New Results in the Linear Cryptanalysis of DES

Igor Semaev
Department of Informatics
University of Bergen, Norway
e-mail: igor@ii.uib.no
phone: (+47)55584279
fax: (+47)55584199

May 23, 2014

Abstract

Two open problems on using Matsui's Algorithm 2 with multiple linear approximations posed earlier by Biryukov, De Cannière and M. Quisquater at Crypto'04 are solved in the present paper. That improves the linear cryptanalysis of 16-round DES reported by Matsui at Crypto'94.

keywords: linear cryptanalysis, multiple linear approximations, success probability, MRHS linear equations, gluing algorithm.

1 Introduction

Linear Cryptanalysis is one of the major techniques in the cryptanalysis of symmetric ciphers. It was introduced and then improved in [10, 11] as an attack to DES by Matsui, though some similar ideas appeared independently in [9] as well. Linear Cryptanalysis is a known plain-text attack and it exploits that certain linear combinations, called approximations, modulo 2 of the plaint-text, cipher-text and key bits are zeros with some a priori computed probability. Two attacks Algorithm 1 and Algorithm 2 were suggested in [10]. Algorithm 1 uses n-round approximations to attack n-round cipher, while Algorithm 2 uses $n-1$ or $n-2$-round approximations. The latter requires a lower amount of plain-text/cipher-text pairs and is more efficient.

Linear cryptanalysis was extended in different ways by several authors as in [4] and [5], see [6] as well. For instance, [5] made use multivariate approximations instead of one-variate. However only few improvements with relation to DES were published. In [8] a chosen plaint-text linear attack was suggested and in [2] time complexity of the attack first stage was reduced by using Fast Fourier Transform.

For 16-round DES, Matsui shows how to determine candidates for key bits or key-bit linear combinations by Algorithm 2 with 2^{43} plain-text/cipher-text blocks and success probability 0.85, then 2^{43} trials are run to get the correct key [11]. Two 14-round approximations considered statistically independent were there used together. How to improve Algorithm 1 with more than two approximations was shown in [7]. In [1] a framework for using many approximations considered statistically independent was proposed, though no practical cryptanalysis was presented. Two open problems related to Algorithm 2 were there posed. First, how to merge data from different approximations efficiently. Second, how to compute the success probability as a function in the number of available plain-texts and the number of trials in the search phase.

In the present note a solution to those problems is suggested. It is very different from the theory in [5]. The method or its straightforward generalisations are applicable to any cipher. In particular, for 16-round DES we show how by using 2^{43} plain-text/cipher-text blocks and with at most 2^{43} trials in the search phase achieve the success probability 0.8925 with 10 approximations from 14-round DES. Overall running time is essentially 2^{43} DES encryptions. Each of the approximations has at most 19 effective key bits(at most 18-bit subkey and a linear combination of the key bits), so the space amount is negligible. The cryptanalysis is easily implementable and practical.

Using more approximations increases the success probability further. For instance, with the first 24 best approximations it becomes 0.9006, though some approximations may have up to 43 effective key bits and the space requirement is heavy.

For 8-round DES the calculations were checked by experiments. 10^5 random keys were generated, for each of them 1.49×2^{17}, as in [11], random plain-text blocks were encrypted with 8-round DES. For 2^{43} trials in the search phase, the empirical success probability was 0.8933(0.8925 in theory again) with 10 approximation from 6-round DES, each with at most 19 effective key bits.

2 Notation

Let Y be a bit string of some length, then denote

$$\begin{aligned} Y\{i,j,..,k\} &= Y[i] \oplus Y[j] \oplus .. \oplus Y[k], \\ Y[i,j,..,k] &= [\,Y[i], Y[j], .., Y[k]\,]\,. \end{aligned}$$

Let $Y_i, Y_j, .., Y_k$ be bit strings of the same length then

$$Y_{\{i,j,..,k\}}[r] = Y_i[r] \oplus Y_j[r] \oplus .. \oplus Y_k[r].$$

In case of DES we keep the notation of [10]. That is all bit string entries are numbered from right to left, starting with 0, except the key bits numbered as in the DES specification: k_i, where $i = 1, .., 63$ and $i \neq 0 \bmod 8$. According to [10], any 14-round linear approximation implies

