IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2018, No. 2, pp. 79–110. DOI:10.13154/tosc.v2018.i2.79-110

Separable Statistics and Multidimensional Linear Cryptanalysis

Stian Fauskanger[1] and Igor Semaev[2]

[1] Norwegian Defence Research Establishment (FFI), PB 25, 2027 Kjeller, Norway
stian@fauskanger.me
[2] Department of Informatics, University of Bergen, Bergen, Norway
igor@ii.uib.no

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 the encryption algorithm 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 may 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 that will 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. With $2^{41.81}$ plaintext blocks and success rate 0.83 (computed theoretically) we found $2^{41.46}$ (which is close to the theoretically predicted number $2^{41.81}$) key-candidates to 56-bit DES key. Search tree to compute the statistic values which fall into the critical region incorporated $2^{45.45}$ nodes in the experiment and that is at least theoretically inferior in comparison with the final brute force. To get success probability 0.85, which is a fairer comparison to Matsui's results, we would need $2^{41.85}$ data and to brute force $2^{41.85}$ key-candidates. That compares favourably with 2^{43} achieved by Matsui.

Keywords: Separable statistics · Multidimensional linear cryptanalysis · DES

1 Introduction

Linear Cryptanalysis is a statistical approach in the cryptanalysis of symmetric ciphers. It is a known plaintext attack which does not require any special plaintext/ciphertext pairs and therefore is a very important tool in practical cryptanalysis. It was introduced by Matsui in [20, 21] as an attack to DES. Davies and Murphy came up with another approach in statistical cryptanalysis in [5]. Linear Cryptanalysis exploits the fact that an xor of certain plaintext, ciphertext and key bits is zero with some a priori computed probability p different from $1/2$. Such combinations were called linear approximations in [20]. The probability itself somehow depends on the cipher key bits. The method is more efficient if p 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 plaintext/ciphertext 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 variations Algorithm 1 and Algorithm 2 were suggested in [20]. 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

Received: 2018-03-01, Accepted: 2018-05-01, Published: 2018-06-07

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 plaintext/ciphertext 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 $n = 2^{43}$ plaintext/ciphertext 64-bit blocks and success probability 0.85, then 2^{43} encryptions are performed to find the correct key [21]. 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.

The expression linear approximation though well settled in the current literature on cryptanalysis does not seem quite precise. Formally, one can say it measures how ciphertext is being approximated by the plain-text. However this intuition is not very helpful within the subject. We believe that the linear cryptanalysis and its modifications are based on the view that a linear approximation, or generally, any string $\mathtt{x}$ of the encryption algorithm internal bits is an ordinary random variable and does approximate nothing. Rather, a priori computed p, or generally, a probability distribution is an approximation to the real probability (distribution). The latter is a complicated function in many (or all) cipher key bits. However the approximate probability (distribution) p commonly depends on a small set of the cipher key bits (linear combinations of the key bits) and that makes the method work. For this reason we put that expression in quotation marks in what follows. A more detailed discussion on the meaning of a "linear approximation" and why using several approximations to the same $\mathtt{x}$ does not improve on the cryptanalysis is in Appendix 1 below.

Only few improvements with relation to DES have been published since Matsui's work. In [19] a chosen plaintext linear attack was suggested and in [7] time complexity of the attack's first stage was reduced by using Fast Fourier Transform. It was experimentally found in [11, 12] 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. The success probability was determined experimentally with 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 [18]. In [2] a framework for using many "linear approximations" considered statistically independent was proposed, though no practical cryptanalysis of 16-round DES was presented, where the sub-keys relevant to the observations on "linear approximations" were considered disjoint as in [21]. Linear cryptanalysis was further extended in different ways in [14, 1, 15], see [17]. For instance, [1, 15] made use multidimensional analysis instead of one-dimensional. A good survey of publications on using multiple "linear approximations" is in [16]. Recently, a series of papers on linear cryptanalysis of PRESENT were published, see for instance [6, 4, 3]. 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 general methods for computing joint a priori distributions (approximate joint distributions) of multiple "linear approximations" in block ciphers were published before. If 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 it has a low complexity. The same limitation holds in multidimensional linear cryptanalysis of [15]. The present work solves the deficiency by giving formulae to compute multidimensional probability distributions in Feistel ciphers. The method presented in Section 10 of this work is general and based on clear mathematical foundation. It is a direct generalisation of how Matsui calculated probabilities of his "linear approximations" in [20]. The approach is applicable to any round ciphers. These formulas may be further analysed to derive useful information on how the key bits involved in the trail affect those distributions. That was done in Section 11 in case of DES.

Similar ideas were earlier used to compute joint probability distributions of some particular bits and study how those distributions depend on the cipher key for DES in [5, 9] and for PRESENT in [8]. Those methods are based on a number of heuristic assumptions and simplifications. In particular, the calculation in [8] was done for 15-round cipher, where key bits involved in each round on the trail are the same.

In theory, it is possible, as it is suggested in [15, 6], to derive an approximate multidimensional distribution from the correlations of linear functions defined on its domain. For instance, that may be a span of a set of strong "linear approximations". The known distributions of those "linear approximations" are not exact, where the accuracy of the approximations and the number of the key bits involved depend on chosen trails. So to get an applicable joint distribution the trails are to be somehow compliant with each other, otherwise that may result in the final multidimensional distribution depends on too many key bits. From the point of view of the present work, the approach may require an approximate description of the cipher under question by using a big trail (appropriate auxiliary event in terminology of Section 10.4) which incorporates all trails for particular "linear approximations".

Another direction is to study joint distributions of correlations between "linear approximations" as functions of randomised cipher key, see the latest version of [4] and references in there. As that seems a difficult line to follow, the analysis is based on various hypotheses on joint behaviour of the above correlations, which are difficult to justify for a specific cipher as well. To get a practical key-recovery attack it is more natural to assume that the cipher key one wants to find is fixed while available plaintexts are randomly generated as it is in the original linear cryptanalysis and other statistical attacks as in [5]. We rather need to know how a priori distributions (approximate a priori distributions) depend on the cipher key in exact terms and that is achieved in the present work for Feistel ciphers.

Two open problems related to Algorithm 2 were posed in [2]. First, how to merge data (find cipher key) from analysing different "linear approximations" efficiently. Second, how to compute the success probability as a function in the number of available plaintexts and the number of trials in the search phase. A solution to these problems was found in [28]. 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 8 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 [28]. Instead of "linear approximations" certain projections (sub-strings of bits or multidimensional linear functions) of the encryption internal states are used. In contrast with [28] we do not here assume the projections are statistically independent. We are able to compute their approximate joint a priori distributions and therefore predict correctly success probability of the attack besides 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 4.

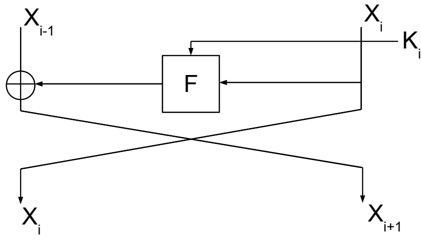

Figure 1: One Feistel round

2 Feistel Cipher and DES

The methods introduced in this paper are general and applicable to many ciphers. In Sections 11 and 12 we show how they work for DES as an example. In this section DES details are given.

Let X_0, X_1 be plaintexts blocks of bit-length r each and $K_i, i = 1, \dots, R$ round keys of bit-length s. Then for $i = 1, \dots, R$ 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.1. The output of the R-th round X_{R+1}, X_R is the ciphertext.

In case of DES we have $r = 32$ and $s = 48$, and the number of encryption rounds is 16. We keep the notation of [20], in particular, 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 $i = 1, \dots, 63$ and $i \neq 0 \bmod 8$. Besides, we ignore the initial permutation. See [29] for DES specification.

3 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.2. 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 combinations of the key bits) denoted $\mathtt{key}$ in Fig.2, see Section 11 how this dependence looks for some particular vector $\mathtt{x}_1$ in DES. On the other hand, the observation on $\mathtt{x}$ depends on the available plaintext/ciphertext blocks and some key bits from the first and the last rounds denoted $\mathtt{Key}$ in Fig.2. Assume one guesses relevant key bits $\bar{K} = (\mathtt{key}, \mathtt{Key})$. If the guess was correct, then the observation follows a priori distribution (correct key assumption). If not, then the observation follows a distribution which is close to the uniform distribution. We assume it is uniform (wrong key assumption) 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, at least that is true in [20] and in our experiments with DES, see Section 11. Those assumptions were used by many authors before and their correctness is supported by the experiments with DES in the present work. The setting described here is the setting of the multidimensional linear cryptanalysis which originates from [1]. However this work does not suggest any way to compute joint a priori distributions of the encryption algorithm internal bits and so the method was not implemented.

Theoretically, according to [1, 15], 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

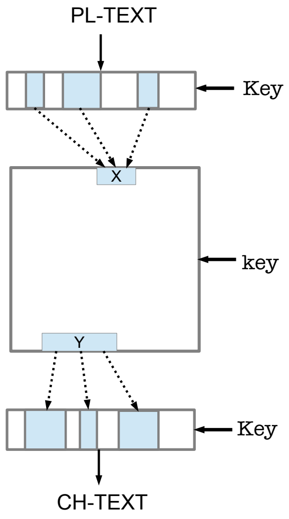

Figure 2: Round Cipher Cryptanalysis

of $\bar{K}$ according to Neyman and Pearson [24]. However the method is not efficient if the size of $\bar{K}$ (the number of linearly independent combinations in $\bar{K}$) is large. In this case one has to rank $2^{|\bar{K}|}$ values of the key bits involved according to the value of the statistic.

At the same time the distribution of some projections (sub-vectors or generally any functions) $h_i(\mathtt{x})$ and observations on them may depend on a much lower number of the key bits $\bar{K}_i = (\mathtt{key}_i, \mathtt{Key}_i)$. That holds for DES, see Section 11 below, and may hold for other modern block cipher based on small S-boxes. A good key schedule, e.g., round keys are not linear functions of the main key, may reduce the negative effect of small S-boxes in this sense. However that requires a separate investigation.

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 this can be accomplished by solving efficiently an optimisation problem without brute forcing the values of $\bar{K}$. Also we answer what the size of the set of $\bar{K}$-candidates is. To this end we will use a novel statistic which reflects the structure of the cipher round. The statistic is a linear combination of LLR statistics for different projections and we do not need that they are statistically independent.

