptanalysis concept in several ways, little work concerning its success probability and its complexity has been done, and while it is widely accepted that linear cryptanalysis of DES, given 2^{43} known plaintext-ciphertext pairs, has a success probability of 85 % within a complexity of 2^{43} DES evaluations, it was conjectured that this value is pessimistic [9, 3].

Motivated by this fact, by the parallel implementation concept of Biham [1] and the actual 64-bit processor performances, we propose in this paper a theoretical and experimental complexity analysis. By using a fast DES routine implemented for the Intel MMX architecture, the production part of the attack has been run several time, virtually breaking a total of 21 keys.

This paper is organized as follows: in §2, we recall some theoretical background on the attack. In §3, we describe briefly the design of the fast DES routine and the attack implementation. In §4, we discuss and complete the success probability and complexity model. In §5, we discuss some issues on the linear expression biases, the piling-up approximation and the wrong-key randomization hypothesis, comparing the known theoretical results to our experimental ones and finally we give in §6 our experimental results.

2 Matsui's Attack

In this paper, we deal with the improved attack [7] proposed by Matsui against DES. The attack's core is unbalanced linear expressions, i.e. equations involving a modulo two sum of plaintext and ciphertext bits on the left and a modulo two sum of key bits on the right. Such an expression is unbalanced if it is satisfied with probability [1] $p = \frac{1}{2} + \kappa\epsilon$ with $0 < \epsilon \leq \frac{1}{2}$ and $\kappa \in \{-1, 1\}$ when the plaintexts and the key are independent and chosen uniformly at random and where κ depends on the key value.
Given some plaintext bits $\mathsf{P}_{i_1}, \ldots, \mathsf{P}_{i_r}$, ciphertext bits $\mathsf{C}_{j_1}, \ldots, \mathsf{C}_{j_s}$ and key bits $\mathsf{K}_{k_1}, \ldots, \mathsf{K}_{k_t}$, and using the notation $\mathsf{X}_{l_1} \oplus \mathsf{X}_{l_2} \oplus \ldots \oplus \mathsf{X}_{l_u} = \mathsf{X}_{[l_1,\ldots,l_u]}$, we can write a linear expression $\mathcal{L}$ as

$$\mathcal{L} : \mathsf{P}_{[i_1,\ldots,i_r]} \oplus \mathsf{C}_{[j_1,\ldots,j_s]} = \mathsf{K}_{[k_1,\ldots,k_t]} \tag{1}$$

Matsui's improved attack operates on 14 rounds using two biased linear expressions which collect statistical information on 26 bits out of the first and last round subkeys. The remaining 30 unknown key bits have to be searched exhaustively. The linear expression (1) involves thus two terms of F-function and can be rewritten as

$$\begin{aligned} \mathcal{L} : \mathsf{P}_{[i_1,\ldots,i_r]} \oplus \mathsf{C}_{[j_1,\ldots,j_s]} \oplus F^{(1)}_{[l_1,\ldots,l_u]}\left(\mathsf{P}, \mathsf{K}^{(1)}\right) \oplus \\ F^{(16)}_{[m_1,\ldots,m_v]}\left(\mathsf{C}, \mathsf{K}^{(16)}\right) = \mathsf{K}_{[k_1,\ldots,k_t]} \end{aligned} \tag{2}$$

where $F^{(1)}_{[l_1,\ldots,l_u]}\left(\mathsf{P}, \mathsf{K}^{(1)}\right)$ is the modulo two sum of some bits resulting from the F-function output in the first round and $\mathsf{K}^{(1)}$ is the subkey of round 1. A similar notation is used for the last F-function.
The attack main idea is related to the following assumption:

Assumption 1 (Wrong-key randomization hypothesis [3]). *For any linear expression $\mathcal{L}$ operating on n rounds for which*

$$\left|\Pr\left[\mathcal{L} = 0 \,|\, \mathsf{K}^{(1)} = k^{(1)}, \ldots, \mathsf{K}^{(n)} = k^{(n)}\right] - \frac{1}{2}\right|$$