$$\Phi_i(D^i, K^i) = l^i(K) \tag{1}$$

with probability p_i for some explicit function Φ_i, where $l^i(K)$ is a linear function in 56 variables K, the cipher key bits, D^i are some plain-text, cipher-text bits and K^i is a subkey. The $|K^i| + 1$ bits $K^i, l^i(K)$ are called effective key bits for the approximation (1). Remark that in [10] only K^i are called effective. Let X_{i-1}, X_i denote the input to the i-th round and X_{i+1}, X_i denote the i-th round output. So X_0, X_1 and X_{17}, X_{16} are plain-text and cipher-text blocks respectively, where the initial permutation is ignored. Let K_j be 48-bit round key at round j. Then

$$\begin{aligned} & X_0\{7,18,24\} \oplus X_{17}\{15\} \oplus X_{16}\{7,18,24,29\} \\ \oplus\ & F_1(X_1, K_1)\{7,18,24\} \oplus F_{16}(X_{16}, K_{16})\{15\} \\ =\ & K_{\{3,5,7,9,11,13,15\}}[22] \oplus K_{\{4,8,12\}}[44] \end{aligned} \tag{2}$$

holds with probability $\frac{1}{2} - \frac{78125}{137438953472} = 1/2 - 5.6843 \times 10^{-7}$. That is the largest possible in absolute value bias over all approximations with at most one active S-box in rounds $2, 3, \dots, 15$, see equation (33) in [10]. So $\Phi_1(D^1, K^1) = l^1(K)$, where

$$\begin{aligned} D^1 &= X_0\{7,18,24\} \oplus X_{17}\{15\} \oplus X_{16}\{7,18,24,29\}, X_1[11,..,16], X_{16}[27,..,31,0], \\ K^1 &= K[3,4,18,22,25,28,37,39,42,54,57,59], \\ l^1(K) &= K\{4,7,13,14,39,45,46,49,50,59\}. \end{aligned}$$

(2) is listed first in Table 7 as well.

3 Constructing and solving MRHS linear equations

For each of the linear approximations (1) a critical region for rejecting wrong candidates for effective key bits $K^i, l^i(K)$ is defined. The success is not to reject the correct key. Success probability is easily computed for each region. As the approximations are considered independent, the overall success probability $1-\beta$ is the product of the success probabilities for the approximations taken separately. With critical regions one collects a system of Multiple Right Hand Side (MRHS) [13] linear equations in the key bits, one equation per an approximation. The equations are solved with Gluing Algorithm and the solutions are then brute forced one after the other as they are generated. The solutions to the system contain the correct key with probability $1-\beta$. The method is characterised by the number of plain-text/cipher-text blocks, the complexity of solving the MRHS system, the number of trials in the search phase and the success probability. The latter is maximised by choosing critical regions such that the number of final trials is prescribed.

3.1 Critical regions and error probabilities

Let n be the number of plain-texts and $\nu = \nu(a)$ denote the number of plain-texts, where $\Phi_i(D^i, a) = 0$. Let $p_i < \frac{1}{2}$. As the key K is fixed, there are two possibilities for $l^i(K)$.

First, assume $l^i(K) = 0$. By (1) we have $\mathbf{Pr}(\Phi_i(D^i, K^i) = 0) = p_i$ for correct K^i. One rejects $K^i = a$ if

$$\frac{\nu(a)}{n} \geq \frac{1}{2} + x_i,$$

otherwise $K^i = a$ is accepted. We compute the probability of rejecting $K^i = a$ under the condition that was correct and then wrong.

$$\beta_i = \mathbf{Pr}(\text{reject}\, K^i = a |\, \text{correct}\, K^i = a\, \text{and}\, l^i(K) = 0) = \mathbf{Pr}\left(\nu/n \geq 1/2 + x_i |\, p_i\right).$$

That is the probability the normalised number of successes in Bernoulli trials with success probability p_i is at least $1/2 + x_i$. So

$$\beta_i = \mathbf{Pr}\left(\frac{\nu - np_i}{\sqrt{np_iq_i}} \geq \sqrt{n}\,\frac{1/2 - p_i + x_i}{\sqrt{p_iq_i}}\,|\, p_i\right) \approx \mathbf{Pr}\left(N(0,1) \geq \sqrt{n}\,\frac{1/2 - p_i + x_i}{\sqrt{p_iq_i}}\right) \tag{3}$$

by de Moivre-Laplace theorem [3], where $q_i = 1 - p_i$ and $N(0,1)$ denotes a standard normal random variable. Then

$$\alpha_i = \mathbf{Pr}(\text{reject}\, K^i = a |\, \text{wrong}\, K^i = a\, \text{and}\, l^i(K) = 0) = \mathbf{Pr}\left(\nu/n \geq 1/2 + x_i |\, 1/2\right).$$

