On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2

Andrey Bogdanov[1] and Elmar Tischhauser[2]

[1] Technical University of Denmark
anbog@dtu.dk
[2] KU Leuven and iMinds, Belgium
elmar.tischhauser@esat.kuleuven.be

Abstract. This paper aims to improve the understanding of the complexities for Matsui's Algorithm 2 — one of the most well-studied and powerful cryptanalytic techniques available for block ciphers today.
We start with the observation that the standard interpretation of the wrong key randomisation hypothesis needs adjustment. We show that it systematically neglects the varying bias for wrong keys. Based on that, we propose an adjusted statistical model and derive more accurate estimates for the success probability and data complexity of linear attacks which are demonstrated to deviate from all known estimates. Our study suggests that the efficiency of Matsui's Algorithm 2 has been previously somewhat overestimated in the cases where the adversary attempts to use a linear approximation with a low bias, to attain a high computational advantage over brute force, or both. These cases are typical since cryptanalysts always try to break as many rounds of the cipher as possible by pushing the attack to its limit.
Surprisingly, our approach also reveals the fact that the success probability is *not* a monotonously increasing function of the data complexity, and can decrease if more data is used. Using less data can therefore result in a more powerful attack.
A second assumption usually made in linear cryptanalysis is the key equivalence hypothesis, even though due to the linear hull effect, the bias can heavily depend on the key. As a further contribution of this paper, we propose a practical technique that aims to take this into account.
All theoretical observations and techniques are accompanied by experiments with small-scale ciphers.

Keywords: Block ciphers, linear cryptanalysis, data complexity, wrong key randomisation hypothesis, key equivalence, linear hull effect

1 Introduction

Linear cryptanalysis proposed by Matsui [25, 26], besides differential cryptanalysis [5], has been a seminal cryptanalytic technique used to attack block ciphers since two decades now. It was linear cryptanalysis that both theoretically and practically broke the former U.S. encryption standard DES. This might suggest

that NSA could have been unaware of the entire power of this attack, at least at the design time of DES back in the 1970s.

Numerous papers investigated the questions of how to improve linear cryptanalysis on the one hand and how to design ciphers resistant to linear cryptanalysis on the other. With the establishment of such block cipher design approaches as the wide-trail design strategy [9], which eventually lead to the design of the current U.S. encryption standard AES [9], the cryptographers were given reliable tools to construct ciphers that are arguably resistant against the classical flavours of linear cryptanalysis.

The extension of linear cryptanalysis to take advantage of multiple approximations revived the field [16–18, 32]. Lately, increasingly more works [23, 27, 32] have been dedicated to the study of the linear hull effect [28, 29] – the fact that depending on the key, the efficiency of linear cryptanalysis may significantly vary. Also in terms of the success probability and data complexity estimation, a lot of detailed works have been published [2, 3, 6, 19–21, 33]. The fact that many published attacks have data and time requirements beyond practical reach implies that the question of how to accurately estimate their complexity (and hence determine which attack actually is a valid attack) is of great importance to the security of block ciphers.

Our contributions. In this paper, we aim to obtain a more accurate estimation of success probability and data complexity of linear attacks using Matsui's Algorithm 2 — a question fundamental to symmetric-key cryptanalysis. Our contributions are as follows:

- **New wrong key randomisation hypothesis:** Informally speaking, the wrong key randomisation hypothesis says that by partially decrypting/encrypting with a wrong key up to the boundary of the linear approximation, the adversary faces a randomly drawn permutation instead of the expected cipher structure with rounds peeled off. The standard interpretation of this hypothesis in linear cryptanalysis seems to have been to replace a randomly drawn permutation which varies for every wrong key candidate with the expected behaviour among all permutations. We demonstrate that this can be misleading and result in underestimated complexity in some cases. Those cases are likely to occur when the adversary tries to exploit a linear approximation with low bias, or to attain a high advantage over the brute force, or both. These cases are actually very typical since cryptanalysts always try to break as many rounds of the cipher as possible by pushing the attack to the limit.
- **More data does not necessarily mean higher probability of success:** As a surprising consequence of the adjusted wrong key randomisation hypothesis, our analysis reveals that the success probability in general is not a monotonous function of the data complexity. This means that sometimes, using less data can result in a better success probability of a linear attack. This is backed up by experimental results confirming the non-monotonous behaviour of the success rate.

– **Linear hull vs linear trails:** The general methodologies to evaluate the complexity of linear attacks at hand assume the exact bias or its good estimate is given to the adversary. Practically speaking, however, this is never the case for almost any real-world cipher. This is due to the fact that for a relatively large block size (e.g. longer 50 bits) it is challenging to exactly evaluate the bias even for one known key. That is why most linear attacks base their complexity estimates on one or several known trails[3] (rather than on the entire linear hull bias). In this context, we make two observations in this paper. First, we propose to split the linear hull into a signal part and a noise part. The signal part is then sampled for random cipher keys to obtain a more reliable evaluation of the impact of those trails. Second, we statistically model the noise part to make the estimation of complexity more realistic.

The remainder of the paper is organized as follows. Some brief background on block ciphers, linear cryptanalysis and previous work in this direction is given in Section 2. In Section 3, the new model for the data complexity of linear attacks based on the new wrong key randomisation hypothesis is developed. The non-monotonicity of the success rate as function of data complexity is studied in Section 4. Section 5 proposes a method of computing the data complexity of a linear attack and presents experimental results. Our refined key equivalence hypothesis is presented in Section 6. Section 7 proposes a practical algorithm implementing the new key equivalence hypothesis for key-alternating ciphers. We conclude in Section 8.

2 Preliminaries

2.1 Notation

We denote by $\mathbb{F}_2 = \{0, 1\}$ the finite field with two elements and the n-dimensional vector space over $\mathbb{F}_2$ by $\mathbb{F}_2^n$. The canonical scalar product of two vectors $a, b \in \mathbb{F}_2^n$ is denoted by $a^T b$.

We denote by $\mathcal{N}(\mu, \sigma^2)$ the normal distribution with mean μ and variance σ^2. The probability density and cumulative distribution function of the standard normal distribution $\mathcal{N}(0, 1)$ are denoted by $\phi(x)$ and $\Phi(x)$, respectively.

2.2 Block ciphers and linear cryptanalysis

Block ciphers. A *block cipher* is a mapping $E : \mathbb{F}_2^n \times \mathbb{F}_2^\kappa \to \mathbb{F}_2^n$ with the property that $E_k \stackrel{\text{def}}{=} E(\cdot, k)$ is a bijection of $\mathbb{F}_2^n$ for every $k \in \mathbb{F}_2^\kappa$. If $y = E_k(x)$, we refer to x as the *plaintext*, k as the *key* and y as the *ciphertext* of x under the key k. We call n the *block length* and κ the *key size* of the cipher.

