Separable Statistics and Multidimensional Linear Cryptanalysis

S. Fauskanger and I. Semaev

October 9, 2017

Abstract

Multidimensional linear cryptanalysis of block ciphers is improved in this work by introducing a number of new ideas. Firstly, formulae is given to compute approximate multidimensional distributions of encryption internal bits. Conventional statistics like LLR(Logarithmic Likelihood Ratio) do not fit to work in Matsui's Algorithm 2 for large dimension data, as the observation depend on too many cipher key bits. So, secondly, a new statistic which reflects the structure of the cipher round is constructed instead. Thirdly, computing the statistic values which fall into a critical region is presented as an optimisation problem for which an efficient algorithm is suggested. The algorithm works much faster than brute forcing all relevant key bits to compute the statistic. An attack for 16-round DES was implemented. We got an improvement over Matsui's attack on DES in data and time complexity keeping success probability the same.

1 Introduction

Linear Cryptanalysis is a statistical approach in the cryptanalysis of symmetric ciphers. It is a known plain-text attack which does not require any special plain-text/cipher-text pairs and therefore is a very important tool in practical decryption. It was introduced by Matsui in [17, 18] as an attack to DES. Another approach in statistical cryptanalysis was earlier published in [16]. Linear Cryptanalysis exploits the fact that an xor of certain plaint-text, cipher-text and key bits is zero with some a priori computed probability different from 1/2. Such combinations were called "linear approximations" in [17]. The probability itself somehow depends on the cipher key. The method is more efficient if the probability is far from 1/2, one says a "linear approximation" is more biased in this case. The attack is characterised by the number of necessary plain-text/cipher-text pairs(data complexity), by the complexity of ranking relevant sub-keys according to the value of a statistic and the size of the final brute force(time complexity), and by success probability.

Two attacks Algorithm 1 and Algorithm 2 were suggested in [17]. Algorithm 1 uses r-round approximations, while Algorithm 2 uses $r-1$ or $r-2$-round approximations to attack r-round cipher. In Algorithm 2 an observation on "linear approximations" may depend on some key bits from the first and the last rounds of the cipher and the "linear approximations" themselves are generally more biased. So one may recover more cipher key bits at a lower price, in other words, the method requires a lower amount of plain-text/cipher-text pairs and is more efficient.

For 16-round DES, Matsui shows how to determine candidates for relevant key bits or key bit linear combinations by Algorithm 2 with $n = 2^{43}$ plain-text/cipher-text blocks and success probability 0.85, then 2^{43} trials are run to get the correct key [18]. The success probability was found experimentally for 8-round cipher with 10^4 attack applications and then extrapolated to 16-round DES. Two 14-round "linear approximations" were there used together.

Only few improvements with relation to DES have been published since Matsui's work. In [15] a chosen plaint-text linear attack was suggested and in [5] time complexity of the attack first stage was reduced by using Fast Fourier Transform. It was experimentally found in [6, 7] that time complexity of Matsui's attack on DES may be decreased with a better ranking of the values of relevant sub-key bits, though data complexity and success probability remain the same. However, the success probability was determined experimentally with only 21 attack applications, which does not seem enough to justify the figure 0.85.

How to improve Algorithm 1 with more than two "linear approximations" the distribution of which depend on the same key bits was shown in [14]. In [1] a framework for using many "linear approximations" considered statistically independent was proposed, though no practical cryptanalysis of 16-round DES was presented. The sub-keys relevant to the observations on "linear approximations" were considered disjoint as in [18]. Linear cryptanalysis was further extended in different ways in [9] and [10], see [13]. For instance, [10] made use multidimensional analysis instead of one-dimensional. A good survey of publications on using multiple "linear approximations" is in [11]. Recently, a series of papers on linear cryptanalysis of PRESENT were published, see for instance [4, 3, 2], which provide with a new insight into the area of multivariate and multidimensional linear cryptanalysis of block ciphers. Most of the methods are based on the assumption that the "linear approximations" are statistically independent, which may be true only to some extent. On the other hand, no algorithm for computing joint a priori distributions(approximate joint distributions) of multiple "linear approximations" in block ciphers was published. As a priori distribution is unknown, it looks difficult to predict the success probability of relevant statistical attacks. An attack with low success probability has limited usefulness even if has a low complexity. The same limitation holds in multidimensional linear cryptanalysis of [10]. The present work solves this deficiency by giving formulae to compute multidimensional probability distributions in Feistel ciphers, see Section 7. Similar formulae hold for SPN type ciphers, though that is not presented in the paper.

Two open problems related to Algorithm 2 were posed in [1]. First, how to merge data from different "linear 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. A solution to these problems was found in [24]. In particular, an attack for 16-round DES with 2^{43} data and same amount of the final brute force trials, and with success probability 0.89 was there described. The probability was predicted by theoretical means and the prediction was found correct experimentally for a similar method in case of 8-round DES with 10^5 method applications. The attack uses 10 best 14-round "linear approximations", considered statistically independent. The distributions of those "linear approximations" and observations on them depend on 53 DES key bits. By solving a particular optimisation problem(stated in its generality in Section 5 of the present work) one finds a set of size 2^{40} of 53-bit key-candidates at price $\approx 2^{40}$ computations, that is without brute forcing 2^{53} values of the statistic. The probability that a correct 53-bit sub-key is in this set is 0.89.

The present work is far and away generalisation of [24]. Instead of "linear approximations" certain projections(sub-strings of bits or multidimensional linear functions) of the encryption internal states are used. In contrast with [24] we do not here suppose the projections to be statistically independent. We are able to compute their approximate joint a priori distributions and therefore predict correctly success probability of the attack in addition to other things. We implemented our method and got improvement over Matsui's result on 16-round DES in data and time complexity while success probability remains the same, see Section 3.

2 The Problem

Let $\mathtt{x}$ be a vectorial random variable which incorporates some bits from the encryption first round output and some input bits to the last round as $\mathtt{x} = (X, Y)$ in Fig.1. Like in Matsui's linear cryptanalysis, an approximate distribution of $\mathtt{x}$ may be a priori computed from the encryption algorithm specification. It commonly depends on a relatively low number of the cipher key bits(linear combination of the key bits) denoted $\mathtt{key}$. On the other hand, the observation on $\mathtt{x}$ depends on the available plain-text/cipher-text blocks and some key bits from the first and the last rounds denoted $\mathtt{Key}$. Assume one guesses relevant key bits $\bar{K} = \mathtt{key}, \mathtt{Key}$. If the guess was correct, then the observation follows a priori distribution. If not, then the observation follows a distribution which is close to the uniform distribution. We assume it is uniform by ignoring the case when the guess on $\mathtt{Key}$ was correct but the guess on $\mathtt{key}$ was not. In that case the observation usually follows a permuted a priori distribution. The assumption was used by many authors before and its correctness is supported by experiments with DES in the present work as well.

A multidimensional variation of the linear cryptanalysis developed in [12, 10] may be applicable. One can use a logarithmic likelihood ratio (LLR) statistic, which depends on both the distribution and the observation, so it depends on $\bar{K}$. That provides the most powerful statistical test to distinguish correct and incorrect values of $\bar{K}$. However the method is not efficient if the size(rank) of $\bar{K}$ is large. Really, one is to range $2^{|\bar{K}|}$ values of

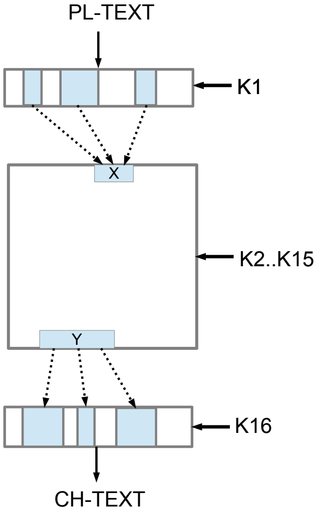

Figure 1: A 16-round Cipher Cryptanalysis

the key bits according to the value of the statistic.

At the same time the distribution of the projections(functions) $h_i(\mathtt{x})$ and observations on them may depend on a much lower number of the key bits $\bar{K}_i = \mathtt{key}_i, \mathtt{Key}_i$. At least that holds for DES, see Section 8 below, and other modern block cipher based on small S-boxes. For DES the values of $\bar{K}_i$ are linear projections of a $\bar{K}$-value. The sub-keys $\bar{K}_i$ which affect the distributions and the observations for the projections h_i may partly coincide or be linearly dependent. In this paper we consider how by observing the values of several projections $h_i(\mathtt{x})$ reconstruct a set of $\bar{K}$-candidates which contains the correct value with a prescribed success probability. We show that can be accomplished by solving efficiently an optimisation problem without brute forcing the values of $\bar{K}$. Also we answer what the size of set of $\bar{K}$-candidates is. To this end we will use a novel statistic which reflects the structure of the cipher round. The new statistic is a linear combination of LLR statistics for different projections and we do not need that they are statistically independent.

3 Our Contributions

This paper contains the following contributions

1. An approximate probabilistic description of a Feistel ciphers is suggested in Section 7. A convolution type formula for computing approximate probability distribution of multidimensional random variables $\mathtt{x}$ constructed with internal bits of the encryption

is there derived.

2. A novel statistic which combines the LLR statistics for different projections $h_i(\mathtt{x})$ is used in this cryptanalysis, see Section 4. The statistic is approximately separable, which allows to analyse the observation on different projections separately. If several statistically independent $\mathtt{x}$ are available, several such separable statistics may be used simultaneously. In this cryptanalysis of DES we use two vectorial random variables $\mathtt{x}_1, \mathtt{x}_2$ produced by DES symmetry and considered independent, see Section 8, so two separable statistics are used.

3. The distribution of the statistic under a correct value of $\bar{K}$ is determined in Section 4.2. We find the distribution of the statistic under incorrect key assumption. Also a critical region and the success probability of the attack are defined in Section 6.1. The latter is the probability that the value of the statistic computed under the correct value of $\bar{K}$ falls into this region. The number of incorrect values of $\bar{K}$ for which the value of the statistic falls into the region is computed as well. Those values are to be brute forced.

4. We represent the problem of reconstructing $\bar{K}$-values which fall into the critical region from $\bar{K}_i$-values as an optimisation problem stated in Section 5. So a general algorithm to solve the problem described in Section 5.2 is applicable. It is based on the idea of gluing of $\bar{K}_i$-values developed in [22, 21].

5. Our approach allows to find the number of necessary plain-text/cipher-text blocks, given desired success probability and the number of $\bar{K}$-candidates to be brute forced.