4 Our Contributions

This paper contains the following contributions.

1. An approximate probabilistic description of Feistel ciphers is suggested in Section 10 and a convolution type formula for computing approximate probability distribution of multidimensional random variables $\mathtt{x}$ constructed with internal bits of the encryption algorithm is there derived.

2. A novel statistic which combines LLR statistics for different projections $h_i(\mathtt{x})$ is used in this cryptanalysis, see Section 7. 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 14-bit vectorial random variables $\mathtt{x}_1, \mathtt{x}_2$ produced by DES symmetry and considered independent, see Section 11, so two separable statistics are used.

3. The distribution of the statistic under correct and incorrect values of $\bar{K}$ is determined in Section 7.2. A critical region and success probability of the attack are defined in Section 9.1. The latter is the probability that the statistic value 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 too and used to predict the time complexity of the attack.

4. We represent a problem of reconstructing $\bar{K}$-values which fall into the critical region from $\bar{K}_i$-values as an optimisation problem stated in Section 8. A general algorithm to solve that problem is described in Section 8.2. It is based on the idea of gluing of $\bar{K}_i$-values developed in [26, 25].

5. Our approach allows to find the number of necessary plaintext/ciphertext blocks, given desired success probability and the number of $\bar{K}$-candidates to brute force.

6. The attack was implemented for 16-round DES, see Section 12 and it provides an improvement over Matsui's results. We used two independent separable statistics, each based on 14 of 10-bit projections with 54 DES key bits involved overall, see the next section for a summary.

5 Summary of the Attack for DES

We compute approximate a priori distributions of 14-round DES input/output 14-bit vectors $\mathtt{x}_1, \mathtt{x}_2$ in Section 11. The vector $\mathtt{x}_1$ incorporates all variables relevant to Matsui's best "linear approximation" after adding some more variables, then $\mathtt{x}_2$ is produced from $\mathtt{x}_1$ by DES symmetry.

For each $\mathtt{x}_t$, $t = 1, 2$ some 14 of its 10-bit projections $h_{ti} = h_{ti}(\mathtt{x}_t)$ are used. A priori distribution of h_{ti} depends on 3 key bits (linear combinations of the key-bits). The observation on the projection is a function of the plaintext/ciphertext bits and round keys from the first and the last rounds. It depends on at most 18 key bits. An LLR statistic LLR_{ti} is constructed for each projection h_{ti}. It depends on a sub-key of size at most 21, see Section 12.1. So at most 2^{21} values of the sub-key are ranked by the statistic LLR_{ti}. That gives weights for the values of those sub-keys.

The vector $s_t = (LLR_{t\,1}, \ldots, LLR_{t\,14})$ depends on the cipher key bits and may have two multivariate Normal distributions: one for a correct guess and another one for an incorrect guess on those key bits. Following Neyman-Pearson approach, we construct another LLR statistic to distinguish these Normal distributions. The final LLR statistic $\mathcal{S}_t = \sum_{i=1}^{14} \omega_{ti}\, LLR_{ti}$ linearly depends on the values of LLR_{ti}. So it is separable, see Section 7. That property is important for a tree search algorithm in Section 8.

Two separable statistics coming from $\mathtt{x}_1$ and $\mathtt{x}_2$ are considered independent, see the next section for a discussion on this assumption. They are used simultaneously. One sets two thresholds z_1 and z_2 (in fact, $z_1 = z_2 = z$ as s_1, s_2, and, therefore, $\mathcal{S}_1, \mathcal{S}_2$ are equally distributed), and defines the critical region: $\mathcal{S}_1 > z_1, \mathcal{S}_2 > z_2$, in order to provide a desirable success probability (e.g., 0.83) of the attack. That defines a statistical test. Overall, there are 54 independent key bits which affect the distributions and observations for all the projections h_{ti}, see Section 12.2. The total weight of a 54 -bit sub-key is a tuple of two real numbers, each of them is a linear combination of the weights of (≤ 21)-bit sub-keys for relevant projections h_{ti}. The 54-bit key candidates that pass the test are computed with the tree search algorithm in Section 8, by creating $2^{45.45}$ nodes in the experiment. Creating a node is a very simple operation. An important feature of the method is that no need to check all 2^{54} sub-key values to decide. The number of the DES 56-bit key candidates is $2^{41.46}$ (close to the theoretically predicted number $2^{41.81}$). In order to mount the attack one needs $n = 2^{41.81}$ plaintext/ciphertext blocks.

To get success probability 0.85, which is a fairer comparison to Matsui's results, we would need $n = 2^{41.85}$ and to brute force $2^{41.85}$ keys.

6 Assumptions

In this section we summarise the assumption underlying the statistic and the attack.

1. A sample of n plaintext blocks uniformly and independently generated, and their encryptions with the same cipher key are available.

2. Correct key assumption. Under correct relevant key bits the distribution of encryption internal vectors (e.g., 14-bit vectors $\mathtt{x}_1, \mathtt{x}_2$ in Section 11 in case of DES) involved in the attack is close to an approximate distribution a priori computed by Theorem 1. Technically, we use $X_0, X_1, .., X_R$ are uniformly distributed and an event $\mathcal{C}$ has happened. Depending on the event, one may compute the exact or approximate probabilities of various events in the encryption algorithm. The exact probabilities depend on all key bits and are difficult to handle, while approximate probabilities may only depend on few key bits.

3. Incorrect key assumption. Under incorrect relevant key bits the distribution of the above vectors (e.g., 14-bit vectors $\mathtt{x}_1, \mathtt{x}_2$ in Section 11 in case of DES) is close to the uniform distribution. Those distributions may vary. So relaxing the assumption in line with [3] may lead to an improvement.

4. We use limit distributions (produced by Central Limit Theorem) of the main statistic S in Section 7.2 in order to compute the success probability and the attack complexity.

5. In this particular cryptanalysis of DES the vectors $\mathtt{x}_1, \mathtt{x}_2$ in Section 11 incorporate different internal bits of the encryption and so considered statistically independent. So two separable statistics considered independent are used. Ideally, we would need a joint distribution of $\mathtt{x}_1, \mathtt{x}_2$ which is a 28-bit vector. Though feasible it would take more time. On the other hand, we wanted to demonstrate that our method is flexible and several independent separable statistics may be used simultaneously.

7 Separable Statistics

Let $\mathtt{x}$ be a vectorial random variable and an observation $\nu = (\nu_1, \dots, \nu_m)$ on m projections, which are sub-vectors and generally any functions in $\mathtt{x}$, is available. Here ν_i denotes a vector of observations on the outcomes of the projection $h_i(\mathtt{x})$. We do not assume the projections are statistically independent. In this cryptanalysis ν_i is a function in available plaintext/ciphertext blocks and the key bits $\mathtt{Key}_i$. By the statistic we mean a function which depends on the observation ν. A statistic $\mathcal{S}(\nu)$ is called separable if it can be represented as

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

where $S_i(\nu_i)$ are statistics computed for different $h_i(\mathtt{x})$. This property allows analysing data ν in parts by analysing ν_i separately. The notion was introduced in [23] to study statistical tests to distinguish discrete distributions. In this cryptanalysis the statistic $S_i(\nu_i)$ depends on a priori distribution of $h_i(\mathtt{x})$ and therefore on the key bits $\mathtt{key}_i$ besides the observation ν_i and the key bits $\mathtt{Key}_i$. So $S_i(\nu_i)$ depends on $\bar{K}_i$. The statistic $\mathcal{S}(\nu)$ depends on available plaintext/ciphertext blocks and the key bits $\bar{K}$, which incorporate all $\bar{K}_i$. That defines the statistic's domain. In fact, S_i are weighted LLR statistics for $h_i(\mathtt{x})$. To get the main statistic $\mathcal{S}(\nu)$ the Neyman-Pearson approach is applied again in Section 7.2. We will write $\mathcal{S}(\bar{K})$ and $\mathcal{S}_i(\bar{K}_i)$ instead of $\mathcal{S}(\nu)$ and $\mathcal{S}_i(\nu_i)$ to stress the dependence of the statistics on the sub-key $\bar{K}$ and $\bar{K}_i$ respectively. So (1) may be written as

$$\mathcal{S}(\bar{K}) = \sum_{i=1}^{m} S_i(\bar{K}_i).$$

One decides a value of $\bar{K}$ is correct if $\mathcal{S}(\bar{K}) > z$ for some threshold z. That defines the critical region. If the distribution of $\mathcal{S}(\bar{K})$ 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 final key search.

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 8. An efficient algorithm to solve it is introduced in Section 8.2. The algorithm implements walking over a search tree by creating new nodes if certain linear inequalities, implications of $\mathcal{S}(\bar{K}) > z$, are satisfied and takes advantage of the fact that the statistic is separable. The computation cost is much lower than $2^{|\bar{K}|}$. One may use several statistically independent $\mathtt{x}$, so several statistics of that kind may be used simultaneously.

Another statistic is derived in Appendix 3. That is based on a more direct application of the Neyman-Pearson approach. However it is separable only for statistically independent projections. That is not true for the bunches of the projection (28) and (29) 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.

7.1 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 $Q = (q_1, \ldots, q_N)$ for non-zero p_i, q_i. Let

$$V(n) = (V_1, V_2, \ldots, V_N)$$

denote outcome frequencies for $\mathtt{x}$ after n trials, so that $\sum_{j=1}^{N} V_j = n$. In other words, V_j is the number of times the outcome j was hit.

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 $\nu_{ib} = \sum_{h_i(a)=b} V_a$. Thus $\nu = \nu(n) = (\nu_1, \ldots, \nu_m)$ is a vector of observations on $(h_1(\mathtt{x}), \ldots, h_m(\mathtt{x}))$.

7.2 Main Statistic

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

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

and the sum is over a such that $h_i(a) = b$. 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 have $p_{ib}, q_{ib} \neq 0$. We consider the LLR (Logarithmic Likelihood Ratio) statistic for h_i

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

According to Neyman-Pearson lemma [24], LLR_i provides with the most powerful test to distinguish the distributions P_i and Q_i by observing independent samples.