[1] In the literature, this non-linearity measure is often called *linear probability*, and expressed as $\mathrm{LP}^f(a,b) = \left(2\Pr[a \cdot x = b \cdot f(x)] - 1\right)^2$, where a and b are the masks selecting the plaintext and ciphertext bits, respectively. In this paper, we will refer to the *bias* ϵ for simplicity reasons.

is large for virtually all values $k^{(1)}, \ldots, k^{(n)}$ *of the round keys, the following is true: for virtually all possible full keys* $(k^{(1)}, \ldots, k^{(n)})$ *and for all estimates* $\hat{k}$ *of the last round key,*

$$\frac{\left|\Pr\left[\mathcal{L} = 0 \mid \mathsf{K} = k_r\right] - \frac{1}{2}\right|}{\left|\Pr\left[\mathcal{L} = 0 \mid \mathsf{K} = \hat{k}\right] - \frac{1}{2}\right|} \gg 1 \quad \forall \hat{k} \neq k_r \tag{3}$$

where k_r *is the right key.*

Intuitively, the decryption of the first and the last round with wrong subkey candidates can be considered as two rounds more of encryption. Thus, the plaintext and the ciphertext will be less dependent, and the linear expressions less biased. The first linear cryptanalysis phase (see Fig. 1) consists in evaluating the bias of both linear expressions for all possible subkey candidates and for all known plaintext-ciphertext pairs. In a second phase (Fig. 2), the two lists of subkey candidates corresponding each to a linear expression are sorted in a maximum-likelihood manner, combined, and the missing bits are finally searched exhaustively for each pair of subkey candidate until the right key is found.
The complexity $\mathcal{C}$ of the attack is then related to the number of needed DES encryptions in the exhaustive search part while its success probability $\mathcal{P}_{\mathcal{C}}$ within a given complexity $\mathcal{C}$ is *also* related to the success while guessing the right part of both linear expressions.

```
N = number of known plaintext-ciphertext pairs at disposal.
for linear expressions L_1 and L_2 do
    for all subkey candidates k̂_i, 1 ≤ i ≤ 2^12 do
        C_{k̂_i} = number of times out of N where left part of (2) is equal to 0 when
        K = k̂_i.
    end for
end for
```

Figure 1: Matsui's algorithm 2 [7] (phase 1)

3 Implementation of the Attack

The linear cryptanalysis attack against DES, except the exhaustive search part, has been implemented as described in [7]. After having determined the rank of the right subkey candidate in the final list, it is not difficult to compute [2] the expected complexity (in DES function evaluations) of the exhaustive search part:

$$\mathrm{E}[\hat{\mathcal{C}}] = (r-1) \cdot 2^{30} + 2^{29}$$

[2] The strategy used to combine the two lists of 13-bit subkey candidates is Matsui's proposed one [7]: sort the pairs by increasing $r = i \cdot j$ (see lines 12-13 of Fig. 2), where i and j are the respective ranks in the 13-bit subkey lists.

for linear expressions $\mathcal{L}_1$ and $\mathcal{L}_2$ **do**
  Sort the $C_{\hat{k}_i}$'s by decreasing $\left|\frac{N}{2} - C_{\hat{k}_i}\right|$ and rename them $C_j^*, 1 \leq j \leq 2^{12}$.
  for $1 \leq j \leq 2^{12}$ **do**
    /* κ is defined in Sect. 2 (expected bias of $\mathcal{L}$) */
    if $\left(C_j^* - \frac{N}{2}\right)\kappa > 0$ **then**
      Guess $\mathsf{K}_{[k_1,\ldots,k_t]} = 0$
    else
      Guess $\mathsf{K}_{[k_1,\ldots,k_t]} = 1$
    end if
  end for
end for
Form 2^{24} $(C_i^*, C_j^*)_r$ pairs where $r := i \cdot j$.
Sort them by increasing r and rename them $D_k, 1 \leq k \leq 2^{24}$.
for $1 \leq k \leq 2^{24}$ **do**
  Fix the key bits given by D_k and search exhaustively the remaining 30 bits of K until the right key is found.
end for

Figure 2: Matsui's algorithm 2 [7] (phase 2)

where r is the rank in the list D of subkey candidates. The complexity's estimation error has thus a maximal value of 2^{29} DES evaluations, which is negligible almost all the time.