6. The attack was implemented for 16-round DES, see Section 9. We used two independent separable statistics, each based on 14 of 10-bit projections with 54 DES key bits involved overall. With $n = 2^{41.8}$ plain-text blocks and success rate 0.85(computed theoretically) we found $2^{39.8}$(also predicted theoretically) key-candidates to 54-bit DES sub-key. That makes $2^{41.8}$ of 56-bit DES keys to brute force. Though search tree to compute statistics values which fall into the critical region incorporated $2^{45.5}$ nodes, constructing one node(checking two linear inequalities with real numbers) is an inferior operation in comparison with one DES encryption. So the attack provides an improvement over Matsui's cryptanalysis of DES in [18].

4 Separable Statistics

Let an observation $\nu = (\nu_1, \ldots, \nu_m)$ on m projections(functions) $h_i(\mathtt{x})$ be available, where ν_i denotes a vector of observations on the outcomes of $h_i(\mathtt{x})$. We do not assume the projections are statistically independent. In this cryptanalysis ν_i is a function in available plain-text/cipher-text blocks and $\bar{K}_i$. By the statistic we mean a function which depends on the observation ν. A statistic $S(\nu)$ is called separable if it can be represented as

$$S(\nu) = \sum_{i=1}^{m} S_i(\nu_i). \tag{1}$$

This property allows analysing data ν in parts by analysing ν_i separately. The notion was introduced in [20] to study statistical test to distinguish polynomial distributions. One decides the value of $\bar{K}$ is correct if $S(\nu) > z$ for some threshold z. That defines a critical region. If the distribution of S is known, then the value z is determined by a prescribed success probability. One can also determine the average number of wrong values of $\bar{K}$ which pass the test as well. That defines the complexity of the attack.

The values of $\bar{K}_i$ which agree on common key bits or, more generally, common linear subspaces of the key bits are to be combined to get a value of $\bar{K}$ which falls into the critical region. That is an instance of the optimisation problem described in Section 5. An efficient algorithm to solve it is introduced in Section 5.2. The algorithm takes advantage of the fact that the statistic is separable. The algorithm implements walking over a search tree by creating new nodes if certain linear inequalities, implications of $S(\nu) > z$, are satisfied. The computation cost is much lower than $2^{|\bar{K}|}$. One may take advantage of several statistically independent $\mathtt{x}$, so several statistics of that kind may be used simultaneously.

Another statistic is derived in Appendix 2. That is based on a more direct application of Neyman-Pearson approach. However it is separable only for statistically independent projections. That is not true for the bunches of the projection (27) and (28) in this cryptanalysis of DES as all the projections inside each bunch are statistically dependent. Therefore, the second statistic does not fit well within this cryptanalysis and won't be used.

4.1 Notation

Let $\mathtt{x}$ be a random variable with N outcomes denoted $1, 2, \ldots, N$. Assume $\mathtt{x}$ may have two probability distributions: $P = (p_1, \ldots, p_N)$ and $Q = (q_1, \ldots, q_N)$. Let

$$v(n) = (v_1, v_2, \ldots, v_N)$$

denote outcome frequencies for $\mathtt{x}$ after n trials, so that $\sum_{j=1}^{N} v_i = n$. Also let $h_i, i = 1, \ldots, m$ be functions defined on $\{1, 2, \ldots, N\}$ with values in $\{1, 2, \ldots, N_i\}$. We call them projections and let

$$\nu_i = \nu_i(n) = (\nu_{i1}, \ldots, \nu_{iN_i}),\ i = 1, \ldots, m,$$

denote outcome frequencies for $h_i(\mathtt{x})$ after n trials, so $\sum_{j=1}^{N_i} \nu_{ij} = n$. We therefore have $\nu_{ib} = \sum_{h_i(a)=b} v_a$. Also let $\nu = \nu(n) = (\nu_1, \ldots, \nu_m)$.

4.2 Main Statistic

Let $\mathtt{x}$ have distribution P. Then $P_i = (p_{i1}, \ldots, p_{iN_i})$ denotes the distribution of $h_i(\mathtt{x})$, where

$$p_{ib} = \mathbf{Pr}(h_i(\mathtt{x}) = b) = \sum_{h_i(a)=b} p_a.$$

Similarly, if $\mathtt{x}$ is distributed according to Q, then $Q_i = (q_{i1}, \ldots, q_{iN_i})$ is the distribution of $h_i(\mathtt{x})$. For each i and b we assume $p_{ib}, q_{ib} \neq 0$. Consider the LLR statistic for h_i

$$LLR_i(\nu_i) = \sum_{b=1}^{N_i} \nu_{ib} \ln\left(\frac{q_{ib}}{p_{ib}}\right) = \sum_{a=1}^{N} v_a \ln\left(\frac{q_{ih_i(a)}}{p_{ih_i(a)}}\right). \tag{2}$$

By a standard argument, see for instance [12], $LLR_i(\nu_i) = \sum_{j=1}^{n} R_{it}$, where R_{it} are identically distributed random variables. Let μ_{iP}, σ_{iP} denote the expectation and the variance of R_{it} under condition that ν_i follows the distribution P_i. By [12], if the distributions P and Q are close enough, then $\mu_{iP} \approx -\mu_{iQ}$ and $\sigma_{iP} \approx \sigma_{iQ}$. We will prove a more general statement.

Let $s(\nu) = (LLR_1(\nu_1), \ldots, LLR_m(\nu_m))$. Then $s(\nu) = \sum_{t=1}^{n} R_t$, where $R_t = (R_{1t}, .., R_{mt})$ are identically distributed vectorial random variables. The expectation of R_t under condition that ν follows the distribution P is $\mu_P = (\mu_{1P}, \ldots, \mu_{mP})$. Let C_P denote the covariance matrix of R_t. Let the distributions P and Q be close enough, then $\mu_Q \approx -\mu_P$ and $C_P \approx C_Q$ by the following Lemma.

Lemma 1 *Let $q_a = p_a + \epsilon_a$, where $|\epsilon_a/p_a| \leq \delta$ for $a = 1, \ldots, N$. Then $\mu_P = -\mu_Q + O(\delta^3)$ and $C_P = C_Q + O(\delta^3)$ for small enough δ.*

Proof By definition, $\mu_{iQ} = \sum_{b=1}^{N_i} q_{ib} \ln\left(\frac{q_{ib}}{p_{ib}}\right)$ and $\mu_{iP} = \sum_{b=1}^{N_i} p_{ib} \ln\left(\frac{q_{ib}}{p_{ib}}\right)$. By expanding ln,

$$\ln\left(\frac{q_{ib}}{p_{ib}}\right) = \ln\left(1 + \frac{\varepsilon_{ib}}{p_{ib}}\right) = \frac{\varepsilon_{ib}}{p_{ib}} - \frac{1}{2}\frac{\varepsilon_{ib}^2}{p_{ib}^2} + O(\delta^3), \tag{3}$$

where $\varepsilon_{ib} = \sum_{h_i(a)=b} \varepsilon_a$ and as

$$|\varepsilon_{ib}| = |\sum_{h_i(a)=b} \varepsilon_a| = |\sum_{h_i(a)=b} p_a(\varepsilon_a/p_a)| \leq \delta \sum_{h_i(a)=b} p_a = \delta p_{ib}.$$

Then

$$\begin{aligned} \mu_{iQ} + \mu_{iP} &= \sum_{b=1}^{N_i} (q_{ib} + p_{ib}) \ln\left(\frac{q_{ib}}{p_{ib}}\right) \\ &= \sum_{b=1}^{N_i} (2p_{ib} + \varepsilon_{ib}) \left(\frac{\varepsilon_{ib}}{p_{ib}} - \frac{1}{2}\frac{\varepsilon_{ib}^2}{p_{ib}^2} + O(\delta^3)\right) = O(\delta^3). \end{aligned}$$

That implies $\mu_P = -\mu_Q + O(\delta^3)$. Similarly, we get $\mu_{iP} = O(\delta^2)$ and so $\mu_{iQ} = O(\delta^2)$. Let $\mathtt{x}$ have distribution P. By c_{ijP} we denote an entry of C_P, the covariance between R_{it} and R_{jt}. It does not depend on n. From (2)

$$c_{ijP} = \sum_a p_a \ln\left(\frac{q_{ih_i(a)}}{p_{ih_i(a)}}\right) \ln\left(\frac{q_{jh_j(a)}}{p_{jh_j(a)}}\right) - \mu_{iP}\mu_{jP}. \tag{4}$$

So

$$c_{ijQ} - c_{ijP} = \sum_{a=1}^{N} \varepsilon_a \ln\left(\frac{q_{ih_i(a)}}{p_{ih_i(a)}}\right) \ln\left(\frac{q_{jh_j(a)}}{p_{jh_j(a)}}\right) + O(\delta^5)$$

as by above $\mu_{iQ}\mu_{jQ} = \mu_{iP}\mu_{jP} + O(\delta^5)$. By (3), $\ln\left(\frac{q_{ib}}{p_{ib}}\right) = O(\delta)$ and by condition $|\epsilon_a| \le \delta p_a$. Therefore, $c_{ijQ} - c_{ijP} = O(\delta^3)$. That proves the lemma.

By Central Limit Theorem, for large enough n the vector $s(\nu)$ is distributed as multivariate normal random variable $\mathbf{N}(n\mu_P, nC_P)$ or $\mathbf{N}(n\mu_Q, nC_Q)$. To distinguish between P and Q by observing the value of ν one may distinguish between the normal distributions above. Assume the matrices C_P and C_Q are invertible. Then the normal distributions have densities. That always happens in our experiments with DES. A normalised logarithmic likelihood ratio statistic is

$$S(\nu) = \frac{1}{4n}\left(-\left[s(\nu) - n\mu_Q\right] C_Q^{-1} \left[s(\nu) - n\mu_Q\right]^T + \left[s(\nu) - n\mu_P\right] C_P^{-1} \left[s(\nu) - n\mu_P\right]^T\right).$$

Generally, it is a quadratic function in $s(\nu)$. As $C = C_P \approx C_Q$ the statistic is approximately linear. Let $\mu = \mu_Q$. We take into account that $\mu_P \approx -\mu$. By expanding brackets in the expression for $S(\nu)$ we get

$$S(\nu) \approx s(\nu) C^{-1} \mu^T = \sum_{i=1}^{m} S_i(\nu_i), \tag{5}$$

where $S_i(\nu_i) = \omega_i LLR_i(\nu_i)$ for some coefficients ω_i, entries of $C^{-1}\mu^T$. Denote $a = n\,\mu C^{-1}\mu^T$, then $a > 0$. The expectation of $S(\nu)$ is $\approx \pm a$ and its variance is $\approx a$. So if $\mathtt{x}$ follows Q, then $S(\nu)$ is distributed approximately as $\mathbf{N}(a, a)$. If $\mathtt{x}$ follows P, then $S(\nu)$ is distributed approximately as $\mathbf{N}(-a, a)$. Therefore the approximation (5) to the statistic $S(\nu)$ is separable. That property will be used in the search algorithm in Section 5.2 and in the cryptanalysis of DES, see Section 9.

