Linear Cryptanalysis of DES with Asymmetries

Andrey Bogdanov and Philip S. Vejre(✉)

Technical University of Denmark, Kongens Lyngby, Denmark
{anbog,psve}@dtu.dk

Abstract. Linear cryptanalysis of DES, proposed by Matsui in 1993, has had a seminal impact on symmetric-key cryptography, having seen massive research efforts over the past two decades. It has spawned many variants, including multidimensional and zero-correlation linear cryptanalysis. These variants can claim best attacks on several ciphers, including PRESENT, Serpent, and CLEFIA. For DES, none of these variants have improved upon Matsui's original linear cryptanalysis, which has been the best known-plaintext key-recovery attack on the cipher ever since. In a revisit, Junod concluded that when using 2^{43} known plaintexts, this attack has a complexity of 2^{41} DES evaluations. His analysis relies on the standard assumptions of right-key equivalence and wrong-key randomisation.

In this paper, we first investigate the validity of these fundamental assumptions when applied to DES. For the right key, we observe that strong linear approximations of DES have more than just one dominant trail and, thus, that the right keys are in fact *inequivalent* with respect to linear correlation. We therefore develop a new right-key model using Gaussian mixtures for approximations with several dominant trails. For the wrong key, we observe that the correlation of a strong approximation after the partial decryption with a wrong key still shows much *non-randomness*. To remedy this, we propose a novel wrong-key model that expresses the wrong-key linear correlation using a version of DES with more rounds. We extend the two models to the general case of multiple approximations, propose a likelihood-ratio classifier based on this generalisation, and show that it performs better than the classical Bayesian classifier.

On the practical side, we find that the distributions of right-key correlations for multiple linear approximations of DES exhibit exploitable *asymmetries*. In particular, not all sign combinations in the correlation values are possible. This results in our improved multiple linear attack on DES using 4 linear approximations at a time. The lowest computational complexity of $2^{38.86}$ DES evaluations is achieved when using $2^{42.78}$ known plaintexts. Alternatively, using 2^{41} plaintexts results in a computational complexity of $2^{49.75}$ DES evaluations. We perform practical experiments to confirm our model. To our knowledge, this is the best attack on DES.

Keywords: Linear cryptanalysis · DES · Mixture models · Right-key equivalence · Wrong-key randomisation · Linear hull · Multiple linear

T. Takagi and T. Peyrin (Eds.): ASIACRYPT 2017, Part I, LNCS 10624, pp. 187–216, 2017.
https://doi.org/10.1007/978-3-319-70694-8_7

1 Introduction

Accepted as a standard in 1976 by the National Bureau of Standards (later NIST), DES can now celebrate its fortieth birthday. Being a highly influential cipher, it has inspired much cryptanalysis. Triple-DES is still massively deployed in conservative industries such as banking. Moreover, it is used to secure about 3% of Internet traffic [1].

The first attack on the full DES came in 1992, where Biham and Shamir demonstrated that *differential cryptanalysis* enabled a key recovery using 2^{47} *chosen plaintexts* in time 2^{37} [2]. The year after, in 1993, Matsui introduced a new cryptanalytic technique, *linear cryptanalysis*, which DES proved especially susceptible to. While the first iteration of the attack required 2^{47} *known plaintexts* [21], Matsui soon improved his attack to only require 2^{43} known texts, taking 2^{43} time to recover the key. This complexity estimate was lowered to 2^{41} by Junod in [17]. In [18], Knudsen and Mathiassen lower the complexity to 2^{42} plaintexts, however this attack uses *chosen plaintexts*.

In this paper we present the first successful attack on full DES using multiple linear approximations. By developing new models for the correlation distributions, and by exploiting asymmetries in the right-key distribution, we obtain an improved key-recovery attack. Using $2^{42.78}$ known plaintexts, the attack recovers the key in time equal to $2^{38.86}$ DES encryptions.

1.1 Previous Work and Problems

Linear cryptanalysis has proven to be widely applicable, and has spawned many variants and generalisations. Amongst them are differential-linear cryptanalysis [19], multiple linear cryptanalysis [3,16], multidimensional linear cryptanalysis [14,15], zero-correlation linear cryptanalysis [5,6], multivariate linear cryptanalysis [8], etc. These techniques have successfully been applied to a wide range of ciphers, including Serpent [14,23], PRESENT [8,9], Camellia and CLEFIA [4], and CAST-256 [27].

Matsui first introduced the concept of a *linear approximation* of a block cipher in [21]. If we denote the encryption of a plaintext $\mathcal{P}$ using key K by $\mathcal{C} = E_K(\mathcal{P})$, then a linear approximation of this cipher is a pair of masks, (α, β), which indicate some bits of the plaintext and ciphertext. The idea is to find α and β such that the sum of plaintext bits indicated by α is strongly correlated to the sum of ciphertext bits indicated by β. A measure of the strength of a linear approximation is the *linear correlation*, defined by

$$C_K(\alpha, \beta) = 2 \cdot \Pr(\langle \alpha, x \rangle \oplus \langle \beta, E_K(x) \rangle = 0) - 1,$$

where $\langle \cdot, \cdot \rangle$ is the canonical inner product. Matsui showed how an approximation with linear correlation that deviates significantly from zero can be used to attack the cipher, and found such approximations for DES. The attack procedure was formalised as Algorithm 2, in which an attacker obtains plaintext-ciphertext pairs over r rounds of a cipher. The attacker then guesses the outer round keys in order to encrypt/decrypt the outer rounds, and compute the correlation over $r - 2$ rounds.

Standard assumptions for linear cryptanalysis on DES. In [17] Junod revisited Matsui's attack, and concluded that Matsui's original complexity was slightly overestimated. Junod instead estimated that the attack could be performed in time 2^{41} using the same number of known plaintexts. Central to both Matui's and Junod's analysis are two assumptions.

Assumption A (Right-Key Equivalence). *For a linear approximation* (α, β)*, the magnitude of the correlation,* $|C_K(\alpha, \beta)|$*, does not deviate significantly from its expected value over all keys, that is,* $|C_K(\alpha, \beta)| = \mathrm{E}(|C_K(\alpha, \beta)|)$*.*

Problem 1: Insufficient Right-Key Distribution: The assumption of right-key equivalence is usually the result of assuming that the magnitude of the linear correlation is determined by a single dominant trail. This further implies that the linear correlation only takes on two values over the key space. However, in [24], Nyberg first introduced the concept of a *linear hull*, i.e. the collection of all trails of a linear approximation, and showed that Assumption A is not true in general. In [7], Bogdanov and Tischhauser gave a refined version of Assumption A, which takes a larger part of the hull into account. However, to the best of our knowledge, no thorough exploration of the right-key distribution for DES has been conducted, and it is unclear how accurate Assumption A is in this context.

Assumption B (Wrong-Key Randomisation). *In the context of Algorithm 2, the correlation of a linear approximation* (α, β) *is equal to 0 for all wrong guesses of the outer round keys.*

Problem 2: Unrealistic Wrong-Key Distribution: The assumption of wrong-key randomisation implies that if an attacker guesses the wrong outer round keys in Algorithm 2, the resulting texts pairs behave in a completely random way, i.e. the linear correlation will be equal to zero. A refined version of this assumption was given by Bogdanov and Tischhauser in [7], where the wrong-key distribution was given as the Gaussian distribution $\mathcal{N}(0, 2^{-n})$, where n is the block size. This distribution matches that of an ideal permutation. Neither of these assumptions have been verified for DES. Indeed, DES exhibits very strong linear approximations, and it is not clear if a wrong key guess is sufficient to make the linear correlation close to that of an ideal permutation.

Linear cryptanalysis of DES with multiple approximations. While several models for using multiple approximations for linear cryptanalysis have been proposed, see e.g. [3,8,14–16,26], the application to DES has been very limited. In [16], Kaliski and Robshaw specifically note that their approach is limited when applied to DES. In [26], Semaev presents an alternative approach, but does not obtain better results than Matsui's original attack.

The most promising attempt was given in [3] by Biryukov et al. Under Assumption A, when using M approximations, the key space can be partitioned into at most 2^M key classes based on the signs of the M linear correlations. This allowed Biyukov et al. to describe the correlation of each key class as an

M-variate normal distribution $\mathcal{N}_M(\boldsymbol{\mu}_i, 1/N \cdot \mathbf{I})$, where $\mathbf{I}$ is an $M \times M$ identity matrix, and the mean vector is given by

$$\boldsymbol{\mu}_i = (s_{i,1}|C_K(T_1)|, \dots, s_{i,M}|C_K(T_M)|)^\top,$$

where $s_{i,j} \in \{-1, 1\}$ describes the sign combination of the i'th key class. Based on this, they developed a Bayesian classifier, in order to decide between a correct or incorrect guess of the round keys in Algorithm 2.

Problem 3: Applying Multiple Linear Cryptanalysis to DES: While Biryukov et al. demonstrate that their method of using multiple approximations can potentially reduce the complexity of Matsui's attack, they also note that the structure of DES makes it difficult to arbitrarily use a large number of approximations. As such, they did not present a new attack on DES. Similar observations were made by Kaliski and Robshaw in [16]. To the best of our knowledge, no other variants of linear cryptanalysis which uses multiple approximations have been able to outperform Matsui's original attack.

1.2 Our Contributions

More Accurate Right-Key Model for DES. In Sect. 3 we consider Problem 1, i.e. the fundamental problem of the DES right-key distribution. We enumerated over 1000 trails for the linear approximation used by Matsui, and calculated the resulting correlation distribution for 1 million keys. We demonstrate in Sect. 3.2 that while this distribution does have two modes symmetric around zero, each mode does not consist of a single value, as predicted by Assumption A. Indeed, it is not even the case that each mode takes on a simple Gaussian distribution. As such, one cannot consider different keys to have equivalent behaviour.

We therefore develop a new model for the right-key distribution in Sect. 3.3. This model is given below, and expresses the distribution as a mixture of Gaussian components. An example of this model applied to DES is shown in Fig. 1.

Model A (Right-Key Equivalence for One Approximation). *Consider a linear approximation (α, β) of r rounds of DES. The distribution of the linear correlation $C_K(\alpha, \beta)$ over the key space is approximately given by a Gaussian mixture for some weights λ_i and components $\mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, \dots, \ell$.*

Applying this model to the approximations used by Matsui, we show that it is able to accurately describe the observed distribution. Moreover, it is interesting to note that the component associated with the dominant trail *only accounts for 30% of the correlation, contrasting Assumption* A. We furthermore apply the mixture model to describe the full correlation distribution observed during an attack. We note that when the number of texts used in the attack is small, the right-key distribution originally given by Matsui is a good approximation. However, we stress that the cryptanalyst should carefully examine the right-key distribution when this is not the case.

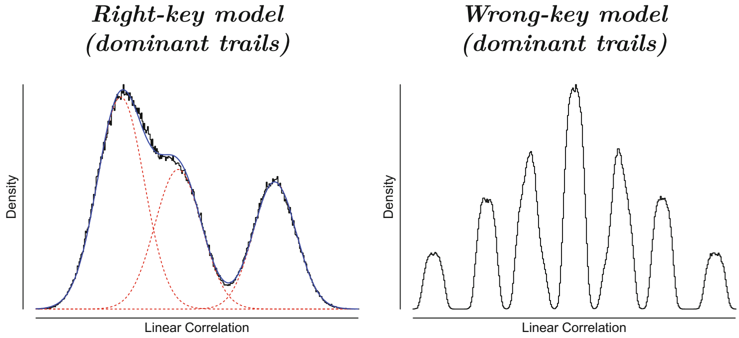

Fig. 1. Our new models for the distributions of linear correlation over the key space for DES. The distributions are expressed as Gaussian mixtures. The model shows a deviation from the standard assumptions of right-key equivalence and wrong-key randomisation.