By a standard argument, see for instance [1], for independently generated samples we get $LLR_i(\nu_i) = \sum_{t=1}^{n} R_{it}$, where R_{it} are independent identically distributed random variables. R_{it} takes the value $\ln\left(\frac{q_{ib}}{p_{ib}}\right)$ with probability p_{ib} for $i = 1, \ldots, N_i$, or with probabilitiy q_{ib} for $i = 1, \ldots, N_i$. In this cryptanalysis the independence is provided by the independence of the plaintext blocks, see Section 6.

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 [1], if the distributions P_i and Q_i are close enough, then $\mu_{iP} \approx -\mu_{iQ}$ and $\sigma_{iP} \approx \sigma_{iQ}$. In this section we will prove a more general statement.

Let $s(\nu) = (LLR_1(\nu_1), \ldots, LLR_m(\nu_m))$. Then, by the argument above, $s(\nu) = \sum_{t=1}^{n} R_t$, where $R_t = (R_{1t}, \ldots, R_{mt})$ are independent 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 $\mu_{iP} = \sum_{b=1}^{N_i} p_{ib} \ln\left(\frac{q_{ib}}{p_{ib}}\right)$. We remark $q_{ib} = p_{ib} + \varepsilon_{ib}$, where $\varepsilon_{ib} = \sum_{h_i(a)=b} \varepsilon_a$. By expanding the logarithm,

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

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,

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

$$= \quad -\frac{1}{2} \sum_{b=1}^{N_i} p_{ib} \left(\frac{\varepsilon_{ib}}{p_{ib}}\right)^2 + O(\delta^3) = O(\delta^2).$$

and so $\mu_{iQ} = O(\delta^2)$. Let $\mathtt{x}$ have the distribution P. By c_{ijP} we denote an entry of C_P, the covariance between R_{it} and R_{jt}. One can see R_{it} takes the values $\ln\left(\frac{q_{ih_i(a)}}{p_{ih_i(a)}}\right)$ with probability p_a. By definition,

$$c_{ijP} = \sum_{a=1}^{N} p_a \ln\left(\frac{q_{ih_i(a)}}{p_{ih_i(a)}}\right) \ln\left(\frac{q_{jh_j(a)}}{p_{jh_j(a)}}\right) - \mu_{iP}\mu_{jP}. \tag{4}$$

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 $\mu_{iQ}\mu_{jQ} = \mu_{iP}\mu_{jP} + O(\delta^5)$ by the above argument. By (3), $\ln\left(\frac{q_{ib}}{p_{ib}}\right) = O(\delta)$ and by the condition $|\epsilon_a| \leq \delta p_a$. 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 a 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. That always happens in our experiments with DES, though the determinants are fairly small. Then the normal distributions have densities. A normalised logarithmic likelihood ratio statistic is

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

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

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

where $S_i(\nu_i) = \omega_i LLR_i(\nu_i)$ for some coefficients ω_i, entries of $C^{-1}\mu^T$. Therefore the approximation (5) to the statistic $\mathcal{S}(\nu)$ is separable. That property will be used in the search algorithm in Section 8.2 and in the cryptanalysis of DES, see Section 12. Denote $u = n\,\mu C^{-1}\mu^T$, then $u > 0$. The expectation of $s(\nu)$ is $n\mu$ (under Q) and its covariance matrix is nC. So the expectation of $\mathcal{S}(\nu)$ is $\approx \pm u$ and its variance is $\approx u$. So if $\mathtt{x}$ follows Q, then $\mathcal{S}(\nu)$ is distributed approximately as $\mathbf{N}(u, u)$. If $\mathtt{x}$ follows P, then $\mathcal{S}(\nu)$ is distributed approximately as $\mathbf{N}(-u, u)$.

An heuristic argument to justify the distributions of the statistic $\mathcal{S}(\nu)$ is given in this section. Though the distributions work well in our experiments with DES, it is an open problem to get a rigorous proof.

8 Optimization Problem

In this Section we give an algorithm to solve a particular optimization problem. This algorithm is used to construct a set of $\bar{K}$-values such that $S(\bar{K}) > z$ in Section 9.

Let $A_i, i = 1, \ldots, m$ 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. Note that, in this section, n represents the number of variables not the number of plaintext blocks. Let $X = (x_1, \ldots, x_n)$ be a vector of unknowns of length n. We consider a system of inclusions (a system of MRHS equations according to [25])

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

where $\{a_{i1},\ldots,a_{it_i}\}$ are given vectors of length r_i over the same field. Let S_i be a weight function on the right hand side vectors in (6). If a is a vector of length r_i and $a \notin \{a_{i1},\ldots,a_{it_i}\}$, then we set $S_i(a) = -\infty$. The function S_i may be vectorial defined over real numbers including $-\infty$, and it should be of the same dimension for every i. Let A be a matrix composed of a basis of the space generated by the rows in all A_i. To simplify the notation we assume that $\mathrm{rank}(A) = n$. The problem is to find all values of X such that the following vectorial inequality holds

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

for some vectorial threshold z. One can consider that problem over any field, in other words, the entries of X may take values from any field. The only limitation is the number of vectors on the right hand sides of (6) are 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 $\mathrm{rank}(A) \le n$ is reducible to the case where $\mathrm{rank}(A) = n$ by rewriting (6) in new variables $Y = AX$.

8.1 Example of the Problem

Let a system of 3 MRHS equations in variables $X = (x_1, x_2, x_3)$ with weights be given, where the S_i, $i = 1, 2, 3$, are all of dimension 1.

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

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

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

8.2 Algorithm

The algorithm is described in terms of linear functions not vectors. Thus A_iX are vectorial linear functions and AX is a basis of the linear space generated by the entries in all A_iX. Assume a sequence of the subspaces generated by sets of 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 and $T_r = AX$. The choice of (9) affects the time complexity of the algorithm below. In particular, 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 $\langle T_{ji}\rangle = \langle T_j\rangle \cap \langle A_iX\rangle$ by its basis T_{ji}. One can assume that $T_{ri} = A_iX$. For each value $T_{ji} = a_i$ the maximum of S_i achieved upon that fixation of T_{ji} is stored. We denote that maximum by $d_{ji}(a_i)$. If $T_{ji} = 0$, then the maximum is denoted d_{ji}. Formally,

   $$d_{ji}(a_i) = \max_{T_{ji}=a_i} S_i(A_iX).$$

   For each j and i one keeps $2^{|T_{ji}|} \le 2^{r_i}$ real numbers $d_{ji}(a_i)$.

2. We set $T_0 = 0$. Then 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
$$\sum_{i=1}^{m} d_{j,i}(a_i) > z. \tag{10}$$
Let (10) hold. If $j = r$, then to find the solution X one solves the system of linear equations $a = AX$ as in this case $a_i = A_i X$ and $S_i(A_i X) = d_{ri}(a_i)$, and (7) holds. 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 [27]. It is justified by the following lemma.

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

Proof. By the definition of a_i and $d_{j,i}(a_i)$, we have
$$\begin{aligned} d_{j,i}(a_i) &= \max_{T_j = a} S_i(A_i X), \\ d_{j-1,i}(b_i) &= \max_{T_{j-1} = b} S_i(A_i X). \end{aligned}$$

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

By Lemma 2, the inequality $\sum_{i=1}^{m} S_i(A_i X) > z$ implies the inequalities (10) for any $1 \leq j \leq r$ as $d_{ri}(a_i) = S_i(A_i X)$, where $a_i = A_i X, a = T_r = AX$. Therefore we won't reject a value of X by the decision rule (10) for any $j = 1, \ldots, r$ if it satisfies (7).

8.3 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
$$\begin{aligned} &T_{11} = \{0\},\, T_{12} = \{0\},\, T_{13} = \{x_1\}, \\ &T_{21} = \{x_2\},\, T_{22} = \{x_1 + x_2\},\, T_{23} = \{x_1\}, \\ &T_{31} = \{x_1 + x_3, x_2\},\, T_{32} = \{x_1 + x_2\},\, T_{33} = \{x_1, x_2 + x_3\}. \end{aligned}$$

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

.

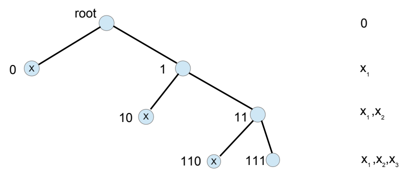

Figure 3: The Search Tree

The search tree is presented in Fig. 3. 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

$$d_{11} + d_{12} + d_{13}(1) > 1.3 \,.$$

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

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

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

$$d_{31}(11) + d_{32}(0) + d_{33}(11) > 1.3 \,.$$

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

$$d_{31}(01) + d_{32}(0) + d_{33}(10) > 1.3 \,.$$

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. 3 incorporates 6 nodes besides the root and one is to check 6 inequalities. The brute force requires to check 8 inequalities (8).

9 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 (27) in the cryptanalysis of DES below. They depend on different internal bits of the encryption and therefore may be considered independently distributed. We use some of their 10-bit linear projections.

Let n plaintext/ciphertext 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 $\mathrm{rank}(\bar{K}) = 54$.

For each possible value $\bar{K}_{ti}$ one computes the value $S_{ti}(\bar{K}_{ti}) = \omega_{ti} LLR_{ti}(\nu_{ti}, \bar{K}_{ti})$ by (2). One then combines the values of $\bar{K}_{ti}$ into a value of $\bar{K}$ such that

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

for all t and some thresholds z_t to be defined later from a prescribed success probability. One can easily represent all (11) together as a vectorial inequality (7). Therefore the algorithm from Section 8.2 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.

9.1 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 $\mathcal{S}_t(\bar{K})$ on the left hand side of (11) has the normal distribution $\mathbf{N}(u_t, u_t)$ for every t if $\mathtt{x}_t$ follows a priori distribution. Here $u_t = n\,\mu_t C_t^{-1} \mu_t^T$, where $n\mu_t$ and nC_t are the expectation vector and covariance matrix of the vectorial random variables $s_t(\nu_t)$ constructed with LLR statistics for $h_{ti}(\mathtt{x}_t), i = 1, \ldots, m_t$, see Section 7.2. For each t the success is not to miss the correct value of $\bar{K}_t$. The probability of success is computed by

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

where $\mathbf{N}(u_t, u_t)$ denotes a random variable as well. 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 $\bar{K}_{ti}$ are not correct. 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 $\mathcal{S}_t(\bar{K})$ has normal distribution $\mathbf{N}(-u_t, u_t)$. The fraction of incorrect $\bar{K}$ which pass the test for one t is

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

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 with maximum success probability. One searches for z_t such that $\prod_t (1 - \alpha_t) = 2^{s-56}$ to maximise the success probability $\prod_t (1 - \beta_t)$.