That is the probability the normalised number of successes in Bernoulli trials with success probability $1/2$ is at least $1/2 + x_i$. Then

$$\alpha_i = \mathbf{Pr}\left(\frac{\nu - n/2}{\sqrt{n/4}} \geq 2\sqrt{n}\, x_i \,|\, 1/2\right) \approx \mathbf{Pr}\left(N(0,1) \geq 2\sqrt{n}\, x_i\right). \tag{4}$$

Assume $l^i(K) = 1$ now, then $\mathbf{Pr}(\Phi_i(D^i, K^i) = 0) = q_i$ for correct K^i. One rejects $K^i = a$ if

$$\frac{\nu(a)}{n} \leq \frac{1}{2} - x_i,$$

otherwise $K^i = a$ is accepted. Similarly

$$\beta_i = \mathbf{Pr}(\text{reject}\, K^i = a |\, \text{correct}\, K^i = a\, \text{and}\, l^i(K) = 1) \approx \mathbf{Pr}\left(N(0,1) \geq \sqrt{n}\,\frac{1/2 - p_i + x_i}{\sqrt{p_iq_i}}\right),$$

and

$$\alpha_i = \mathbf{Pr}(\text{reject}\, K^i = a |\, \text{wrong}\, K^i = a\, \text{and}\, l^i(K) = 1) \approx \mathbf{Pr}\left(N(0,1) \geq 2\sqrt{n}\, x_i\right)$$

are true again. Therefore for $p_i < 1/2$ in both cases

$$\beta_i \approx \mathbf{Pr}\left(N(0,1) \geq \sqrt{n}\,\frac{1/2 - p_i + x_i}{\sqrt{p_i q_i}}\right), \quad \alpha_i \approx \mathbf{Pr}\left(N(0,1) \geq 2\sqrt{n}\,x_i\right). \tag{5}$$

Let $p_i > 1/2$, then the critical region is $\frac{\nu(a)}{n} \leq \frac{1}{2} - x_i$ for $l^i(K) = 0$ and $\frac{\nu(a)}{n} \geq \frac{1}{2} + x_i$ for $l^i(K) = 1$. Therefore for $p_i > 1/2$ in both cases

$$\beta_i \approx \mathbf{Pr}\left(N(0,1) \geq \sqrt{n}\,\frac{1/2 - q_i + x_i}{\sqrt{p_i q_i}}\right), \quad \alpha_i \approx \mathbf{Pr}\left(N(0,1) \geq 2\sqrt{n}\,x_i\right). \tag{6}$$

We summarise the criterion in Table 1.

Table 1: The criterion

$l^i(K)$	p_i	reject $K^i = a$ if
0	$< 1/2$	$\nu(a) \geq (1/2 + x_i)n$
1	$> 1/2$	$\nu(a) \geq (1/2 + x_i)n$
0	$> 1/2$	$\nu(a) \leq (1/2 - x_i)n$
1	$< 1/2$	$\nu(a) \leq (1/2 - x_i)n$

3.2 Collecting MRHS linear equations

Assume $l^i(K) = 0$ and let $a_{01}, \ldots, a_{0s}$ be the accepted values for K^i. Assume $l^i(K) = 1$ and let $a_{11}, \ldots, a_{1t}$ be the accepted values for K^i. One writes this fact as

$$\begin{array}{c} K^i \\ l^i(K) \end{array} = \left[\begin{array}{cccccc} a_{01} & \ldots & a_{0s} & a_{11} & \ldots & a_{1t} \\ 0 & \ldots & 0 & 1 & \ldots & 1 \end{array}\right], \tag{7}$$

where the columns on the right hand side are acceptable values for $\begin{array}{c} K^i \\ l^i(K) \end{array}$. Thus a MRHS linear equation is constructed, see definitions in [13]. For either $l^i(K) = 0$ or $l^i(K) = 1$ the probability that $K^i = a$ is $(1 - \alpha_i)$. Any $|K^i| + 1$-bit string is in the right hand side of (7) with probability $(1 - \alpha_i)$. Hence the equation (7) has $(1 - \alpha_i)2^{|K^i|+1}$ right hand sides on the average. The probability the correct value for $\begin{array}{c} K^i \\ l^i(K) \end{array}$ is a column on the right hand side of (7) is $1 - \beta_i$, the success probability when using this approximation. A system of such equations, one from each of the approximations is thus collected. Let N approximations be available, so we have (7) for $i = 1, \ldots, N$.