New Wrong-Key Model for DES. In Sect. 4 we consider Problem 2. In order to obtain a wrong-key model that more accurately describes the case of a wrong key guess in Algorithm 2, we propose the following new approach.

Model B (Non-random Wrong-Key Distribution). *Consider an Algorithm 2 style attack on r rounds of DES using a linear approximation (α, β) over $r - 2$ rounds. Let R_K be the keyed round function of DES, and let $E^\star_K$ denote the r-round encryption function. For a wrong guess of the outer round keys, the correlation will be distributed as for the cipher*

$$E'_K(x) = R^{-1}_{K_a}(E^\star_K(R^{-1}_{K_b}(x))), \qquad (1)$$

where K_a and K_b are chosen uniformly at random.

This model accurately matches the situation of guessing the wrong outer round keys in an Algorithm 2 attack. We enumerated over 900 trails for the linear approximation used by Matsui for the cipher E', and calculated the resulting correlation distribution for 1 million keys. The result is shown in Fig. 1. While the distribution has mean zero, the shape of the distribution does not match Assumption B, nor that of the revised version by Bogdanov and Tischhauser, as its variance is much larger than 2^{-n}. As is the case for the right-key distribution, the wrong-key distribution is also not a simple Gaussian, but rather some Gaussian mixture. Again, for low data complexities, we demonstrate that a Gaussian model is sufficient to describe the wrong-key distribution observed during an attack, but advise caution when the data complexity is close to full codebook.

Multiple Linear Cryptanalysis with Asymmetries. In Sects. 5 and 6 we remedy Problem 3. We develop a classifier for M approximations based on the likelihood-ratio of the right-key and wrong-key distributions developed in Sects. 3 and 4. This classifier is given by

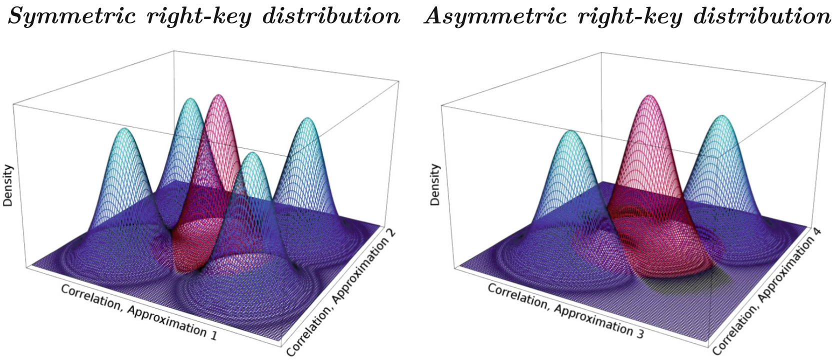
Symmetric right-key distribution *Asymmetric right-key distribution*

Fig. 2. An illustration of the difference between a symmetric and an asymmetric joint distribution of linear correlation for two approximations over the key space. The right-key distribution is blue, while the wrong-key distribution is red. (Color figure online)

$$\Lambda(\mathbf{x}) = \frac{\sum_{i=1}^{\ell} \lambda_i \phi_M(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + (2^{-n} + 1/N)\mathbf{I})}{\phi_M(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_W + (2^{-n} + 1/N)\mathbf{I})},$$

where ϕ is the probability density function (PDF) of the Gaussian distribution. The wrong-key distribution is a simple M-variate Gaussian. The right-key distribution is a mixture of at most 2^M, M-variate components based on the signs of the M correlations. In contracts to the work in [3], we do not partition the key space, but express the correlation distribution over the entire key space. Also in contrast to this work, our classifier directly takes the wrong-key distribution into account. We demonstrate how this improves the classifier.

We make the interesting observation that if the right-key distribution is asymmetric, that is, if the number of components is less than 2^M, we obtain a stronger classifier. This situation is demonstrated in Fig. 2. From this example, one can get an intuitive understanding of how an asymmetric distribution makes it easier to distinguish between right-key and wrong-key. We therefore propose the term *symmetry factor*, namely the ratio between number of components and 2^M, and conjecture that a lower symmetry factor will result in a stronger attack.

First Successful Multiple Linear Cryptanalysis of DES. By using the asymmetric classifier in Sect. 6, we give the first attack on full DES using multiple linear approximations which improves Matsui's original attack. We use two sets of four linear approximations. Using $2^{42.78}$ known plaintexts, the attack recovers the key in time equal to $2^{38.86}$ encryptions, with a success probability of 85%. This is 4.4 times faster than Junod's estimate of Matsui's attack, and uses $2^{40.2}$ fewer texts. We confirm these results by measuring the actual correlation distributions using this number of texts for 1300 random keys, and computing the resulting advantage of our classifier. We find that the model fits our practical results very well. Alternatively, we can lower the data complexity to 2^{41}, and recover the key in time $2^{49.76}$, with a success probability of 80%. Our attack is compared to previous attacks on full DES in Table 1.

Table 1. Comparison of key-recovery attacks on full DES.

Technique	Data complexity	Time complexity	Success probability	Attack scenario	Source
Differential	$2^{47.00}$	$2^{37.00}$	58%	Chosen plaintext	[2]
Linear	$2^{43.00}$	$2^{43.00}$	85%	Known plaintext	[22]
Linear	$2^{43.00}$	$2^{41.00}$	85%	Known plaintext	[17]
Multiple linear	$2^{42.78}$	$2^{38.86}$	85%	Known plaintext	Sect. 6
Multiple linear	$2^{41.00}$	$2^{49.76}$	80%	Known plaintext	Sect. 6

2 Linear Cryptanalysis of DES

In 1993, Matsui introduced the concept of linear cryptanalysis and applied it to derive a key-recovery attack on the full 16-round DES [21,22]. In this section, we briefly outline the attack. We then give an overview of the assumptions Matsui made in his analysis, and show the resulting complexity of the attack. Moreover, we show a variant of linear cryptanalysis due to Biryukov et al. [3], which will be important for the remaining part of this work.

2.1 Basics of Linear Cryptanalysis

We consider a block cipher with block length n and key length κ. We denote the encryption of plaintext $\mathcal{P} \in \mathbb{F}_2^n$ under key $K \in \mathbb{F}_2^\kappa$ by $E_K(\mathcal{P})$. The idea of linear cryptanalysis is to find a *linear approximation* $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ such that the magnitude of its *linear correlation*, defined by

$$C_K(\alpha, \beta) = 2 \cdot \Pr(\langle \alpha, x \rangle \oplus \langle \beta, E_K(x) \rangle = 0) - 1,$$

is large. Here, $\langle \cdot, \cdot \rangle$ denotes the canonical inner product on $\mathbb{F}_2^n$. Thus, the correlation is a measure of how often the parity bit $\langle \alpha, \mathcal{P} \rangle$ of the plaintext is equal to the parity bit $\langle \beta, \mathcal{C} \rangle$ of the ciphertext. We expect a strong cipher to only have approximations with linear correlation close to 0, and hence a correlation value that deviates significantly from 0 indicates a weakness of the cipher.

For Feistel ciphers, such as DES, the linear correlation of an approximation (α, β) can be calculated by considering so called *linear trails* of the cipher. We define a single-round linear trail of DES as the triple $(u, t, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m \times \mathbb{F}_2^n$, where m is the size of a single round key. The linear correlation of this single-round trail is then defined as

$$C_{K_r}(u, t, v) = 2 \cdot \Pr(\langle u, x \rangle \oplus \langle v, R_{K_r}(x) \rangle = \langle t, K_r \rangle) - 1,$$

where R_{K_r} is the DES round-function using the r'th round key K_r. We now define a linear trail T over r rounds as a collection of single-round trails (u_i, t_i, u_{i+1}), $i = 0, \ldots, r-1$, as well as the *correlation contribution* of the trail T as [11,12] $C_K(T) = \prod_{i=0}^{r-1} C_{K_i}(u_i, t_i, u_{i+1})$. We will also make use of the concept of an associated *key trail* $\bar{T}$ of a trail T. The key trail is defined as the concatenation of the t_i, $i = 0, \ldots, r-1$.

Daemen and Rijmen demonstrated that the correlation contribution of a trail can be written as [11,12]

$$C_K(T) = (-1)^{s_T \oplus \langle \bar{T}, \bar{K} \rangle} |C_K(T)|, \tag{2}$$

where s_T is a sign bit specific to the trail T, and $\bar{K}$ denotes the concatenation of the round keys K_i. Moreover, under the assumption of independent round keys, $|C_K(T)|$ is independent of the key. Thus, the correlation contribution of a trail T has a fixed magnitude for all keys, but the sign is determined by the round key bits indicated by the key trail $\bar{T}$. Finally, Daemen and Rijmen give the correlation over all r rounds for some approximation (α, β) as [11,12]

$$C_K(\alpha, \beta) = \sum_{u_0 = \alpha, u_r = \beta} C_K(T) = \sum_{u_0 = \alpha, u_r = \beta} (-1)^{s_T \oplus \langle \bar{T}, \bar{K} \rangle} |C_K(T)|, \tag{3}$$

i.e. the sum of the correlation contributions of all trails from α to β.

2.2 Matsui's Approach

Matsui's key observation was that DES exhibits linear trails where the correlation contribution deviates significantly from zero. Consider the full 16-round DES, let $\mathcal{P}$ be the plaintext, and let $\mathcal{C}$ be the ciphertext. Let $[i_0, \ldots, i_\ell]$ denote an element in $\mathbb{F}_2^n$ whose i_j'th components are 1, $j = 0, \ldots, \ell$, while all other components are 0. Then, over 14 rounds of DES, the approximations

$$\gamma_1 = ([7, 18, 24], [7, 18, 24, 29, 47]) \quad \text{and} \quad \delta_3 = ([15, 39, 50, 56, 61], [39, 50, 56]),$$

both have trails with correlation contribution $C_K(T) = \pm 2^{-19.75}$ [22]. From (2) we can determine one bit of information if we know the sign of $C_K(T)$, namely the parity $\langle \bar{T}, \bar{K} \rangle$ of the round key bits indicated by the key trail $\bar{T}$. Let k_f denote the key-bits of round key K_0 required to partially encrypt a plaintext $\mathcal{P}$ one round and calculate $\langle \alpha, R_{K_0}(\mathcal{P}) \rangle$, and let k_b denote the key-bits of round key K_{r-1} required to partially decrypt the ciphertext $\mathcal{C}$ one round and calculate $\langle \beta, R^{-1}_{K_{r-1}}(\mathcal{C}) \rangle$. Matsui developed the following general approach in order to determine $|k_f| + |k_b| + 1$ key bits, formalised as Algorithm 2.

Algorithm 2

1. Obtain N plaintext-ciphertext pairs.
2. For each guess of the key-bits (k_f, k_b), partially encrypt/decrypt each plaintext-ciphertext pair $(\mathcal{P}, \mathcal{C})$ and calculate the number of times L_i the input parity $\langle \alpha, R_{R_0}(\mathcal{P}) \rangle$ is equal to the output partiy $\langle \beta, R^{-1}_{R_{r-1}}(\mathcal{C}) \rangle$ for the i'th guess, $i = 1, \ldots, 2^{|k_f| + |k_b|}$.