10 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, see Section 2 for the definitions.

10.1 Notation

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

10.2 Multivariate Distributions

Assume the plaintext 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 ciphertext X_{R+1}, X_R and any internal bits in the encryption algorithm are then random variables. Given strings of indices (masks) Ω_0, Ω_1, Ω_R,Ω_{R+1}, our goal is to compute a priori distribution of

$$Z = X_0[\Omega_0], X_1[\Omega_1], X_R[\Omega_R], X_{R+1}[\Omega_{R+1}], \tag{15}$$

which is to be used in this cryptanalysis below. Then Z is a vectorial random variable of $|\Omega_0| + |\Omega_1| + |\Omega_R| + |\Omega_{R+1}|$ 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 as it was done for one-bit "linear approximations" in [20].

10.3 Exact Probabilistic Description of a Feistel Cipher

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

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

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

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

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.

10.4 Approximate Probabilistic Description of a Feistel Cipher

We define a larger event $\mathcal{C}_\Gamma$, which means $\mathcal{C}$ implies $\mathcal{C}_\Gamma$, see for instance (17) below and then put $\mathbf{Pr}(\mathcal{E}|\mathcal{C}) \approx \mathbf{Pr}(\mathcal{E}|\mathcal{C}_\Gamma) = \frac{\mathbf{Pr}(\mathcal{E},\mathcal{C}_\Gamma)}{\mathbf{Pr}(\mathcal{C}_\Gamma)}$. That is an approximate description of the cipher. It depends on the event $\mathcal{C}_\Gamma$. Obviously, by taking another event we will have another approximate description of the cipher. As our goal is to compute an approximate distribution of (15), a relevant event $\mathcal{C}_\Gamma$ is to be taken. This approach was already implicitly used by Matsui in [20] to compute probability of his "linear approximations", see Section 10.5. The accuracy of so defined approximate descriptions is unclear. It is even unclear how to measure that accuracy. There are two important parameters which play a role: the quality of the distribution and the number of the key bits which affect the distribution (and the number of the key bits which affect the observation in case of Matsui's Algorithm 2). The quality of the distribution may be measured by its Euclidean distance to an uniform distribution. That measure is called quadratic imbalance in [1]. Intuitively, a better approximate distribution should depend on a lager set of the key bits, see for instance Section 11.1, where another marginally better approximate distribution of $\mathtt{x}_1$ defined by (24) is constructed. However, using such distributions may reduce the efficiency of an attack as they may depend on a significantly larger set of the key bits. At the same time by an informal argument in Appendix 1 any two very good approximate distributions are essentially the same, in particular they essentially depend on the same key bits. That is in accordance with this paper experiments: by computing approximate distributions for the same vector with different trails, one gets an uniform distribution or the distributions which are very close to each other. That may probably mean that using more than one approximate distribution won't provide with any advantage, though that requires further investigation. Anyway, the approach gives good results in practice in the original linear cryptanalysis [20] and in the present work.

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 $\mathbf{Pr}(Z = A|\mathcal{C}_\Gamma)$ for $\mathcal{C}_\Gamma$ defined by

$$X_{i-1}[\Gamma_i] \oplus X_{i+1}[\Gamma_i] = F_i(X_i, K_i)[\Gamma_i], \quad i = 1, \dots, R. \tag{17}$$

We see $\mathbf{Pr}(\mathcal{C}_\Gamma) = 2^{-\sum_{i=1}^{R} |\Gamma_i|}$. One says $\Gamma = (\Gamma_1, \dots, \Gamma_R)$ are output masks for multivariate round approximations (called round sub-vectors here) in R consecutive rounds respectively. Let's denote by Δ_i input masks. The sequence of Γ_i, Δ_i defines a trail, see Section 10.6 for definitions. Trails are classically used to compute probability distributions of one-bit "linear approximations" for DES in [20]. The approximate distribution of (15) does not depend on the input masks Δ_i in the internal rounds, that is for $i = 2, \dots, R-1$, if the trail satisfies some natural conditions, see Section 10.6. Such trails will be called regular. We remark that the probability $\mathbf{Pr}(Z = A|\mathcal{C}_\Gamma)$ only depends on the key bits involved in the right hand sides of (17).

10.5 Approximate Distributions in Matsui's Work

A similar approach was implicitly used by Matsui [20] when computing the distribution of one-bit "linear approximations" to DES encryption algorithm. He used the event $\mathcal{C}'_\Gamma$:

$$X_{i-1}\{\Gamma_i\} \oplus X_{i+1}\{\Gamma_i\} = F_i(X_i, K_i)\{\Gamma_i\}, \quad i = 1, \dots, R,$$

where Γ_i were output masks for round "linear 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 $R = 3$ and $\Gamma = (\{7, 18, 24, 29\}, \emptyset, \{7, 18, 24, 29\})$. Under assumption that $X_0, \dots, X_4$ are uniformly and independently distributed, the probability of $\mathcal{C}'_\Gamma$ is $1/4$. We find $\mathbf{Pr}(f = 0|\mathcal{C}) \approx \mathbf{Pr}(f = 0|\mathcal{C}'_\Gamma) \approx 0.70$ as stated in [20], see Appendix 2 for details.

10.6 Regular Trails

Let $\Gamma_i, \Delta_i, \Theta_i \subseteq \{0, 1, \dots, r-1\}$ and $\Lambda_i \subseteq \{0, 1, \dots, s-1\}$. The sequence of $|\Gamma_i| + |\Delta_i|$-bit strings

$$X_i[\Delta_i], F_i[\Gamma_i], \quad i = 1, \dots, R \tag{19}$$

is called a trail. The members of the trail are called round sub-vectors, they are vectorial functions in X_i. Index subsets Δ_i, Γ_i are called input and output masks for the round sub-vectors ("linear approximation" for the round function in the terminology of [20]), see Fig. 4. The distribution of round sub-vectors are easy to derive from the definition of the round function. Our goal is to compute the joint distribution of some input and output bits (15) for R-round Feistel cipher by using a certain trail.

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

$$\Theta_i \cap (\Gamma_{i-1} \cup \Gamma_{i+1}) \subseteq \Delta_i \subseteq \Theta_i, \quad i = 1, \dots, R, \tag{20}$$

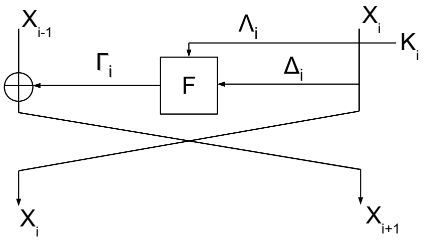

Figure 4: Masks in one Feistel round

where $\Gamma_0 = \Gamma_{R+1} = \emptyset$. We recall that the endpoints of the trail are fixed as we are to compute the distribution of (15). It is easy to check the following statement.

Lemma 3. *Let $n > 3$, then for any strings of indices $\Omega_0, \Omega_1, \Omega_R, \Omega_{R+1}$ in (15) there exists a regular trail (19), such that*

$$\Omega_0 = \Gamma_1, \Omega_1 = \Gamma_2 \cup \Delta_1, \Omega_R = \Gamma_{R-1} \cup \Delta_R, \Omega_{R+1} = \Gamma_R.$$

Proof. We put

i	Γ_i	Δ_i
1	Ω_0	$\Theta_1 \cap \Omega_1$
2	Ω_1	Θ_2
$3 \leq i \leq R-2$	any	Θ_i
$R-1$	Ω_R	Θ_{R-1}
R	Ω_{R+1}	$\Theta_R \cap \Omega_R$

That proves the lemma. □

For $R = 3$ a regular trail exists if and only if $\Omega_3 \setminus \Theta_3 \subseteq \Omega_1$ and $\Omega_1 \setminus \Theta_1 \subseteq \Omega_3$. Generally, there is a large variety of certain auxiliary events $\mathcal{C}_\Gamma$, or equivalently, regular trails for computing approximations to the actual distribution of (15). Those trails produce generally different distributions, and, in particular, the distributions may depend on different key bits.

10.7 Convolution Formula for the Distribution

Assume a regular trail (19), where $\Gamma = (\Gamma_1, \dots, \Gamma_R)$ are output masks. We now produce a convolution type formula to calculate an approximate distribution of the vector

$$Z = X_0[\Gamma_1], X_1[\Gamma_2 \cup \Delta_1], X_R[\Gamma_{R-1} \cup \Delta_R], X_{R+1}[\Gamma_R] \tag{21}$$

for that trail. That is we give a formula to calculate $\mathbf{Pr}(Z = A|\mathcal{C}_\Gamma)$, where $\mathcal{C}_\Gamma$ is defined by (17). Lemma 4 below states that the distribution does not depend on Δ_i, where $i = 2, \dots, R-1$. To simplify notation, we put $\Gamma_0 = \emptyset, \Gamma_{R+1} = \emptyset$ and denote

$$\mathbf{q}_i(b, a, k) = \mathbf{Pr}(X_i[\Delta_i] = b, F_i[\Gamma_i] = a \,|\, K_i[\Lambda_i] = k)$$

the probability distribution of round sub-vectors. In DES, if only non-adjacent S-boxes are involved in the trail (19), then by the definition of F_i we have $\mathbf{q}_i(b, a, k) = \mathbf{q}_i(b \oplus k[\Delta_i], a, 0)$. We denote the latter by $\mathbf{q}_i(b \oplus k[\Delta_i], a)$. The values of $Z = X_0[\Gamma_1], X_1[\Gamma_2 \cup \Delta_1], X_R[\Gamma_{R-1} \cup \Delta_R], X_{R+1}[\Gamma_R]$ are respectively denoted by $A = A_0, A_1, A_R, A_{R+1}$.