Block ciphers are often constructed as iterated mappings based on round functions $\rho_i[k_i]$. Let R denote the number of rounds. A key scheduling algorithm

[3] We are aware of the earlier term *linear characteristic* [4] but prefer to use the term *linear trail* throughout the paper.

expands the encryption key k into R round keys $K \stackrel{\text{def}}{=} (k_0, \ldots, k_{R-1})$. The ciphertext y of $x_0 = x$ is then obtained as $y = x_R$ with $x_{i+1} = \rho_i[k_i](x_i)$. If the iteration can be written as a sequence of unkeyed rounds and bitwise addition of the round keys by XOR, the cipher is called a *key-alternating cipher* [8,9].

Note that ciphers following the substitution-permutation network design are key-alternating by definition. However, also some Feistel ciphers [13] can be written as key-alternating ciphers [10]. This includes the well-known Feistel ciphers CLEFIA [35], CAMELLIA [1], Piccolo [34], SMS4 [24], KASUMI [14].

Linear cryptanalysis and Matsui's Algorithm 2. A *linear approximation* (α, β) of a vectorial Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is an ordered pair of n-bit *masks* α and β. It is said to hold with probability $p \stackrel{\text{def}}{=} \Pr_{x \in \mathbb{F}_2^n}(\alpha^T x = \beta^T f(x))$. The deviation of p from $1/2$ is called the *bias* $\epsilon \stackrel{\text{def}}{=} p - 1/2$. The *correlation* of a linear approximation (α, β) is $C \stackrel{\text{def}}{=} 2p - 1 = 2\epsilon$. The quantity $LP \stackrel{\text{def}}{=} C^2$ is called the *linear probability* of (α, β).

Linear cryptanalysis [25, 26] is a known plaintext attack exploiting linear relations between bits of the plaintext and ciphertext holding with absolute bias $|\epsilon| > 0$. Note that in the known plaintext model, the plaintexts are assumed to be sampled independently and uniformly at random from the plaintext space, which implies that repetitions can occur [20, 33].

In this paper, we consider linear attacks using Matsui's Algorithm 2. We describe the attack for the case where subkey bits of the last round are attacked. The adversary observes a number N of plaintext/ciphertext pairs encrypted under the same cipher key k and chooses a linear approximation (α, β) with $|\epsilon| > 0$ for the remaining first $R - 1$ rounds. Suppose that the bits of $x_{R-1} = \rho[k_R]^{-1}(y)$ selected by β depend on M bits of k_R and the attacker wants to recover a subset of $m \leq M$ of them. For each of the 2^m possible values of the target subkey bits, the adversary (partially) decrypts the N ciphertexts and tests whether the linear approximation $\alpha^T x = \beta^T \rho[k_R]^{-1}(y)$ holds. In this way, a counter T_i is maintained for each key candidate k_i, $0 \leq i < 2^m$. After this step, the key candidates are ranked in increasing order of the absolute value of the sample bias $|\widehat{\epsilon}_i| \stackrel{\text{def}}{=} |T_i/N - 1/2|$. Following [33], if the correct key k_r is ranked among the highest 2^l out of the 2^m key candidates with probability P_S, we say that the attack provides an *advantage* of $a \stackrel{\text{def}}{=} m - l$ bits over exhaustive search with *success probability* P_S.

Linear trails and hulls. A linear approximation (α, β) for an iterative block cipher of R rounds with round functions ρ_i can actually be decomposed into R connecting linear approximations for the intermediate steps: $U = [(\alpha, u_1), (u_1, u_2), \ldots, (u_{R-1}, \beta)]$ with $u_i \in \mathbb{F}_2^n$. For each fixed value of the u_i, such a sequence is called a *linear trail* [9] or *linear characteristic* [25, 26]. The approximation (α, β) can permit many trails with the same input mask α and output mask β, but different intermediate masks. The collection of all such trails is called the *linear hull* (α, β) [28, 29].

2.3 Previous analyses of the data complexity of linear attacks

The data complexity of both Matsui's Algorithm 2 and the more general problem of distinguishing the distributions for the right and the wrong keys have been extensively studied in the literature [2, 3, 6, 19–21, 25, 26, 33].

In his original papers, using a normal approximation to the binomial distribution, Matsui [25, 26] estimates the data complexity to be of the order $|2\epsilon|^{-2}$ and gives estimations which multiple of this is required to obtain a certain success probability. This analysis has been systematized and deepened by Junod [19]. Furthermore, Junod and Vaudenay [21] have proven that Matsui's key ranking procedure is optimal for the case of Algorithm 2 using a single linear approximation.

In his important work, Selçuk [33] presented a thorough statistical analysis of the data complexity of linear and differential attacks based on a model of Junod [19] and a normal approximation for order statistics. This yields practical closed formulas for the success probability P_S and data complexity N of a linear attack when an advantage of a bits is sought:

Theorem 1 ([33, Theorem 2]). *Let P_S be the probability that a linear attack on an m-bit subkey, with a linear approximation of probability p, with N known plaintext blocks, delivers an a-bit or higher advantage. Assuming that the linear approximation's probability to hold is independent for each key tried and is equal to $1/2$ for all wrong keys, one has for sufficiently large m and N:*

$$P_S = \Phi\left(2\sqrt{N}|p - 1/2| - \Phi^{-1}(1 - 2^{-a-1})\right). \tag{1}$$

Corollary 1 ([33, Corollary 1]). *With the assumptions of Theorem 1,*

$$N = \left((\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-a-1}))/2\right)^2 \cdot |p - 1/2|^{-2} \tag{2}$$

plaintext blocks are needed in a linear attack to accomplish an a-bit advantage with a success probability of P_S.

Other published estimates include analyses by Junod [20], Baignères, Junod and Vaudenay [2], Baignères and Vaudenay [3], and Blondeau, Gérard and Tillich [6]. Those estimates are summarised in Table 1, with $D(p||q)$ denoting the Kullback-Leibler divergence between two binomial distributions with probabilities p and q.

Note that throughout the literature, the assumption is made that decrypting with a wrong key results in a zero bias for the linear approximation. As we will see, this constitutes a simplified view of the problem.