3. For each counter L_i, if $L_i > N/2$, guess that the sign bit $\langle \bar{T}, \bar{K} \rangle = s_T$, otherwise guess that $\langle \bar{T}, \bar{K} \rangle = s_T \oplus 1$.
4. For any counter L_i with $|T_i - N/2| > \varGamma$, for a predetermined value $\varGamma$, guess the remaining $\kappa - (|k_f| + |k_b| + 1)$ bits of the master key K, and determine the correct value of K through trial encryption.

For his attack on DES, Matsui performed Algorithm 2 once for γ_1 and once for δ_3, determining 26 bits before guessing the remaining 30 bits of K. In his analysis of the success rate and complexity of the attack, Matsui assumed that the linear correlation of the approximations γ_1 and δ_3 were only determined by a single trail T. The idea is that the correlation contribution of T is much larger than that of all other trails – a so called *dominant trail*. We will call the associated key trail $\bar{T}$ of such a trail a *dominant key trail*. In the presence of such a dominant trail, $C_K(\alpha, \beta)$ only takes on two values over the key space. This can be seen from Eq. (3), as the case of a dominant trail implies that this sum only has one term. Under this assumption, Matsui concluded that when using 2^{43} texts, there is an 85% probability of recovering the key at a time complexity of 2^{43} DES encryptions. In a later analysis of Matsui's attack [17], Junod concluded that the actual computational complexity is closer to 2^{41} DES encryptions.

2.3 Biryukov et al. – Multiple Approximations

A natural extension of Matsui's linear cryptanalysis is to attempt to use multiple linear approximations simultaneously. The first attempt at developing such a framework was by Kaliski and Robshaw in [16]. This work has the limitation that all linear approximations must have the same dominant key trail, and the approximations were assumed to be statistically independent. Moreover, as Kaliski and Robshaw note, the application of this method to DES is very limited.

Another approach was undertaken by Biryukov et al. in [3]. Here, the approximations can in principle be picked arbitrarily, but the framework still requires the assumption of one dominant trail for each approximation, and independence between approximations. Due to these restrictions, the foundations of multidimensional linear cryptanalysis was developed in e.g. [14,15]. While this approach has been applied with great success to a large range of ciphers, no results have been shown on DES. Thus, Matsui's single linear cryptanalysis still provides the best results on this cipher.

Let us briefly reconsider the method by Biryukov et al., assuming the use of M linear approximations. The idea is to partition the key space into at most 2^M classes based on the parity of the $\langle \bar{T}_i, \bar{K} \rangle$, where $\bar{T}_i$ is the dominant key trail of the i'th approximation. An Algorithm 2 type attack is then performed: For each guess of the key-bits (k_f, k_b), the vector $(L_{i,1}, \ldots, L_{i,M})$ is calculated, and the likelihood of that vector belonging to each of the key classes is computed. The right guess of (k_f, k_b) should yield one class with high likelihood, and the class then indicates at most M parity bits, $\langle \bar{T}_i, \bar{K} \rangle$. Central to the analysis of [3] are the following two assumptions:

Assumption 1 (Right-Key Equivalence). *For a linear approximation (α, β), the magnitude of the correlation, $|C_K(\alpha, \beta)|$, does not deviate significantly from its expected value over all keys, that is, $|C_K(\alpha, \beta)| = \mathrm{E}(|C_K(\alpha, \beta)|)$.*

Assumption 2 (Wrong-Key Randomisation). *For Algorithm 2, the correlation of a linear approximation (α, β) is 0 for all wrong guesses of (k_f, k_b).*

The assumption of right-key equivalence implies that the linear approximation has one dominant trail, say T, and consequently the distribution of the correlation over the key space only takes on two values, namely $\pm|C_K(T)|$. Thus, the natural partitioning of the key space for M approximations is the partitioning induced by the sign of the correlations, i.e. the vector $((-1)^{\langle \bar{T}_1, \bar{K} \rangle}, \dots, (-1)^{\langle \bar{T}_M, \bar{K} \rangle})$. In practice however, the correlations are calculated from the counters $L_{i,j}$. The joint distribution of the resulting measured correlations, for some specific key class, is given in [3] as an M-variate normal distribution, described in the following model.

Model 1 (Right-Key Partitioning for Multiple Approximations [3]). *Consider a set of linear approximations $(\alpha_1, \beta_1), \dots, (\alpha_M, \beta_M)$ of r rounds of DES. Then, the key space can be partitioned into at most 2^M key classes based on the signs of the correlations. The undersampled distribution of the linear correlation vector, using N texts and restricted to the i'th key class, denoted by $C_i^N(\boldsymbol{\alpha}, \boldsymbol{\beta})$, is an M-variate normal distribution*

$$C_i^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{N}_M(\boldsymbol{\mu}_i, 1/N \cdot \mathbf{I}).$$

The mean vector of the i'th key class is given by $\boldsymbol{\mu}_i[j] = s_{i,j}|C_K(T_i)|$, where $s_{i,j} \in \{-1, 1\}$ describes the sign combination of the i'th key class, $j = 1, \dots, M$.

Based on this model, a Bayesian classifier is constructed. We refer to Sect. 5 for the details. While the approach presented by Biryukov et al. seems promising, it has yet to result in an improved attack on DES. To the best of our knowledge, no other variants of linear cryptanalysis which uses multiple approximations have been able to outperform Matsui's original attack. Moreover, while updated versions of Assumptions 1 and 2 have been applied to other ciphers, no such work exists for DES. In the following, we address these concerns. We consider the right-key distribution in Sect. 3, and the wrong-key distribution in Sect. 4. Using the results obtained in these sections, we develop an improved linear attack on DES in Sects. 5 and 6.

3 Right-Key Correlation for DES: Key Inequivalence

In this section, we consider the correlation distribution of DES approximations over the key space. In Sect. 3.1, we consider current models for this distribution, as well as the undersampled distribution. In Sect. 3.2, we enumerate a large number of trails for DES, and show that, contrary to Assumption 1, the absolute value of the correlation does vary significantly as the key changes. In fact, the correlation distribution has a complicated structure. In Sect. 3.3, we develop a new model for this correlation based on Gaussian mixtures, which is able to accurately describe this structure. Moreover, we extend the model to describe the full undersampled correlation distribution over keys for multiple approximations.

3.1 The Correlation Distribution of a Single Approximation

As mentioned, most linear cryptanalysis of DES assumes that each linear approximation has one dominant trail, determining the magnitude of the absolute correlation. This idea is effectively expressed by Assumption 1. Consider, for example, one of the approximations used by Matsui, γ_1. This approximation has a primary trail T_A over 14 rounds of DES with correlation contribution $C_K(T_A) = \pm 2^{-19.75}$. In [24], Nyberg first introduced the concept of a linear hull, i.e. the collection of all trails of a linear approximation, and showed that Assumption 1 is not true in general. For γ_1, the trail with second largest correlation contribution, T', has contribution $C_K(T') = \pm 2^{-25.86}$. While the contribution from this trail is not large enough to change the sign of the linear correlation $C_K(\gamma_1)$, or increase/decrease the magnitude of the correlation much, it does not match the model given in Assumption 1. When including the second trail, the correlation distribution does not take on only two distinct values, but four.

Signal/noise decomposition. In order to refine Assumption 1, Bogdanov and Tischhauser considered a *signal/noise decomposition* of the hull in [7]. Consider a situation in which d dominant trails of an approximation (α, β) are known. We call this collection of trails the *signal*, and define the *signal correlation* as the sum of their correlation contributions

$$C'_K(\alpha, \beta) = \sum_{i=1}^{d} (-1)^{s_{T_i} \oplus \langle \bar{T}_i, \bar{K} \rangle} |C_K(T_i)|.$$

The remaining part of the hull is unknown, and is modelled as *noise*, with the distribution $\mathcal{N}(0, 2^{-n})$. Then, the refined right-key equivalence assumption of [7] states that the correlation of (α, β) is given by the sum of the signal correlation and the noise:

$$C_K(\alpha, \beta) = C'_K(\alpha, \beta) + \mathcal{N}(0, 2^{-n}).$$

Since the approximations we will typically consider in the context of DES have quite high correlation, the addition of the noise term will not make a significant difference. However, we include it for completeness.

Undersampling. The cryptanalyst is most often not interested in having to obtain the full codebook to exactly measure the linear correlation $C_K(\alpha, \beta)$. Therefore, the undersampled distribution is of great interest. Let

$$C_K^N(\alpha, \beta) = \frac{2}{N} \#\{x_i, i = 1, \ldots, N | \langle \alpha, x_i \rangle \oplus \langle \beta, E_K(x_i) \rangle = 0\} - 1$$

be the empirical value of $C_K(\alpha, \beta)$ measured using N text pairs. Here, we assume that x_i is drawn uniformly at random with replacement from $\mathbb{F}_2^n$. Matsui first considered the distribution of $C_K^N(\alpha, \beta)$ over the key space under Assumption 1. In this case, Matsui used the Gaussian distribution $C_K^N(\alpha, \beta) \sim \mathcal{N}(C_K(\alpha, \beta), 1/N)$. While no proof is given in [21], one can show this result via a Gaussian approximation to the binomial distribution, assuming that $|C_K(\alpha, \beta)|$ is small.

3.2 Exploring the Signal Distribution of DES

On the basis of the signal/noise model, we now turn our attention to the signal distribution of DES approximations. By computing the signal correlation C'_K for a large number of trails, we are able to get a good idea of the actual distribution of the correlation C_K. We first describe how the signal trails were enumerated.

Our trail enumeration algorithm. We implemented a bounded breadth-first search in order to enumerate trails of DES approximations over 14 rounds. The algorithm consists of two search phases and a matching phase. Consider an approximation (α, β). The first search phase searches for trails in the forward direction, from round one to round seven. The search starts with α as an input mask to the first round, and then finds t and v such that the single round trails (α, t, v) has non-zero correlation. This process is then repeated for each trail with v as input mask to the second round, etc. The second search phase is similar, but searches backwards from β.

The searches are bounded in two ways. First, we only consider trails that activate at most three S-Boxes in each round. Second, we limit the number of trails which are kept in each round. This is done in such a way that only the trails with largest absolute correlation contribution are kept. This ensures a locally optimal choice, although no such guarantee can be made globally. The number of trails kept is determined by the *branching factor* B, such that in the i'th round of the search, $i \cdot B$ trails are kept.

After the two search phases, each trail found in the forward direction is matched to any trail in the backwards direction which shares the same mask in the middle. In this way, we obtain a number of trails of (α, β) over 14 rounds. Globally optimal trails will have a good chance of being enumerated if the branching factor B is chosen sufficiently large. In the following, we set $B = 1$ million, which means that we can find at most 7 million trails in each search direction. Note that the number of trails eventually discovered by the algorithm highly depends on the number of rounds and the approximation under consideration. We performed the enumeration for the eight approximations given in Table 2 using 20 Intel Xeon Processor E5-2680 cores. The enumeration took about 8 CPU hours.