Theorem 1. *Let $X_0, \ldots, X_R$ be distributed independently and uniformly at random, and* (19) *be a regular trail. Then*

$$\mathbf{Pr}(Z = A|\mathcal{C}_\Gamma) = \frac{2^{\sum_{i=2}^{R-1}|\Gamma_i|}}{2^{\sum_{i=1}^{R}|(\Gamma_{i-1}\cup\Gamma_{i+1})\setminus\Delta_i|}} \sum_{A_2,\ldots,A_{R-1}} \prod_{i=1}^{R} \mathbf{q}_i(A_i[\Delta_i], (A_{i-1} \oplus A_{i+1})[\Gamma_i], k_i), \quad (22)$$

where the sum is over $A_i = A_i[\Gamma_{i-1} \cup \Gamma_{i+1} \cup \Delta_i]$ (the bits of A_i are indexed by the members of $\Gamma_{i-1} \cup \Gamma_{i+1} \cup \Delta_i$) and $K_i[\Lambda_i] = k_i$.

Proof. By conditional and total probability formulas,

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

where the sum is over $A_j = A_j[\Gamma_{j-1} \cup \Gamma_{j+1} \cup \Delta_j]$, $j = 2, \ldots, R-2$, and as the events

$$\mathcal{A}_1 = \begin{pmatrix} Z & = & A_0, A_1, A_R, A_{R+1}, \\ X_i[\Gamma_{i-1} \cup \Gamma_{i+1} \cup \Delta_i] & = & A_i,\ i = 2, \ldots, R-1, \\ \mathcal{C}_\Gamma & & \end{pmatrix}$$

and

$$\mathcal{A}_2 = \begin{pmatrix} X_i[\Delta_i], F_i[\Gamma_i] & = & A_i[\Delta_i], (A_{i-1} \oplus A_{i+1})[\Gamma_i], \\ X_0[\Gamma_1] & = & A_0, \\ X_i[(\Gamma_{i-1} \cup \Gamma_{i+1}) \setminus \Delta_i] & = & A_i[(\Gamma_{i-1} \cup \Gamma_{i+1}) \setminus \Delta_i], \\ X_{R+1}[\Gamma_R] & = & A_{R+1}, \\ i & = & 1, \ldots, R \end{pmatrix}$$

are equivalent. We took into account that the event $\mathcal{C}_\Gamma$ is defined by $X_{i-1}[\Gamma_i] \oplus X_{i+1}[\Gamma_i] = F_i(X_i, K_i)[\Gamma_i], \quad i = 1, \ldots, R$. By $\mathcal{E}_1$ we denote the event

$$X_i[\Delta_i], F_i[\Gamma_i] = A_i[\Delta_i], (A_{i-1} \oplus A_{i+1})[\Gamma_i], \quad i = 1, \ldots, R,$$

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

$$\begin{aligned} X_0[\Gamma_1] &= A_0, \\ X_i[(\Gamma_{i-1} \cup \Gamma_{i+1}) \setminus \Delta_i] &= A_i[(\Gamma_{i-1} \cup \Gamma_{i+1}) \setminus \Delta_i],\ i = 1, \ldots, R, \\ X_{R+1}[\Gamma_R] &= A_{R+1}. \end{aligned}$$

By the definition of a regular trail, no variables in $X_i[(\Gamma_{i-1} \cup \Gamma_{i+1}) \setminus \Delta_i]$ are relevant to $X_i[\Delta_i], F_i[\Gamma_i]$. Really, only $X_i[\Theta_i \cup \Delta_i]$ are relevant to $X_i[\Delta_i], F_i[\Gamma_i]$. The sets $\Theta_i \cup \Delta_i$ and $(\Gamma_{i-1} \cup \Gamma_{i+1}) \setminus \Delta_i$ have empty intersection as $\Theta_i \cap (\Gamma_{i-1} \cup \Gamma_{i+1}) \subseteq \Delta_i$. So the events $\mathcal{E}_1, \mathcal{E}_2$ are independent as they depend on different bits of X_i, $i = 1, \ldots, R$. We can now split the latter probability into a product. Then

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

As

$$\begin{aligned} \mathbf{Pr}\left(\mathcal{E}_1\right) &= \prod_{i=1}^{R} \mathbf{q}_i(A_i[\Delta_i], (A_{i-1} \oplus A_{i+1})[\Gamma_i], k_i), \\ \mathbf{Pr}\left(\mathcal{E}_2\right) &= 2^{-(|\Gamma_1|+|\Gamma_R|+\sum_{i=1}^{R}|\Gamma_{i-1}\cup\Gamma_{i+1}\setminus\Delta_i|)}, \end{aligned}$$

and $\mathbf{Pr}\left(\mathcal{C}_\Gamma\right) = 2^{-\sum_{i=1}^{R}|\Gamma_i|}$ we get

$$\mathbf{Pr}(Z = A|\,\mathcal{C}_\Gamma) = \frac{2^{\sum_{i=2}^{R-1}|\Gamma_i|}}{2^{\sum_{i=1}^{R}|(\Gamma_{i-1}\cup\Gamma_{i+1})\setminus\Delta_i|}} \sum_{A_2,\ldots,A_{R-1}} \prod_{i=1}^{R} \mathbf{q}_i(A_i[\Delta_i], (A_{i-1}\oplus A_{i+1})[\Gamma_i], k_i).$$

That finishes the proof. □

10.8 Distribution Properties

The conditions of Theorem 1 are satisfied if, for instance, $\Delta_i = \Theta_i \cap (\Gamma_{i-1}\cup\Gamma_{i+1})$. 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

$$X_0\{\Gamma_1\} \oplus X_1\{\Gamma_2 \cup \Delta_1\} \oplus X_R\{\Gamma_{R-1}\cup\Delta_R\} \oplus X_{R+1}\{\Gamma_R\}$$

by summing round approximations $X_i\{\Delta_i\} \oplus F_i\{\Gamma_i\}$ the masks Γ_i, Δ_i are to satisfy $\Gamma_{i-1}\oplus\Gamma_{i+1} = \Delta_i$, see [20]. 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* Δ_i.

Proof. We have

$$\Theta_i \cap (\Gamma_{i-1}\cup\Gamma_{i+1}) \subseteq \Delta_i \subseteq \Theta_i,\ i = 2,\ldots,R-1.$$

Let $\Delta_i' = \Theta_i \cap (\Gamma_{i-1}\cup\Gamma_{i+1})$. Then $\Delta_i' \subseteq \Delta_i$ and $(\Gamma_{i-1}\cup\Gamma_{i+1})\setminus\Delta_i = (\Gamma_{i-1}\cup\Gamma_{i+1})\setminus\Delta_i'$. The statement follows from

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

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

Lemma 4 implies that to reduce calculation cost one can take $\Delta_i = \Theta_i\cap(\Gamma_{i-1}\cup\Gamma_{i+1}), i = 2,\ldots,n-1$ for a regular trail (19). That produces the same distribution by (22). Also we call a regular trail (19) reduced if

$$\Gamma_{i-1}\setminus\Delta_i = \Gamma_{i+1}\setminus\Delta_i.$$

for all $i = 2\ldots R-1$. 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 $A_i = A_i[\Gamma_{i-1}\cup\Gamma_{i+1}\cup\Delta_i]$ only affect

$$\begin{gathered}\mathbf{q}_{i-1}(A_{i-1}[\Delta_{i-1}], (A_{i-2}\oplus A_i)[\Gamma_{i-1}], k_{i-1}),\\ \mathbf{q}_i(A_i[\Delta_i], (A_{i-1}\oplus A_{i+1})[\Gamma_i], k_i),\\ \mathbf{q}_{i+1}(A_{i+1}[\Delta_{i+1}], (A_i\oplus A_{i+2})[\Gamma_{i+1}], k_{i+1}),\end{gathered}$$

in (22). Therefore if $\Gamma_{i-1}\setminus\Delta_i \neq \Gamma_{i+1}\setminus\Delta_i$ 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 $\Delta_i, \Gamma_i = \emptyset, \emptyset$ or the round vector $X_i[\Delta_i], F_i[\Gamma_i]$ is uniformly distributed. Similarly to the proof of Lemma 4, 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.*

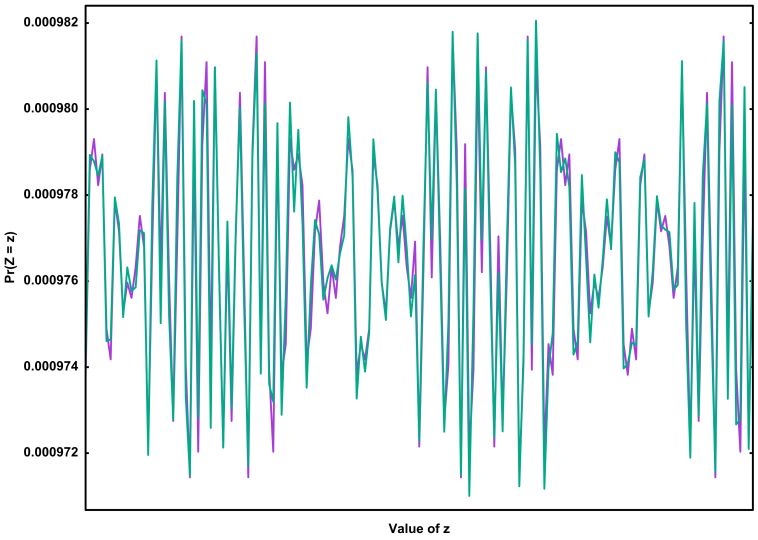

Figure 5: Theoretical and empirical distributions in DES

10.9 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 $1 < i < R$ and the approximate distributions of

$$\begin{aligned} Z_1 &= X_0[\Gamma_1], X_1[\Gamma_2 \cup \Delta_1], X_i[\Gamma_{i-1} \cup \Delta_i], X_{i+1}[\Gamma_i], \\ Z_2 &= X_i[\Gamma_{i+1}], X_{i+1}[\Gamma_{i+2} \cup \Delta_{i+1}], X_R[\Gamma_{R-1} \cup \Delta_R], X_{R+1}[\Gamma_R] \end{aligned}$$

be already computed based on events $\mathcal{C}_{\Gamma'}, \mathcal{C}_{\Gamma''}$, where $\Gamma' = (\Gamma_1, \dots, \Gamma_i)$ and $\Gamma'' = (\Gamma_{i+1}, \dots, \Gamma_R)$. Then the approximate distribution of Z is computed by

Corollary 1.

$$\begin{aligned} & \mathbf{Pr}(Z = A_0, A_1, A_R, A_{R+1} \,|\mathcal{C}_\Gamma) \\ = \; & 2^{|\Gamma_i|} \sum_{A_i, A_{i+1}} \mathbf{Pr}\left(Z_1 = A_0, A_1, A_r[\Gamma_{i-1} \cup \Delta_i], A_{i+1}[\Gamma_i] \,|\, \mathcal{C}_{\Gamma'}\right) \\ \times \; & \mathbf{Pr}\left(Z_2 = A_i[\Gamma_{i+1}], A_{i+1}[\Gamma_{i+2} \cup \Delta_{i+1}], A_n, A_{n+1} |\, \mathcal{C}_{\Gamma''}\right). \end{aligned}$$

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

Fig. 5 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 6-round DES internal bits. 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 plaintext blocks for one randomly chosen cipher key. We realise the distributions are oscillating around $2^{-10} \approx 0.000976$ and are very close, almost indistinguishable. We got a number of such figures besides this one, all of them look similar.

