Multidimensional Linear Cryptanalysis of Reduced Round Serpent

Miia Hermelin[1], Joo Yeon Cho[1], and Kaisa Nyberg[12]

[1] Helsinki University of Technology
[2] Nokia Research Center, Finland

Abstract. Various authors have previously presented different approaches how to exploit multiple linear approximations to enhance linear cryptanalysis. In this paper we present a new truly multidimensional approach to generalise Matsui's Algorithm 1. We derive the statistical framework for it and show how to calculate multidimensional probability distributions based on correlations of one-dimensional linear approximations. The main advantage is that the assumption about statistical independence of linear approximations can be removed. Then we apply these new techniques to four rounds of the block cipher Serpent and show that the multidimensional approach is more effective in recovering key bits correctly than the previous methods that use a multiple of one-dimensional linear approximations.

1 Introduction

Linear cryptanalysis introduced by Matsui in [1] has become one of the most important cryptanalysis methods for symmetric ciphers. Matsui analysed the DES block cipher using a linear approximation of the known data bits, which holds with a large correlation independently of the key, and presented two ways of exploiting this property: Algorithm 1 which determines one bit from the secret key and Algorithm 2 which recovers a part of the last (or first) round key bits. Originally, only one approximative linear relation was used. In [2], two approximations were used to reduce the amount of data needed for the attack. This idea was developed further by Kaliski and Robshaw in [3], and later by Biryukov, et al., in [4], where the goal was to use several linear approximations simultaneously in order to recover more key bits with equal amount of data. In both [3] and [4] the fundamental assumption was that the approximations are statistically independent. This assumption is hard to verify in practice. The main contribution of this paper is to remove this assumption.

In [5], Baignères, et al., analysed the statistical properties of multidimensional linear approximations without the assumption of statistical independence. They proved that by using multiple approximations, less data is needed to have the same level of test as with only one approximation. However, their target system was a block cipher, which was assumed to have a Markovian property [6]. Consequently, no practical way of building the probability distributions for the purposes of Matsui's Algorithm 1 can be found.

In [7] Englund and Maximov calculated directly the multidimensional probability distribution needed for the distinguisher. However, their calculations become infeasible

for systems with word-size of 64 or more. In this paper, it will be shown how one-dimensional linear approximations can be combined to determine the multidimensional linear approximation and the corresponding probability distribution. The method can be applied to both stream and block ciphers of any word size.

The goal of this paper is to present a key recovery attack by generalising Algorithm 1 to the multidimensional case. This algorithm will be compared with the method suggested by Biryukov, et al., in [4] and the experimental results presented in [8].

The structure of this paper is as follows: In Sect. 2 the notation and the theoretical basics needed in this paper are given. Section 3 starts with showing how linear one-dimensional approximations can be used to make multidimensional linear approximations. Using the results of [5] it is then shown that it is advantageous to use multiple approximations instead of just one. The rest of the Sect. 3 shows how to generalise Matsui's Algorithm 1. Section 4 shows how the method can be applied to the block cipher Serpent. The results will also be compared to those presented in [8], where Biryukov's method was applied to Serpent. Finally, Sect. 5 draws conclusions.

2 Probability Distribution of a Boolean Function

We will denote the space of n-dimensional binary vectors by V_n. The inner product is defined for $a = (a^1, \ldots, a^n), b = (b^1, \ldots, b^n) \in V_n$ as $a \cdot b = a^1 b^1 + \cdots + a^n b^n$, where $+$ is sum modulo 2.

A function $f : V_n \to V_1$ is called a Boolean function. A function $f : V_n \to V_m$ with $f = (f_1, \ldots, f_m)$, where f_i are Boolean functions is called a vector Boolean function of dimension m. A linear Boolean function from $V_n \to V_m$ is represented by an $m \times n$ binary matrix U. The m rows of U are denoted by $u_1, \ldots, u_m$, where each u_i is a binary vector of length n.

A random variable (r.v.) is denoted by boldface, capital letters, e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \ldots$. The abbreviation i.i.d. will mean independent and identically distributed.