3.3 Solving MRHS linear equations

Though there are several methods to solve MRHS linear equations in [13], we use Gluing Algorithm. The Algorithm starts with (7) at $i = 1$. Formally, at each step a new MRHS linear equation

$$M_i K = [c_{i,1}, \ldots, c_{i,s_i}] \, , \tag{8}$$

is constructed, where M_i is a matrix of size $r_i \times |K|$ of full rank and where $c_{i,1}, \ldots, c_{i,s_i}$ are r_i-bit strings. (8) is a gluing of the previous equation

$$M_{i-1} K = \left[c_{i-1,1}, \ldots, c_{i-1,s_{i-1}}\right]$$

and (7) for $i = 2, \ldots, N$. In fact, the method is implementable as a walk over a search tree: the current c_{i-1,j_1} is sought to be extended to some c_{i,j_2} with a table look up and few Xor's, see Section 6.

As the right hand sides of the initial equations (7) are generated independently, a particular c appears in the right hand side of (8) with probability $\prod_{j=1}^{i}(1-\alpha_j)$ by Lemma 1, Section 6. Therefore $s_i = 2^{r_i} \prod_{j=1}^{i}(1-\alpha_j)$ on the average. The complexity of constructing the final equation (8), that is for $i = N$, is proportional to

$$s_1 + s_2 + \ldots + s_N = 2^{r_1}(1-\alpha_1) + 2^{r_2}(1-\alpha_1)(1-\alpha_2) + \ldots + 2^{r_N}\prod_{i=1}^{N}(1-\alpha_i) \tag{9}$$

Xor's of bit strings of various length, see (15) in Section 6. One then solves the linear system $M_N K = c$ for each generated c to find key candidates to brute force. The number of the key candidates is $2^{|K|}\prod_{i=1}^{N}(1-\alpha_i)$ on the average.

3.4 Success probability

To brute force $2^{|K|-t}$ key candidates, there should be

$$\prod_{i=1}^{N}(1-\alpha_i) = 2^{-t}. \tag{10}$$

The success probability is

$$1-\beta = \prod_{i=1}^{N}(1-\beta_i). \tag{11}$$

By choosing critical regions, that is $x_1, \ldots, x_N$, (11) is maximised under (10). We do not provide any analytic formula for this maximum here. At least for relatively low N as below that is computed by a reasonable search.

4 16-round DES with 10 approximations

Let $n = 2^{43}$ and we want to brute force at most 2^{43} key candidates. One takes 10 linear approximations (1) with the best biases listed in the rows $I = [1, .., 7, 10, 11, 24]$ of Table 7, each with at most 19 effective key bits. For each approximation (1), that is by using 10 counters as in [10], the frequencies λ_A of $D^i = A$ over all available plain-text/cipher-text blocks are collected. The cost per one plain-text is negligible in comparison with one 16-round DES encryption. Then

$$\nu(a) = \sum_A \lambda_A \, (1 \oplus \Phi_i(A, a)), \tag{12}$$

where a is mostly Xor'ed to A by the DES round function construction. So Fast Fourier Transform is used to compute the convolution (12) in time $|K^i| \, 2^{|K^i|+1}$ for all values of K^i as in [2] and for each of those approximations.

By searching $\alpha = [\alpha_i, i \in I]$ such that $\prod_{i \in I}(1 - \alpha_i) = 2^{-13}$, the success probability $\prod_{i \in I}(1 - \beta_i)$ is maximised to 0.8925 at α_i and x_i shown in Table 2. Table 3 presents the growth of the rank r_i and the number of right hand sides s_i in the equations (8) produced by the Gluing Algorithm. The complexity of the Gluing Algorithm is at most 2^{40} of 17-bit Xor's by (15). One does not need essentially more memory than to keep initial equations (7) as the algorithm is implementable, after some precomputation, with a search tree, see [14] and Section 6. As a right hand side c of the final equation is produced, one finds $2^{56-r_{10}} = 8$ key candidates, solutions to $M_{10}K = c$, to brute force. The matrix M_{10} is almost diagonal except one row contains two non-zero entries. So each c specifies key bits k_i for