11 Multinomial Distributions for 14-round DES

One of the two best "linear approximations" for 14-round DES found by Matsui in [20] is $X_2\{24, 18, 7\} \oplus X_{15}\{15\} \oplus X_{16}\{24, 18, 7, 29\}$. We included all those bits in (24), and

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

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

added some more bits as intuitively probability distribution on larger vectors reveals more information on the cipher key. However this increases the number of key bits $\texttt{key}_1$ involved in the distribution and key bits $\texttt{Key}_1$ from the first and the last round keys involved in the observation. One is to keep the number of the key bits involved relatively low. Adding some new bits does not increase the number of the key bits involved while adding some others really does. For instance, adding $X_2[29]$ keeps the number of the key bits the same, adding $X_{15}[16, 14, 13, 12, 11]$ enlarges $\texttt{key}_1$ by $K_{15}[23, 21, 20, 19, 18]$ and $\texttt{Key}_1$ by 30 bits from K_{16}. We got 14-bit string

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

Approximate a priori distribution of $\texttt{x}_1$ 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:

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

denoted $\texttt{key}_1$. The distribution is a permutation of the distribution, where $\texttt{key}_1$ is a zero-string by the following Lemma.

Lemma 6. $\mathbf{Pr}(\texttt{x}_1 = A_2, A_{15}, A_{16}\,|k, k_{15}) = \mathbf{Pr}(\texttt{x}_1 = A_2 \oplus k, A_{15} \oplus k_{15}, A_{16}|0, 0\,)$, *where* A_2, A_{15}, A_{16} *are* $4, 6, 4$*-bit strings respectively.*

Proof. We will prove the lemma by applying Theorem 1. To this end we denote

$$\begin{aligned}
\mathbf{p_1}(a, b) &= \mathbf{Pr}(X[4], S_5[3, 2, 1, 0] = a, b),\\
\mathbf{p_2}(a, c) &= \mathbf{Pr}(X[2], S_1[2] = a, c),\\
\mathbf{p_3}(d, b) &= \mathbf{Pr}(X[5, 4, 3, 2, 1, 0], S_5[3, 2, 1, 0] = a, b),
\end{aligned}$$

where a, b, c, d are $1, 4, 2, 6$-bit strings respectively, and $X[5, 4, 3, 2, 1, 0]$ denote all input bits to a respective S-box. Remark that input/output bits are numbered from right to left as in [20]. By Theorem 1 with the trail shown in Table 1,

$$\begin{aligned}
\mathbf{Pr}(\texttt{x}_1 = A_2, A_{15}, A_{16}|\texttt{key}_1\,) \approx\ & 4 \sum \prod_{i=3,5,7,9,11,13} \mathbf{p_1}(A_i \oplus K_i[22], A_{i-1} \oplus A_{i+1})\\
& \times \prod_{i=4,8,12} \mathbf{p_2}(A'_i \oplus K_i[44], A_{i-1} \oplus A_{i+1})\\
& \times\ \mathbf{p_3}(A_{15} \oplus k_{15}, A_{14} \oplus A_{16}),
\end{aligned}$$

where the sum is over 1-bit $A_3, A_5, A_7, A_9, A_{11}, A_{13}$ and 4-bit $A_4, A_6, A_8, A_{10}, A_{12}, A_{14}$, and A'_4, A'_8, A'_{12} denote right-most bits of A_4, A_8, A_{12} respectively. By the definition of the distributions $\mathbf{p_1}(a, b), \mathbf{p_2}(a, c)$, see Appendix 4, we get

$$\begin{aligned}
\mathbf{p_1}(a \oplus 1, b) &= \frac{1}{2^4} - \mathbf{p_1}(a, b),\\
\mathbf{p_2}(a \oplus 1, c \oplus 1) &= \mathbf{p_2}(a, c).
\end{aligned}$$

We transform the expression for $\mathbf{Pr}(\mathtt{x}_1 = A_2, A_{15}, A_{16}\,)$ by introducing new variables and applying those properties. We get the probability depends on k, k_{15} such that $\mathbf{Pr}(\mathtt{x}_1 = A_2, A_{15}, A_{16}|k, k_{15}\,)$ is

$$\begin{aligned}
&\approx \;\; 4\sum \prod_{i=5,7,9,11,13} \mathbf{p_1}(A_i, A_{i-1} \oplus A_{i+1}) \times \prod_{i=8,12} \mathbf{p_2}(A'_i, A_{i-1} \oplus A_{i+1}) \\
&\times \;\; \mathbf{p_1}(A_3, A_2 \oplus A_4)\, \mathbf{p_2}(A'_4 \oplus k, A_3 \oplus A_5)\, \mathbf{p_3}(A_{15} \oplus k_{15}, A_{14} \oplus A_{16}) \\
&= \;\; 4\sum \prod_{i=5,7,9,11,13} \mathbf{p_1}(A_i, A_{i-1} \oplus A_{i+1}) \times \prod_{i=8,12} \mathbf{p_2}(A'_i, A_{i-1} \oplus A_{i+1}) \\
&\times \;\; \mathbf{p_1}(A_3, A_2 \oplus A_4 \oplus k)\, \mathbf{p_2}(A'_4, A_3 \oplus A_5)\, \mathbf{p_3}(A_{15} \oplus k_{15}, A_{14} \oplus A_{16}). \\
&= \;\; \mathbf{Pr}(\mathtt{x}_1 = A_2 \oplus k, A_{15} \oplus k_{15}, A_{16}|0, 0\,). \qquad (26)
\end{aligned}$$

That implies the lemma. □

In the known-plaintext attack we do not observe the bits of (24). They are internal to the encryption algorithm and depend on the first and the last round keys. (24) can be computed by

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

while $X_{16}[24, 18, 7, 29]$ is a part of the ciphertext. Thus the observation depends on some plaintext/ciphertext 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 $\mathtt{Key}_1$. In theory, one can apply a multidimensional linear analysis developed in [15]. Likelihood Ratio statistic will then depend on $\mathtt{key}_1, \mathtt{Key}_1$: overall 44 key bits and one linear combination of the key bits. That makes 2^{45} variants for $\mathtt{key}_1, \mathtt{Key}_1$ to rank 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

$$\mathtt{x}_2 = (X_{15}[24, 18, 7, 29], X_2[16, 15, 14, 13, 12, 11], X_1[24, 18, 7, 29]), \qquad (27)$$

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 $\mathtt{key}_2$. The observation on (27) depends on a 37-bit sub-key from K_1 and K_{16} denoted $\mathtt{Key}_2$. Again, (27) can be computed by

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

while $X_1[24, 18, 7, 29]$ is a part of the plaintext. As above we can not afford using $\mathtt{x}_2$ directly in multidimensional linear analysis. 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 (28) and (29) below are considered independent too.

A natural question to ask is how to find the best possible strings of bits (as $\mathtt{x}_1$ and $\mathtt{x}_2$ for 14-round DES, for instance), which provide with the most efficient key-recovery attack.

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

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

In the original linear cryptanalysis that is the best "linear approximations" for the full or truncated cipher. That seems a very difficult problem as we do not have a ready measure for this superiority. The problem is not completely solved even in Matsui's linear cryptanalysis as his algorithm in [22] does ignore dependencies between "linear approximations" for different S-boxes in one DES round. So there is a theoretical possibility to find even better "linear approximations". In multidimensional linear cryptanalysis the situation is more complicated as the number of the parameters increases, one of most important is the number of the key bits (or key bit linear combinations) involved besides the quality of the distribution itself. Related problem is given a string $\mathtt{x}$ of the encryption internal bits, find a superior trail to compute an approximate distribution for $\mathtt{x}$. That problem looks easier, an informal argument in Appendix 1 shows that all good trails provide with essentially the same approximation, which essentially depends on the same key bits.

11.1 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 with uniform distribution. Quadratic imbalance is the Euclidean distance between two distributions, see [1]. Therefore the new distribution is more powerful (though marginally) when it comes to decide on incorrect cipher key bits. However we remark that in the trail presented in Table 2 the masks Δ_i, Γ_i are generally larger sets than relevant masks in Table 1. So this approximation depends on a significantly larger number of the key bits and therefore trade off seems negative for the efficiency of the attack. For those reasons the distribution is not used in the present analysis. We don't know how to find the best trail but the one in Table 1 works well in practice. By Appendix 1 argument there is essentially only one good approximation to the distribution of $\mathtt{x}_1$.

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

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

$$X_2[24,18,7,29], X_{15}[i,j], X_{16}[24,18,7,29], \tag{28}$$

for different $i,j \in \{16,15,14,13,12,11\}$ except $i=16, j=11$, where the distribution of (28) is uniform. When it is not uniform the distribution generally 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. However if i,j incorporates 16 or 11 then the distribution depends on 2 key bits and all of them are permutations of the same distribution. For instance, the distribution of $X_2[24,18,7,29], X_{15}[16,15], X_{16}[24,18,7,29]$ depends on

$$K_{\{3,5,7,9,11,13,15\}}[22] \oplus K_{\{4,8,12\}}[44], K_{15}[23].$$

This follows from the properties of the distribution $\mathbf{p_4}(d,b) = \mathbf{Pr}(X[5,4], S_5[3,2,1,0] = d,b)$, namely, $\mathbf{p_4}(d+1,b) = \frac{1}{32} - \mathbf{p_4}(d,b)$, see Appendix 4. All 14 such 10-bit vectors are used. The observation on (28) 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 up to 2^{21} variants of the observation and distribution on (28) and that number is affordable. In practice, because of repeated key bits from the key schedule, there are between 2^{17} and 2^{21} variants depending on i and j.