The computational most intensive part of the attack being data encryption, the involved DES routine speed is a key parameter regarding the time needed to process 2^{43} plaintexts. We have thus implemented a very fast DES routine using the bitslicing concept [1] and some attack-related optimizations. Our routine has been designed for the Intel MMX architecture which has eight 64-bit registers at disposal. Although this platform has several drawbacks regarding a bitsliced implementation [8], it has the advantage of being very common.

Kwan's gate representation of the S-boxes [5] builds the core of the implementation, the other parts of the cipher (key schedule, permutations, ...) being hard-coded. By eliminating parts of the cipher unrelated to the attack and by using advanced optimization techniques like instruction pairing, prefetching of the data and code unrolling, we managed to get an encryption speed of 183 Mbps on an Intel Pentium III clocked at 666 MHz. This represents 232.7 clock cycles for encrypting one block of data. One can hardly compare this number with existing good implementations [3], because of the optimizations related to the attack; however, using classical available implementations for our purposes would have resulted in poorer performances.

[3] A DES routine was implemented for similar purposes in [12] on other platforms; they report 62 Mbps on a Ultra SPARC 200 MHz and 336 Mbps on a Alpha 21164A 500 MHz. The significant speed difference on the latter platform is due to the large number of available 64-bit registers (and thus to a lesser number of slow memory accesses).

The attack has run 21 times, using the idle time of 8 to 16 computers; this represents between 3 and 6 days for a single run.

4 Success Probability

In this section, we address a general way to characterize the probability distribution of the rank of the right 13-bit subkey in the list of candidates given by a linear approximation $\mathcal{L}$.

4.1 Rank Probability

As the complexity $\mathcal{C}$ of the attack is closely related to the rank of the right subkey in the candidates list, we address first the problem of estimating the rank distribution.

Let $W_1, \ldots, W_n$ be n independent and identically distributed continuous random variables having $f_W(x)$ and $F_W(x)$ as common density function and distribution function, respectively. Let R be a continuous random variable independent of the W_i's and having $f_R(x)$ and $F_R(x)$ as density and distribution function. Sort these $n+1$ random variables in non-increasing order and rename them $Z_{(1)} > Z_{(2)} > \ldots > Z_{(n+1)}$. Finally, let Ψ be a discrete random variable taking values on $\{1, \ldots, n+1\}$ which models the rank of R in the sorted list: $\Psi = \psi \Leftrightarrow Z_{(\psi)} = R$. The distribution of Ψ and its expected value are given by the following theorem, whose proof is given in Appendix A.

Theorem 1. *Under previous assumptions and for* $1 \leq \psi \leq n \in \mathbb{N}$*, the distribution function of* Ψ *is equal to*

$$\Pr[\Psi \leq \psi] = \int_{-\infty}^{+\infty} B_{n+1-\psi,\psi}(F_W(x)) f_R(x) dx$$

and

$$\mathrm{E}[\Psi] = 1 + n \left(1 - \int_{-\infty}^{+\infty} f_R(x) F_W(x) dx\right)$$

where

$$B_{a,b}(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1} dt$$

is the incomplete beta function of order (a, b).

In order to be able to compute the densities of the estimated biases[4], we first have to make the following assumptions [13]; the two first ones are heuristic in nature, while the last one is motivated by the law of large numbers. C_{k_r} (C_{k_w}) will denote a random variable modeling the counter value (as defined at line 4 of Fig. 1) in the case of a right (wrong) subkey candidate and N is the number of known plaintext-ciphertext pairs.

[4] The mean and standard deviation of the counters and the respective biases of the linear expression being linearly related, we will use in the following the bias terminology.

Assumption 2. *The bias*

$$B_w = \left| \frac{1}{2} - \frac{C_{k_w}}{N} \right|$$

of a linear expression evaluated with wrong subkey candidates has a distribution independent of the key value.

Assumption 3. *The bias*

$$B_r = \left| \frac{1}{2} - \frac{C_{k_r}}{N} \right|$$

of a linear expression evaluated with the right subkey candidates has a distribution independent of the distribution defined in Assumption 2 and independent of the key value.