5 Optimization Problem

Let $A_i, i = 1, .., m$ be matrices of size $r_i \times n$ over binary finite field and of rank r_i which are relatively low in comparison with n. Let X be a vector of unknowns of length n. We consider a system of inclusions(a system of MRHS equations according to [21])

$$A_i X \in \{a_{i1}, .., a_{it_i}\}, \tag{6}$$

where $\{a_{i1}, .., a_{it_i}\}$ are vectors of length r_i over the same field. Let S_i be a weight function on the right hand side vectors in (6). The function S_i may be vectorial of the same dimension for every i and have negative entries. Let A be a basis of the space generated by the rows in all A_i. To simplify the notation we assume that $\text{rank}(A) = n$. The problem is to find all values $X = x$ such that the following vectorial inequality holds

$$\sum_{i=1}^{m} S_i(A_i x) > z \tag{7}$$

for some vectorial threshold z. One can consider that problem over any field, the only limitation is the number of vectors in the right hand side of (6) is finite. The problem may be solved by brute force in case of a finite field by trying all values of X. We now suggest a method that works faster. General case $\text{rank}(A) \leq n$ is reducible to the case where $\text{rank}(A) = n$ by rewriting (6) in new variables $Y = AX$.

5.1 Example of the Problem

Let a system of 3 MRHS equations with weights be given.

$x_1 + x_3$	x_2	S_1
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.1

,

$x_1 + x_2$	S_2
0	0.5
1	0.1

,

x_1	$x_2 + x_3$	S_3
0	0	0.4
0	1	0.5
1	0	0.7
1	1	0.1

.

One is to find all x_1, x_2, x_3 such that

$$S_1(x_1 + x_2, x_3) + S_2(x_1 + x_3) + S_3(x_1, x_2 + x_3) > 1.3. \tag{8}$$

The solution is $x_1, x_2, x_3 = 1, 1, 1$.

5.2 Algorithm

The algorithm is described in terms of linear functions not vectors. Thus $A_i X$ is a vectorial linear function. Let AX be a basis of the linear space generated by all entries in $A_i X$. That is A is a basis of the linear space generated by rows in all A_i. Assume a sequence of the subspaces generated by linearly independent basis functions T_j such that

$$\langle 0 \rangle = \langle T_0 \rangle \subseteq \langle T_1 \rangle \subseteq \langle T_2 \rangle \subseteq \ldots \subseteq \langle T_r \rangle = \langle AX \rangle. \tag{9}$$

One can assume that T_{j-1} is a subset of T_j. The choice of (9) affects the time complexity of the algorithm below. It is important to keep the growth of the dimension stable, for instance, $\dim\langle T_j \rangle - \dim\langle T_{j-1} \rangle = 1$.

1. (precomputation) For each j, i one defines the subspace $\langle T_{ji} \rangle = \langle T_j \rangle \cap \langle A_i X \rangle$ by its basis T_{ji}. For each value $T_{ji} = a$ the maximum of S_i achieved upon that fixation of T_{ji} is stored. We denote that maximum by $d_{j,i}(a)$. If $T_{ji} = 0$, then the maximum is denoted $d_{j,i}$. For each j and i one keeps $2^{|T_{ji}|} \leq 2^{r_i}$ numbers $d_{ji}(a)$.

2. We start the search with $j = 1$ and implement the following recursive step. Let for some $j \geq 1$ the value of $T_{j-1} = b$ be already determined. We will determine a value for T_j. Take any value $T_j = a$ that extends the value of $T_{j-1} = b$. For each i, as $\langle T_{ji} \rangle \subseteq \langle T_j \rangle$, compute the value $T_{ji} = a_i$ and look up $d_{ji}(a_i)$. Check
$$\sum_{i=1}^{m} d_{ji}(a_i) \geq z. \tag{10}$$
Let (10) hold. If $j = r$, then as $\langle T_r \rangle = \langle AX \rangle$ a solution is found. Another value for T_r is then examined or one backtracks, that is $j \leftarrow j - 1$ and one repeats the step. If $j < r$ then $j \leftarrow j + 1$ and one repeats the step. If (10) does not hold, then another value for T_j is examined or one backtracks.

The algorithm is an adaptation of a gluing type algorithm from [23]. It is justified by the following lemma.

Lemma 2 *Let the value $T_j = a$ be an extension of the value $T_{j-1} = b$, then*
$$\sum_{i=1}^{m} d_{j-1,i}(b_i) \geq \sum_{i=1}^{m_t} d_{j,i}(a_i)$$
for any i.

Proof. If the value $T_j = a$ is an extension of the value $T_{j-1} = b$, then $d_{j-1,i}(b_i) \geq d_{ji}(a_i)$ for any i. That implies the statement.

As $\langle T_r \rangle = AX$, we have $d_{ri}(a_i) = S_i(a)$ for any value $T_r = a$. By Lemma 2, the inequalities $\sum_{i=1}^{m} S_i(a_i) \geq z$ imply the inequalities (10) for any j. Therefore we won't reject a solution a by the decision rule (10) for any $j = 1, \ldots, r$ if it satisfies $S(a) \geq z$.

5.3 Example of the Problem Solution

Let $T_1 = \{x_1\}, T_2 = \{x_1, x_2\}, T_3 = \{x_1, x_2, x_3\}$. We define
$$\begin{aligned}
&T_{11} = \{0\},\ T_{12} = \{0\},\ T_{13} = \{x_1\},\\
&T_{21} = \{x_2\},\ T_{22} = \{x_1 + x_2\},\ T_{23} = \{x_1\},\\
&T_{31} = \{x_1 + x_3, x_2\},\ T_{22} = \{x_1 + x_2\},\ T_{23} = \{x_1, x_2 + x_3\}.
\end{aligned}$$

After the precomputation

d_{1i}	
d_{11}	0.3
d_{12}	0.5
$d_{13}(0)$	0.5
$d_{13}(1)$	0.7

,

d_{2i}	
$d_{21}(0)$	0.3
$d_{21}(1)$	0.2
$d_{22}(0)$	0.5
$d_{22}(1)$	0.1
$d_{23}(0)$	0.5
$d_{23}(1)$	0.7

,

d_{3i}	
$d_{31}(00)$	0.1
$d_{31}(01)$	0.2
$d_{31}(10)$	0.3
$d_{31}(11)$	0.1
$d_{32}(0)$	0.5
$d_{32}(1)$	0.1
$d_{33}(00)$	0.4
$d_{33}(01)$	0.5
$d_{33}(10)$	0.7
$d_{33}(11)$	0.1

.

The search tree is presented in Fig. 2. We demonstrate how it is constructed. To construct the first node one sets $x_1 = 0$ and checks if

$$d_{11} + d_{12} + d_{13}(0) > 1.3\,.$$

This is false, one backtracks, sets $x_1 = 1$ and checks

$$d_{11} + d_{12} + d_{23}(1) > 1.3\,.$$

This is true, one extends $x_1, x_2 = 10$ and checks

$$d_{21}(0) + d_{22}(1) + d_{23}(1) > 1.3\,.$$

This is false, one backtracks, puts $x_1, x_2 = 11$ and checks

$$d_{21}(1) + d_{22}(0) + d_{23}(01) > 1.3\,.$$

This is true, so one puts $x_1, x_2, x_3 = 110$ and checks

$$d_{31}(11) + d_{32}(0) + d_{23}(11) > 1.3\,.$$

This is false, so one backtracks, puts $x_1, x_2, x_3 = 111$ and checks

$$d_{31}(01) + d_{32}(0) + d_{23}(10) > 1.3\,.$$

That is true, so $x_1, x_2, x_3 = 111$ is the only solution to the problem. The complexity is determined by the number of constructed nodes. The tree in Fig. 2 incorporates 6 nodes besides the root and one is to check 6 inequalities. The brute force requires to check 8 inequalities (8).

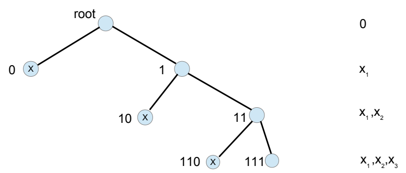

Figure 2: The Search Tree

6 Application in Cryptanalysis

Let a number of statistically independent vectors $\mathtt{x}_t$ be given along with their projections $h_{ti}(\mathtt{x}_t), i = 1, \ldots, m_t$. For instance, $\mathtt{x}_1, \mathtt{x}_2$ are 14-bit vectors (24) and (26) in the cryptanalysis of DES below. They depend on different internal bits of the encryption and therefore may be considered independently distributed. We use their 10-bit linear projections.

Let n plain-text/cipher-text pairs be available. The observation on $h_{ti}(\mathtt{x}_t)$ is a string of frequencies ν_{ti} of length N_{ti}. In this cryptanalysis of DES $N_{ti} = 2^{10}$. Let's denote $\bar{K}_{ti} = \mathtt{key}_{ti}, \mathtt{Key}_{ti}$, where $\mathtt{key}_{ti}$ are key bits which affect a priori distribution of $h_{ti}(\mathtt{x}_t)$, and $\mathtt{Key}_{ti}$ are those key bits from the first and the last round keys which affect the observation on $h_{ti}(\mathtt{x}_t)$. Therefore $\bar{K}_{ti}$ are linear functions(at least in case of DES) in unknown cipher key bits. Let $\bar{K}$ be a list of linearly independent functions in all $\bar{K}_{ti}$. For DES cryptanalysis with $\mathtt{x}_1, \mathtt{x}_2$ we have $\text{rank}(\bar{K}) = 54$.

For each possible value $\bar{K}_{ti}$ one computes the value $S_{ti}(\bar{K}_{ti}) = \omega_{ti} LLR_{ti}(\nu_{ti}, \bar{K}_{ti})$. One then combines the values of $\bar{K}_{ti}$ into a value of $\bar{K}$ such that

$$\sum_{i=1}^{m_t} S_{ti}(\bar{K}_{ti}) \geq z_t \tag{11}$$

for all t and some thresholds z_t to be defined later. One can easily represent all (11) together as a vectorial inequality (7). Therefore the algorithm from Section 5.2 is applicable.