Computing the Signal Distribution. Using the algorithm described above, we enumerated 1126 trails of the approximation γ_1 over 14 rounds, and calculated the signal correlation

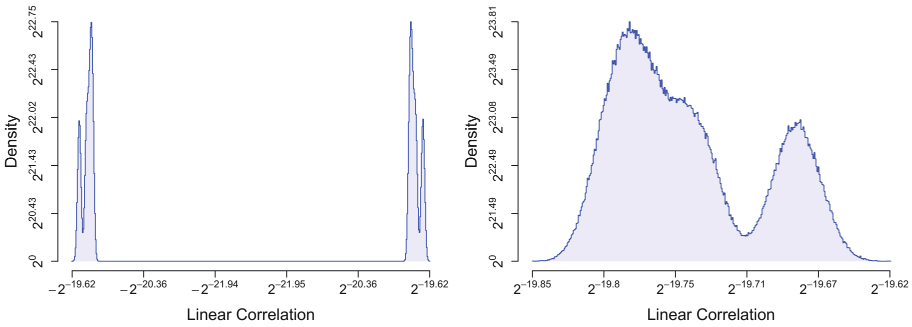

Fig. 3. The signal distribution of linear correlation for the approximation γ_1 over 14 rounds of DES. The signal correlation was calculated using 1126 trails and 1 million randomly drawn keys. The trails had an absolute correlation contribution between $2^{-43.61}$ and $2^{-19.75}$. The left plot shows the two main modes, symmetric around zero. The right plot shows only the positive half of the distribution.

$$C'_K(\gamma_1) = \sum_{i=1}^{1126} (-1)^{s_{T_i} \oplus \langle \bar{T}_i, \bar{K} \rangle} |C_K(T_i)|,$$

for 1 million randomly drawn keys. The trails we found have an absolute correlation contribution between $2^{-43.61}$ and $2^{-19.75}$, and include the dominant trail used by Matsui in [22]. The resulting distribution can be seen in Fig. 3.

The left part of the figure shows the full distribution over the key space. At this scale, the distribution resembles the one described in Sect. 2; there are two very prominent modes symmetric around zero, with peaks around $\pm 2^{-19.75}$, corresponding to the correlation contribution of the dominant trail. However, the right part of the plot, showing the positive half of the distribution, largely contradicts Assumption 1 of key equivalence. While the mean of the distribution is $2^{-19.75}$, it also has a non-negligible standard deviation of $2^{-24.71}$. Moreover, the distribution is not Gaussian. The correlations cluster around three values, namely $2^{-19.79}$, $2^{-19.75}$, and $2^{-19.68}$. Interestingly, the probability density is larger around the cluster with the lowest correlation value.

Under the signal/noise model, adding the noise distribution $\mathcal{N}(0, 2^{-n})$ gives us a good estimate of the actual distribution of the correlation $C_K(\gamma_1)$. However, due to the large variance of the signal distribution, the effect of the noise term is negligible in this case. Thus, the distribution in Fig. 3 should be quite close to the actual distribution. This poses a fundamental problem, as none of the analysis of linear cryptanalysis applied to DES accounts for this type of distribution. Indeed, it is not clear how the distribution of the undersampled correlation, C_K^N, looks, which is essential to know when determining the complexity of linear attacks.

3.3 A New Mixture Model for Single and Multiple Approximations

To relieve the problems discussed in Sect. 3.2, we now propose a model for the correlation distribution based on *Gaussian mixtures*. Consider a distribution in

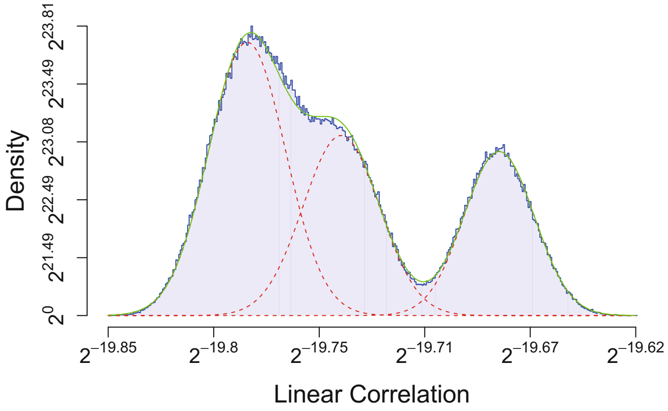

Fig. 4. A Gaussian mixture fitted to the correlation distribution of the linear approximation γ_1 over 14 rounds of DES. The individual components are shown in red, the mixture density is shown in green, and the measured distribution is shown in blue. Under this model, only 30% of the distribution is attributed to the Gaussian component associated with the dominant trail. (Color figure online)

which each sample is drawn from one of ℓ Gaussian distributions. Each Gaussian is called a *component*. The probability of the sample being drawn from the i'th component is λ_i, usually called the *weights*, with $\sum \lambda_i = 1$. The probability density function (PDF) of such a distribution is given by

$$f(x) = \sum_{i=1}^{\ell} \lambda_i \phi(x; \mu_i, \sigma_i^2),$$

where $\phi(x; \mu_i, \sigma_i^2)$ is the PDF of the i'th Gaussian distribution, having mean μ_i and variance σ_i^2 [20]. We will denote the distribution itself by $\mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, \ell)$. We then propose the following model.

Model 2 (Right-Key Inequivalence for One Approximation). *Consider a linear approximation (α, β) of r rounds of DES. The distribution of the linear correlation $C_K(\alpha, \beta)$ over the key space is approximately given by a Gaussian mixture for some weights λ_i and components $\mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, \ldots, \ell$. That is,*

$$C_K(\alpha, \beta) \sim \mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, \ell).$$

We note that the signal/noise decomposition easily applies to this model. If we determine that the signal correlation follows a Gaussian mixture, i.e. $C'_K(\alpha, \beta) \sim \mathcal{M}(\lambda'_i, \mu'_i, \sigma_i^{2\prime}, \ell')$ for some appropriate parameters, then we can approximate the actual correlation distribution by adding the noise distribution:

$$C_K(\alpha, \beta) \sim \mathcal{M}(\lambda'_i, \mu'_i, \sigma_i^{2\prime}, \ell') + \mathcal{N}(0, 2^{-n}).$$

We apply Model 2 to the distribution obtained in Sect. 3.2. The result of fitting a Gaussian mixture model with three components to the positive part of the signal distribution is shown in Fig. 4. We first note that the mixture model fits the measured signal distribution quite well. The parameters are

$$\begin{aligned} \lambda_1 &= 0.45, \quad \mu_1 = 2^{-19.79}, \quad \sigma_1^2 = 2^{-52.40}, \\ \lambda_2 &= 0.30, \quad \mu_2 = 2^{-19.75}, \quad \sigma_2^2 = 2^{-52.37}, \\ \lambda_3 &= 0.25, \quad \mu_3 = 2^{-19.68}, \quad \sigma_3^2 = 2^{-52.68}. \end{aligned}$$

The second mixture component has mean equal to the correlation contribution of the dominant trail, but this component only contributes to 30% of the full distribution. In fact, the main part of the contribution, 45%, can be attributed to the first component, which has a slightly lower mean. This demonstrates that considering only the contribution of the dominant trail can be misleading, even when the remaining trails have a far lower correlation contribution. In general, one should consider as large a part of the hull as possible. Nevertheless, for attacks with relatively low data complexity, the actual distribution can easily be hidden, as we shall see next.

The undersampled mixture. In Sect. 3.2, we recalled that under the assumption of a dominant trail, the distribution of the undersampled correlation C_K^N is given by the Gaussian $\mathcal{N}(C_K, 1/N)$. We state the following equivalent result in the setting of Model 2 and give an outline of the proof.

Theorem 1 (Undersampled distribution). *Assuming Model 2, the undersampled correlation distribution of an approximation (α, β) obtained using N random text pairs is given by*

$$C_K^N(\alpha, \beta) \sim \mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, \ell) + \mathcal{N}(0, 1/N).$$

Proof. For any fixed key k, C_k^N is distributed as $\mathrm{Bin}(N, C_k)$ over the random text sample, which can be approximated by $\mathcal{N}(C_k, 1/N)$ if C_k is small. That is, $C_K^N \mid K = k \sim \mathcal{N}(C_k, 1/N)$. The PDF of the compound distribution C_K^N, i.e. without the conditioning on K, is given by

$$p_{C_K^N}(y) = \int \phi(y; x, 1/N) \cdot \sum_{i=1}^{\ell} \lambda_i \phi(x; \mu_i, \sigma_i^2) dx,$$

which can be shown to be equal to

$$p_{C_K^N}(y) = \sum_{i=1}^{\ell} \lambda_i \phi(y; \mu_i, \sigma_i^2 + 1/N).$$

This is a Gaussian mixture where each component can be written as $\mathcal{N}(\mu_i, \sigma_i^2) + \mathcal{N}(0, 1/N)$. But since we add the second distribution with probability one, the same distribution can be obtained by first drawing from the original mixture, and then adding the distribution $\mathcal{N}(0, 1/N)$, finishing the proof. □

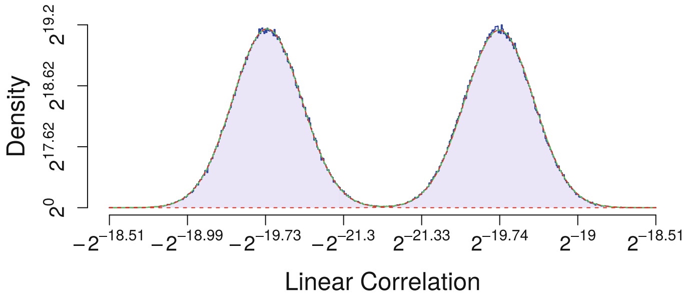

Fig. 5. The distribution of the undersampled linear correlation of γ_1, $C'_K + \mathcal{N}(0, 2^{-n}) + \mathcal{N}(0, 1/N)$, over 14 rounds of DES, with $N = 2^{43}$. C'_K was measured using 1126 trails over 1 million randomly drawn keys. A Gaussian mixture with two components have been fitted to the distribution. The components are shown in red, while the full distribution is shown in green. (Color figure online)

If the number of texts N is relatively large, the model can be somewhat simplified. If we wanted to apply Model 2 and Theorem 1 directly to the case of γ_1, we would model the measured correlation as

$$C_K^N(\gamma_1) = \mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, 6) + \mathcal{N}(0, 2^{-n}) + \mathcal{N}(0, 1/N), \tag{4}$$

using six components for the Gaussian mixture. However, the details of the mixture are easily lost at high levels of undersampling, as can be seen in Fig. 5. Here, we have shown the distribution

$$C'_K(\gamma_1) + \mathcal{N}(0, 2^{-n}) + \mathcal{N}(0, 1/N),$$

where $N = 2^{43}$. The resulting distribution can be described as a Gaussian mixture with two components, instead of six. Each component has variance roughly equal to $1/N$, and the means are $\pm 2^{-19.75}$, i.e. the correlation contribution of the dominant trail. This agrees with the models used by e.g. Matsui and Biryukov, et al., but we stress that this is only true when N is relatively small compared to the linear correlation. In particular, for ciphers with strong dominant trails, $1/N$ needs to be larger than the variance of the positive/negative part of the distributions. For values of N close to the full codebook, this is not true (unless the approximation is extremely weak), and the distribution of C_K cannot be ignored. However, this simplification will help greatly when we consider the joint distribution of multiple approximations in the next subsection.

The Gaussian mixture of multiple approximations. Model 2 and the results of Sect. 3.3 can be generalised to consider the case of multiple linear approximations. Let $C_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$ denote the vector of correlations of M linear approximations, $(C_K(\alpha_1, \beta_1), \ldots, C_K(\alpha_M, \beta_M))^\top$. In the following, we will restrict ourselves to the case where the signal distributions, $C'_K(\alpha_i, \beta_i)$, each have two distinct modes: one positive and one negative. This allows us to split the joint signal distribution, $C'_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$, into at most 2^M components determined by the signs of $C'_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$. In the case of relatively low values of N, we propose the following model.