Assumption 4. *The distributions of* $\frac{C_{k_r}}{N}$ *and* $\frac{C_{k_w}}{N}$ *are well approximated by a normal law.*

We denote in the following the normal law density with mean μ and variance σ^2 by $\phi_{(\mu,\sigma^2)}$ and the corresponding cumulative distribution function by $\Phi_{(\mu,\sigma^2)}$. Because the cryptanalyst ignores the linear expression's right part, she is more interested in the absolute value of the biases. Noting that if X is a normal law $\phi_{(\mu,\sigma^2)}$, the density of $Y = |X - a|$, $a \leq \mu$ is given by $f_Y^{(\mu,\sigma^2)}(y, a) = \phi_{(\mu,\sigma^2)}(y + a) + \phi_{(\mu,\sigma^2)}(a - y)$ for $0 \leq y \leq +\infty$, the bias densities in case of wrong and right subkey candidates are respectively given by

$$f_W(x) = f^{(\mu_w,\sigma_w^2)}(x, \tfrac{1}{2}) \tag{4}$$

$$f_R(x) = f^{(\mu_r,\sigma_r^2)}(x, \tfrac{1}{2}) \tag{5}$$

with

$$\mu_r = \mathrm{E}\left[\frac{C_{k_r}}{N}\right] = \frac{1}{2} + \kappa\epsilon_r \qquad \mu_w = \mathrm{E}\left[\frac{C_{k_w}}{N}\right] = \frac{1}{2} + \kappa\epsilon_w$$

$$\sigma_r^2 = \mathrm{Var}\left[\frac{C_{k_r}}{N}\right] \approx \frac{1}{4N} \qquad \sigma_w^2 = \mathrm{Var}\left[\frac{C_{k_w}}{N}\right] \approx \frac{1}{4N}$$

where $\kappa \in \{-1, +1\}$ depends of the unknown key bits and C_{k_r} (C_{k_w}) is the random variable modeling the value of the counter corresponding to the (a) right (wrong) subkey. Fig. 4.1 gives some numerical evaluations of Theorem 1 for these densities while the following table gives the expected rank for various amounts of known plaintext-ciphertext pairs at disposal. Here, we assume that $\epsilon_r = 1.19 \cdot 2^{-21}$ is equal to the piling-up lemma approximation and that $\epsilon_w = 0$.

N	2^{43}	$2^{42.5}$	2^{42}	2^{41}	2^{40}
$\mathrm{E}[\Psi]$	71.3	182.5	361.9	847.3	1311.6

We note that Theorem 1 gives exactly the same values as Matsui's experimental computations [7] regarding the cumulative rank probability of the right subkey candidate.