We call a value of $\bar{K}$ which passes the test (11) a $\bar{K}$-candidate. After the test each $\bar{K}$-candidate is extended to a key-candidate(56-bit key in case of DES). All such key-candidates are to be brute forced. The algorithm's success is that (11) is true for the correct value of $\bar{K}$. We now analyse the success probability of the method and the number of $\bar{K}$-candidates.

6.1 Success probability and the number of $\bar{K}$-candidates

Assume the value of $\bar{K}$ is correct. Then the value of $\bar{K}_{ti}$ is correct too. The observation on every $h_{ti}(\mathtt{x}_t)$ has a distribution derived from a priori distribution of $\mathtt{x}_t$. The statistic $S_t(\nu_t) = \sum_{i=1}^{m_t} S_{ti}(\nu_{ti})$ on the left hand side of (11) has then normal distribution $\mathbf{N}(a_t, a_t)$

for every t if ν_t follows a priori distribution. Here $a_t = n\,\mu_t C_t^{-1} \mu_t^T$, where $n\mu_t$ and nC_t are the expectation and covariance matrix of the vectorial random variables $s_t(\nu_t)$ constructed with LLR statistics for $h_{ti}(\mathtt{x}_t), i = 1, .., m_t$, see Section 4.2. For each t the success is not to miss the correct value of $\bar{K}_t$, where the probability of success is computed by

$$1 - \beta_t = \mathbf{Pr}(\mathbf{N}(a_t, a_t) \geq z_t) = \frac{1}{\sqrt{2a_t\pi}} \int_{-\infty}^{-z_t} e^{-\frac{(y-a_t)^2}{2a_t}} dy. \tag{12}$$

As $\mathtt{x}_t$ are independent, the success probability of the whole method is then $\prod_t 1 - \beta_t$.

If the value of $\bar{K}$ is incorrect we assume that all $\bar{K}_t$ are not correct. Otherwise, the number of $\bar{K}$-values for which the latter is not true is negligible. So one can assume that the observation on every $h_{ti}(\mathtt{x}_t)$ is uniformly distributed and the statistic $\sum_{i=1}^{m_t} S_{ti}(\nu_{ti})$ has normal distribution $\mathbf{N}(-a_t, a_t)$. The fraction of incorrect $\bar{K}_t$ which pass the test for one t is

$$1 - \alpha_t = \mathbf{Pr}(\mathbf{N}(-a_t, a_t) \geq z_t) = \frac{1}{\sqrt{2a_t\pi}} \int_{-\infty}^{-z_t} e^{-\frac{(y+a_t)^2}{2a_t}} dy. \tag{13}$$

The fraction of incorrect $\bar{K}$ which pass the test for all t is $\prod_t 1 - \alpha_t$ as $\mathtt{x}_t$ are independent. The number of $\bar{K}$-candidates is on the average

$$2^{|\bar{K}|} \prod_t 1 - \alpha_t. \tag{14}$$

So the number of the cipher key values to brute force, that is the number of key-candidates, is $2^{56} \prod_t 1 - \alpha_t$ in case of DES. Assume one wants to brute force 2^s key candidates. One searches for z_t such that $\prod_t 1 - \alpha_t = 2^{s-56}$ to maximise the success probability $\prod_t 1 - \beta_t$.

7 Multivariate Probability Distribution in Feistel Ciphers

Based on the analysis of the encryption algorithm we get a priori probability distributions of internal bits in Feistel Ciphers.

7.1 Notation

Let Y be a bit string of some length, then we denote $Y\{i, j, .., k\} = Y[i] \oplus Y[j] .. \oplus Y[k]$ and $Y[i, j, .., k] = [\,Y[i], Y[j], .., Y[k]\,]$. Let $Y_i, Y_j, .., Y_k$ be bit strings of the same length then $Y_{\{i,j,..,k\}}[r] = Y_i[r] \oplus Y_j[r] \oplus .. \oplus Y_k[r]$.

Let X_0, X_1 be plain-texts blocks of bit-length r each and $K_i, i = 1, \ldots, n$ round keys of bit-length s. Then for $i = 1, \ldots, n$ the blocks X_{i-1}, X_i is an input to the i-th round of the encryption algorithm, where X_{i+1}, X_i is the output, and $X_{i+1} = X_{i-1} \oplus F_i(X_i, K_i)$ for some function F_i see Fig.3. The output of the n-th round X_{n+1}, X_n is the cipher-text.

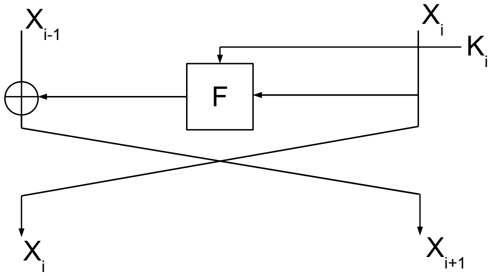

Figure 3: One Feistel round

In case of DES we have $r = 32$ and $s = 48$, and the number of encryption rounds is 16. We keep the notation of [17]. All bit string entries are numbered from right to left, starting with 0. In case of DES the key bits numbered as in its specification: k_i, where $i = 1, .., 63$ and $i \neq 0 \bmod 8$. Besides, we ignore the initial permutation.

7.2 Multivariate Distributions

Assume the plain-text X_0, X_1 is taken uniformly at random from the set of all $2r$-bit strings and the cipher key we want to recover is fixed. The cipher-text X_{n+1}, X_n and any internal bits in the encryption algorithm are then random variables. Our first goal is to compute a priori distribution of

$$Z = X_0[\sigma_0], X_1[\sigma_1], X_n[\sigma_n], X_{n+1}[\sigma_{n+1}], \tag{15}$$

which is to be used in this cryptanalysis below. (15) is a vectorial random variable of $|\sigma_0|+|\sigma_1|+|\sigma_n|+|\sigma_{n+1}|$ bit length. The sought distribution depends on the cipher key and its exact calculation is a very difficult task. Instead, we will construct an approximation to that distribution which depends on a lower number of the key bits.

7.3 Exact Probabilistic Description of a Feistel Cipher

Let $X_0, X_1, \ldots, X_{n+1}$ be now random independently generated r-bit blocks and $K_1, \ldots, K_n$ fixed round keys of bit-length s. Let's consider the event $\mathcal{C}$:

$$X_{i-1} \oplus X_{i+1} = F_i(X_i, K_i), \quad i = 1, \ldots, n. \tag{16}$$

By induction, $\mathbf{Pr}(\mathcal{C}) = 2^{-rn}$. The exact probability of an event $\mathcal{E}$ which happens in the encryption algorithm is

$$\mathbf{Pr}(\mathcal{E}|\mathcal{C}) = \frac{\mathbf{Pr}(\mathcal{E}\mathcal{C})}{\mathbf{Pr}(\mathcal{C})} = 2^{rn}\mathbf{Pr}(\mathcal{E}\mathcal{C}).$$

The event $\mathcal{C}$ depends on the whole cipher key, so it is difficult to calculate $\mathbf{Pr}(\mathcal{E}|\mathcal{C})$ by this formula. Instead, a relaxed version of (16) will be used.

7.4 Approximate Probabilistic Description of a Feistel Cipher

We define a larger event $\mathcal{C}_\alpha$, which means $\mathcal{C}$ implies $\mathcal{C}_\alpha$, see for instance (17) below and then put $\mathbf{Pr}(\mathcal{E}|\mathcal{C}) \approx \mathbf{Pr}(\mathcal{E}|\mathcal{C}_\alpha) = \frac{\mathbf{Pr}(\mathcal{E}\mathcal{C}_\alpha)}{\mathbf{Pr}(\mathcal{C}_\alpha)}$. That is an approximate description of the cipher. It depends on the event $\mathcal{C}_\alpha$. Obviously, by taking another event we will have another approximate description of the cipher. Our goal is to compute an approximate distribution of (15). So a relevant event $\mathcal{C}_\alpha$ is to be taken. The accuracy of so defined approximate description is unclear. It is even unclear how to measure that accuracy. In spite of this, the approach gives good results in practice and was already implicitly used in [17], see Section 7.5.

For Z defined by (15) and a bit string A of the same length, we will derive a formula to compute the exact value of $\mathbf{Pr}(Z = A|\mathcal{C}_\alpha)$ for certain $\mathcal{C}_\alpha$ defined by

$$X_{i-1}[\alpha_i] \oplus X_{i+1}[\alpha_i] = F_i(X_i, K_i)[\alpha_i], \quad i = 1, \ldots, n. \tag{17}$$

We see $\mathbf{Pr}(\mathcal{C}_\alpha) = 2^{-\sum_{i=1}^{n}|\alpha_i|}$. One says $\alpha = (\alpha_1, \ldots, \alpha_n)$ are output masks for multivariate round approximations(called round sub-vectors here) in n consecutive rounds respectively. Let's denote by β_i input masks. The sequence of α_i, β_i defines a trail, see Section 7.6 for definitions. Trails are classically used to compute probability distributions of one-bit "linear approximations" for DES in [17]. The approximate distribution of (15) does not depend on the input masks β_i in the internal rounds, that is for $i = 2, \ldots, n-1$, if the trail satisfies some natural conditions, see Section 7.6. Such trails will be called regular. We remark the probability $\mathbf{Pr}(Z = A|\mathcal{C}_\alpha)$ only depends on the key bits involved in the right hand sides of (17).

7.5 Approximate Distributions in Matsui's work

A similar approach was implicitly used by Matsui [17] when computing the distribution of one-bit "linear approximations" to DES encryption algorithm. He used the event $\mathcal{C}'_\alpha$:

$$X_{i-1}\{\alpha_i\} \oplus X_{i+1}\{\alpha_i\} = F_i(X_i, K_i)\{\alpha_i\}, \quad i = 1, \ldots, n,$$

where α_i were output masks for round approximations. For instance, for 3-round DES in Figure 4 of Matsui's work one wants to compute the distribution of

$$f = X_0\{7, 18, 24, 29\} \oplus X_4\{7, 18, 24, 29\} \oplus X_1\{15\} \oplus X_3\{15\} \oplus K_1\{22\} \oplus K_3\{22\}. \tag{18}$$

Let $n = 3$, and $\alpha = (\{7, 18, 24, 29\}, \emptyset, \{7, 18, 24, 29\})$. Under assumption that $X_0, \ldots, X_4$ are uniformly and independently distributed, the probability of $\mathcal{C}'_\alpha$ is 1/4. We find $\mathbf{Pr}(f = 0|\mathcal{C}) \approx \mathbf{Pr}(f = 0|\mathcal{C}'_\alpha) \approx 0.70$ as stated in [17], see Appendix 1 for details.