Let $\mathbf{Y}$ be a r.v. in V_m, and denote by $p_\eta = \Pr(\mathbf{Y} = \eta)$. Then the probability distribution (p.d.) of $\mathbf{Y}$ is the vector $p = (p_0, \ldots, p_{2^m-1})$. Let $f : V_n \to V_m$ be a vector Boolean function, and let $\mathbf{X}$ be a r.v. in V_n with the 2^n-dimensional uniform distribution vector $\theta_n = 2^{-n}(1, \ldots, 1)$. Then we associate with f a r.v. $\mathbf{Y} = f(\mathbf{X})$ in V_m with a probability distribution $p(f) = (p_0(f), \ldots, p_{2^m-1}(f))$, where $\Pr(f(\mathbf{X}) = \eta) = p_\eta(f)$, $\eta \in V_m$. This p.d. is called the probability distribution of f and is denoted by $p(f)$. We may also abbreviate $p_\eta(f)$ by p_η if the function is clear from the context. Two Boolean functions f and g are called statistically independent if the associated r.v.'s are statistically independent.

The correlation between a binary r.v. $\mathbf{X}$ and zero is defined as $\Pr(\mathbf{X} = 0) - \Pr(\mathbf{X} = 1)$. The correlation of a Boolean function $g : V_n \to V_1$ to zero shall be referred to as the correlation (of g) and is defined as

$$2^{-n}\left(\#\{\,\xi \mid g(\xi) = 0\,\} - \#\{\,\xi \mid g(\xi) = 1\,\}\right) = 2\Pr\left(g(\mathbf{X}) = 0\right) - 1,$$

where $\mathbf{X}$ is uniformly distributed.

Capacity was defined by Biryukov in [4] where they showed that it was inversely proportional to the data complexity of their distinguishing attack. We will now generalise the definition.

Definition 1. *Let $p = (p_0, \ldots, p_M)$ and $q = (q_0, \ldots, q_M)$ be two p.d.'s. Their (mutual) capacity is then*

$$C(p, q) = \sum_{\eta=0}^{M} \frac{(p_\eta - q_\eta)^2}{q_\eta}. \tag{1}$$

If $M = 2^m - 1$ and $q = \theta_m$ is uniform then $C(p, \theta_m) = 2^m \|p - \theta_m\|_2^2$ will be called the capacity of p and we will denote it by $C(p)$. It can also be called the Squared Euclidean Imbalance [5].

In the next section, we will see that the generalised capacity will be inversely proportional to the data complexity of a multidimensional linear distinguisher.

3 Multidimensional Approximation of Boolean Functions

3.1 From One-Dimensional Probability Distributions to Multiple Dimensions

Let $f : V_\ell \to V_n$ be a vector Boolean function and binary vectors $w_i \in V_n$ and $u_i \in V_\ell$, $i = 1, 2, \ldots, , m$ be linear masks such that the paired masks (u_i, w_i) are linearly independent. Let us define functions g_i by

$$g_i(\xi) \coloneqq w_i \cdot f(\xi) + u_i \cdot \xi, \tag{2}$$

and assume g_i's have correlations ρ_i, $i = 1, 2, \ldots, m$. We will call these correlations the base-correlations, and the corresponding linear approximations of f the base-approximations. We want to find the p.d. of the m-dimensional linear expression

$$g(\xi) \coloneqq W f(\xi) + U\xi,$$

where $W = (w_1, \ldots, w_m)$, $U = (u_1, \ldots, u_m)$ and $g = (g_1, \ldots, g_m)$. Let the p.d. of g be p. Assume that we have the correlations $\rho(a)$ of all the linear mappings $a \cdot g$ of g, $a \in V_m$. If $e_i = (0 \ldots 010 \ldots 0)$ with 1 at the ith position, then $\rho(e_i) = \rho_i, i = 1, \ldots, m$. We will call the correlations $\rho(a), a \neq e_i$ the combined correlations of f and the corresponding approximations the combined approximations. Recall the following lemma from [9].

Lemma 1. *Let $g = (g_1, \ldots, g_m) : V_n \to V_m$ be a vector-valued Boolean function and p it's p.d. Then*

$$2^n p_\eta = 2^{-m} \sum_{a \in V_m} \sum_{\xi \in V_n} (-1)^{a \cdot (g(\xi) + \eta)}.$$

The correlations $\rho(a)$ can be written as

$$\rho(a) = 2^{-n} \sum_{\xi \in V_n} (-1)^{a \cdot g(\xi)}.$$

Using this and Lemma 1 we get the following corollary that connects p and the one-dimensional correlations $\rho(a)$:

Corollary 1. *Let $g : V_n \to V_m$ be a Boolean function with p.d. p and one-dimensional correlations $\rho(a)$ of $a \cdot g$. Then*

$$p_\eta = 2^{-m} \sum_{a \in V_m} (-1)^{a \cdot \eta} \rho(a).$$

The following corollary is obtained using Parseval's theorem. An equivalent form of it can be found in [5], where the proof was based on the inverse Walsh-Hadamard transform of the deviations ϵ_η from the uniform distribution, $\epsilon_\eta = p_\eta - 2^m$.