Table 2: 16-round DES, probabilities α_i and critical region constants x_i

α_i	x_i
0.9310	$-2.5009175980 \times 10^{-7}$
0.9310	$-2.5009175980 \times 10^{-7}$
0.7120	$-9.4786295453 \times 10^{-8}$
0.7120	$-9.4786295453 \times 10^{-8}$
0.3235	$7.7660577131 \times 10^{-8}$
0.3235	$7.7660577131 \times 10^{-8}$
0.1460	$1.7836882403 \times 10^{-7}$
0.1460	$1.7836882403 \times 10^{-7}$
0.0372	$3.0250394719 \times 10^{-7}$
0.0372	$3.0250394719 \times 10^{-7}$

$i \in \{1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29, 30, 31, 33, 34,$

Table 3: Gluing Algorithm complexity

i	1	2	3	4	5	6	7	8	9	10
r_i	13	26	37	42	42	42	42	42	48	53
$\log_2(s_i)$	9.14	18.28	27.48	30.69	30.12	29.56	29.33	29.10	35.05	40.00

$$35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 62, 63\}$$

and $k_{23} \oplus k_{61}$. That makes 2^{43} trials on the average for all right hand sides together during the search phase.

5 8-round DES experiments with 10 approximations

In this Section the results of the experiments on 8-round DES are reported. Let $n = 1.49 \times 2^{17}$ and we want to brute force at most 2^{43} key candidates. One takes 10 linear approximations (1) with the best biases listed in the rows $I = [1, .., 7, 10, 11, 24]$ of Table 6, where 1-round approximations are from Table 5. By searching $\alpha = [\alpha_i, i \in I]$ such that $\prod_{i \in I}(1 - \alpha_i) = 2^{-13}$, one maximises the success probability $\prod_{i \in I}(1 - \beta_i)$ to 0.8925 at α_i and x_i shown in Table 4. The correct K is rejected if $\nu(K^i)$ is in the critical region defined by $l^i(K), p_i, x_i$ for at least one i.

For each of 10^5 random 56-bit keys K, n random plain-texts were generated and encrypted with 8-round DES. The frequency of "the correct K is accepted", i.e. empirical success probability, was 0.8933. That is very close to the success probability computed theoretically.

6 Gluing Algorithm

Let

$$A_1X = [B_1], \quad A_2X = [B_2] \tag{13}$$

be a system of two MRHS linear equations, where X is a column vector of unknowns, A_1, A_2 are matrices of full rank, that is no linearly dependent rows, with v_1, v_2 rows respectively. Let B_1, B_2 be matrices whose columns are possible right-hand sides for A_iX. Therefore $X = x$ is a solution to (13) if the columns A_1x, A_2x belong to the columns of B_1, B_2 respectively. Let u_i be the number of columns in B_i.

There exists an equation $MX = [L]$ whose solutions are all common solutions to (13). $MX = [L]$ is constructed by the following steps. One can assume A_1 is already triangulated.

Table 4: 8 round DES, probabilities α_i and critical region constants x_i

α_i	x_i
0.9300	$-1.6699273321 \times 10^{-3}$
0.9300	$-1.6699273321 \times 10^{-3}$
0.7150	$-6.4277918792 \times 10^{-4}$
0.7150	$-6.4277918792 \times 10^{-4}$
0.3255	$5.1174922072 \times 10^{-4}$
0.3255	$5.1174922072 \times 10^{-4}$
0.1470	$1.1871782307 \times 10^{-3}$
0.1470	$1.1871782307 \times 10^{-3}$
0.0369	$2.0217334996 \times 10^{-3}$
0.0369	$2.0217334996 \times 10^{-3}$

1. Triangulate the concatenation of A_1, A_2 by a linear transform U and get:
$$M = U\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} A_1 \\ U_1A_1 + U_2A_2 \end{pmatrix}, \quad U = \begin{pmatrix} E & 0 \\ U_1 & U_2 \end{pmatrix},$$
where E is an identity matrix. Then
$$\bar{B}_1 = U\begin{pmatrix} B_1 \\ 0 \end{pmatrix} = \begin{pmatrix} B_1 \\ U_1B_1 \end{pmatrix}, \quad \bar{B}_2 = U\begin{pmatrix} 0 \\ B_2 \end{pmatrix} = \begin{pmatrix} 0 \\ U_2B_2 \end{pmatrix}.$$

2. Let c_1, c_2 be columns of $\bar{B}_1, \bar{B}_2$ respectively. Then $c_1 \oplus c_2$ defines a right hand side to MX if and only if c_1, c_2 coincide on the entries, where M has a zero row. Let s be the number of resulting possible right hand sides to MX.