Model 3 (Right-Key Mixture for Multiple Approximations). *Consider a set of linear approximations $(\alpha_1, \beta_1), \ldots, (\alpha_M, \beta_M)$ of r rounds of DES. The undersampled distribution of the linear correlation vector over the key space, $C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta})$, is approximately given by an M-variate Gaussian mixture, namely*

$$C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{M}_M(1/\ell, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + 1/N \cdot \mathbf{I}, \ell),$$

where $\ell \leq 2^M$. Moreover, the parameters of the mixture components are given by

$$\begin{aligned} \boldsymbol{\mu}_i &= \mathrm{E}(C_K(\boldsymbol{\alpha}, \boldsymbol{\beta}) | s_{i,j} \cdot C_K(\alpha_i, \beta_i) > 0, j = 1, \ldots, M), \\ \boldsymbol{\Sigma}_i &= \mathrm{Cov}(C_K(\boldsymbol{\alpha}, \boldsymbol{\beta}) | s_{i,j} \cdot C_K(\alpha_i, \beta_i) > 0, j = 1, \ldots, M), \end{aligned}$$

where $s_{i,j} \in \{-1, 1\}$ describes the sign combination of the i'th component.

As for the case of a single approximation, the signal/noise decomposition applies to this model, resulting in an undersampled distribution of the form

$$C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{M}_M(1/\ell, \boldsymbol{\mu}'_i, \boldsymbol{\Sigma}'_i + (2^{-n} + 1/N)\mathbf{I}, \ell).$$

The signal parameters, $\boldsymbol{\mu}'_i$ and $\boldsymbol{\Sigma}'_i$, can be estimated by enumerating an appropriate number of trails and then calculating $C'_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$ for a large number of keys.

This model bears some resemblance to the one given by Biryukov et al. in [3]. While both models use the signs of the correlation vector to split the distribution into several Gaussians, our model captures the entire key space in one distribution, whereas the model in [3] partitions the key space into at most 2^M parts which are considered separately. Additionally, we do not make any assumption about the independence of the linear approximations. As such, $\boldsymbol{\Sigma}_i$ need not be diagonal matrices, and not all 2^M sign combinations need to be present. While the possibility of $\ell < 2^M$ is briefly mentioned in [3], all experiments were done such that $\ell = 2^M$. As we shall see in Sect. 5, the case of $\ell < 2^M$ allows for stronger attacks. Moreover, an improved attack on full DES was not presented in [3]. We apply our model to obtain a key-recovery attack on full DES in Sect. 6. First, however, we turn our attention to the wrong-key distribution.

4 Wrong-Key Correlation for DES: Non-random Behaviour

In this section, we consider the correlation distribution of DES approximations in the case of a wrong key guess in Algorithm 2. This distribution is essential, as the effectiveness of the algorithm is determined by how different the right-key and wrong-key distributions are. In Sect. 4.1, we consider the current models for the wrong-key distribution. In Sect. 4.2, we develop a new model for the wrong-key distribution of DES, and show that the distribution obtained under this model deviates significantly from that considered in Sect. 4.1. Nevertheless, as for the right-key in Sect. 3, we show that the deviation has little impact when the number of texts used in the attack is relatively small.

4.1 The Current Ideal Wrong-Key Distribution

The assumption of wrong-key randomisation, Assumption 2, used by Matsui in [22] and by Biryukov et al. in [3], predicts that a wrong guess of the outer round keys in Algorithm 2 should result in an approximation with correlation zero. This is motivated by the idea that if we encrypt/decrypt using the wrong key, we are doing something equivalent to encrypting two extra rounds. This should result in a linear correlation much closer to zero, as we are essentially considering the correlation over $r+4$ rounds instead of r rounds. However, as shown by Daemen and Rijmen in [13], even a linear approximation of an ideal permutation will approximately have the correlation distribution

$$C_K(\alpha, \beta) \sim \mathcal{N}(0, 2^{-n}),$$

where n is the blocksize. Since we intuitively cannot do "worse" than an ideal cipher, the correlation of a wrong guess should follow this distribution. This consideration led Bogdanov and Tischhauser to present an updated wrong-key randomisation hypothesis in [7], in which the wrong key correlation follows this ideal Gaussian distribution. However, if we consider the case of DES where, even over 14 rounds, strong linear approximations exist, the wrong-key correlation might not be close to the ideal distribution. We consider this problem next.

4.2 A New Non-random Wrong-Key Distribution

Consider the scenario in which an attacker obtains a plaintext-ciphertext pair computed over r rounds of a cipher, and attempts to encrypt the plaintext one round, and decrypt the ciphertext one round, in order to calculate the correlation of an approximation over $r-2$ rounds. If the attacker uses the wrong round keys for the encryption/decryption, she essentially obtains a plaintext/ciphertext pair of some related cipher with $r+2$ rounds. Motivated by this, we propose the following wrong-key model for linear cryptanalysis on DES.

Model 4 (Non-random Wrong-Key Distribution). *Consider an Algorithm 2 style attack on r rounds of DES using a linear approximation (α, β) over $r-2$ rounds. Let R_K be the keyed round function of DES, and let $E_K^\star$ denote the r-round encryption function. For a wrong guess of the outer round keys, the correlation will be distributed as for the cipher*

$$E'_K(x) = R_{K_a}^{-1}(E_K^\star(R_{K_b}^{-1}(x))), \tag{5}$$

where K_a and K_b are chosen uniformly at random.

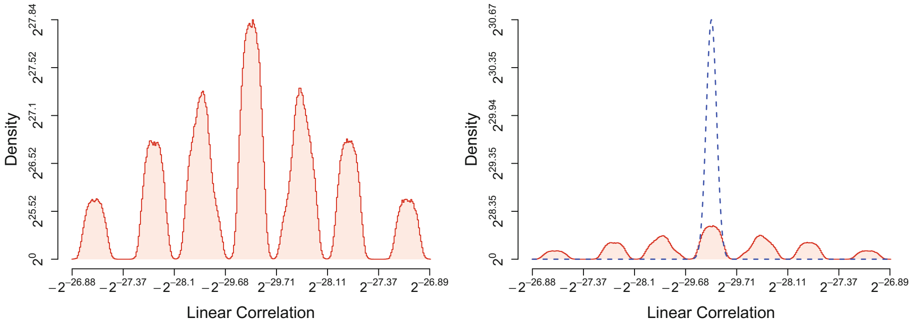

Fig. 6. The distribution of linear correlation for the approximation γ_1 over 18 rounds of DES with randomly chosen outer round keys. The correlation was calculated using 954 trails and 1 million randomly drawn keys. The distribution is close to zero, but the variance is $2^{-56.08}$. To the right, the distribution is compared to that of an ideal permutation, i.e. the Gaussian $\mathcal{N}(0, 2^{-64})$.

For DES, where encryption and decryption are similar, this can reasonably be simplified to $E'_K(x) = E_K^{r+2}$, where the outer round keys are randomly chosen.

In light of this, we considered the approximation γ_1 over 18 rounds of DES, with randomly chosen outer round keys. Using the algorithm described in Sect. 3.2, with $B = 1$ million, we enumerated 954 trails of this approximation. Using 20 Intel Xeon Processor E5-2680 cores, the enumeration took about 15 CPU hours. We then calculated the resulting signal correlation for 1 million keys. The trails had an absolute correlation contribution between $2^{-45.84}$ and $2^{-28.75}$. The distribution is shown in Fig. 6. We note that the result is similar for the other approximations given in Table 2.

As was the case for the right-key distribution, this wrong-key distribution appears to be approximately a Gaussian mixture. More importantly, while the distribution is symmetric around zero, the variance is much larger than that of an ideal permutation: $2^{-56.08}$ compared to 2^{-64}. This shows that, while the added four rounds make the correlation weaker, the assumption of a resulting ideal distribution is optimistic. For attacks that use a data complexity close to the full codebook, this assumption could result in a overestimate of success probability or an underestimate of attack complexity. Moreover, if the cryptanalyst only appends/prepends one round to the approximation, this effect could be significant.

The undersampled distribution. While the distribution in Fig. 6 is far from ideal, the actual distribution of the correlation matters little if the level of undersampling is significant. If we apply signal/noise decomposition and Theorem 1 to our estimate of the wrong-key distribution, with the number of texts $N = 2^{43}$, we obtain the result shown in Fig. 7. We see here that it is sufficient to use a single Gaussian distribution to approximate the undersampled wrong-key correlation distribution. If this distribution is similar for other approximations, it will

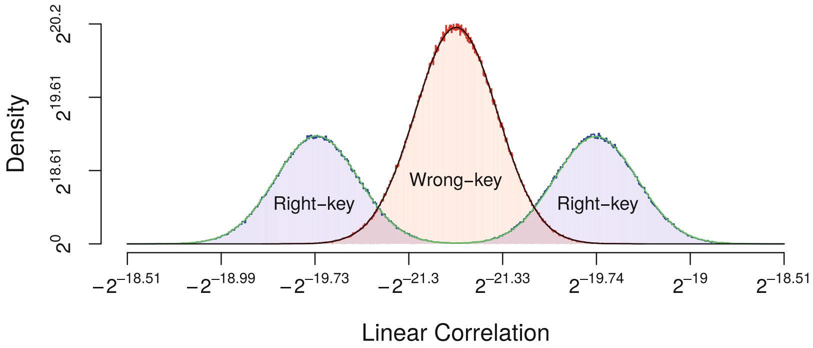

Fig. 7. Undersampled right-key (blue) and wrong-key (red) distributions for the approximation γ_1 with $N = 2^{43}$. The signal distributions were measured using 1 million randomly drawn keys. A Gaussian mixture has been fitted to the right-key distribution (green), while a single Gaussian distribution was fitted to the wrong-key distribution (black). (Color figure online)

be sufficient to model the joint wrong-key correlation distribution of M approximations as an M-variate Gaussian distribution. Thus, if $\mathbf{\Sigma}_W$ is the covariance matrix of the signal correlation of the M approximations over E'_K, then the undersampled wrong-key distribution will approximately be given by

$$C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_W + (2^{-n} + 1/N)\mathbf{I}),$$

if $1/N$ is sufficiently large.

Using Model 3 for the right-key and Model 4 for the wrong-key distribution, we develop a classifier that uses both these distributions in the following section.

5 Classifying Keys Using Asymmetric Distributions

In Sect. 3, we developed a model for the linear correlation distribution of a correct key-guess in Algorithm 2, namely a multivariate Gaussian mixture model. In Sect. 4, we similarly developed a simple multivariate Gaussian model for the linear correlation distribution of a wrong key-guess. Using these two distributions, we now develop a classifier based on the likelihood-ratio, which can be used in Algorithm 2 to decide between potential right and wrong key guesses. We first present the classifier given in [3] in Sect. 5.1. We then introduce our new classifier in Sect. 5.2, and compare the performance of the two in Sect. 5.3.

In the following, we will consider the two sets of four linear approximations over 14 rounds of DES given in Table 2. While it is difficult to visualise the joint distribution of more than three approximations, Fig. 8 shows the pairwise joint distributions of the approximations γ_1, γ_2, γ_3, and γ_4, as well as the marginal distributions, for $N = 2^{43}$. Note that the joint distributions of γ_1 and γ_3, as well as that of γ_2 and γ_4, only have two components. We will explore this phenomenon in Sect. 5.4, and show that such distributions can improve our classifier.