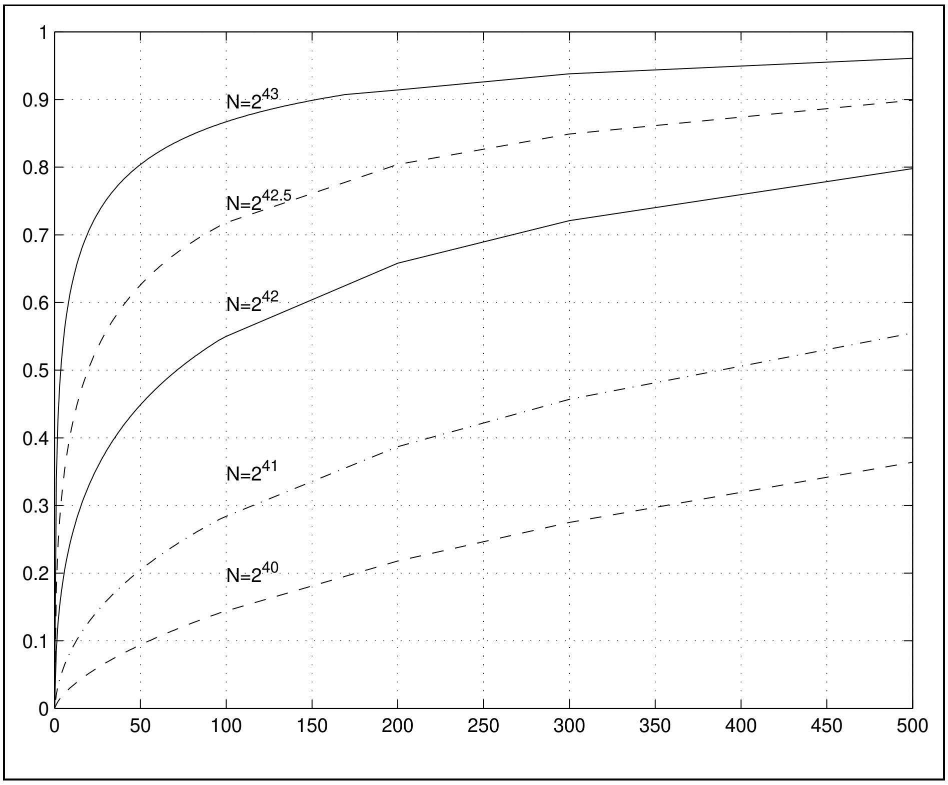

Figure 3: Rank distribution $\Pr[\Psi \leq \psi]$ for various amounts N of plaintext-ciphertext pairs.

4.2 Success Probability

The attack's success probability $\mathcal{P}_{\mathcal{C}}$ within a given complexity $\mathcal{C}$ is also dependent on the error probability while guessing the bit of information about $\mathsf{K}_{[k_1,\ldots,k_t]}$. Using the same assumptions as during the previous computations, it is easy to compute this error probability (in the case where $\kappa = +1$ and $\mathsf{K}_{[k_1,\ldots,k_t]} = 0$, the other ones being symmetric).

$$p_{wg} = \Pr\left[\text{“}\mathsf{K}_{[k_1,\ldots,k_t]} \text{ wrongly guessed”}\right] = \Phi_{(\mu_r,\sigma_r^2)}\left(\frac{N}{2}\right) \tag{6}$$

The following table gives some numerical approximations for various N:

N	2^{43}	$2^{42.5}$	2^{42}	2^{41}	2^{40}
p_{wg}	0.0004	0.0023	0.0086	0.0462	0.1170

5 Experimental Linear Expressions Biases

A key parameter regarding the linear cryptanalysis success is of course the bias of the involved linear expression(s). As it is infeasible to compute the exact bias of a linear expression, one uses implicit assumptions, such as the wrong-key randomization one and the independence of data between two successive rounds. The incidence of these assumptions has been well discussed in the literature [9, 2–4]. Although several situations where these assumptions can fail have been suggested and discussed, it is accepted that the linear expression real bias should be well approximated in case the of DES.
The experimental results go in this direction. We have computed the sample means of the experimental biases $\hat{B}_r$ and $\hat{B}_w$, which can be compared to the expected values of densities (4) and (5).
In case of right key, the sample mean is equal to $5.5 \cdot 10^{-7}$ with a standard deviation of $0.2 \cdot 10^{-7}$. This value has to be compared with the one given by the piling-up approximation and (5), $\mathrm{E}[B_r] = 5.674 \cdot 10^{-7}$. As a first observation, one can note that the *linear hull effect* [9] is not visible for DES, the mean experimental bias being not perceptibly greater than the piling-up lemma approximation.
Our experiments provide furthermore a good opportunity to confirm the validity of Assumption 1. The sample mean in case of wrong subkey candidates, averaged over all the wrong subkeys and all experiments, is equal to $1.38 \cdot 10^{-7}$ with a standard deviation of $0.03 \cdot 10^{-7}$. This value has to be compared with $\mathrm{E}[B_w] = 1.345 \cdot 10^{-7}$ given by $\epsilon_w = 0$ and (4). Obviously, as one could expect, the mean seems to be slightly greater than for a perfect cipher and thus the plaintext and ciphertext are still correlated. However, the bias values for the wrong candidates are not on the same scale as those for the right candidates, confirming the validity of Assumption 1 for DES.

6 Experimental Results

It is widely accepted that linear cryptanalysis of DES, given 2^{43} known plaintext-ciphertext pairs, has a success probability of $\mathcal{P}_{\mathcal{C}_\mathcal{A}} = 85\%$ within a complexity of $\mathcal{C}_\mathcal{A} = 2^{43}$ DES encryptions, which are values given in [7]. Our experimental results suggest a lower complexity.