3. $M, \bar{B}_1, \bar{B}_2$ are found, $MX = [L]$ is constructed by u_1 look ups and s of v_2-bit Xor's.

Assume now a system
$$A_1X = [B_1],\ A_2X = [B_2], \dots,\ A_NX = [B_N], \tag{14}$$
where A_i are of full rank. Starting with the first equation, one constructs $M_iX = [L_i]$ $(i = 1, \dots, N)$ by gluing of $M_{i-1}X = [L_{i-1}]$ and $A_iX = [B_i]$. As all linear algebra steps may be precomputed, the algorithm reduces to look ups and vector Xor's, see below. The constructing of the columns L_i is implementable recursively with a search tree and therefore one only keeps the initial equations (14) after the precomputation and the current column at each step. Finally, the system solutions are the solutions to $M_NX = [L_N]$. The complexity is $\sum_{i=1}^{N} s_i$ Xor's of bit strings of various length and $\sum_{i=1}^{N-1} s_i$ look ups, where s_i is the number of columns in L_i.

We consider the algorithm in detail. Let v_i be the number of rows in A_i and u_i the number of columns in B_i.

Precomputation. Initially, $M_1 = A_1$ is already triangulated and let $\bar{B}_1 = B_1$. For $i = 2, \ldots, N$ the matrix $\left(\begin{array}{c} M_{i-1} \\ A_i \end{array}\right)$ is triangulated by computing a matrix M_i of size $(\sum_{t=1}^{i} v_t) \times |X|$:

$$\begin{aligned} M_i &= U \left(\begin{array}{c} M_{i-1} \\ A_i \end{array}\right) = \left(\begin{array}{c} M_{i-1} \\ U_1 M_{i-1} + U_2 A_i \end{array}\right), \\ U &= \left(\begin{array}{cc} E & 0 \\ U_1 & U_2 \end{array}\right), \end{aligned}$$

where E is an identity matrix. Then matrices $\bar{B}_j$ of size $(\sum_{t=1}^{i} v_t) \times u_j$ for $j = 1, \ldots, i$ are computed by

$$\begin{aligned} \bar{B}_j &\leftarrow U \left(\begin{array}{c} \bar{B}_j \\ 0 \end{array}\right) = \left(\begin{array}{c} \bar{B}_j \\ U_1 \bar{B}_j \end{array}\right), \quad j = 1, \ldots, i-1, \\ \bar{B}_i &= U \left(\begin{array}{c} 0 \\ B_i \end{array}\right) = \left(\begin{array}{c} 0 \\ U_2 B_i \end{array}\right). \end{aligned}$$

The precomputation is finished.

Let b_j be a column in $\bar{B}_j$, $j = 1, \ldots, i$. Then $c_i = b_1 \oplus \ldots \oplus b_i$, after reducing to the first $\sum_{j=1}^{i} v_j$ entries, is a correct right hand side for $M_i X$ if and only if the entries of c_i in the positions, where M_i has a zero row, are zeros.

So given $c_{i-1} = b_1 \oplus \ldots \oplus b_{i-1}$, one looks up b_i such that c_{i-1}, b_i coincide on the entries in the positions $\sum_{t=1}^{i-1} v_t + 1, \ldots, \sum_{t=1}^{i} v_t$, where M_i has zero rows. The first $\sum_{j=1}^{i-1} v_j$ entries of b_i are zeros, so $c_i = c_{i-1} \oplus b_i$ is computed with $\sum_{j=i}^{N} v_j$-bit Xor's. Let s_i be the number of the right hand sides to $M_i X$, then the complexity of the algorithm is at most

$$\sum_{i=1}^{N} s_i \, (\sum_{j=i}^{N} v_j) \tag{15}$$

bit Xor's and $\sum_{i=1}^{N-1} s_i$ look ups.

Example. Let $X = (x_1, x_2, x_3, x_4)$ and there are three equations:

$$\begin{aligned} \left(\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right) X &= \left[\begin{array}{cc} 0 & 1 \\ 0 & 1 \end{array}\right], \left(\begin{array}{cccc} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) X = \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array}\right], \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right) X &= \left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array}\right]. \end{aligned}$$

After precomputation

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \bar{B}_1 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad \bar{B}_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \bar{B}_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

One is to combine columns b_i in $\bar{B}_i$ such that $c_2 = b_1 \oplus b_2$ has zero in the position 4 and $c_3 = c_2 \oplus b_3$ has zero in the position 6. So the final MRHS linear equation is

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} X = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and the solutions are $X = (0,0,0,0), (0,1,1,0)$. There is a better way to solve those particular equations. Each of them is equivalent to an ordinary linear equation:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} X = 0, \quad \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} X = 0, \quad \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} X = 0,$$

see [13] for detail. The solution follows.