Table 2. The top table specifies two sets of four linear approximations over 14 rounds of DES, and gives the correlation contribution of their dominant trail, as well as the sign bit of that trail. The bottom table specifies the set of non-zero bits of the associated dominant key trails, where t_i^j is the j'th bit of t_i.

Linear approximation	Dominant key trail	$\lvert C_K(T_\cdot)\rvert$	$s_{T_\cdot}$
$\gamma_1 = ([7,18,24],[7,18,24,29,47])$	$\bar{T}_A$	$2^{-19.75}$	1
$\gamma_2 = ([7,18,24],[7,18,24,29,44,48])$	$\bar{T}_B$	$2^{-20.48}$	1
$\gamma_3 = ([7,18,24,29],[7,18,24,47])$	$\bar{T}_A$	$2^{-20.75}$	0
$\gamma_4 = ([7,18,24,29],[7,18,24,44,48])$	$\bar{T}_B$	$2^{-20.07}$	1
$\delta_1 = ([15,39,50,56],[39,50,56,61])$	$\bar{T}_C$	$2^{-20.75}$	0
$\delta_2 = ([12,16,39,50,56],[39,50,56,61])$	$\bar{T}_D$	$2^{-20.07}$	1
$\delta_3 = ([15,39,50,56,61],[39,50,56])$	$\bar{T}_C$	$2^{-19.75}$	1
$\delta_4 = ([12,16,39,50,56,61],[39,50,56])$	$\bar{T}_D$	$2^{-20.48}$	1

Key trail	Non-zero key mask bits	Key trail	Non-zero key mask bits
$\bar{T}_A$	$\{t_1^{22}, t_2^{44}, t_3^{22}, t_5^{22}, t_6^{44}, t_7^{22}, t_9^{22}, t_{10}^{44}, t_{11}^{22}, t_{13}^{22}\}$	$\bar{T}_B$	$\bar{T}_A \backslash t_{13}^{22} \cup \{t_{13}^{19}, t_{13}^{23}\}$
$\bar{T}_C$	$\{t_0^{22}, t_2^{22}, t_3^{44}, t_4^{22}, t_6^{22}, t_7^{44}, t_8^{22}, t_{10}^{22}, t_{11}^{44}, t_{12}^{22}\}$	$\bar{T}_D$	$\bar{T}_C \backslash t_0^{22} \cup \{t_0^{19}, t_0^{23}\}$

5.1 The Bayesian Classifier of Biryukov et al.

Consider an Algorithm 2 style attack using M linear approximations. Let $\mathcal{K}_R$ denote the space of correct guesses of the key-bits (k_f, k_b), and let $\mathcal{K}_W$ denote the space of wrong guesses. We have to classify each key-guess as either an incorrect guess or a potential correct guess, based on the measured linear correlation vector $\mathbf{x}$. Let $f_R(\mathbf{x}) = \Pr(\mathbf{x} \mid (k_f, k_b) \in \mathcal{K}_R)$ be the PDF of the right-key correlation distribution. We define the Bayesian classifier, BC, as the following decision rule

$$BC(\mathbf{x}) = \begin{cases} \text{If } B(\mathbf{x}) > \Gamma, \text{ decide that } (k_f, k_b) \in \mathcal{K}_R, \\ \text{otherwise, decide that } (k_f, k_b) \in \mathcal{K}_W, \end{cases}$$

where $B(\mathbf{x}) = f_R(\mathbf{x})$. Under Model 3, $B(\mathbf{x})$ is given as the Gaussian mixture

$$B(\mathbf{x}) = \sum_{i=1}^{\ell} \lambda_i \phi_M(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + (2^{-n} + 1/N)\mathbf{I}).$$

This exact classifier is not described in [3], but it is essentially identical to the one developed there. The difference is that in [3], each component of f_R is considered separately, and so ℓ scores are produced for each key guess. The classifier BC should be functionally equivalent to this approach, but this representation allows for easy comparison to the likelihood-ratio classifier we propose next.

5.2 Our Likelihood Classifier

We now propose a new classifier based in the likelihood-ratio. As opposed to the Bayesian classifier, the likelihood classifier directly takes the wrong-key distribution into account. To this end, let $f_W(\mathbf{x}) = \Pr(\mathbf{x} \mid (k_f, k_b) \in \mathcal{K}_R)$ be the PDF

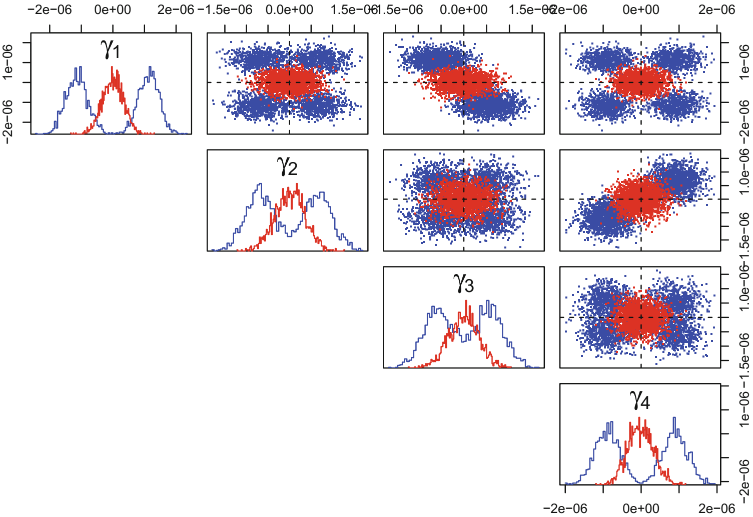

Fig. 8. Histograms and pairwise distributions of the undersampled correlations of approximations $\gamma_1, \dots, \gamma_4$ given in Table 2. The right-key distributions are shown in blue, the wrong-key distributions are shown in red. The number of texts is $N = 2^{43}$. Note that since γ_1 and γ_3 have the same dominant key trail, their joint distribution only has two components. Likewise for γ_2 and γ_4. (Color figure online)

of the wrong-key correlation distribution. Then the *likelihood-ratio* is defined as $\Lambda(\mathbf{x}) = f_R(\mathbf{x})/f_W(\mathbf{x})$. For the right-key and wrong-key distributions described in Sects. 3 and 4, this is equal to

$$\Lambda(\mathbf{x}) = \frac{\sum_{i=1}^{\ell} \lambda_i \phi_M(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + (2^{-n} + 1/N)\mathbf{I})}{\phi_M(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_W + (2^{-n} + 1/N)\mathbf{I})},$$

where $\mathbf{x}$ is an observed value of correlations for M approximations. A large value of $\Lambda(\mathbf{x})$ will then indicate a likely correct key guess, while a low value will indicate a wrong key guess. Thus, we define the *likelihood classifier LC* as the following decision rule

$$LC(\mathbf{x}) = \begin{cases} \text{If } \Lambda(\mathbf{x}) > \Gamma, \text{decide that } (k_f, k_b) \in \mathcal{K}_R, \\ \text{otherwise, decide that } (k_f, k_b) \in \mathcal{K}_W. \end{cases}$$

In light of this definition, two important concepts are the success probability and advantage of the classifier. Formally, we define the success probability and advantage, respectively, as

$$P_S = 1 - \Pr(\Lambda(\mathbf{x}) < \Gamma \mid (k_f, k_b) \in \mathcal{K}_R), \tag{6}$$

$$a = -\log_2(\Pr(\Lambda(\mathbf{x}) \geq \Gamma \mid (k_f, k_b) \in \mathcal{K}_W)), \tag{7}$$

in accordance with the usual definition [25]. We usually choose Γ such that we achieve a certain success probability. Under our proposed model, the involved probabilities cannot be explicitly stated. Thus, we must rely on simulations to calculate these values. Since simulating values from a Gaussian distribution is easy, this is not a problem. Using this approach, we now compare the performance of the likelihood classifier and the Bayesian classifier.

5.3 Decision Boundaries

The likelihood classifier LC divides the M-dimensional cube $[-1,1]^M$ into two regions separated by the *decision boundary*, namely where $\Lambda(\mathbf{x}) = \Gamma$. On one side of the decision boundary, observations are classified as belonging to the right-key distribution, while observations from the other side are classified as belonging to the wrong-key distribution. By visualising this decision boundary, we can get a better understanding of the classifier.

In the following, we consider the eight approximations given in Table 2, over 14 rounds of DES. We enumerated between 1100 and 1400 trails for each approximation and calculated the signal correlations for 1 million random keys, in order to estimate $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. The same was done over E'_K, where between 950 and 1100 trails were enumerated, in order to estimate $\boldsymbol{\Sigma}_W$. For each data point, we added noise drawn from $\mathcal{N}_M(\mathbf{0}, (2^{-n} + 1/N)\mathbf{I})$, according to the signal/noise decomposition and Theorem 1. This allows us to simulate $\Lambda(\mathbf{x})$ and $B(\mathbf{x})$ for varying values of N and calculate the resulting decision boundary and advantage.

Consider the pair of approximations γ_1 and δ_1 and let $N = 2^{43}$. We simulate $\Lambda(\mathbf{x})$ and $B(\mathbf{x})$ for each data point as described above, and then fix a threshold value for each classifier such that $P_S = 0.90$, cf. Eq. (6). The resulting decision boundaries, as well as the related probability distributions, are shown in Fig. 9. In this case, the likelihood classifier obtains an advantage of 5.5 bits, while the Bayesian classifier only has an advantage of 3.1 bits. By considering the decision boundary, it is clear why this is the case. Since the Bayesian classifier only uses information about the right-key distribution, it simply creates a decision boundary around each component of the mixture which is large enough to obtain the desired success probability. In view of the information that is available to the classifier, this makes sense, since observations close to the mean of component have a larger chance of being a correct key guess. Because of this, the parts of the right-key distribution which is farthest away from the wrong-key distribution is also discarded as unlikely candidates. This in turn requires the decision boundary to be wider than actually needed, and the advantage is therefore quite low due to an increased number of false positives.

The likelihood classifier on the other hand does use information about the wrong-key distribution. The decision boundary is created such that there is a good boundary between each component and the wrong-key distribution. Any observation that is sufficiently far away from the wrong-key distribution is deemed a likely correct key guess, no matter how extreme the observation is in the right-key distribution. Thus, extreme points in the right-key distribution are not “wasted”, allowing for a tight decision boundary around the wrong-key distribution, yielding a larger advantage.