6.1 Rank and Guessing Error Probabilities

Each of the 21 experiments provides two statistical samples. Following table summarizes our results about the ranks of the right subkey candidates for various amounts N of known plaintext-ciphertext pairs and compare them to the theoretical expectations (values in smaller characters) given by Theorem 1.

N	2^{43}	$2^{42.5}$	2^{42}	2^{41}	2^{40}
$\psi \leq 5$	20 (22)	13 (13.5)	7 (7.6)	0 (2.3)	0 (0.8)
$\psi \leq 10$	27 (25.8)	16 (17.1)	9 (10.5)	2 (3.6)	0 (1.3)
$\psi \leq 50$	33 (33.6)	26 (26.2)	18 (18.8)	5 (8.6)	2 (3.9)
$\psi \leq 150$	38 (37.7)	34 (32.3)	24 (25.7)	10 (14.3)	5 (7.7)
$\psi \leq 300$	42 (39.4)	39 (35.7)	31 (30.3)	17 (19.2)	14 (11.6)
$\psi \leq 600$	42 (40.8)	40 (38.5)	35 (34.6)	25 (24.7)	22 (16.8)
$\mathrm{E}[\psi]$	38 (71)	129 (182)	302 (362)	654 (847)	1121 (1312)

We observe that Theorem 1 seems to give a pessimistic rank expected value. It is difficult to explain this fact because of the small statistical sample size. Furthermore, we have noticed that Theorem 1 is very sensitive numerically. For instance, the expected rank $\mathrm{E}[\Psi]$ is equal to 113 and to 39 when we assume that $\epsilon_r = 1.1 \cdot 2^{-21}$ and $\epsilon_r = 1.3 \cdot 2^{-21}$, respectively.
The experimental results regarding the remaining bit guessing error probability are summarized in the following table. The number n_{wg} of cases where the guessing phase was unsuccessful is reported, together with the theoretical expected values given by (6) which are given in smaller characters.

N	2^{43}	$2^{42.5}$	2^{42}	2^{41}	2^{40}
n_{wg}	0 (0.02)	0 (0.10)	0 (0.36)	0 (1.94)	1 (4.91)

One can see that (6) is a bit pessimistic, which can be explained a new time by the arguments developed below. We note furthermore that the success probability $\mathcal{P}_\mathcal{C}$ of the linear cryptanalysis of DES within a given complexity $\mathcal{C}$ seems not to be so dependent on the guessed bit of information about the key and that the key factor regarding $\mathcal{P}_\mathcal{C}$ is the given upper bound $\mathcal{C}$.

6.2 Complexity of the Attack

An exhaustive table of our experimental results regarding the complexity is given in Appendix B. Key facts (mean, median, maximal and minimal $\hat{\mathcal{C}}$) are summarized in the following table where a value of x means 2^x DES evaluations:

N	2^{43}	$2^{42.5}$	2^{42}	2^{41}	2^{40}
$\mu_{\hat{\mathcal{C}}}$	41.4144	47.1516	48.9504	50.2121	51.4154
$\hat{\mathcal{C}}_{med}$	38.1267	41.8023	44.2949	48.5492	51.0533
$\hat{\mathcal{C}}_{min}$	32.1699	29.0000	36.5157	43.8552	41.9750
$\hat{\mathcal{C}}_{max}$	45.4059	51.2973	52.3671	52.1953	53.1000

Our experimental results lead to the following observations:

- Given 2^{43} known plaintext-ciphertext pairs, our experiments have a complexity of less than 2^{41} DES evaluation with a success probability of 86 % where more than the half of the cases have a complexity less than 2^{39}.

Furthermore, if an attacker is ready to decrease her success probability, the complexity drops dramatically (less than 2^{34} DES evaluations with a success probability of 10 %).
– Given $2^{42.5}$ known plaintext-ciphertext pairs (i.e. with 30 % less pairs), half of the experiments have a complexity less than 2^{42} DES evaluations.
– With only 2^{40} pairs at disposal, the complexity is far lower than an exhaustive search.