7.6 Regular Trails

Let $\alpha_i, \beta_i, \gamma_i \subseteq \{0, 1, \dots, r-1\}$ and $\delta_i \subseteq \{0, 1, \dots, s-1\}$. The sequence of $|\alpha_i| + |\beta_i|$-bit strings

$$X_i[\beta_i], F_i[\alpha_i], \quad i = 1, \dots, n \tag{19}$$

is called a trail. The members of the trail are called round sub-vectors. The distribution of round sub-vectors are easy to derive from the definition of the round function as it was done in [17] for one-bit "linear approximations". Our goal is to compute the joint distribution of some input and output bits (15) for n-round Feistel cipher by using a certain trail.

Let $K_i[\delta_i]$ and $X_i[\gamma_i]$ denote the round key bits and input bits relevant to the function $F_i[\alpha_i]$. For instance, in case of DES the key bits $K_i[23..18]$ and input bits $X_i[16..11]$ are relevant to $F_i[24, 18, 7, 29]$. We call the trail (19) regular if

$$\gamma_i \cap (\alpha_{i-1} \cup \alpha_{i+1}) \subseteq \beta_i \subseteq \gamma_i, \quad i = 1, \dots, n, \tag{20}$$

where $\alpha_0 = \alpha_{n+1} = \emptyset$. It is easy to check the following statement.

Lemma 3 *Let $n > 3$, then for any strings of indicies $\sigma_0, \sigma_1, \sigma_n, \sigma_{n+1}$ in* (15) *there exists a regular trail* (20)*, such that*

$$\sigma_0 = \alpha_1, \sigma_1 = \alpha_2 \cup \beta_1, \sigma_n = \alpha_{n-1} \cup \beta_n, \sigma_{n+1} = \alpha_n.$$

Proof. We put

i	α_i	β_i
1	σ_0	$\gamma_1 \cap \sigma_1$
2	σ_1	γ_2
$3 \le i \le n-2$	any	γ_i
$n-1$	σ_n	γ_{n-1}
n	σ_{n+1}	$\gamma_n \cap \sigma_n$

That proves the lemma.

For $n = 3$ a regular trail exists if and only if $\sigma_3 \setminus \gamma_3 \subseteq \sigma_1$ and $\sigma_1 \setminus \gamma_1 \subseteq \sigma_3$. Generally, there is a large variety of certain auxiliary events $\mathcal{C}_\alpha$ or equivalently regular trails for computing approximations to the actual distribution of (15). Those trails produce generally different distributions, and, in particular, the latter may depend on different key bits.

7.7 Convolution Formula for the Distribution

Assume a regular trail (19), where $\alpha = (\alpha_1, \dots, \alpha_n)$ are output masks. We now produce a convolution type formula to calculate an approximate distribution of the vector

$$Z = X_0[\alpha_1], X_1[\alpha_2 \cup \beta_1], X_n[\alpha_{n-1} \cup \beta_n], X_{n+1}[\alpha_n] \tag{21}$$

for that trail. We will see that the distribution does not depend on β_i, where $i = 2, \ldots, n-1$. The method introduced in Section 7.3 is used. We give a formula to calculate $\mathbf{Pr}(Z = A|\mathcal{C}_\alpha)$, where $\mathcal{C}_\alpha$ is defined by (17). To simplify notation, we put $\alpha_0 = \emptyset, \alpha_{n+1} = \emptyset$ and denote

$$\mathbf{q}_i(b, a, k) = \mathbf{Pr}(X_i[\beta_i] = b, F_i[\alpha_i] = a \,|\, K_i[\delta_i] = k_i)$$

the probability distribution of round sub-vectors. In case of DES, if only non-adjacent S-boxes are involved in the trail (19), then by the definition of F_i we have $\mathbf{q}_i(b, a, k) = \mathbf{q}_i(b \oplus k, a, 0)$. We denote the latter by $\mathbf{q}_i(b \oplus k, a)$. The values of $Z = X_0[\alpha_1], X_1[\alpha_2 \cup \beta_1], X_n[\alpha_{n-1} \cup \beta_n], X_{n+1}[\alpha_n]$ are respectively denoted by $A = A_0, A_1, A_n, A_{n+1}$.

Theorem 1 *Let* (19) *be a regular trail, then*

$$\mathbf{Pr}(Z = A|\mathcal{C}_\alpha) = \frac{2^{\sum_{i=2}^{n-1} |\alpha_i|}}{2^{\sum_{i=1}^{n} |(\alpha_{i-1} \cup \alpha_{i+1}) \setminus \beta_i|}} \sum_{A_2, \ldots, A_{n-1}} \prod_{i=1}^{n} \mathbf{q}_i(A_i[\beta_i], (A_{i-1} \oplus A_{i+1})[\alpha_i], k_i), \quad (22)$$

where the sum is over $A_i = A_i[\alpha_{i-1} \cup \alpha_{i+1} \cup \beta_i]$ *and* $K_i[\delta_i] = k_i$.

Proof. By conditional and total probability formulas,

$$\begin{aligned} &\mathbf{Pr}(Z = A|\mathcal{C}_\alpha) = \mathbf{Pr}(\mathcal{C}_\alpha)^{-1}\mathbf{Pr}(Z = A, \mathcal{C}_\alpha) \\ &= \mathbf{Pr}(\mathcal{C}_\alpha)^{-1} \sum_{A_2, \ldots, A_{n-1}} \mathbf{Pr}\left(\mathcal{A}_1\right) \\ &= \mathbf{Pr}(\mathcal{C}_\alpha)^{-1} \sum_{A_2, \ldots, A_{n-1}} \mathbf{Pr}\left(\mathcal{A}_2\right), \end{aligned} \quad (23)$$

where the sum is over $A_j = A_j[\alpha_{j-1} \cup \alpha_{j+1} \cup \beta_j]$, $j = 2, .., n-2$, and as the events

$$\mathcal{A}_1 = \begin{pmatrix} Z & = & A_0, A_1, A_n, A_{n+1}, \\ X_i[\alpha_{i-1} \cup \alpha_{i+1} \cup \beta_i] & = & A_i, \, i = 2, \ldots, n-1, \\ \mathcal{C}_\alpha & & \end{pmatrix}$$

and

$$\mathcal{A}_2 = \begin{pmatrix} X_i[\beta_i], F_i[\alpha_i] & = & A_i[\beta_i], (A_{i-1} \oplus A_{i+1})[\alpha_i] \\ X_0[\alpha_1] & = & A_0, \\ X_i[(\alpha_{i-1} \cup \alpha_{i+1}) \setminus \beta_i] & = & A_i[(\alpha_{i-1} \cup \alpha_{i+1}) \setminus \beta_i], \\ X_{n+1}[\alpha_n] & = & A_{n+1}, \\ i & = & 1, \ldots, n. \end{pmatrix}$$

are equivalent. By $\mathcal{E}_1$ we denote the event

$$X_i[\beta_i], F_i[\alpha_i] = A_i[\beta_i], (A_{i-1} \oplus A_{i+1})[\alpha_i], \quad i = 1, \ldots, n,$$

and by $\mathcal{E}_2$ the event

$$X_0[\alpha_1] = A_0,\ X_i[(\alpha_{i-1}\cup\alpha_{i+1})\setminus\beta_i] = A_i[(\alpha_{i-1}\cup\alpha_{i+1})\setminus\beta_i],\ X_{n+1}[\alpha_n] = A_{n+1}, \quad i = 1, \ldots, n.$$

By definition of regular trail, no variables in $X_i[(\alpha_{i-1}\cup\alpha_{i+1})\setminus\beta_i]$ are relevant to $X_i[\beta_i], F_i[\alpha_i]$. So the events $\mathcal{E}_1, \mathcal{E}_2$ are independent as they depend on different bits of X_i, $i = 1, \ldots, n$. We can now split the latter probability into a product. Then

$$\mathbf{Pr}(Z = A|\,\mathcal{C}_\alpha) = \mathbf{Pr}(C_\alpha)^{-1} \sum_{A_2,\ldots,A_{n-1}} \mathbf{Pr}\left(\mathcal{E}_1\right)\mathbf{Pr}\left(\mathcal{E}_2\right).$$

As $\mathbf{Pr}\left(\mathcal{E}_2\right) = 1/2^{|\alpha_1|+|\alpha_n|+\sum_{i=1}^{n}|\alpha_{i-1}\cup\alpha_{i+1}\setminus\beta_i|}$ and $\mathbf{Pr}\left(\mathcal{C}_\alpha\right) = 1/2^{\sum_{i=1}^{n}|\alpha_i|}$ we get

$$\mathbf{Pr}(Z = A|\,\mathcal{C}_\alpha) = \frac{2^{\sum_{i=2}^{n-1}|\alpha_i|}}{2^{\sum_{i=1}^{n}|(\alpha_{i-1}\cup\alpha_{i+1})\setminus\beta_i|}} \sum_{A_2,\ldots,A_{n-1}} \prod_{i=1}^{n} \mathbf{q}_i(A_i[\beta_i], (A_{i-1}\oplus A_{i+1})[\alpha_i], k_i).$$

That finishes the proof.

7.8 Distribution Properties

The conditions of Theorem 1 are satisfied if, for instance, $\beta_i = \gamma_i \cap (\alpha_{i-1}\cup\alpha_{i+1})$. That is an extension of the conditions upon which the distribution of one-bit "linear approximation" was computed by Matsui. To calculate the distribution of

$$X_0\{\alpha_1\} \oplus X_1\{\alpha_2 \cup \beta_1\} \oplus X_n\{\alpha_{n-1}\cup\beta_n\} \oplus X_{n+1}\{\alpha_n\}$$

by summing round approximations $X_i\{\beta_i\} \oplus F_i\{\alpha_i\}$ the masks α_i, β_i are to satisfy $\alpha_{i-1} \oplus \alpha_{i+1} = \beta_i$, see [17]. We now study properties of regular trails and relevant distributions.

Lemma 4 *Let* (19) *be a regular trail, then the distribution* (22) *does not depend on* β_i*, where*

$$\gamma_i \cap (\alpha_{i-1}\cup\alpha_{i+1}) \subseteq \beta_i \subseteq \gamma_i,\ i = 2, \ldots, n-1.$$

Proof. Let $\beta_i' = \gamma_i \cap (\alpha_{i-1}\cup\alpha_{i+1})$. Then $\beta_i' \subseteq \beta_i$ and $(\alpha_{i-1}\cup\alpha_{i+1})\setminus\beta_i = (\alpha_{i-1}\cup\alpha_{i+1})\setminus\beta_i'$ by (20). The statement follows from

$$\sum_{A_i[\beta_i\setminus\beta_i']} \mathbf{q}_i(A_i[\beta_i], (A_{i-1}\oplus A_{i+1})[\alpha_i], k_i) = \mathbf{q}_i(A_i[\beta_i'], (A_{i-1}\oplus A_{i+1})[\alpha_i], k_i)$$

as all other terms in (22) do not depend on $A_i[\beta_i \setminus \beta_i']$.