2.4 Distribution of biases in Boolean permutations

Daemen and Rijmen [11] have proved the following characterisation of the distribution of correlation of a fixed linear approximation over the set of all n-bit permutations:

Estimate for data complexity N	Reference
$N \approx \frac{2\Phi^{-1}\left(\frac{(1-P_S)+2^{-a-1}}{2}\right)^2}{D(p\|\|0.5)}$	Baignères, Junod and Vaudenay [2], Theorem 6
$N \approx -\frac{\ln \max\{1-P_S, 2^{-a-1}\}}{D(p\|\|0.5)}$	Baignères and Vaudenay [3], Corollary 4
$N' = -\frac{\ln\left(\frac{\nu \cdot 2^{-a-1}}{\sqrt{D(p\|\|0.5)}}\right) + 0.5\ln\left(-\ln\left(\nu \cdot 2^{-a-1}\right)\right)}{D(p\|\|0.5)}$ with $\nu = \left((p-0.5)\sqrt{2\pi(1-p)}\right) / \left(\sqrt{p}/2\right)$	Blondeau, Gérard and Tillich [6], Theorem 2

Table 1: Estimates for the data complexity of a linear attack based on a linear approximation with probability p, success probability P_S and advantage a.

Fact 1 ([11, Theorem 4.7]). *Consider a fixed nontrivial linear approximation (α, β) with $\alpha, \beta \neq 0$. When $n \geq 5$, the distribution of the correlation $C_{\alpha,\beta}$ over all n-bit permutations can be approximated by the following distribution up to continuity correction:*

$$C_{\alpha,\beta} \sim \mathcal{N}(0, 2^{-n}). \tag{3}$$

Since $C = 2\epsilon$, this immediately implies

Corollary 2. *With the assumptions of Fact 1,*

$$\epsilon_{\alpha,\beta} \sim \mathcal{N}(0, 2^{-n-2}). \tag{4}$$

3 Improved key randomisation hypothesis and success rate

3.1 More accurate wrong key randomisation

In Matsui's Algorithm 2 using a linear approximation (α, β) and N known plaintexts, a counter T_i is maintained for each key candidate k_i. For each of the N texts, the counter T_i is incremented if the approximation holds for a text when performing a trial decryption with the key k_i.

The distribution of T_i has a crucial impact on the precision of estimating the data complexity of Matsui's Algorithm 2. First of all, the distribution of the T_i for the wrong keys has to be determined. It has been generally assumed that once the ciphertext is partially decrypted using a wrong key, the resulting permutation – over which the linear approximation is checked – turns into a randomly chosen permutation. This is called *the wrong key randomisation hypothesis* [15, 19].

Multiple works have used this hypothesis and it usually proves to reflect the reality. However, in its basic formulation it does not explicitly specify which

distribution to assume for the T_i. In the sequel, we argue that this is exactly the point where the standard interpretation of the wrong key randomisation hypothesis needs adjustment.

To the best of our knowledge, all previous complexity evaluations of Matsui's Algorithm 2 have used the hypothesis that for all wrong keys k_w, $0 \leq w \neq r < 2^m$, the approximation (α, β) will hold with probability of exactly $1/2$, that is with bias zero. This constitutes the best scenario from the attacker's point of view:

Hypothesis 1 (Standard wrong key randomisation hypothesis). *Consider a nontrivial linear approximation $\mathcal{L} = (\alpha, \beta)$ with absolute bias $|\epsilon| \gg 0$ for virtually all possible cipher keys. Let k_r be the right subkey guess. Then, for virtually all cipher keys and for all wrong subkey guesses $k_w \neq k_r$:*

$$\left| \Pr(\mathcal{L} \textit{ holds} \mid k_w) - \frac{1}{2} \right| = 0.$$

In this case, making the usual independence assumption, the distribution of the wrong key counters T_w is given by a binomial distribution with probability $p = 1/2$ and N repetitions. For sufficiently large N, this can be very closely approximated by a normal distribution with mean $Np = N/2$ and variance $Np(1-p) = N/4$. The sample bias $\widehat{\epsilon}_w = T_w/N - 1/2$ of the wrong keys is therefore assumed to be approximately distributed as $\mathcal{N}(0, 1/(4N))$.

Though the standard formulation of the wrong key randomisation hypothesis is inspired by the intention to make the approximation (α, β) behave as for a randomly drawn n-bit permutation, the distribution of the $\widehat{\epsilon}_w$ is not completely adequate. In fact, it is known (see Fact 1 and Corollary 2) that the bias of (α, β) over the n-bit permutations is not constantly zero, but instead follows a known distribution over the wrong keys. We therefore postulate:

Hypothesis 2 (Adjusted wrong key randomisation hypothesis). *Consider a nontrivial linear approximation $\mathcal{L} = (\alpha, \beta)$ with absolute bias $|\epsilon| \gg 0$ for virtually all possible cipher keys. Let k_r be the right subkey guess. Then, for virtually all cipher keys and for all wrong subkey guesses $k_w \neq k_r$:*

$$\left| \Pr(\mathcal{L} \textit{ holds} \mid k_w) - \frac{1}{2} \right| = \mathcal{N}(0, 2^{-n-2}).$$

The following lemma, which is a new result, takes this into account.

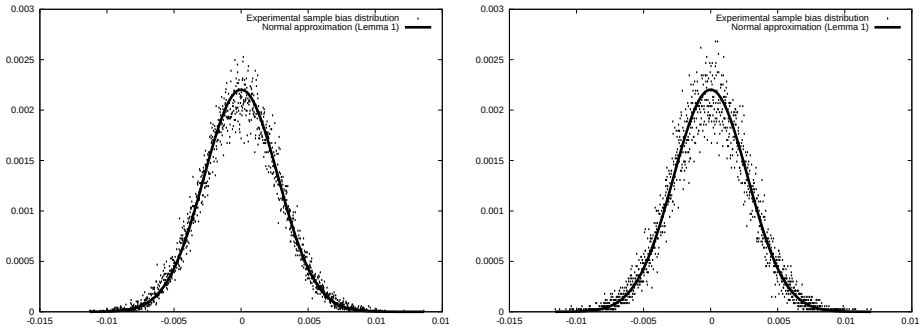
Lemma 1. *In a linear attack with Matsui's Algorithm 2 on an n-bit block cipher using N known plaintexts, the sample bias $\widehat{\epsilon}_w$ of the wrong keys approximately follows a normal distribution with mean zero and variance $1/4 \cdot (1/N + 1/2^n)$:*

$$\widehat{\epsilon}_w \sim \mathcal{N}(0, 1/4 \left(\frac{1}{N} + \frac{1}{2^n} \right)). \tag{5}$$

Proof. See the full version [7] of this paper for the proof.

Previous interpretations of the wrong key randomisation hypothesis have therefore used the mean zero instead of the full distribution $\mathcal{N}(0, 2^{-n-2})$ for the bias when decrypting with a wrong key. For the sample bias of the wrong keys, this resulted in using $\mathcal{N}(0, 1/(4N))$ instead of $\mathcal{N}(0, 1/4\left(\frac{1}{N} + \frac{1}{2^n}\right))$, implying that the distributions for the right key and the wrong keys were assumed to only differ in the mean, but had the same variance. While this arguably simplifies the analysis, the possible impact of this simplification has to be investigated.