Lemma 1 *Let $MX = [L]$ be a gluing of $A_i X = [B_i]$, $i = 1, \ldots, N$ and γ_i, γ be the probability a particular bit-string is a column in B_i, L respectively. Assume the columns in B_i are independently chosen, then*

$$\gamma = \prod_{i=1}^{N} \gamma_i.$$

Proof. Each column in L is determined by a column from each of B_i. Vice versa, a column in L determines those columns in B_i uniquely. That implies the lemma.

References

[1] A. Biryukov, C. De Cannière, and M. Quisquater, *On Multiple Linear Approximations,* in CRYPTO'04(M.Franklin ed.), LNCS vol. 3152, Springer, 2004, pp. 1–22.

[2] B. Collard, F. X. Standaert, and J.-J. Quisquater, *Improving the Time Complexity of Matsui's Linear Cryptanalysis,* in ICISC'07(K.-H. Nam and G. Rhee eds.), LNCS vol. 4717, Springer, 2007, pp. 77–88.

[3] W. Feller, *An Introduction to Probability Theory and its Applications, 3rd ed.*, vol. 1, John Wiley & Sons, 1968.

[4] C. Harpes, G. Kramer, and J. Massey, *A generalisation of linear cryptanalysis and the applicability of Matsui's piling-up lemma*, in Eurocrypt'95 (L.C. Guillou and J.-J. Quisquater eds.), LNCS vol. 921, Springer, 1995, pp. 24–38.

[5] M. Hermelin, *Multidimensional Linear Cryptanalysis*, PhD thesis, Aalto University-School of Science and Technology, Finland, 2010.

[6] P. Junod and A. Canteaut(eds.), *Advanced Linear Cryptanalysis of Block and Stream Ciphers*, IOS Press, 2011.

[7] B. S. Kaliski and M. J. Robshaw, *Linear cryptanalysis using multiple approximations,* in CRYPTO'94 (Y. Desmedt, ed.), LNCS vol. 839, Springer, 1994, pp. 26–39.

[8] L. R. Knudsen and J. E. Mathiassen, *A chosen-plaintext linear attack on DES,* in FSE'00 (B. Schneier, ed.), LNCS vol. 1978, Springer, 2001, pp. 262–272.

[9] D. Davies and S. Murphy, *Pairs and Triples of DES S-Boxes*, J. Cryptology, vol. 8(1995), pp. 1–25.

[10] M. Matsui, *Linear Cryptanalysis of DES Cipher(I)*, preprint, 1993.

[11] M. Matsui, *The First Experimental Cryptanalysis of the Data Encryption Standard*, in CRYPTO'94 (Y.Desmedt, ed), LNCS 839, Springer, 1994, pp. 1-11.

[12] M. Matsui, *On the correlation between the order of S-boxes and the strength of DES*, in Eurocrypt'94(A. De Santis ed.), LNCS 950, Springer, 1995, pp. 366-375.

[13] H. Raddum and I. Semaev, *Solving Multiple Right Hand Sides linear equations*, Des., Codes Cryptogr., vol. 49 (2008), pp. 147–160 , Springer.

[14] I. Semaev, *On solving sparse algebraic equations over finite fields*, Des. Codes Cryptogr., vol. 49 (2008), pp. 47–60, Springer.

7 Appendix: 6-round and 14-round DES Linear Approximations

Linear approximations of 14-round DES with bias in absolute value $\geq 2.2737 \times 10^{-7}$ are listed in Table 7. They are produced according to an algorithm in [12], where at most one S-box was active in each of the rounds. A row consists of 17 entries: the approximation number in the list, which 1-round approximation from Table 5 was used in each of the 14 rounds, where "0" means a trivial approximation $0 = 0$ in that round, the number of the effective key bits $K^i, l^i(K)$, and finally 10^7 times the approximation bias. Similarly, linear approximations of 6-round DES with bias in absolute value $\geq 1.5259 \times 10^{-3}$ are listed in Table 6.

Each row in Table 5 has 5 entries: the 1-round approximation number in the list, the S-box number, input and output masks and the approximation bias.