Lemma 4 implies that to reduce calculation cost one can take $\beta_i = \gamma_i \cap (\alpha_{i-1}\cup\alpha_{i+1})$ for a regular trail (19). That produces the same distribution by (22). Also we call a regular trail (19) reduced if

$$\alpha_{i-1} \setminus \beta_i = \alpha_{i+1} \setminus \beta_i.$$

for all $i = 2 \dots n-1$. It is not difficult to see that if the trail is not reduced, then one can construct another trail which gives the same distribution for (21) or the distribution itself degenerates into a distribution of a sub-vector of (21). This follows from the fact that the bits $A_i = A_i[\alpha_{i-1} \cup \alpha_{i+1} \cup \beta_i]$ only affect

$$\begin{aligned}
&\mathbf{q}_{i-1}(A_{i-1}[\beta_{i-1}], (A_{i-2} \oplus A_i)[\alpha_{i-1}], k_{i-1}),\\
&\mathbf{q}_{i}(A_{i}[\beta_{i}], (A_{i-1} \oplus A_{i+1})[\alpha_{i}], k_{i}),\\
&\mathbf{q}_{i+1}(A_{i+1}[\beta_{i+1}], (A_{i} \oplus A_{i+2})[\alpha_{i+1}], k_{i+1}),
\end{aligned}$$

in (22). Therefore if $\alpha_{i-1} \setminus \beta_i \neq \alpha_{i+1} \setminus \beta_i$ the trail (19) may be reduced and (22) gives the same distribution with another trail or the distribution of a sub-vector of Z.

We say H_i holds if $\beta_i, \alpha_i = \emptyset, \emptyset$ or round vector $X_i[\beta_i], F_i[\alpha_i]$ is uniformly distributed. Similarly, one proves

Lemma 5 *Let* (19) *be a regular trail and H_i, H_{i+1} or H_i, H_{i+2} hold simultaneously for some i. Then* (22) *provides a uniform distribution.*

7.9 Recurrent Formula

The computation with Theorem 1 might be tedious for $n = 14$ or 15. So one can use a convolution type formula based on splitting the encryption into two parts. Let $1 < r < n$ and the distribution of

$$\begin{aligned}
Z_1 &= X_0[\alpha_1], X_1[\alpha_2 \cup \beta_1], X_r[\alpha_{r-1} \cup \beta_r], X_{r+1}[\alpha_r],\\
Z_2 &= X_r[\alpha_{r+1}], X_{r+1}[\alpha_{r+2} \cup \beta_{r+1}], X_n[\alpha_{n-1} \cup \beta_n], X_{n+1}[\alpha_n]
\end{aligned}$$

be already computed. We denote $\alpha' = (\alpha_1, \dots, \alpha_r)$ and $\alpha'' = (\alpha_{r+1}, \dots, \alpha_n)$.

Corollary 1

$$\begin{aligned}
&\mathbf{Pr}(Z = A_0, A_1, A_n, A_{n+1} \,|\mathcal{C}_\alpha)\\
=\ & 2^{|\alpha_r|} \sum_{A_r, A_{r+1}} \mathbf{Pr}\left(Z_1 = A_0, A_1, A_r[\alpha_{r-1} \cup \beta_r], A_{r+1}[\alpha_r] \,|\, \mathcal{C}_{\alpha'}\right)\\
\times\ & \mathbf{Pr}\left(Z_2 = A_r[\alpha_{r+1}], A_{r+1}[\alpha_{r+2} \cup \beta_{r+1}], A_n, A_{n+1} | \,\mathcal{C}_{\alpha''}\right).
\end{aligned}$$

Lemma 1 is proved by splitting the product in (22) and summing the first part over $A_2, \dots, A_{r-1}$ and the second part over $A_{r+2}, \dots, A_{n-1}$ and using the theorem again.

Fig. 4 shows theoretical and empirical a priori distributions for the 10-bit block $X_2[24, 18, 7, 29], X_7[16, 14], X_8[24, 18, 7, 29]$ of DES internal bit. Approximate theoretical distribution was computed with Corollary 1 by using an appropriate trail. This distribution depends on 3 key bits. The empirical distribution was produced by encrypting 2^{39} randomly and independently generated 64-bit plain-text blocks for one randomly chosen cipher key. We realise the distributions are very close, almost indistinguishable.

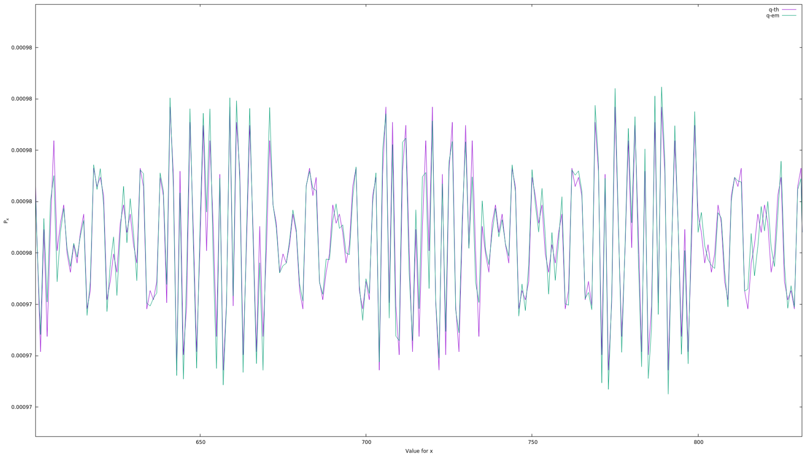

Figure 4: Theoretical and empirical distributions in DES

8 Multinomial Distributions for 14-round DES

One of two best "linear approximations" for 14-round DES found by Matsui in [17] is $X_2\{24,18,7\} \oplus X_{15}\{15\} \oplus X_{16}\{24,18,7,29\}$. We took all those bits in above. More bits may be added with increasing of the number of the key bits from the first and the last round keys involved. We got 14-bit string

$$\mathtt{x}_1 = (X_2[24,18,7,29], X_{15}[16,15,14,13,12,11], X_{16}[24,18,7,29]). \tag{24}$$

Approximate a priori distribution of $\mathtt{x}_1$ was computed by using Theorem 1 and Corollary 1 with the trail shown in Table 1. The computation took only a few seconds on a common computer. The distribution depends on the value of 7-bit string:

Table 1: Trail for computing the distribution of (24)

round i	β_i, α_i
$2,6,10,14$	$\emptyset, \emptyset$
$3,5,7,9,11,13$	$\{15\}, \{24,18,7,29\}$
$4,8,12$	$\{29\}, \{15\}$
15	$\{16,\ldots,11\}, \{24,18,7,29\}$

$$K_{\{3,5,7,9,11,13\}}[22] \oplus K_{\{4,8,12\}}[44], K_{15}[23,22,21,20,19,18]. \tag{25}$$

We denote that by $\texttt{key}_1$. The distribution of (24) is a permutation of the distribution, where $\texttt{key}_1$ is a zero-string. In the known-plain-text attack we do not observe the bits of (24). The latter are internal to the encryption algorithm and depend on the first and the last round keys

$$\begin{aligned} X_2[24,18,7,29] &= X_0[24,18,7,29] \oplus S_5(X_1[16..11] \oplus K_1[23...18]),\\ X_{15}[16] &= X_{17}[16] \oplus S_3(X_{16}[24...19] \oplus K_{16}[35...30]),\\ &\ldots,\\ X_{15}[11] &= X_{17}[11] \oplus S_8(X_{16}[4...31] \oplus K_{16}[5...0]), \end{aligned}$$

and $X_{16}[24,18,7,29])$ is a part of the cipher-text. Thus the observation depends on some plain-text/cipher-text bits, 36 bits of the last round key K_{16} and 6 bits of the first round key K_1. As some key bits repeat, the observation effectively depends on a 39-bit sub-key denoted $\texttt{Key}_1$. In theory, one can apply a multidimensional linear analysis developed in [10]. Likelihood ratio statistic will then depend on $\texttt{key}_1, \texttt{Key}_1$: overall 44 key bits and one linear combination of the key bits. That makes 2^{45} variants for $\texttt{key}_1, \texttt{Key}_1$ to range by the value of the statistic and won't give any advantage over Matsui's analysis of DES even if one uses Fast Fourier Transform to compute the statistic. By DES symmetry one gets the distribution of

$$\mathtt{x}_2 = (X_{15}[24,18,7,29], X_2[16,15,14,13,12,11], X_1[24,18,7,29]), \tag{26}$$

which depends on $K_{\{4,6,8,10,12,14\}}[22] \oplus K_{\{5,9,13\}}[44], K_2[23,22,21,20,19,18]$ denoted by $\texttt{key}_2$ The observation on (26) depends on a 37-bit sub-key from K_1 and K_{16} denoted $\texttt{Key}_2$.

$$\begin{aligned} X_{15}[24,18,7,29] &= X_{17}[24,18,7,29] \oplus S_5(X_{16}[16..11] \oplus K_{16}[23...18]),\\ X_2[16] &= X_0[16] \oplus S_3(X_1[24...19] \oplus K_1[35...30]),\\ &\ldots,\\ X_2[11] &= X_0[11] \oplus S_8(X_1[4...31] \oplus K_1[5...0]) \end{aligned}$$

$X_1[24,18,7,29]$ is a part of the plain-text. As above we can not afford using $\mathtt{x}_2$. Instead of $\mathtt{x}_1, \mathtt{x}_2$, two bunches of their 10-bit projections will be defined in this section. We get overall 28 14-round input/output sub-vectors, whose multinomial distributions will be used to attack 16-round DES later in this paper. As $\mathtt{x}_1$ and $\mathtt{x}_2$ depend on disjoint sets of the encryption algorithm internal bits, they are here considered independently distributed. The observation on two bunches of 10-bit sub-vectors (27) and (28) below are considered independent too.

8.1 Another Trail

Another approximate distribution of $\mathtt{x}_1$ was computed by using another trail shown in Table 2. It has a negligibly larger quadratic imbalance. However we remark that in the trail presented in Table 2 the masks β_i, α_i are generally larger sets than relevant masks in Table 1. So this approximation depends on a significantly larger number of the key bits. The distribution is marginally different from the distribution produced with the trail in Table 1. For those reasons the distribution is not used in the present analysis.