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

By DES symmetry, 10-bit projections

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

of $\mathtt{x}_2$ may be used for the reason above. The distribution of (29) 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. There are between 2^{15} and 2^{21} variants of the observation and distribution on (29) depending on i' and j'.

12 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 $\mathbf{N}(u, u)$ for $u = n\mu C^{-1}\mu^T$, 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. Then we find the threshold z such that the number of plaintext/ciphertext pairs n equals to the number of 56-bit keys for the final brute force. To this end we are to solve 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). In particular $t = 2$,

$$\begin{aligned}\alpha_1 = \alpha_2 &= 0.99257519589049966079368,\\ \beta_1 = \beta_2 &= 0.07804160334301441307569 9,\\ n &= 3972370584411 \approx 2^{41.85},\\ a &= 3.71408966211822134028 88,\\ z &= 0.98061363072909915519076.\end{aligned}$$

Due to bad planning when generating random plaintext blocks to use in our experimental implementation on full DES, we generated 16 independent sets of 2^{39} random messages and their corresponding ciphertexts (with a fixed key). Our experimental implementation of the attack was therefore run with $n = 7 \times 2^{39} \approx 2^{41.81}$. In this case we get

$$\begin{aligned}(1 - \beta_1)^2 &= 0.83,\\ 2^{56}(1 - \alpha_1)^2 &= n,\\ \alpha_1 = \alpha_2 &= 0.99269207541645714573241,\\ \beta_1 = \beta_2 &= 0.08819413839590442011356 8,\\ n &= 3848290697216 \approx 2^{41.81},\\ a &= 3.59807736756681697046 22,\\ z &= 1.03359967632868626438 19.\end{aligned}$$

We have published supplemental material in [10]. From there one can download: the probability distributions of $\mathtt{x}_1$ and $\mathtt{x}_2$; the actual value of the statistics from our experiment $\mathcal{S}_1(\bar{K})$ and $\mathcal{S}_2(\bar{K})$; the weights ω_i for $i = 1, ..., 14$; the vector of the means μ for $LLR_i(\nu_i)$, $i = 1, ..., 14$; the covariance matrix C, and its inverse C^{-1}.

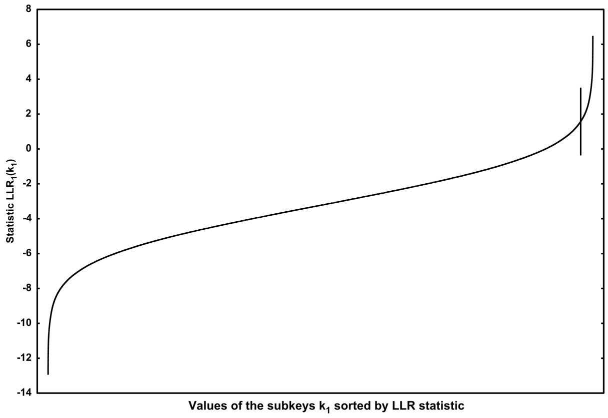

Figure 6: LLR-values for h_1

12.1 One of 28 Projections

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

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

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. 6. The values k_1 are there sorted by $LLR_1(k_1)$ in ascending order.

With $n = 2^{41.81}$ plaintext/ciphertext pairs the expectation of LLR_1 for correct k_1 is 3.23905, for incorrect -3.23905. Experimental value for the correct key is 1.57123, it is presented by the vertical line in Fig. 6. 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.

12.2 Search Tree Complexity

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

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

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 $\bar{K}_i$ (as a set of linear functions) drops upon the fixation of x by a constant. For instance, x_2 relevant to 14 (maximal number) of $\bar{K}_i$, etc.

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 (30). 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

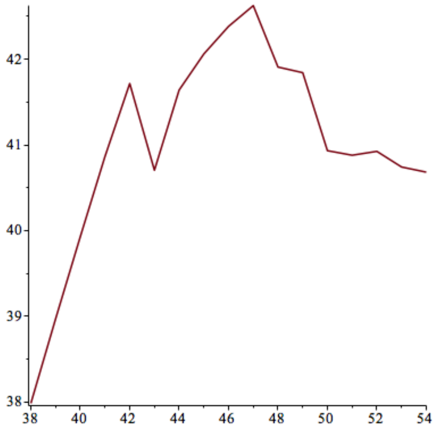

Figure 7: Search tree complexity

the complexity (the number of nodes) of the tree. Search algorithm from Section 8.2 is then run.

The number of examined values of T_j (tree nodes at level j), $j = 38, \ldots, 54$, in $\log_2$ scale are presented in Fig. 7. Overall number of nodes is $2^{45.45} << 2^{54}$. 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 $2^{39.46}$, so the number of 56-bit DES keys to brute force is $2^{41.46}$ again close to what was predicted by our theory. Constructing one node requires few bit xor's and few additions with low precision real numbers, see Section 9. So search tree complexity (constructing $2^{45.45}$ nodes) is lower in bit operations than final brute force of $2^{41.46}$ DES keys. In fact, our implementation works slower than that as the source code is not as optimized as DES libraries are and 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^{31} low precision real numbers (less than 9GB of memory).

12.3 Possible Improvements

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

1. Obviously, one may get better result by using larger strings $\mathtt{x} = (X, Y)$ of the encryption internal bits, see Fig.2.

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. By choosing a different z_j for each level, each slightly less than the statistic for correct key, we managed to reduce the number of visited nodes to $< 2^{34}$. Clearly the success probability in this case is smaller, but it is an indication that these methods may work in practice.

3. The number of nodes in the search tree may probably be further reduced by choosing another sequence of T_j. It is still unclear how to do that in general.

4. Another statistics for the projections $h_1, \ldots, h_m$ may be used. For instance, let the key bits $\mathtt{key}_i$ affect a priori distribution of h_i, and $\mathtt{Key}_i$ affect the observation on h_i.

Then we define

$$LLR_i^*(\texttt{Key}_i) = \max_{\texttt{key}_i} LLR_i(\bar{K}_i).$$

We here neglect that $\texttt{Key}_i$ and $\texttt{key}_i$ may have some key bits in common. 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 [21, 22]. Also one can choose any subset of $\bar{K}_i$ instead of $\texttt{key}_i$ in the definition of LLR_i^* above.

13 Conclusion

Detailed contributions of the present work are presented in Section 4. Three main points are summarised below as well. Firstly, the use of separable statistics. Secondly, efficient algorithm for computing the statistic values by solving an optimisation problem based on gluing together partial information on the cipher key, so there no need to compute the statistic for all values of the key bits involved. Thirdly, formulae for joint approximate distributions of internal bits in Feistel ciphers. Only in the end they are combined to a concrete attack on DES improving on Matsui's results. The combination of the first two may be of independent interest in reducing time complexity of key recovery attacks in other ciphers.

Acknowledgments

The computations were performed on resources provided by UNINETT Sigma2 – the National Infrastructure for High Performance Computing and Data Storage in Norway. Also we are grateful to the reviewers of ToSC for a number of suggestions which helped to improve largely the presentation.

References

[1] 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.

[2] 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.

[3] 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.

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

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

[6] J.Y. Cho, *Linear Cryptanalysis of Reduced-Round PRESENT,* in CT-RSA (J. Pieprzyk ed.), LNCS vol. 5985, Springer, 2010, pp. 302–317.

[7] 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.

[8] B. Collard, F.-X. Standaert, *A Statistical Saturation Attack against the Block Cipher PRESENT,* in CT-RSA 2009 (M. Fischlin ed.), LNCS vol. 5473, Springer, 2009, pp. 195–210.

[9] S. Fauskanger and I. Semaev, *Statistical and Algebraic Properties of DES,* in Inscrypt 2015 (D.Lin et al. eds), LNCS 9589, Springer, 2016, pp.93–107.

[10] S. Fauskanger, I. Semaev, *Separable Statistics and Multidimensional Linear Cryptanalysis - Supplemental Material*, Norstore., [Data set available at the following urls until May 2028],
https://doi.org/10.11582/2018.00013 or
https://archive.norstore.no/pages/public/datasetDetail.jsf?id=10.11582/2018.00013

[11] P. Junod, *On the complexity of Matsui's Attack*, in Selected Areas in Cryptography, (S. Vaudenay S. and A. M. Youssef eds) LNCS vol. 2259, Springer, 2001, pp. 199–211.

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

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

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.

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

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

[22] 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.

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

[24] J. Neyman and E. S. Pearson, *On the Problem of the Most Efficient Tests of Statistical Hypotheses.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. vol. 231 (1933), pp. 289–337.

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

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

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

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

[29] D. R. Stinson, *Cryptography: Theory and Practice*, Second Edition, Chapman & Hall/CRC, Boca Raton, 2002.

14 Appendix 1. Meaning of "Linear Approximation"

We believe that the term "linear approximation" though well settled in the current literature is not very precise. By definition, "linear approximation" is a linear Boolean function in plaintext/ciphertext bits. As the plaintext is random and the cipher key is constant, the function takes 0-value with some probability $q = q(K)$. The latter depends on all cipher key bits K and is hardly possible to utilise. In particular, it looks that there is no way to represent given "linear approximation" in a form suitable for an efficient key recovery attack. Instead, in Matsui's linear cryptanalysis one finds, by using an appropriate trail, an approximate probability $p = p(k)$ which depends on only one linear combination of the key bits k. One then mounts an attack (Algorithm 1) which recovers this linear combination by statistical analysis.

One can consider any vectorial function $\mathtt{x}$ in plaintext/ciphertext bits which has some distribution $q(K)$. Assume its approximate distribution $p(k)$ which depends on a few key bits k. The approximation has a common mathematical meaning: $q(K) - p(k) = \delta(K)$, where δ is a function with real values for any value of K. One can say that the approximate distribution $p(k)$ is "good" if $|\delta(K)|$ is small in comparison with $p(k)$ for any value of K. Therefore to express what a "linear approximation" does approximate in precise terms one sees that it is rather the probability (distribution) q is being approximated by the probability (distribution) p than anything else.