Table 5: 1-round approximations

1	5	16	14	$\frac{5}{32}$
2	1	4	4	$-\frac{1}{32}$
3	5	16	15	$-\frac{5}{16}$
4	5	34	14	$-\frac{1}{4}$
5	5	34	15	$-\frac{3}{16}$
6	1	27	4	$-\frac{5}{32}$
7	5	9	14	$-\frac{1}{8}$
8	1	20	4	$-\frac{1}{8}$
9	1	38	4	$\frac{1}{8}$
10	1	39	4	$-\frac{1}{8}$
11	1	44	4	$\frac{1}{8}$
12	1	53	4	$-\frac{1}{8}$
13	1	62	4	$-\frac{1}{8}$

Table 6: Best 6-round approximations

1	0	1	2	3	0	3	13	-3.8147
2	3	0	3	2	1	0	13	-3.8147
3	0	3	2	1	0	4	19	-3.0518
4	4	0	1	2	3	0	19	-3.0518
5	0	1	2	3	0	5	19	-2.2888
6	5	0	3	2	1	0	19	-2.2888
7	0	3	2	1	0	1	13	1.9073
8	6	3	0	3	2	4	41	- 1.9073
9	4	2	3	0	3	6	40	- 1.9073
10	1	0	1	2	3	0	13	1.9073
11	0	3	2	1	0	7	17	-1.5259
12	8	3	0	3	2	4	28	-1.5259
13	9	3	0	3	2	4	37	1.5259
14	10	3	0	3	2	4	43	- 1.5259
15	11	3	0	3	2	4	32	1.5259
16	12	3	0	3	2	4	40	- 1.5259
17	13	3	0	3	2	4	41	- 1.5259
18	4	2	3	0	3	8	27	- 1.5259
19	4	2	3	0	3	9	37	1.5259
20	4	2	3	0	3	10	43	- 1.5259
21	4	2	3	0	3	11	32	1.5259
22	4	2	3	0	3	12	39	- 1.5259
23	4	2	3	0	3	13	40	- 1.5259
24	7	0	1	2	3	0	17	- 1.5259

Table 7: Best 14-round approximations

1	0	1	2	3	0	3	2	1	0	1	2	3	0	3	13	- 5.6843
2	3	0	3	2	1	0	1	2	3	0	3	2	1	0	13	- 5.6843
3	0	3	2	1	0	1	2	3	0	3	2	1	0	4	19	- 4.5475
4	4	0	1	2	3	0	3	2	1	0	1	2	3	0	19	- 4.5475
5	0	1	2	3	0	3	2	1	0	1	2	3	0	5	19	- 3.4106
6	5	0	3	2	1	0	1	2	3	0	3	2	1	0	19	- 3.4106
7	0	3	2	1	0	1	2	3	0	3	2	1	0	1	13	2.8422
8	6	3	0	3	2	1	0	1	2	3	0	3	2	4	41	- 2.8422
9	4	2	3	0	3	2	1	0	1	2	3	0	3	6	40	- 2.8422
10	1	0	1	2	3	0	3	2	1	0	1	2	3	0	13	2.8422
11	0	3	2	1	0	1	2	3	0	3	2	1	0	7	17	- 2.2737
12	8	3	0	3	2	1	0	1	2	3	0	3	2	4	28	- 2.2737
13	9	3	0	3	2	1	0	1	2	3	0	3	2	4	37	2.2737
14	10	3	0	3	2	1	0	1	2	3	0	3	2	4	43	- 2.2737
15	11	3	0	3	2	1	0	1	2	3	0	3	2	4	32	2.2737
16	12	3	0	3	2	1	0	1	2	3	0	3	2	4	40	- 2.2737
17	13	3	0	3	2	1	0	1	2	3	0	3	2	4	41	- 2.2737
18	4	2	3	0	3	2	1	0	1	2	3	0	3	8	27	- 2.2737
19	4	2	3	0	3	2	1	0	1	2	3	0	3	9	37	2.2737
20	4	2	3	0	3	2	1	0	1	2	3	0	3	10	43	- 2.2737
21	4	2	3	0	3	2	1	0	1	2	3	0	3	11	32	2.2737
22	4	2	3	0	3	2	1	0	1	2	3	0	3	12	39	- 2.2737
23	4	2	3	0	3	2	1	0	1	2	3	0	3	13	40	- 2.2737
24	7	0	1	2	3	0	3	2	1	0	1	2	3	0	17	- 2.2737