Experimental verification. Even in the new form presented in Lemma 1, the wrong key randomisation hypothesis remains an idealisation. In order to verify that it reflects the reality with reasonable accuracy, we have experimentally determined the distribution of the sample bias over 2^{16} wrong keys for two structurally very different small-scale ciphers with a block length of 20 bits: SmallPresent-20 [22] with 8 rounds, and RC6-5/6/10 [31] with four 5-bit words, 6 rounds and an 80-bit key. In both cases, the number of samples was $N = 2^{16}$. As illustrated in Figure 1 the resulting distributions follow the theoretical estimate of (5) quite closely in both cases. Note that the scattering of data points occurs due to the fact that we are basically using a histogram with bin size one, and deal with raw data instead of averaging.



(a) SmallPresent-20, 8 rounds (b) RC6-5/6/10

Fig. 1: Experimental distribution of the sample bias over 2^{16} wrong keys and 2^{16} texts for SmallPresent and small-scale RC6.

3.2 Probability of success

In this section, we study the implications of Lemma 1 for the success probability in linear cryptanalysis with Matsui's Algorithm 2. This leads to a new formula for the success probability of a linear attack.

Theorem 2. *Consider a linear attack with Matsui's Algorithm 2 on an n-bit block cipher ($n \geq 5$) using a linear approximation with bias $\epsilon \neq 0$ and sufficiently*

large $N \leq 2^n$ known plaintexts. Denote by P_S the probability that this attack succeeds with an advantage of $a > 0$ bits over exhaustive key search. Then

$$P_S \approx \Phi\left(2\sqrt{N}|\epsilon| - \sqrt{1 + \frac{N}{2^n}}\Phi^{-1}(1 - 2^{-a-1})\right). \tag{6}$$

Proof. See the full version of this paper [7] for the proof.

Note that the difference between (6) and Selçuk's formula (1) lies in the factor $\sqrt{1 + \frac{N}{2^n}}$ of the term $\Phi^{-1}(1-2^{-a-1})$. Since Φ is monotonously increasing, our estimate for P_S is always smaller or equal to (1), and the resulting data complexity required for a certain advantage and P_S will always be at least as big as the one of (2).

The biggest deviations between both models occur when the influence of the second term $\sqrt{1 + \frac{N}{2^n}} \cdot \Phi^{-1}(1 - 2^{-a-1})$ grows. This can happen if the adversary seeks a particularly big advantage a, or when the number of known plaintexts gets close to 2^n. Both cases typically occur when the cryptanalyst is aiming for the maximum possible number of rounds that can be broken by his respective linear attack.

4 Non-monotonicity of success rate as function of data complexity

Consider any fixed given combination of the bias ϵ, the block length n and the advantage a. The success probability of a linear attack is then a function of the number of known plaintexts N only and can hence be expressed as $P_S(N)$. Even though our estimate for $P_S(N)$ given by Theorem 2 is always smaller or equal to Selçuk's formula (1), the addition of the second term results in a function that is not necessarily monotonously increasing in N anymore.

From (6), we can derive

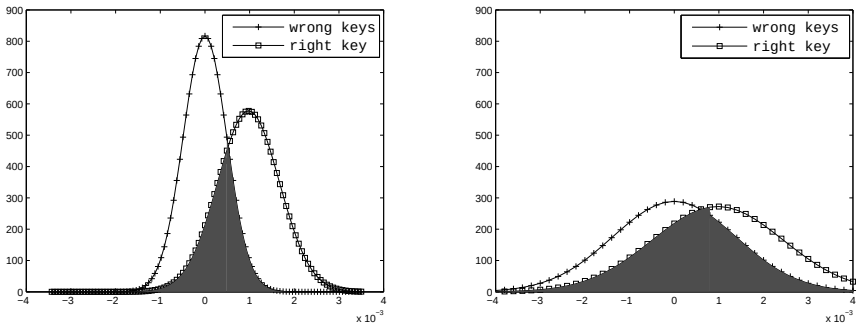
Proposition 1. *For fixed ϵ, a and n, the success probability $P_S(N)$ with respect to the data complexity as given by Eq. (6) attains a relative maximum at*

$$\widehat{N} \stackrel{\text{def}}{=} \frac{4|\epsilon|^2 \cdot 2^{2n}}{\left(\Phi^{-1}(1 - 2^{-a-1})\right)^2 - 4|\epsilon|^2 \cdot 2^{2n}}. \tag{7}$$

Proposition 1 implies that our model can in certain cases predict a decrease in success probability for an increased number of known plaintexts. While this may seem counterintuitive at first, one has to take into account that the success probability depends on the overlapping area between two approximately normal distributions, namely $\mathcal{N}(\epsilon, \frac{1}{4N})$ for the right key and $\mathcal{N}(0, \frac{1}{4}\left(\frac{1}{N} + \frac{1}{2^n}\right))$ for the wrong keys. In the context of small ϵ and large N of the order 2^n, increasing N can actually result in increasing the overlapping area, and hence decrease the success probability. This can be seen as a direct consequence of linear cryptanalysis being a known plaintext attack: Since the observed plaintexts are sampled

independently and uniformly at random from the plaintext space, the probability of duplicates increases with N, up to a point where adding more samples to the statistic only amplifies the noise. An attack exploiting a fixed linear approximation with fewer known plaintexts could therefore be more efficient than with more given samples.

A given advantage a corresponds to a fixed threshold T for distinguishing the distributions, with the type I error $\mathcal{E}_I = 1 - P_S$ varying with N, and fixed type II error $\mathcal{E}_{II} = 2^{-a-1}$. Having $P_S(N) = P_S(N')$ for $N \neq N'$ is therefore equivalent to having the same overlapping area $\mathcal{E}_I + \mathcal{E}_{II}$ between the distributions for N and N' samples. This is depicted in Figure 2: the overlapping area $\mathcal{E}_I + \mathcal{E}_{II}$ between the two Gaussian distributions in Figure 2a and 2b is the same for different values of N.



(a) Situation for $N = 2^{20}$. (b) Situation for $N = 2^{17}$.

Fig. 2: Example of an equal overlapping area between $\mathcal{N}(\epsilon, \frac{1}{4N})$ and $\mathcal{N}(0, \frac{1}{4}\left(\frac{1}{N} + \frac{1}{2^n}\right))$ for $n = 20, \epsilon = 2^{-10}$ and different values of N.