Even if we have to take these experimental results carefully because of the relative small number of statistical samples, they suggest strongly a lower complexity than expected by Matsui in [7] and we risk the following conjecture:

Proposition 1. *Given* 2^{43} *known plaintext-ciphertext pairs, it is possible to recover a DES key using Matsui's linear cryptanalysis within a complexity of* 2^{41} *DES evaluations with a success probability of 85 %.*

7 Conclusion

The first goal of this research was to perform an experimental linear cryptanalysis of DES as many times as possible in order to get a better insight into the real complexity and success probability of this attack. Using a very fast DES function developed for the Intel MMX architecture, we have simulated Matsui's attack 21 times.
Our experimental results suggest a lower complexity than estimated by Matsui. Given 2^{43} known plaintext-ciphertext pairs, the complexity was upper-bounded by 2^{41} DES evaluations with a success probability of 85 %. This has to be compared with the estimated 2^{43}.
We give furthermore a detailed theoretical analysis of the rank probability of the right subkey in the list of candidates, confirming Matsui's experimental results, and we discuss the validity of our theoretical model towards the experimental results, together with several issues regarding past research.

Acknowledgments

The author would like to thank Serge Vaudenay for having suggested this research topic and the SAC'01 anonymous referees for many helpful comments.

References

1. E. Biham, *A fast new DES implementation in software*, FSE '97, LNCS, vol. 1267, Springer-Verlag, 1997, pp. 260–272.
2. U. Blöcher and M. Dichtl, *Problems with the linear cryptanalysis of DES using more than one active S-box per round*, FSE '94, LNCS, vol. 1008, Springer-Verlag, 1995, pp. 265–274.

3. C. Harpes, G. Kramer, and J.L. Massey, *A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma*, Advances in Cryptology - EuroCrypt '95, LNCS, vol. 921, Springer-Verlag, 1995, pp. 24–38.
4. Z. Kukorelly, *The piling-up lemma and dependent random variables*, Cryptography and coding: 7th IMA conference, LNCS, vol. 1746, Springer-Verlag, 1999.
5. M. Kwan, *Reducing the gate count of bitslice DES*, `http://eprint.iacr.org/2000/051.ps`, 2000.
6. M. Matsui, *Linear cryptanalysis method for DES cipher*, Advances in Cryptology - EuroCrypt '93, LNCS, vol. 765, Springer-Verlag, 1993, pp. 386–397.
7. ———, *The first experimental cryptanalysis of the Data Encryption Standard*, Advances in Cryptology - Crypto '94, LNCS, vol. 839, Springer-Verlag, 1994, pp. 1–11.
8. L. May, L. Penna, and A. Clark, *An implementation of bitsliced DES on the pentium MMX ^{TM} processor*, Information Security and Privacy: 5th Australasian Conference, ACISP 2000, LNCS, vol. 1841, Springer-Verlag, 2000.
9. K. Nyberg, *Linear approximation of block ciphers*, Advances in Cryptology - EuroCrypt '94, LNCS, vol. 950, Springer-Verlag, 1995, pp. 439–444.
10. National Bureau of Standards, *Data encryption standard*, U. S. Department of Commerce, 1977.
11. A. Rényi, *Probability theory*, Elsevier, 1970.
12. T. Shimoyama and T. Kaneko, *Quadratic relation of s-box and its application to the linear attack of full round DES*, Advances in Cryptology - Crypto '98, LNCS, vol. 1462, Springer-Verlag, 1998, pp. 200–211.
13. S. Vaudenay, *An experiment on DES statistical cryptanalysis*, 3rd ACM Conference on Computer and Communications Security, ACM Press, 1996, pp. 139–147.

A Proof of Theorem 1

As a first step, let's consider the following situation: let $W_1, W_2, \ldots, W_n$ be n independent and identically distributed continuous random variables having f_W as density function and F_W as distribution function. We arrange the values of $W_1, W_2, \ldots, W_n$ in strictly[5] increasing order and denote them by $W_{(1)} < W_{(2)} < \ldots < W_{(n)}$. The distribution function $F_{W_{(i)}}$ of $W_{(i)}$ is given by the following Lemma whose proof can be found in [11].

Lemma 1. *Under previous assumptions, the distribution function of the i-th smallest random variable is*

$$F_{W_{(i)}}(x) = B_{i,n-i+1}\left(F\left(x\right)\right)$$

where

$$B_{a,b}(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1}\, dt$$

is the incomplete beta function of order (a,b).