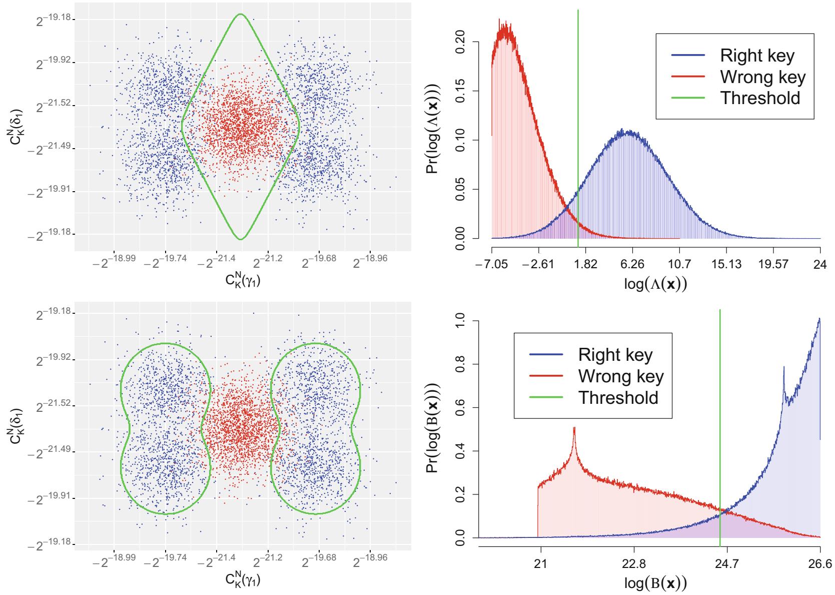

Fig. 9. Left: The joint distribution of $C_K^N(\gamma_1)$ and $C_K^N(\delta_1)$, with $N = 2^{43}$, are shown for both a right key guess (blue) and a wrong key guess (red). The decision boundaries for a success probability of 90% are drawn for the likelihood-ratio classifier (top) and the Bayesian classifier (bottom). Right: The corresponding probability distributions of $\Lambda(\mathbf{x})$ (top) and $B(\mathbf{x})$ (bottom) as well as the threshold value. The likelihood ratio classifier obtains an advantage of 5.5 bits, while the Bayesian classifier obtains an advantage of 3.1 bits. (Color figure online)

For the approximations used here, all sign combinations of the correlation vector are possible. In terms of the mixture model, the number of components is $\ell = 2^M$. We now turn our attention to the case where $\ell < 2^M$.

5.4 Observations on the Asymmetric Distribution

As shown in Sect. 3.2, the sign of the signal correlation $C'_K(\gamma_1)$ for a given key is determined by the parity $\langle \bar{T}_A, \bar{K} \rangle$, where $\bar{T}_A$ is the dominant key trail. Consider the two approximations γ_1 and γ_3 given in Table 2. Both approximations have the same dominant key trail, and since their sign bits s_T are different, the sign of their correlation will therefore always be opposite. In the terminology of Sect. 3.3, the number of components ℓ of the Gaussian mixture is strictly less than 2^M. We will call such a distribution *asymmetric*. On the other hand, the two approximations γ_1 and δ_1 have different dominant key-trails, and therefore all four sign combinations of their correlations are possible. In this case, $\ell = 2^M$, and we call such a distribution *symmetric*.

For γ_1 and δ_1, the decision boundary for the likelihood classifier was shown in Fig. 9. For γ_1 and γ_3, the decision boundary is shown in Fig. 10. Here, the “missing” components in the first and third quadrant are clearly shown, while

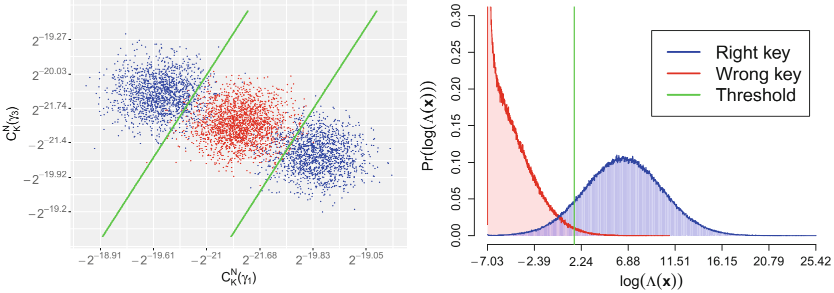

Fig. 10. Left: The joint distribution of $C_K^N(\gamma_1)$ and $C_K^N(\gamma_3)$, with $N = 2^{43}$, are shown for a right key guess (blue) and a wrong key guess (red). The decision boundaries for a success probability of 90% are drawn for the likelihood-ratio classifier. Right: The probability distributions of $\Lambda(\mathbf{x})$ as well as the threshold value. The classifier obtains an advantage of 6.2 bits. (Color figure online)

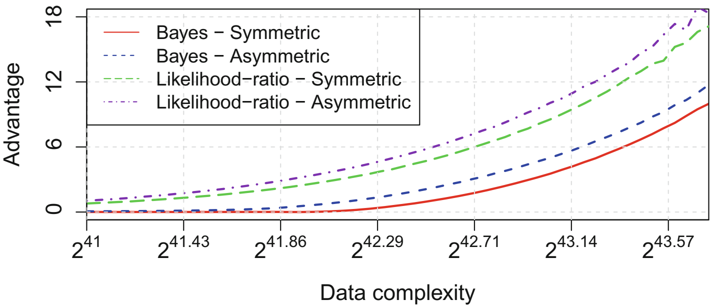

Fig. 11. A comparison of the advantage obtained by using the Bayesian classifier and the likelihood ratio classifier on both symmetric and asymmetric correlation distributions. The symmetric distribution uses the set of approximations $\{\gamma_1, \gamma_2, \delta_1, \delta_2\}$ while the asymmetric distribution uses the set $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$.

the wrong-key distribution is still symmetric around zero. We note that, all else being equal, the classifier on the asymmetric distribution achieves an increased advantage of 0.7 bits. Moreover, the comparison here is fair, since the strength of δ_1 is the same as that of γ_3. The reason for this increase is apparent when we compare the two decision boundaries. For the asymmetric distribution, the decision boundary is such that even extreme points in the wrong-key distribution towards the first and third quadrant are easily classified as wrong key guesses. This decreases the number of false positives, increasing the advantage.

This improvement in the classifier for asymmetric distributions generally extends to higher dimensions, where the effect can be even more pronounced. Indeed, for larger M, ℓ can be much smaller than 2^M. In the example above, we had $\ell = 2$ while $2^M = 4$. Consider now the set of approximations $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$. A shown in Table 2, these approximations only have two distinct dominant key trails, implying that the set has an asymmetric distribution with $\ell = 4 < 2^M = 16$. Figure 11 compares the advantage of this set of approximations to the set

$\{\gamma_1, \gamma_2, \delta_1, \delta_2\}$, which has a symmetric distribution, i.e. $\ell = 2^M = 16$. In general, we observe that the classifiers are stronger for the asymmetric distribution, with an increase in advantage of 1.4 bits for $N = 2^{43}$. Additionally, the better performance of the likelihood classifier is quite clear, consistently obtaining a larger advantage over the Bayesian classifier. For $N = 2^{43}$, the likelihood classifier has an advantage 4.9 bits higher than the Bayesian classifier on both the symmetric and asymmetric distribution. Due to these observations, we propose the term *symmetry factor* for these types of distributions, defined as $\ell/2^M$. A distribution with symmetry factor one is a symmetric distribution, while a symmetry factor less than one indicates an asymmetric distribution. We conjecture that, all else being equal, a lower symmetry factor will result in a stronger classifier.

6 Improved Attack on DES

Using the results from the previous sections, we now mount a key-recovery attack on DES using eight linear approximations. We will use two sets of four linear approximations, $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ and $\{\delta_1, \delta_2, \delta_3, \delta_4\}$ over 14 rounds, as given in Table 2. The attack is mostly identical to Matsui's Algorithm 2. As such, we obtain N plaintext-ciphertext pairs over 16 rounds, guess the key-bits required to partially encrypt/decrypt the texts and compute the linear correlations, and then use the likelihood classifier to categorise each guess as a likely wrong or right key guess. For each guess, we further gain some parity bits of the key based on the signs of the correlations.

6.1 Attack Description

Table 3 shows the key- and text-bits relevant to the attack. For both sets of approximations, we need to know 29 bits of the plaintext/ciphertext, designated $t_{f,\cdot}/t_{b,\cdot}$, and we will guess 24 bits of the first/last round key, designated $k_{f,\cdot}/k_{b,\cdot}$. Moreover, the signs of $C_K^N(\gamma_1)$, $C_K^N(\gamma_4)$, $C_K^N(\delta_3)$, and $C_K^N(\delta_2)$, will allow us to deduce the parity bits p_A, p_B, p_C, and p_D. Thus, the attacker will learn a total of 52 bits of the master key, and will have to guess the remaining 4 bits. In the following, we assume that the distribution parameters $\boldsymbol{\mu}_{i,\cdot}$, $\boldsymbol{\Sigma}_{i,\cdot}$, and $\boldsymbol{\Sigma}_{W,\cdot}$ have been determined before the attack, as described in Sect. 3.3. Moreover, we assume that $\lambda_i = 1/\ell$ for all i. The attack is then given as follows:

- **Distillation**
  1. Obtain N plaintext-ciphertext pairs.
  2. Create two vectors $\mathbf{t}_\gamma$ and $\mathbf{t}_\delta$ of size 2^{29} each. $\mathbf{t}_\gamma[i]$ (similarly $\mathbf{t}_\delta$) is equal to the number of text pairs such that the bits $(t_{f,\gamma}, t_{b,\gamma})$ are equal to i.
- **Analysis**
  1. For each guess of $(k_{f,\gamma}, k_{b,\gamma})$, calculate the vector

$$\mathbf{c}_\gamma = (C_K^N(\gamma_1), C_K^N(\gamma_2), C_K^N(\gamma_3), C_K^N(\gamma_4))^\top,$$

     by partially encrypting/decrypting the data in $\mathbf{t}_\gamma$. Do similarly for the δ-approximations to calculate $\mathbf{c}_\delta$.

Table 3. This table specifies the key/text bits involved in the attack, as well as the parity key bits derived. X^i denotes the i'th bit of X.

Forward key bits guessed				#bits
$k_{f,\gamma}$	$\{K_0^{18}, \ldots, K_0^{23}\}$	$k_{f,\delta}$	$\{K_0^{24}, \ldots, K_0^{35}, K_0^{42}, \ldots, K_0^{47}\}$	6+18
Backward key bits guessed				#bits
$k_{b,\gamma}$	$\{K_{15}^{24}, \ldots, K_{15}^{35}, K_{15}^{42}, \ldots, K_{15}^{47}\}$	$k_{b,\delta}$	$\{K_{15}^{18}, \ldots, K_{15}^{23}\}$	18+6
Plaintext bits stored				#bits
$t_{f,\gamma}$	$\{\mathcal{P}^{11}, \ldots, \mathcal{P}^{16}, \mathcal{P}^{39}, \mathcal{P}^{50}, \mathcal{P}^{56}\}$			9
$t_{f,\delta}$	$\{\mathcal{P}^{0}, \mathcal{P}^{7}, \mathcal{P}^{15}, \ldots, \mathcal{P}^{24}, \mathcal{P}^{27}, \ldots, \mathcal{P}^{31}, \mathcal{P}^{44}, \mathcal{P}^{47}, \mathcal{P}^{48}\}$			20
Ciphertext bits stored				#bits
$t_{b,\gamma}$	$\{\mathcal{C}^{0}, \mathcal{C}^{7}, \mathcal{C}^{15}, \ldots, \mathcal{C}^{24}, \mathcal{C}^{27}, \ldots, \mathcal{C}^{31}, \mathcal{C}^{44}, \mathcal{C}^{47}, \mathcal{C}^{48}\}$			20
$t_{b,\delta}$	$\{\mathcal{C}^{11}, \ldots, \mathcal{C}^{16}, \mathcal{C}^{39}, \mathcal{C}^{50}, \mathcal{C}^{56}\}$			9
Parity bits obtained from signs				
p_A	$K_1^{22} \oplus K_2^{44} \oplus K_3^{22} \oplus K_5^{22} \oplus K_6^{44} \oplus K_7^{22} \oplus K_9^{22} \oplus K_{10}^{44} \oplus K_{11}^{22} \oplus K_{13}^{22}$			
p_B	$p_A \oplus K_{13}^{22} \oplus K_{13}^{19} \oplus K_{13}^{23}$			
p_C	$K_0^{22} \oplus K_2^{22} \oplus K_3^{44} \oplus K_4^{22} \oplus K_6^{22} \oplus K_7^{44} \oplus K_8^{22} \oplus K_{10}^{22} \oplus K_{11}^{44} \oplus K_{12}^{22}$			
p_D	$p_C \oplus K_0^{22} \oplus K_0^{19} \oplus K_0^{23}$			

2. Calculate

$$\Lambda(\mathbf{c}_\gamma) = \frac{\frac{1}{4}\sum_{i=1}^{4} \phi_M(\mathbf{c}_\gamma; \boldsymbol{\mu}_{i,\gamma}, \boldsymbol{\Sigma}_{i,\gamma} + (2^{-n} + 1/N)\mathbf{I})}{\phi_M(\mathbf{c}_\gamma; \mathbf{0}, \boldsymbol{\Sigma}_{W,\gamma} + (2^{-n} + 1/N)\mathbf{I})},$$

   for each guess of $(k_{f,\gamma}, k_{b,\gamma})$. If $\Lambda(\mathbf{c}_\gamma) \leq \Gamma_\gamma$, discard the key guess. Likewise, calculate $\Lambda(\mathbf{c}_\delta)$ for each guess of $(k_{f,\delta}, k_{b,\delta})$. If $\Lambda(\mathbf{c}_\delta) \leq \Gamma_\delta$, discard the key guess.
3. For each surviving key guess, determine the four bits p_A, p_B, p_C, p_D based on the signs of $\mathbf{c}_\gamma$ and $\mathbf{c}_\delta$.