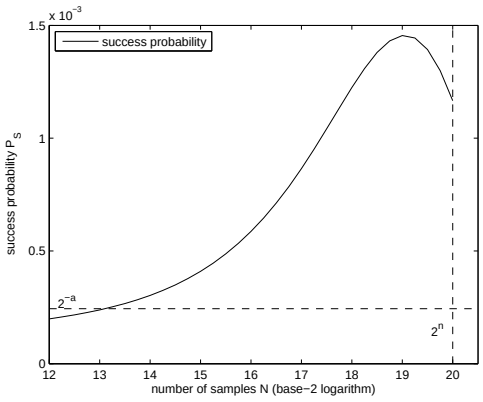
We note that two conditions have to be fulfilled to be able to speak of meaningful (i.e., practically relevant) non-monotonous behaviour: First, the condition $\widehat{N} < 2^n$ has to be satisfied (since N cannot exceed 2^n); and second, one must have $P_S \geq 2^{-a}$, i.e. the success probability of the attack must be higher than two times the false positive rate. Otherwise, the adversary would have to repeat the attack $1/P_S > 2^a$ times and gain only a bits advantage over the exhaustive search.

An example of a parameter combination fulfilling both conditions is $|\epsilon| = 2^{-10}, n = 20$ and $a = 12$, i.e., seeking a large advantage out of an approximation with only marginal bias. In this case, $\widehat{N} \approx 2^{18.75} < 2^{20}$, and $P_S(\widehat{N}) \approx 2^{-9.89} > 2^{-12} = 2^{-a}$. With $P_S(2^{20}) \approx 2^{-10.45}$, this constitutes a meaningful example where using more samples actually decreases the success probability. This theoretical prediction has been verified in the real world using experiments with SmallPresent-20. The recovery of 10000 keys exhibiting exactly the bias

$\epsilon = 2^{-10}$ was attempted for different values of N. The results given in Figure 3 confirm the non-monotonous behaviour.

N	theor. P_S	exp. P_S
2^{17}	0.00072	0.0006
2^{18}	0.00096	0.0009
$2^{18.5}$	0.00104	0.0009
$2^{18.75}$	0.00105	0.0011
2^{19}	0.00104	0.0010
$2^{19.5}$	0.00093	0.0009
2^{20}	0.00071	0.0007

(a) Experimental success probability.



(b) Plot of $P_S(N)$ (Theorem 2).

Fig. 3: Experimental verification of non-monotonous behaviour for SmallPresent-20 with $\epsilon = 2^{-10}$ and $a = 12$.

5 Evaluation of the data complexity

In practice, when evaluating a particular linear attack (where n and ϵ are fixed), it is often interesting to determine the number N of required known plaintexts for certain success probabilities and advantages that are sought by the attacker. In the case of $P_S = 1/2$ and an arbitrary fixed advantage of $a \geq 1$ bits, equation (6) yields a closed formula for N:

Corollary 3. *With the assumptions of Theorem 2, using a linear approximation with bias $|\epsilon| > \Phi^{-1}(1 - 2^{-a-1})/2^{n/2-1}$, the number N of known plaintexts required to obtain an advantage of $a \geq 1$ bits with success probability $P_S = 1/2$ is given by*

$$N \approx 1/\left(\left(2\epsilon/\Phi^{-1}(1 - 2^{-a-1})\right)^2 - 2^{-n}\right). \tag{8}$$

The condition $|\epsilon| > \Phi^{-1}(1 - 2^{-a-1})/2^{n/2-1}$ in Corollary 3 basically prevents the estimate for N from becoming negative. This happens if the sought advantage a is too big for the given bias $|\epsilon|$, resulting in a data requirement of $N > 2^n$ texts, which is clearly impossible and a step outside the model.

For values of P_S different from $1/2$, we can determine N by means of an efficient numerical procedure for given $P_S, a, |\epsilon|$ and n. Note that this procedure is equally applicable to the case $P_S = 1/2$.

Proposition 2. *With the assumptions of Theorem 2, for fixed ϵ, P_S, n and a, the data complexity N can be determined numerically using Algorithm 5.1 up to an absolute error of $1 - 2^{-n}$ in linear time in the block length n.*

Proof. See the full version of the paper [7] for the proof.

Algorithm 5.1 Numerical computation of the data complexity N

Input: Bias ϵ, block length n, success probability $P_S \geq 2^{-a}$, precision bound ν (*)
Output: Data complexity N required for the given parameters.

1: Define $f(N) = 2\sqrt{N}|\epsilon| - \sqrt{1 + \frac{N}{2^n}}\Phi^{-1}(1 - 2^{-a-1})$
2: Calculate $\widehat{N} \leftarrow \left(4|\epsilon|^2 \cdot 2^{2n}\right) / \left(\left(\Phi^{-1}(1 - 2^{-a-1})\right)^2 - 4|\epsilon|^2 \cdot 2^{2n}\right)$
3: $lower \leftarrow 1$, $upper \leftarrow \min\{\widehat{N}, 2^n\}$, $i \leftarrow 0$
4: **while** $|f(lower) - P_S| > 10^{-\nu}$ **and** $i < n$ **do**
5: $\quad mid \leftarrow \frac{lower + upper}{2}$
6: $\quad$ **if** $f(mid) < P_S$ **then**
7: $\quad\quad lower \leftarrow mid$
8: $\quad$ **else**
9: $\quad\quad upper \leftarrow mid$
10: $\quad$ **end if**
11: $\quad i \leftarrow i + 1$
12: **end while**
13: **return** *lower*

*The value of ν in step 4 is used to early-abort fast-converging iterations as soon as an adequate precision is reached. A recommended value is $\nu = 15$.

Algorithm 5.1 runs very efficiently even for large block sizes. For instance, a straightforward Matlab implementation computes the value $N = 2^{126.76}$ for $n = 128$, $|\epsilon| = 2^{-61.9}$, $a = 10$ and $P_S = 0.95$ in about 0.09 seconds on an Intel Core2 Duo E8400.

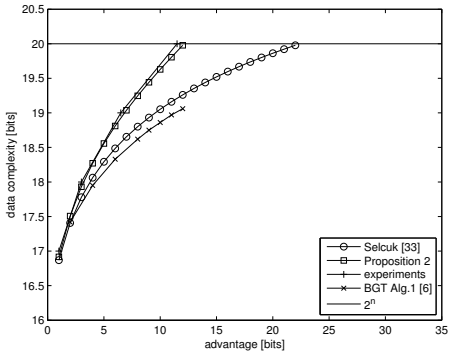
5.1 Experimental results

In this section, we summarise the results of experiments carried out to verify the accuracy of the estimate given by Theorem 2 and Proposition 2 and compare it to other models.