[5] The probability that equal values occur is 0.

By using the previous Lemma and the independence between the involved random variables, we can compute $F_{\Psi}(x)$ as follows:

$$\begin{aligned}
\Pr[\Psi \leq \psi] &= \Pr\left[W_{(\psi)} < R\right] \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{y} f_{W_{(\psi)}}(x) f_R(y)\, dx\, dy \\
&= \int_{-\infty}^{+\infty} B_{n+1-\psi,\psi}\left(F_W(y)\right) f_R(y)\, dy
\end{aligned}$$

By definition, we have

$$\begin{aligned}
\mathrm{E}[\Psi] &= \sum_{\psi=1}^{n+1} \psi \cdot \Pr[\Psi = \psi] \\
&= \Pr[\Psi = 1] + \sum_{\psi=2}^{n+1} \psi \left(\Pr[\Psi \leq \psi] - \Pr[\Psi \leq \psi - 1]\right) \\
&= n + 1 - \sum_{\psi=1}^{n} \Pr[\Psi \leq \psi]
\end{aligned}$$

where

$$\begin{aligned}
\sum_{\psi=1}^{n} \Pr[\Psi \leq \psi] &= \sum_{\psi=1}^{n} \int_{-\infty}^{+\infty} B_{n+1-\psi,\psi}\left(F_W(y)\right) f_R(y)\, dy \\
&= \int_{-\infty}^{+\infty} f_R(y) \sum_{\psi=1}^{n} B_{n+1-\psi,\psi}\left(F_W(y)\right)\, dy
\end{aligned}$$

It is easy to see that

$$\begin{aligned}
\sum_{\psi=1}^{n} B_{n+1-\psi,\psi}\left(F_W(y)\right) &= n \int_{0}^{F_W(y)} \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i}\, dt \\
&= n \int_{0}^{F_W(y)} dt \\
&= n F_W(y)
\end{aligned}$$

and we can thus conclude with

$$
\begin{aligned}
\mathrm{E}\left[\Psi\right] &= n+1-\int_{-\infty}^{+\infty} f_R(y) \sum_{\psi=1}^{n} B_{n+1-\psi,\psi}\left(F_W(y)\right)\, dy \\
&= n+1-n\int_{-\infty}^{+\infty} f_R(y)F_W(y)\, dy \\
&= 1+n\left(1-\int_{-\infty}^{+\infty} f_R(y)F_W(y)\, dy\right)
\end{aligned}
$$

B Complete Experimental Results

This table gives the experimental results regarding the complexity $\hat{\mathcal{C}}$ of each run of the attack for various amounts of plaintext-ciphertext pairs.

Exp	$N=2^{43}$	$N=2^{42.5}$	$N=2^{42}$	$N=2^{41}$	$N=2^{40}$
1	39.1836	38.4818	45.0307	51.3802	51.0533
2	33.2479	41.6346	43.6383	48.0928	43.1913
3	38.6055	41.8023	43.9622	48.5492	51.6012
4	38.1267	34.6147	41.3351	48.7240	51.2041
5	37.4878	29.0000	36.5157	46.1991	52.3685
6	34.0444	44.2753	46.6834	48.5221	50.1937
7	36.4676	45.5732	44.2949	47.3010	51.2913
8	36.1189	44.7722	41.4091	51.6338	52.1143
9	40.3515	47.0565	48.6184	52.1953	53.1000
10	41.6540	41.8682	45.7429	47.9120	41.9750
11	45.4059	51.2973	51.9932	51.8155	52.1972
12	36.1189	43.6633	46.7256	50.3949	49.2317
13	36.4009	36.1189	43.2183	47.0756	46.7680
14	39.0042	42.6736	44.3057	44.7116	47.3256
15	37.6330	39.8572	47.6536	49.5244	52.6439
16	38.9204	36.6653	41.5447	49.1082	49.9939
17	33.5236	38.8502	43.3128	46.1030	48.6798
18	39.8478	47.4938	52.3671	50.6770	50.3675
19	32.1699	31.8074	40.5093	43.8552	48.4968
20	40.7503	38.3729	40.3734	45.2436	52.3101
21	41.8721	44.9063	45.4147	52.0730	52.8571