Table 2: Another trail for computing the distribution of (24)

round i	β_i, α_i
2	$\emptyset, \emptyset$
$3, 5, 7, 9, 11, 13$	$\{16, 15, 14\}, \{24, 18, 7, 29\}$
$4, 6, 8, 10, 12, 14$	$\{29, 24\}, \{16, 15, 14\}$
15	$\{16, \ldots, 11\}, \{24, 18, 7, 29\}$

8.2 First Bunch of 14-round Input/Output Sub-Vectors

Instead of $\mathtt{x}_1$ we use the projections

$$X_2[24, 18, 7, 29], X_{15}[i, j], X_{16}[24, 18, 7, 29], \tag{27}$$

for different $i, j \in \{16, 15, 14, 13, 12, 11\}$ except $i = 16, j = 11$, where the distribution of (27) is uniform. When it is not uniform the distribution depends on 3 key bits $K_{\{3,5,7,9,11,13\}}[22] \oplus K_{\{4,8,12\}}[44], K_{15}[i', j']$, where $K_{15}[i', j']$ denotes a key-mask for $X_{15}[i, j]$ in the 15-th round. We will use 14 such 10-bit vectors. The observation on (27) depends on 12 bits of K_{16} and 6 bits of K_1, that is at most 18 key bits. Therefore one is to examine the values of at most 20 key bits and one linear combination of the key bits. That makes 2^{21} variants of the observation and distribution on (27) and that number is affordable.

8.3 Second Bunch of 14-round Input/Output Sub-Vectors

By DES symmetry, 10-bit projections

$$X_{15}[24, 18, 7, 29], X_2[i, j], X_1[24, 18, 7, 29], \tag{28}$$

of $\mathtt{x}_2$ may be used for the reason above. The distribution of (28) depends on $K_{\{4,6,8,10,12,14\}}[22] \oplus K_{\{5,9,13\}}[44], K_2[i', j']$, where $K_2[i', j']$ denotes a key-mask for $X_2[i, j]$ in the 2-nd round.

9 Implementation Details for 16-round DES

Two independent separable statistics constructed from the above projections of $\mathtt{x}_1$ and $\mathtt{x}_2$ are used. The statistics are identically distributed as one-variate normal random variable

$\mathbf{N}(a,a)$ for $a = n\mu C^{-1}\mu^T$, where μ and C are computed from a priori distribution of $\mathtt{x}_1$(same for $\mathtt{x}_2$) .

We fix required success probability 0.85 and find the threshold z such that the number of plain-text/cipher-text pairs n equals to the number of 56-bit keys for the final brute force by solving the system

$$\begin{aligned}(1-\beta_1)^2 &= 0.85\\ 2^{56}(1-\alpha_1)^2 &= n,\end{aligned}$$

where β_1 and α_1 are defined by (12) and (13). Remark that $t = 2$, and $\beta_1 = \beta_2$, $\alpha_1 = \alpha_2$. In particular, we get $n \approx 2^{41.8}$.

9.1 One of 28 Projections

Let h_1 denote the projection $X_2[24, 18, 7, 29], X_{15}[16, 15], X_{16}[24, 18, 7, 29]$. The observation and distribution of h_1 depend on $\bar{K}_1$ which incorporates 20 unknowns

$$\begin{gathered}x_{63}, x_{61}, x_{60}, x_{53}, x_{46}, x_{42}, x_{39}, x_{36}, x_{31},\\ x_{30}, x_{27}, x_{26}, x_{25}, x_{22}, x_{21}, x_{12}, x_{10}, x_7, x_5,\\ x_{57} + x_{51} + x_{50} + x_{19} + x_{18} + x_{15} + x_{14},\end{gathered}$$

where x_i denote key bits of 56-bit DES key. For each value $\bar{K}_1 = k_1$ the value of $S_1(k_1) = \omega_1 LLR_1(k_1)$ is kept, 2^{20} values overall. $LLR_1(k_1)$ for all values k_1 are shown in Fig. 5. With $n = 2^{41.8}$ plain-text/cipher-text pairs the expectation of LLR_1 for correct k_1 is 4.6649, for incorrect -4.6638. Experimental value for the correct key is 2.2668, it is presented by a vertical line in Fig. 5. There are 23370 values higher than that. We remark that using only h_1 in the cryptanalysis is not efficient enough. One is to brute force $2^{36} \times 23371 > 2^{50.5}$ key-candidates before finding the correct 56-bit DES key. That won't give any advantage over Matsui's results. Similar is true for other 27 projections.

9.2 Search Tree Complexity

54 DES key bits $\bar{K}$ which affect our statistics are

$$\begin{gathered}x_2, x_{19}, x_{60}, x_{34}, x_{10}, x_{17}, x_{59}, x_{36}, x_{42}, x_{27}, x_{25},\\ x_{52}, x_{11}, x_{33}, x_{51}, x_9, x_{23}, x_{28}, x_5, x_{55}, x_{46}, x_{22},\\ x_{62}, x_{15}, x_{37}, x_{47}, x_7, x_{54}, x_{39}, x_{31}, x_{29}, x_{20}, x_{61},\\ x_{63}, x_{30}, x_{38}, x_{26}, x_{50}, x_1, x_{57}, x_{18}, x_{14}, x_{35}, x_{44},\\ x_3, x_{21}, x_{41}, x_{13}, x_4, x_{45}, x_{53}, x_6, x_{12}, x_{43}.\end{gathered} \tag{29}$$

They are taken in an order defined by how many $\bar{K}_i$ those key bits are relevant to. We say a key-bit x relevant to $\bar{K}_i$ if the rank of $\bar{K}_i$ drops upon the fixation of x by a constant. For instance, x_2 relevant to 14(maximal number) of $\bar{K}_i$, etc.

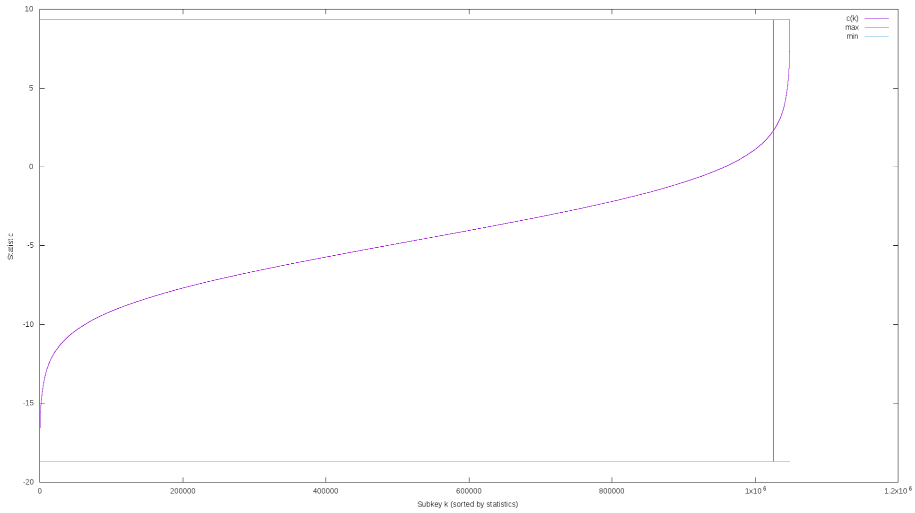

Figure 5: LLR-values for h_1

To construct the search tree one first chooses a sequence $T_1, T_2, \ldots, T_{54}$, where T_{j+1} is produced from T_j by adding one unknown key-bit, which is relevant to the most of $\bar{K}_i$ and which is not in $\langle T_{j+1} \rangle$. The choice is not unique. We use the order defined by (29). That is $T_1 = \{x_2\}, T_2 = \{x_2, x_{19}\}, T_3 = \{x_2, x_{19}, x_{60}\}$,.. The choice of T_j affects significantly the complexity(the number of nodes) of the tree. Search algorithm from Section 5.2 is then run.

The number of examined values of T_j(tree nodes at level j), $j = 38, ..54$, in $\log_2$ scale are presented in Fig. 6. Overall number of nodes is $2^{45.5} << 2^{54}$. So the complexity of finding $\bar{K}$-candidates is much lower than brute forcing all values of $\bar{K}$. The final number of $\bar{K}$-candidates is $2^{39.8}$, so the number of 56-bit DES keys to brute force is $2^{41.8}$ again as it was predicted by our theory. Constructing one node requires few bit xor's and few additions with low precision real numbers, see Section 6. So search tree complexity(constructing $2^{45.5}$ nodes) is lower in bit operations than final brute force of $2^{41.8}$ DES keys. In fact, our implementation works slower than that as we need to access external memory where precomputation results, that is the numbers $d_{ji}(a)$, are kept. At the same time DES encryption is very straightforward. One needs to keep around 2^{26} low precision real numbers.

9.3 Possible Improvements

There are several direction in improving the method practically and theoretically.

1. Obviously, one can get better result by using larger strings $\mathtt{x} = (X, Y)$ of encryption

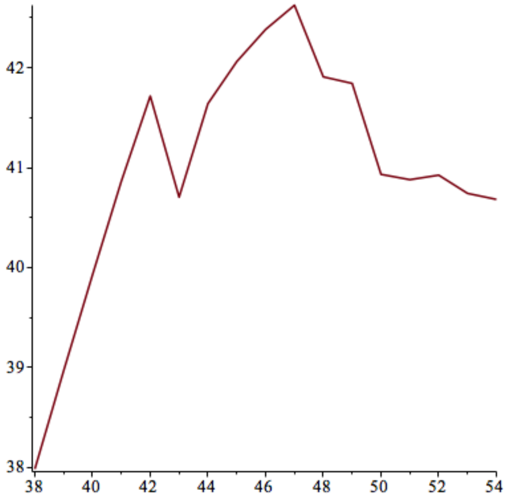

Figure 6: Search tree complexity

internal bits, see Fig.1.

2. There are several practical ways to reduce the number of nodes in the search tree, e.g. by taking a larger threshold z in (10) for low levels(low j) of the tree. However those methods do not guarantee theoretical success probability as Lemma 2 does not apply any more.

3. The number of nodes in the search tree may probably be further reduced by choosing more carefully the sequence of T_j. We do not know how to choose an optimal sequence.

4. One can use another statistics for the projections $h_1, .., h_m$. For instance, let $\bar{K}_{0i} \subset \bar{K}_i$, where the key bits $\bar{K}_{0i}$ affect a priori distribution of h_i, and

$$LLR_i^*(\bar{K}_i \setminus \bar{K}_{0i}) = \max_{\bar{K}_{0i}} LLR_i(\bar{K}_i).$$

Using LLR_i^* instead of LLR_i looks better in practice and in line with Matsui's analysis. However the distribution of $(LLR_1^*, \ldots, LLR_m^*)$ is unknown and therefore the success probability of the method is difficult to predict. One can probably try to compute it experimentally for a truncated cipher and then extrapolate to the full-round one, as similar was done by Matsui in [18, 19].