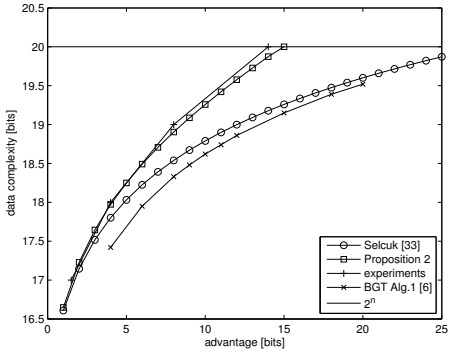
The experiments were first carried out on SmallPresent-20, a small-scale variant [22] of the block cipher PRESENT with block size $n = 20$ bits. The original key schedule algorithm was used. In all experiments, we fixed a linear approximation with a certain bias ϵ and success probability P_S and then analysed the data complexity N which is required to obtain different levels of advantage with this P_S. Each experiment for a certain combination of N and a was averaged over 1000 times to obtain a reliable relation between N and a for this fixed P_S. To verify the independence of our experimental findings from the structure of SmallPresent, all experiments were repeated with RC6-5/r/10, an instantiation of RC6 [31] with a block size of 20 bits and an 80-bit key. The results on

this small-scale variant of RC6 indicate that our model is equally applicable to this substantially different block cipher structure.

In the first experiment on SmallPresent, a linear approximation with bias $\epsilon = 2^{-8.22}$ was used. The success probability was fixed at 0.95. From (6), the influence of the new model for wrong keys is expected to manifest itself already for small advantages given this relatively high P_S and low ϵ (compared to the block length). The results are depicted in Figure 4a. The curve with squares was obtained using Proposition 2 with an estimation of the hull bias averaged over 200 random keys. We can see that the experiments follow the theoretical prediction very closely. The difference to the estimate of [33] is also apparent as soon as $a \geq 6$. For $a = 11$, Selçuk's formula can result in an underestimation of N of factor two. The line with crosses represents the estimate based on Algorithm 1 of [6]. The results of an analogous experiment on RC6-5/8/10 are given in Figure 4b.



(a) SmallPresent, 8 rounds, $n = 20$, $|\epsilon| = 2^{-8.22}, P_S = 0.95$.



(b) RC6-5/8/10, $n = 20$, $|\epsilon| = 2^{-8.09}, P_S = 0.95$.

Fig. 4: Theoretical and experimental evaluation of the data complexity estimate of Proposition 2 for different levels of advantage.

Additional experimental results for different levels of ϵ and a are given the full version of the paper [7].

Our experiments indicate that Theorem 2 and its derivatives are unlikely to decrease the precision in comparison to previous work, since our estimates are more realistic for large a and/or low ϵ but very close to Selçuk's for small advantages and/or high biases. They can hence be used as a universal replacement.

Larger block sizes. Given the experimental evidence supporting the accuracy of the estimates based on Theorem 2, it remains to investigate the impact of the new model for larger block sizes where no practical experiments can be carried out. This is detailed in the full version of the paper [7].

6 Towards a more realistic key equivalence hypothesis

In order to evaluate the success probability and data complexity of a linear attack using Matsui's Algorithm 2, one has to know the exact (absolute value of the) bias $\epsilon(k_r)$ of the used linear approximation (α, β) for the right key k_r.

6.1 Standard key equivalence hypothesis

Dually to the wrong key randomisation hypothesis, a common assumption for the statistical behaviour when partially de- and/or encrypting with the right key is that the bias of the resulting linear approximation does not deviate significantly from its average over all keys [19]. More concretely, this is usually interpreted in the following way [15], which we call the *standard key equivalence hypothesis*:

Hypothesis 3 (Standard key equivalence hypothesis). *Consider a non-trivial linear approximation $\mathcal{L} = (\alpha, \beta)$ with bias $\epsilon(k_r)$. Then $|\epsilon(k_r)|$ is independent of the choice of the key:*

$$|\epsilon(k_r)| = 2^{-\kappa} \sum_{k \in \mathbb{F}_2^\kappa} |\epsilon(k)| \qquad \forall k_r.$$

However, for most practically interesting designs (including the AES, Serpent and PRESENT), it has been shown that this strong form of the key equivalence hypothesis does not hold [8, 9]. In the case of key-alternating ciphers (or Feistel ciphers which can be written as such), the contributions of the biases of the individual trails to the bias of the hull for a fixed key k_r can be explicitly computed [8, 9] as:

$$\epsilon(k_r) = \sum_{u_0 = \alpha, u_r = \beta} (-1)^{d_U \oplus U^T K} |\epsilon_U|, \tag{9}$$

with K denoting the key expansion of the key k_r. This is known as the *linear hull*, which causes the trail biases to be added or subtracted depending on the

value of the key [18, 23, 27, 32]. As a consequence, it usually becomes infeasible to compute the exact value of $|\epsilon_r|$ or even its average over all keys.

Since the standard key equivalence hypothesis does not hold in most cases, the question arises *which value of ϵ_r to take for the evaluation of the attack complexity*. For instance, the work [30] fixes the expanded key of PRESENT to zero to estimate ϵ_r. We note, though, that using a different key for this estimation will result in a different value of ϵ_r and hence change the estimated attack complexity.

In order to address this issue, we need to refine the usual key equivalence hypothesis by taking the influence of linear hulls into account.

To this end, we propose to decompose the linear hull into two parts: a signal part (corresponding to the known dominant trails) and a noise part (consisting of the unknown remainder of the hull). The signal part is then sampled for random cipher keys to obtain a more reliable evaluation of the impact of those dominant trails. We then model the noise part statistically to make the estimation of complexity more realistic.

Previous approaches have omitted the influence of the unknown trails completely, and as has been demonstrated [23,27], the influence of the unknown part of the hull can be very significant. Additionally, we average the data complexity estimate over a number of randomly drawn master keys, as opposed to fixing one specific expanded key. This leads to a refined formulation of the key equivalence hypothesis.

6.2 Refined key equivalence hypothesis

Assume that only one or a small number of dominant trails with high contribution to the absolute bias of the hull (α, β) are known. In a linear attack, to recover (part of) the key k_r, this hull has an unknown bias $\epsilon_r(k_r)$, potentially varying from key to key. For each fixed value of the key, the actual value of $\epsilon_r(k_r)$ can be decomposed into the contributions that stem from the biases of the known trails and the biases of the remaining unknown trails in the hull. We define the former to be the *signal* and the latter the *noise*:

Definition 1. *Consider the linear approximation (α, β) over an R-round iterative block cipher and a fixed cipher key k_r. The bias $\epsilon_r(k_r)$ of the hull (α, β) is given by*

$$\epsilon_r(k_r) = \sum_{u_0=\alpha, u_R=\beta} \epsilon_U(k_r),$$

with $\epsilon_U(k_r)$ denoting the bias of the trail U with key k_r. Suppose some t dominant trails $\mathcal{U} = \{U_1, \ldots, U_t\}$ of the hull (α, β) are known. By defining

$$\epsilon_{U_{signal}}(k_r) \stackrel{\text{def}}{=} \sum_{U \in \mathcal{U}} \epsilon_U(k_r) \tag{10}$$

$$\epsilon_{U_{noise}}(k_r) \stackrel{\text{def}}{=} \sum_{(\alpha,\beta) \setminus \mathcal{U}} \epsilon_U(k_r), \tag{11}$$

we obtain a repartitioning of the above sum as follows:

$$\epsilon_r(k_r) = \epsilon_{U_{signal}}(k_r) + \epsilon_{U_{noise}}(k_r). \tag{12}$$

In contrast to our approach, Röck and Nyberg [32] mention the concept of considering a subset of dominant trails, but do not consider the remainder of the hull.