- **Search**
  1. For each remaining guess of $(k_{f,\gamma}, k_{b_\gamma}, k_{f,\delta}, k_{b,\delta})$, guess the last 4 bits of the master key, and verify the guess by trial encryption.

6.2 Attack Complexity

In the following, we assume that one computational unit is the time it takes to perform one round of DES. The computational complexity of the distillation phase is $\mathcal{O}(N)$, while the memory complexity is $\mathcal{O}(2 \cdot 2^{29})$. For the analysis phase, each C_K^N can be calculated for all key guesses in time $\mathcal{O}((|k_{f,\cdot}| + |k_{b,\cdot}|)2^{|k_{f,\cdot}|+|k_{b,\cdot}|+1.6})$ using the FFT method presented in [10]. In total, step 1 of the analysis phase can be completed in time $\mathcal{O}(2 \cdot 4 \cdot 24 \cdot 2^{25.6}) \approx \mathcal{O}(2^{33.18})$. Step 2 requires the calculation of $\ell + 1$ terms for each key-guess of the type $(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$, to calculate the normal probabilities. Each term can be computed in time $\mathcal{O}(2M^3)$. Thus, step 2 takes a total of $\mathcal{O}(2 \cdot 2^{24} \cdot 5 \cdot 4^3) \approx \mathcal{O}(2^{33.32})$ time. Step 3 takes $\mathcal{O}(2 \cdot 2^{24-a_\gamma} + 2 \cdot 2^{24-a_\delta})$ time, where a_γ and a_δ is the advantage of the classifiers in step 2. The analysis step requires $\mathcal{O}(2^{24-a_\gamma} + 2^{24-a_\delta})$ memory to store the surviving key guesses. The search phase

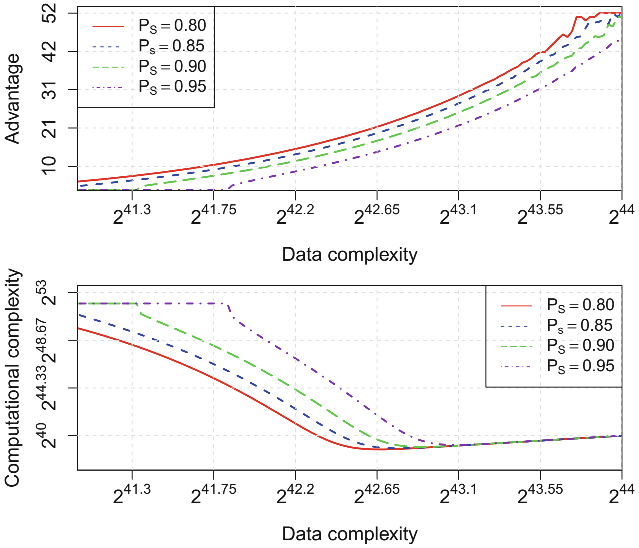

Fig. 12. Top: Combined advantage of the two likelihood classifiers using approximations in Table 2. The success probabilities include the probability of guessing the four parity bits correctly. Bottom: The computational complexity of our key-recovery attack on DES. Each curve has a clear minimum where the trade-off between the data complexity and the strength of the classifiers is optimal.

requires $\mathcal{O}(16 \cdot 2^{48-(a_\gamma+a_\delta)} \cdot 2^{56-52}) = \mathcal{O}(16 \cdot 2^{56-(a_\gamma+a_\delta+4)})$ time and negligible memory. Dividing everything by 16 to get the total number of full DES encryptions, the computational complexity is approximately

$$\mathcal{O}(N \cdot 2^{-4} + 2^{29.18} + 2^{29.32} + 2^{21-a_\gamma} + 2^{21-a_\delta} + 2^{52-(a_\gamma+a_\delta)}).$$

Thus, the attack complexity depends on the advantage of the two classifiers, which in turn depends on the choice of Γ_γ and Γ_δ. Note that step 3 of the analysis phase is not guaranteed to succeed, so the threshold values must be chosen such that the overall success probability of the attack is P_S. Namely, if P_γ and P_δ is the success probabilities of the two classifiers, and Q_γ and Q_δ is the success probabilities of determining the parity bits, then we fix Γ_γ and Γ_δ such that $P_\gamma \cdot P_\delta \cdot Q_\gamma \cdot Q_\delta = P_S$. Using the data obtained in Sect. 5.3, we calculated the total advantage $a_\gamma + a_\delta + 4$ for different N and different values of the success probability P_S. The results are shown in Fig. 12, along with the corresponding attack complexities. For low data complexities, the search phase is dominant, and so the $2^{52-(a_\gamma+a_\delta)}$ term determines the time complexity. For high data complexities, however, the $N \cdot 2^{-4}$ term is dominant. This gives each complexity curve a clear minimum. In a comparison to Matsui's attack, we see that for $P_S = 85\%$, the minimum is achieved at $N = 2^{42.775}$ where the computational complexity is $2^{38.86}$ DES encryptions. This is 17.6 times faster than Matsui's attack estimate (or 4.4 times faster than Junod's estimate of the attack in [17]) using $2^{40.2}$ fewer texts.

6.3 Experimental Verification

While it would be possible to carry out the attack in practice, we would need to do this for many keys to get an idea of the actual advantage, making the experiment infeasible. Instead, we measured the actual values of $\mathbf{c}_\gamma$ and $\mathbf{c}_\delta$ over 14 and 18 rounds of DES (the right key and wrong key, respectively) with $N = 2^{42.78}$ for randomly chosen keys. This can be done in a bitsliced manner, and is therefore faster than performing the actual attack, while giving us all the information we need to verify our model. Using several months of CPU time, we collected 1300 data points for the right key and wrong key distributions. We first note that the observed distributions closely match those predicted by the model in e.g. Figure 8. Moreover, we obtain the advantages $a_\gamma = 6.72$ and $a_\delta = 10.31$, which would give us a complexity of $2^{38.88}$ – very close to that predicted by our model.

References

1. TLS stats from 1.6 billion connections to mozilla.org. https://jve.linuxwall.info/blog/index.php?post/2016/08/04/TLS-stats-from-1.6-billion-connections-to-mozilla.org. Accessed 07 Sep 2017
2. Biham, E., Shamir, A.: Differential cryptanalysis of the full 16-round DES. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 487–496. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_34
3. Biryukov, A., Cannière, C.D., Quisquater, M.: On multiple linear approximations. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 1–22. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_1
4. Bogdanov, A., Geng, H., Wang, M., Wen, L., Collard, B.: Zero-correlation linear cryptanalysis with FFT and improved attacks on ISO standards Camellia and CLEFIA. In: Lange, T., Lauter, K., Lisoněk, P. (eds.) SAC 2013. LNCS, vol. 8282, pp. 306–323. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43414-7_16
5. Bogdanov, A., Rijmen, V.: Zero-correlation linear cryptanalysis of block ciphers. IACR Cryptology ePrint Archive 2011, 123 (2011)
6. Bogdanov, A., Rijmen, V.: Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Des. Codes Cryptogr. **70**(3), 369–383 (2014)
7. Bogdanov, A., Tischhauser, E.: On the wrong key randomisation and key equivalence hypotheses in Matsui's algorithm 2. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 19–38. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43933-3_2
8. Bogdanov, A., Tischhauser, E., Vejre, P.S.: Multivariate linear cryptanalysis: the past and future of PRESENT. IACR Cryptology ePrint Archive 2016, 667 (2016)
9. Cho, J.Y.: Linear cryptanalysis of reduced-round PRESENT. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 302–317. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11925-5_21
10. Collard, B., Standaert, F.-X., Quisquater, J.-J.: Improving the time complexity of Matsui's linear cryptanalysis. In: Nam, K.-H., Rhee, G. (eds.) ICISC 2007. LNCS, vol. 4817, pp. 77–88. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76788-6_7
11. Daemen, J.: Cipher and hash function design strategies based on linear and differential cryptanalysis. Ph.D. thesis, KU Leuven (1995)

12. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-662-04722-4
13. Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. J. Math. Cryptol. **1**(3), 221–242 (2007)
14. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional linear cryptanalysis of reduced round serpent. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 203–215. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70500-0_15
15. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional extension of Matsui's algorithm 2. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 209–227. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03317-9_13
16. Kaliski Jr., B.S., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 26–39. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_4
17. Junod, P.: On the complexity of Matsui's attack. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 199–211. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45537-X_16
18. Knudsen, L.R., Mathiassen, J.E.: A chosen-plaintext linear attack on DES. In: Goos, G., Hartmanis, J., Leeuwen, J., Schneier, B. (eds.) FSE 2000. LNCS, vol. 1978, pp. 262–272. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44706-7_18
19. Langford, S.K., Hellman, M.E.: Differential-linear cryptanalysis. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 17–25. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_3
20. Lindsay, B.G.: Mixture models: theory, geometry and applications. In: NSF-CBMS Regional Conference Series in Probability and Statistics, pp. i–163. JSTOR (1995)
21. Matsui, M.: Linear cryptanalysis method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_33
22. Matsui, M.: The first experimental cryptanalysis of the data encryption standard. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 1–11. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_1
23. Nguyen, P.H., Wu, H., Wang, H.: Improving the algorithm 2 in multidimensional linear cryptanalysis. In: Parampalli, U., Hawkes, P. (eds.) ACISP 2011. LNCS, vol. 6812, pp. 61–74. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22497-3_5
24. Nyberg, K.: Linear approximation of block ciphers. In: Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 439–444. Springer, Heidelberg (1995). https://doi.org/10.1007/BFb0053460
25. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. **21**(1), 131–147 (2008)
26. Semaev, I.A.: New results in the linear cryptanalysis of DES. IACR Cryptology ePrint Archive 2014, 361 (2014). http://eprint.iacr.org/2014/361
27. Zhao, J., Wang, M., Wen, L.: Improved linear cryptanalysis of CAST-256. J. Comput. Sci. Technol. **29**(6), 1134–1139 (2014)