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] C. Blondeau and K. Nyberg,*Joint Data and Key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and Its Impact to Data Complexity,* Cryptology ePrint Archive, 2015/935.

[3] A. Bogdanov, E. Tischhauser, and Ph. S. Vejre, *Multivariate Linear Cryptanalysis: The Past and Future of PRESENT*,Cryptology ePrint Archive, 2016/667.

[4] J.Y. Cho *Linear Cryptanalysis of Reduced-Round PRESENT,* in Topics in Cryptology-CT-RSA 2010, pp. 302–317, Springer (2010)

[5] 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.

[6] P. Junod,*On the complexity of Matsui's Attack*, in Selected Areas in Cryptography, LNCS vol. 2259, Springer, 2001, pp. 199–211.

[7] P. Junod and S. Vaudney,*On the optimality of linear, differential, and sequential distinguisher*, in Eurocrypt 2003, LNCS vol. 2656, Springer, 2003, pp. 17–32.

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

[9] 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.

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

[11] M. Hermelin, K. Nyberg *Linear Cryptanalysis Using Multiple Linear Approximations*, in Advanced Linear Cryptanalysis of Block and Stream Ciphers, P. Junod and A. Canteaut(Eds.), IOS Press, 2011.

[12] T. Baignères, P. Junod and S. Vaudenay, *How Far Can We Go Beyond Linear Cryptanalysis?* in Asiacrypt'04(P. Lee, editor), LNCS vol. 3329, Springer, 2004, pp. 432–450.

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

[14] 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.

[15] 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.

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

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

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

[19] 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.

[20] Yu. I. Medvedev, *Separable Statistics in a Polynomial Scheme. I*, Theory Probab. Appl., 22(1)(1977),pp. 1–15.

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

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

[23] I. Semaev, Improved Agreeing-Gluing Algorithm, Math. in Comp. Science 7(2013), pp. 321–339.

[24] I. Semaev, *New results in the Linear Cryptanalysis of DES,* Cryptology ePrint Archive, 2014/361.

10 Appendix 1. On Matsui's Probability Calculation

In this section we show that the distribution of one-bit "linear approximations" used in [17] may be computed only based on $X_0, X_1, \dots, X_{n+1}$ are independently and uniformly generated and under condition of the auxiliary event $\mathcal{C}'_\alpha$:

$$X_{i-1}\{\alpha_i\} \oplus X_{i+1}\{\alpha_i\} = F_i(X_i, K_i)\{\alpha_i\}, \quad i = 1, \dots, n \tag{30}$$

for some α_i. We will do that in case of 3-round DES represented in Figure 4 of Matsui's work [17]. The general case is similar. We want to compute the distribution of (18). Let

$n = 3$ and $\alpha = (\beta, \emptyset, \beta)$, where $\beta = \{7, 18, 24, 29\}$. Then $\mathcal{C}'_\alpha$ is $F_1\{\beta\} \oplus X_0\{\beta\} \oplus X_2\{\beta\} = 0, F_3\{\beta\} \oplus X_2\{\beta\} \oplus X_4\{\beta\} = 0$. We have

$$\begin{aligned}
\mathbf{Pr}(f = 0|\mathcal{C}) &\approx \mathbf{Pr}(f = 0|\mathcal{C}'_\alpha) = \frac{\mathbf{Pr}(f = 0, \mathcal{C}'_\alpha)}{\mathbf{Pr}(\mathcal{C}'_\alpha)} = 4\,\mathbf{Pr}(f = 0, \mathcal{C}'_\alpha) \\
&= 4 \sum_{a,b,c,d} \mathbf{Pr}\left(\begin{array}{ccc} F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & b, \\ X_1\{15\} & = & c, \\ X_3\{15\} & = & d, \\ f & = & 0, \\ C'_\alpha & & \end{array}\right),
\end{aligned}$$

where the sum is over binary a, b, c, d. We now take into account that $f = [F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{22\}] \oplus [F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{22\}] \oplus [F_1\{\beta\} \oplus X_0\{\beta\} \oplus X_2\{\beta\}] \oplus [F_3\{\beta\} \oplus X_2\{\beta\} \oplus X_4\{\beta\}]$. Let $\bar{F}_1, \bar{F}_3$ be produced from F_1, F_3 by the substitution $X_1\{15\} = c, X_3\{15\} = d$. Then

$$\begin{aligned}
\mathbf{Pr}(f = 0|\mathcal{C}) &\approx 4 \sum_{a,c,d} \mathbf{Pr}\left(\begin{array}{ccc} F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & a, \\ X_1\{15\} & = & c, \\ X_3\{15\} & = & d, \\ \bar{F}_1\{\beta\} \oplus X_0\{\beta\} \oplus X_2\{\beta\} & = & 0, \\ \bar{F}_3\{\beta\} \oplus X_2\{\beta\} \oplus X_4\{\beta\} & = & 0. \end{array}\right) \\
&= 4 \sum_{a,c,d} \mathbf{Pr}\left(\begin{array}{ccc} F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & a, \\ X_1\{15\} & = & c, \\ X_3\{15\} & = & d. \end{array}\right) \\
&\times \mathbf{Pr}\left(\begin{array}{ccc} \bar{F}_1\{\beta\} \oplus X_0\{\beta\} \oplus X_2\{\beta\} & = & 0, \\ \bar{F}_3\{\beta\} \oplus X_2\{\beta\} \oplus X_4\{\beta\} & = & 0. \end{array}\right)
\end{aligned}$$

The probability was split into a product by independence. The last term in the product is 1/4. Therefore,

$$\begin{aligned}
\mathbf{Pr}(f = 0|\mathcal{C}) &\approx \sum_{a,c,d} \mathbf{Pr}\left(\begin{array}{ccc} F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & a, \\ X_1\{15\} & = & c, \\ X_3\{15\} & = & d. \end{array}\right) \\
&= \sum_{a} \mathbf{Pr}\left(\begin{array}{ccc} F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & a. \\ & & \end{array}\right) \\
&= \sum_{a} \mathbf{Pr}(F_1\{\beta\} \oplus X_1\{15\} \oplus K_1\{15\} = a)\,\mathbf{Pr}(F_3\{\beta\} \oplus X_3\{15\} \oplus K_3\{15\} = a)
\end{aligned}$$

$$= \left(\frac{12}{64}\right)^2 + \left(1 - \frac{12}{64}\right)^2 \approx 0.70.$$

11 Appendix 2. Another Statistic

Denote $\nu(n) = (\nu_1, \ldots, \nu_m)$, a vector of length $M = \sum_{i=1}^m N_i$, a concatenation of $\nu_i(n)$, see notation in Section 4.1. We write

$$\nu(n) = \sum_{i=1}^{n} R_i,$$

where R_i are independent identically distributed(as $\nu(1)$) random variables. Assume that $\mathtt{x}$ has distribution P. Then by μ'_P and C'_P we denote the expectation and the covariance matrix for $\nu(1)$. We have

$$\mu'_P = (\mu'_1, \ldots, \mu'_m),$$

where $\mu'_i = \left(\sum_{h_i(a)=1} p_a, \ldots, \sum_{h_i(a)=N_i} p_a\right)$ is the expectation of $\nu_i(1)$. We can split the matrix C'_P into blocks C'_{ij}. Such block represents a covariance matrix for $\nu_i(1)$ and $\nu_j(1)$. By the definition of covariance, it is not difficult to find that equals

$$\begin{aligned} C'_{ij}[b,c] &= \sum_{\substack{h_i(a) = b \\ h_j(a) = c}} p_a - \sum_{h_i(a)=b} p_a \sum_{h_j(a)=c} p_a \\ &= \mathbf{Pr}(h_i(\mathtt{x}) = b, h_j(\mathtt{x}) = c) - \mathbf{Pr}(h_i(\mathtt{x}) = b)\mathbf{Pr}(h_j(\mathtt{x}) = c). \end{aligned}$$

If $h_i(\mathtt{x}), h_j(\mathtt{x})$ for $i \neq j$ are independent random variables, then C'_P is diagonal, because C'_{ij} are zero-matrices. Diagonal blocks C'_{ii} are covariance matrices for $\nu_i(1)$.

By Central Limit Theorem the distribution of $\frac{\nu - n\mu(p)}{\sqrt{n}}$ tends to a multivariate normal distribution $N(0, C'_P)$ with 0 expectations and covariance matrix C'_P. Similarly, if $\mathtt{x}$ has distribution Q, then $\frac{\nu - n\mu(q)}{\sqrt{n}}$ tends to $N(0, C'_Q)$. To decide which distribution P or Q is correct by observing the value of ν, one can apply Neyman-Pearson test. However as the matrices C'_P, C'_Q are singular, the distributions $N(0, C'_P)$ and $N(0, C'_Q)$ do not have densities. A standard solution is to consider a random variable νB for an appropriate matrix B instead of ν. The variable $\frac{\nu B - n\mu B}{\sqrt{n}}$ is distributed as $N(0, C''_P)$ or $N(0, C''_Q)$ for $\mu = \mu'_P$ or μ'_Q accordingly, where $C''_P = BC'_P B^T$ and $C''_Q = BC'_Q B^T$. If those matrices are invertible, the distributions have densities. So Neyman-Pearson statistic $S'(\nu)$ is

$$\frac{1}{n}(- \left[\nu B - n\mu'_P\right] C''^{-1}_P \left[\nu B - n\mu'_P\right]^T + \left[\nu B - n\mu'_Q\right] C''^{-1}_Q \left[\nu B - n\mu'_Q\right]^T). \tag{31}$$

Let $\nu = (\nu_1, \ldots, \nu_m)$, where ν_i is an observation on $h_i(\mathtt{x})$. In case $h_i(\mathtt{x}), i = 1, .., m$ are independent, the matrices C'_P, C'_Q are diagonal, the matrices C''_P, C''_Q are diagonal for some

B too. Then

$$S'(\nu) = \sum_{i=1}^{m} S'_i(\nu_i).$$

In this type of cryptanalysis the observations on ν_i depend on generally different sub-key bits. Therefore one can first examine the values of those key bits separately for each i and arrange them by the value of $S'(\nu_i)$. One then combines the values of the sub-keys to provide $S'(\nu) \geq z'$ for some threshold z' such that those values agree on common key bits as above. However as the projections (27) are dependent(the same is true for (28)), the use of the statistic S' within this cryptanalysis does not seem to give any advantage.