Based on Corollary 2, the noise part $\epsilon_{U_{\text{noise}}}(k_r)$ of the trail contributions can now be modeled to approximately follow a normal distribution $\mathcal{N}(0, 2^{-n-2})$ over the right keys. This leads to our refined key equivalence hypothesis:

Hypothesis 4 (Refined key equivalence hypothesis). *In the setting of Definition 1, the key-dependent bias of a linear approximation in a key-alternating cipher is given by*

$$\begin{aligned}\epsilon(k_r) &= \epsilon_{U_{signal}}(k_r) + \epsilon_{U_{noise}}(k_r) \\ &= \sum_{j=1}^{t} (-1)^{d_{U_j} \oplus U_j^T k_r} |\epsilon_{U_j}| + \mathcal{N}(0, 2^{-n-2}).\end{aligned}$$

Here, d_{U_j} is either 0 or 1, standing for the key-independent part of the sign of the linear trail contribution $|\epsilon_{U_j}|$, while $U_j^T k_r$ deals with the key-dependent part of it.

7 Constructive key equivalence in key-alternating ciphers

This leads to the following algorithm for estimating the data complexity N of a linear attack on an n-bit block cipher using the linear approximation (α, β): We know t trails from the hull and sample $\epsilon_{U_{\text{signal}}}$ over a number of keys by means of (9), each time adding $\epsilon_{U_{\text{noise}}}$ sampled from $\mathcal{N}(0, 2^{-n-2})$. For each tried key, we compute an estimate for N based on this value of ϵ_r. Then the average over all tried keys is taken as the final estimate for N. This procedure is described in Algorithm 7.1.

7.1 Experimenting the signal/noise decomposition

We have performed experiments on SmallPresent-20 to illustrate the effect of the signal/noise decomposition of a linear hull. With a block length of $n = 20$ bits, and an 80-bit key space, it is not feasible to compute the exact distribution or even only the exact average bias of a hull (α, β) over the keys. Since n is small, sampling and averaging over some keys is possible here, but this is not the case anymore for realistic block lengths.

Consider the hull $(\alpha, \beta) = (0x20400, 0x20000)$ over 3 rounds. A branch-and-bound search for trails with $|\epsilon| \geq 2^{-11}$ yields 8 trails from the hull: three with absolute bias $|\epsilon| = 2^{-10}$ and five with $|\epsilon| = 2^{-11}$. Based on this data, the following estimates for the data complexities of a linear attack with $P_S = 0.95$ and varying advantages were computed based on Proposition 2:

Algorithm 7.1 Computation of N using the signal-noise decomposition of the hull for key-alternating ciphers.

Input: Trails $U_j, 1 \leq j \leq t$ from the hull (α, β), their absolute biases $|\epsilon_{U_j}|$, number of keys ℓ to sample.
Input: Block length n, success probability $P_S \geq 2^{-a}$.
Output: Estimate of the data complexity N required for the given parameters.
1: **for** $i = 1, \ldots, \ell$ **do**
2: Select the master key k_i uniformly at random and compute the expanded key.
3: Sample $noise(k_i)$ from $\mathcal{N}(0, 2^{-n-2})$.
4: Compute

$$\begin{aligned}\epsilon(k_i) &= \epsilon_{U_{\text{signal}}}(k_i) + \epsilon_{U_{\text{noise}}}(k_i) \\ &= \sum_{j=1}^{t} (-1)^{d_{U_j} \oplus U_j^T K_i} |\epsilon_{U_j}| + noise(k_i).\end{aligned}$$

5: Compute $N(k_i)$ based on $\epsilon(k_i)$ with Algorithm 5.1.
6: **end for**
7: **return** Average $\overline{N} = \frac{1}{\ell} \sum_{i=1}^{\ell} N(k_i)$.

1. N for $\epsilon_r(k_r)$ of the known trails for one cipher key k_r;
2. N determined with Algorithm 7.1 with $\ell = 200$ keys, but without the noise part;
3. N determined with Algorithm 7.1 with $\ell = 200$ keys;
4. N for an estimation of the hull bias by means of the expected linear probability (ELP), averaged over all keys [28].

Additionally, the actual data complexity was determined experimentally. Each experiment for a certain combination of N and a was averaged over 1000 times to obtain a reliable relation between N and a for this fixed P_S.

The results are depicted in Fig. 5. One observes that summing the trail biases for one key results in a far too optimistic estimation. Averaging the data complexity estimates for the signal trails for 200 keys (but without the noise part) improves the accuracy, but yields an overestimate here. This can be attributed to the impact of two factors: First, the hull must contain more signal trails that are missing in our set of eight trails; and second, the noise impact of the remainder of the hull is not accounted for. Additionally taking the noise into account yields a more realistic estimate. In this specific case though, it is still an overestimate since here obviously the remainder of the hull constructively helps to increase the bias for many keys.

Figure 5 also compares our approach to the estimate based on the ELP, obtained as the sum of the trail ELPs [28]. Note that computing the ELP exactly is infeasible for realistic block ciphers, in contrast to our decomposition approach where only a limited number of dominant trails have to be known.

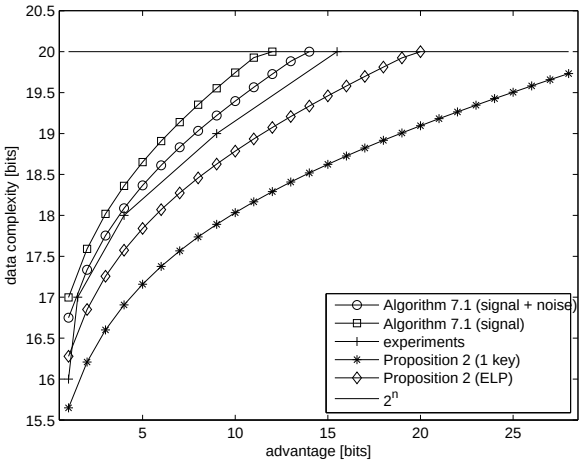


Fig. 5: Theoretical and experimental evaluation of the data complexity with the signal-noise decomposition of Alg. 7.1. Cipher is SmallPresent with 5 rounds, $n = 20, P_S = 0.95$, the signal part contains 8 trails U_j with $2^{-10} \leq |\epsilon_{U_j}| \leq 2^{-11}$. Experimental value of ϵ is $2^{-8.02}$.