Assume another "good" approximation to the distribution: $q(K) - p_1(k_1) = \delta_1(K)$ for another subset of the key bits k_1 and a small function $\delta_1(K)$. By eliminating q,

$$p(k) = p_1(k_1) + \varepsilon(K) \tag{31}$$

for a small $\varepsilon(K) = \delta_1(K) - \delta(K)$. Let's fix common variables (common linear combinations) in k_1 and k (denoted $k_1 \cap k$) by constants. After the fixation $p(k), p_1(k_1)$ become essentially equal constants as $\varepsilon(K)$ is small for any K. So $p(k), p_1(k_1)$ depend essentially on the same variables $k_1 \cap k$ and equal up to a small additive term. We conclude that any "good" trail produces roughly the same approximate distribution $p(k)$ for $\mathtt{x}$. So using several approximate distributions (several trails) for the same $\mathtt{x}$ can not significantly improve on linear cryptanalysis.

Similar is true for multidimensional distributions of encryption internal bits in the present work. Fig. 5 in Section 10.9 demonstrates a real distribution versus an approximate one, they are very close in the above sense. Also two different approximations (by using two different trails) to the distribution of the same $\mathtt{x}_1$ were computed in Section 11. They differ marginally as it is predicted by the argument above. That is why we use only one trail for $\mathtt{x}_1$ in this cryptanalysis.

15 Appendix 2. On Matsui's Probability Calculation

In this section we show that the distribution of one-bit "linear approximations" used in [20] may be computed only based on $X_0, X_1, \ldots, X_{R+1}$ are independently and uniformly generated and under condition of the auxiliary event $\mathcal{C}'_\Gamma$:

$$X_{i-1}\{\Gamma_i\} \oplus X_{i+1}\{\Gamma_i\} = F_i(X_i, K_i)\{\Gamma_i\}, \quad i = 1, \ldots, R \tag{32}$$

for some Γ_i. We will do that in case of 3-round DES represented in Figure 4 of Matsui's work [20]. The general case is similar. We want to compute the distribution of (18). Let $n = 3$ and $\Gamma = (\Delta, \emptyset, \Delta)$, where $\Delta = \{7, 18, 24, 29\}$. Then $\mathcal{C}'_\Gamma$ is $F_1\{\Delta\} \oplus X_0\{\Delta\} \oplus X_2\{\Delta\} = 0, F_3\{\Delta\} \oplus X_2\{\Delta\} \oplus X_4\{\Delta\} = 0$. We have

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

where the sum is over binary a, b, c, d. We now take into account that $f = [F_1\{\Delta\} \oplus X_1\{15\} \oplus K_1\{22\}] \oplus [F_3\{\Delta\} \oplus X_3\{15\} \oplus K_3\{22\}] \oplus [F_1\{\Delta\} \oplus X_0\{\Delta\} \oplus X_2\{\Delta\}] \oplus [F_3\{\Delta\} \oplus X_2\{\Delta\} \oplus X_4\{\Delta\}]$. 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

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

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

$$\begin{aligned}
\mathbf{Pr}(f = 0|\mathcal{C}) &\approx \sum_{a,c,d} \mathbf{Pr}\begin{pmatrix} F_1\{\Delta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\Delta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & a, \\ X_1\{15\} & = & c, \\ X_3\{15\} & = & d. \end{pmatrix} \\
&= \sum_{a} \mathbf{Pr}\begin{pmatrix} F_1\{\Delta\} \oplus X_1\{15\} \oplus K_1\{22\} & = & a, \\ F_3\{\Delta\} \oplus X_3\{15\} \oplus K_3\{22\} & = & a. \end{pmatrix} \\
&= \sum_{a} \mathbf{Pr}(F_1\{\Delta\} \oplus X_1\{15\} \oplus K_1\{22\} = a)\,\mathbf{Pr}(F_3\{\Delta\} \oplus X_3\{15\} \oplus K_3\{22\} = a) \\
&= \left(\frac{12}{64}\right)^2 + \left(1 - \frac{12}{64}\right)^2 \approx 0.70.
\end{aligned}$$

16 Appendix 3. Another Statistic

We will use the notation introduced in Section 7.1. Let's denote $\nu(n) = (\nu_1, \ldots, \nu_m)$, a vector of length $M = \sum_{i=1}^{m} N_i$, a concatenation of $\nu_i(n)$. We can 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 the distribution P. Then by μ_P and C_P we denote here the expectation and the covariance matrix for $\nu(1)$. Remark that the symbols μ_P and C_P are used in Section 7.2 in a different context. We have

$$\mu_P = (\mu_{P,1}, \ldots, \mu_{P,m}),$$

where $\mu_{P,i} = \left(\sum_{h_i(a)=1} p_a, \ldots, \sum_{h_i(a)=N_i} p_a\right)$ 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,

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

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 ν tends to a multivariate normal distribution $\mathbf{N}(n\mu_P, nC_P)$ with expectations $n\mu_P$ and covariance matrix nC_P. Similarly, if $\mathtt{x}$ has the distribution Q, then ν tends to $\mathbf{N}(n\mu_Q, nC_Q)$. To decide which distribution P or Q is correct by observing the value of ν, one can apply the Neyman-Pearson approach. However as the matrices C_P, C_Q are singular, the normal distributions do not have densities.

A standard solution is to consider a truncation of $\nu(n) = (\nu_1, \ldots, \nu_m)$. For instance, let $\nu'(n) = (\nu'_1, \ldots, \nu'_m)$, where ν'_i is produced from ν_i by dropping one entry of the latter. Recall that ν_i is a vector of observations on the values of $h_i(\mathtt{x})$. Then by μ'_P and C'_P we denote here the expectation and the covariance matrix for $\nu'(1)$ when $\mathtt{x}$ follows the distribution P. If $h_i(\mathtt{x}), h_j(\mathtt{x})$ for $i \neq j$ are independent random variables, then C'_P is diagonal and invertible. Similarly, when $\mathtt{x}$ follows the distribution Q and $h_i(\mathtt{x}), h_j(\mathtt{x})$ for $i \neq j$ are independent, then C'_Q is diagonal and invertible.

Therefore, one constructs an LLR statistic to distinguish two multivariate normal distributions:

$$\mathcal{S}'(\nu) = \frac{1}{n}(-\left[\nu' - n\mu'_P\right] C_P'^{-1} \left[\nu' - n\mu'_P\right]^T + \left[\nu' - n\mu'_Q\right] C_Q'^{-1} \left[\nu' - n\mu'_Q\right]^T).$$

In that case

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

is a separable statistic. However as the projections (28) are dependent(the same is true for (29)), the statistic $\mathcal{S}'$ is not applicable within this cryptanalysis.

17 Appendix 4. Some multivariate distributions on DES S-boxes

Let

$$\begin{array}{rcl} \mathbf{p_1}(a,b) & = & \mathbf{Pr}(X[4], S_5[3,2,1,0] = a,b), \\ \mathbf{p_2}(a,c) & = & \mathbf{Pr}(X[2], S_1[2] = a,c), \\ \mathbf{p_4}(d,b) & = & \mathbf{Pr}(X[5,4], S_5[3,2,1,0] = d,b), \end{array}$$

where a, b, c, d are 1-bit, 4-bit, 1-bit and 2-bit binary strings, respectively. The distribution $\mathbf{p_2}(a,c) = \mathbf{Pr}(X[2], S_1[2] = a,c)$ is

$$\begin{pmatrix} \frac{15}{64} & \frac{17}{64} \\ \frac{17}{64} & \frac{15}{64} \end{pmatrix},$$

where the rows are numbered by $a = 0,1$ and the columns by $c = 0,1$. The distribution $\mathbf{p_1}(a,b) = \mathbf{Pr}(X[4], S_5[3,2,1,0] = a,b)$ is

$$\begin{pmatrix} 0 & \frac{1}{16} & \frac{1}{16} & 0 & \frac{3}{64} & 0 & \frac{1}{64} & \frac{1}{16} & \frac{1}{32} & 0 & \frac{1}{32} & \frac{1}{16} & \frac{3}{64} & \frac{3}{64} & \frac{1}{32} & 0 \\ \frac{1}{16} & 0 & 0 & \frac{1}{16} & \frac{1}{64} & \frac{1}{16} & \frac{3}{64} & 0 & \frac{1}{32} & \frac{1}{16} & \frac{1}{32} & 0 & \frac{1}{64} & \frac{1}{64} & \frac{1}{32} & \frac{1}{16} \end{pmatrix},$$

where the rows are numbered by $a = 0,1$ and the columns by $b = 0,\ldots,15$. The distribution $\mathbf{p_4}(d,b) = \mathbf{Pr}(X[5,4], S_5[3,2,1,0] = d,b)$ is

$$\begin{pmatrix} 0 & \frac{1}{32} & \frac{1}{32} & 0 & \frac{1}{32} & 0 & \frac{1}{64} & \frac{1}{32} & 0 & 0 & \frac{1}{64} & \frac{1}{32} & \frac{1}{32} & \frac{1}{64} & \frac{1}{64} & 0 \\ \frac{1}{32} & 0 & 0 & \frac{1}{32} & 0 & \frac{1}{32} & \frac{1}{64} & 0 & \frac{1}{32} & \frac{1}{32} & \frac{1}{64} & 0 & 0 & \frac{1}{64} & \frac{1}{64} & \frac{1}{32} \\ 0 & \frac{1}{32} & \frac{1}{32} & 0 & \frac{1}{64} & 0 & 0 & \frac{1}{32} & \frac{1}{32} & 0 & \frac{1}{64} & \frac{1}{32} & \frac{1}{64} & \frac{1}{32} & \frac{1}{64} & 0 \\ \frac{1}{32} & 0 & 0 & \frac{1}{32} & \frac{1}{64} & \frac{1}{32} & \frac{1}{32} & 0 & 0 & \frac{1}{32} & \frac{1}{64} & 0 & \frac{1}{64} & 0 & \frac{1}{64} & \frac{1}{32} \end{pmatrix},$$

where the rows are numbered by $a = 0,\ldots,3$ and the columns by $b = 0,\ldots,15$.