8 Conclusions

In this paper, we proposed an approach to improving the accuracy of estimating the data complexity and success probability of Matsui's Algorithm 2.

First, we demonstrated that the standard interpretation of the wrong key randomisation hypothesis in linear cryptanalysis implies a simplification that can result in significant overestimations of the attack efficiency. Our adjusted interpretation results in more precise estimates for the success probability and data complexity of linear attacks. The largest improvements compared to previous results occur in the cases where the adversary attempts to use a linear approximation with a low bias, or to attain a high computational advantage over brute force, or both. These cases are particularly relevant in practice since attacks are usually pushed to the limit by recovering many key bits or covering as many rounds of the cipher as possible.

Second, our new analysis of linear attacks reveals that the success probability is not a monotonous function of the data complexity, and can decrease if more data is used. Somewhat surprisingly, using less data can therefore result in a more powerful attack.

Third, we proposed a technique to refine the usual key equivalence hypothesis by taking the linear hull effect into account.

Finally, all theoretical observations and techniques presented in this paper have been verified by experiments with structurally different small-scale ciphers.

Acknowledgments. The authors would like to thank Vincent Rijmen for fruitful discussions and the anonymous referees for their constructive comments.

References

1. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In: Stinson, D.R., Tavares, S.E. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 2012, pp. 39–56. Springer (2000)
2. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In: Lee, P.J. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 3329, pp. 432–450. Springer (2004)
3. Baignères, T., Vaudenay, S.: The complexity of distinguishing distributions (invited talk). In: Safavi-Naini, R. (ed.) ICITS. Lecture Notes in Computer Science, vol. 5155, pp. 210–222. Springer (2008)
4. Biham, E.: On Matsui's Linear Cryptanalysis. In: De Santis [12], pp. 341–355
5. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. J. Cryptology 4(1), 3–72 (1991)
6. Blondeau, C., Gérard, B., Tillich, J.P.: Accurate estimates of the data complexity and success probability for various cryptanalyses. Des. Codes Cryptography 59(1-3), 3–34 (2011)
7. Bogdanov, A., Tischhauser, E.: On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsuis Algorithm 2. IACR ePrint Archive (2013)
8. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Preneel, B. (ed.) FSE. Lecture Notes in Computer Science, vol. 1008, pp. 275–285. Springer (1994)
9. Daemen, J., Rijmen, V.: The Design of Rijndael: AES – The Advanced Encryption Standard. Springer-Verlag (2002)
10. Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. Tech. Rep. 212, IACR eprint Report 2005/212 (2005), `http://eprint.iacr.org/2005/212`
11. Daemen, J., Rijmen, V.: Probability distributions of correlations and differentials in block ciphers. Journal of Mathematical Cryptology 1(3), 221–242 (2007)
12. De Santis, A. (ed.): Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings, Lecture Notes in Computer Science, vol. 950. Springer (1995)
13. Feistel, H.: Cryptography and computer privacy. Scientific American 228, 15–23 (1973)
14. 3rd Generation Partnership Project: Technical specification group services and system aspects, 3G security, specification of the 3GPP confidentiality and integrity algorithms; document 2: KASUMI specification, v3.1.1 (2001)
15. Harpes, C., Kramer, G.G., Massey, J.L.: A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In: Guillou, L.C., Quisquater, J.J. (eds.) EUROCRYPT. Lecture Notes in Computer Science, vol. 921, pp. 24–38. Springer (1995)
16. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional extension of Matsui's Algorithm 2. In: Dunkelman, O. (ed.) FSE. Lecture Notes in Computer Science, vol. 5665, pp. 209–227. Springer (2009)
17. Hermelin, M., Nyberg, K.: Dependent linear approximations: The algorithm of Biryukov and others revisited. In: Pieprzyk, J. (ed.) CT-RSA. Lecture Notes in Computer Science, vol. 5985, pp. 318–333. Springer (2010)
18. Hermelin, M., Nyberg, K.: Linear cryptanalysis using multiple linear approximations. In: Junod, P., Canteaut, A. (eds.) Advanced Linear Cryptanalysis of Block and Stream Ciphers. IOS Press (2011)

19. Junod, P.: On the complexity of Matsui's attack. In: Vaudenay, S., Youssef, A.M. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 2259, pp. 199–211. Springer (2001)
20. Junod, P.: On the optimality of linear, differential, and sequential distinguishers. In: Biham, E. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 2656, pp. 17–32. Springer (2003)
21. Junod, P., Vaudenay, S.: Optimal key ranking procedures in a statistical cryptanalysis. In: Johansson, T. (ed.) FSE. Lecture Notes in Computer Science, vol. 2887, pp. 235–246. Springer (2003)
22. Leander, G.: Small scale variants of the block cipher PRESENT. Tech. Rep. 143, IACR eprint Report 2010/143 (2010), http://eprint.iacr.org/2010/143
23. Leander, G.: On linear hulls, statistical saturation attacks, PRESENT and a cryptanalysis of PUFFIN. In: Paterson, K.G. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 6632, pp. 303–322. Springer (2011)
24. Ltd, B.D.S.T.C.: Specification of SMS4 (in Chinese) (2006), http://www.oscca.gov.cn/UpFile/200621016423197990.pdf
25. Matsui, M.: Linear cryptoanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993)
26. Matsui, M.: The first experimental cryptanalysis of the Data Encryption Standard. In: Desmedt, Y. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 839, pp. 1–11. Springer (1994)
27. Murphy, S.: The effectiveness of the linear hull effect. Tech. Rep. RHUL-MA-2009-19, Royal Holloway (2009)
28. Nyberg, K.: Linear approximations of block ciphers. In: De Santis [12], pp. 439–444
29. Nyberg, K.: Correlation theorems in cryptanalysis. Discrete Applied Mathematics 111(1-2), 177–188 (2001)
30. Ohkuma, K.: Weak keys of reduced-round PRESENT for linear cryptanalysis. In: Jr., M.J.J., Rijmen, V., Safavi-Naini, R. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 5867, pp. 249–265. Springer (2009)
31. Rivest, R., Robshaw, M., Sidney, R., Yin, Y.L.: The RC6 block cipher. In: First Advanced Encryption Standard (AES) Conference. p. 16 (1998)
32. Röck, A., Nyberg, K.: Exploiting linear hull in Matsui's Algorithm 1. In: The Seventh International Workshop on Coding and Cryptography, WCC (April 2011), to appear
33. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptology 21(1), 131–147 (2008)
34. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES. Lecture Notes in Computer Science, vol. 6917, pp. 342–357. Springer (2011)
35. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit blockcipher CLEFIA (extended abstract). In: Biryukov, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 4593, pp. 181–195. Springer (2007)