Corollary 2. *Let g be the Boolean function defined as previously with p.d. p. Then*

$$C(p) = 2^m \sum_\eta \epsilon_\eta^2 = \sum_{a \neq 0} \rho(a)^2.$$

We will need this equality in the next section where we study how linear distinguishing is done in multiple dimensions.

3.2 One vs. Multidimensional Linear Distinguishers

In this section we will present the general statistical framework of multidimensional approximation.

The theory of hypothesis testing can be found for example in [10]. Here we will restrict to the most essential parts of the theory. Assume we have two p.d's p and q, $q \neq p$ and consider two hypotheses: H_0 states that the experimental data $\mathbf{z}^N$ of N words is derived from p and H_1 states that $\mathbf{z}^N$ is derived from q.

In the one-dimensional case, we have a linear approximation such as (2). Let ρ be the correlation of the approximation. The number of bits N_1 needed to distinguish $\mathbf{z}^N$ from a random sequence is λ/ρ^2, where λ depends on the level and the power of the test. It was already noted in [1] that the data complexity N_1 is proportional to $1/\rho^2$. For proof, see [11]. Note that the bias used in [1] is the correlation divided by two.

The data complexity of the attack in [4] using multiple linear approximations, was shown to be proportional to $N_{\text{s.i.}}$, where

$$N_{\text{s.i.}} = \frac{1}{\sum_{i=1}^m \rho_i^2} = \frac{1}{\vec{c}^2}, \tag{3}$$

and $\vec{c}^2$ is the capacity as defined in [4]. This means a significant improvement in data complexity, but relies on the assumption that the base approximations are statistically independent.

Let us next study the case of multiple approximations without the assumption of statistical independence. The log-likelihood ratio (LLR) is defined as follows:

$$l(\mathbf{z}^N) = \sum_{\eta=0}^{M} N(\eta) \log \frac{p_\eta}{q_\eta}, \tag{4}$$

where p and q are defined as in Definition 1 and $N(\eta)$ is the experimental frequency of the value η in $\mathbf{z}^N$. The LLR was used as the distinguisher in [5] to proof the following theorem.

Theorem 1. *Let us have a hypothesis testing problem with H_0 stating that the data $\mathbf{z}^N$ is drawn i.i.d. from p.d. p and H_1 stating that the data is drawn from $q \neq p$. Assume that the p.d's are close to each other:*

$$|q_\eta - p_\eta| \ll q_\eta, \text{ for all } \eta. \tag{5}$$

Then the amount of data needed for distinguishing the hypotheses is proportional to

$$N = \frac{\lambda}{C(p,q)}, \tag{6}$$

where λ depends on the level and the power of the test.

If we want to distinguish a distribution of some data related to a cipher from that of a truly random source we will use the previous hypothesis test with q as the ciphers p.d. and p as the uniform distribution. Using (2) we will see that $N_{\text{s.i.}}$ given by (3) is actually greater than the true amount of data needed for $m \leq n$ linear approximations, since by using Corollary 2, the latter is proportional to

$$N_m = \frac{\lambda}{C(q)} = \frac{\lambda}{\sum_{a \neq 0} \rho(a)^2}.$$

In an "optimal case" we can make an m-dimensional approximation where all the correlations $\rho(a)$ are (in absolute) value equal to the maximal one-dimensional correlations. If N_1 is the data requirement for one approximation, then $N_m = N_1/(2^m - 1)$. On the other hand, if only a single one-dimensional approximation has a large correlation, then $N_m \approx N_1$ and it is not useful to use multiple approximations.

In [5] Markovian block ciphers were analysed using multidimensional distinguishers on the probability distributions related to the Markovian transition probabilities averaged over the keys. Hence, their main goal was to improve the efficiency of Algorithm 2. Next, we will generalise Matsui's Algorithm 1 to the multidimensional case. In the practical experiments we use Corollary 1 to determine the related multidimensional probability distributions from the correlations of the one-dimensional linear approximations.

3.3 Key Recovery Attack

We will show how to find m key bits of the key K using a multidimensional version of Algorithm 1. Let $\mathbf{X}$ be a uniformly distributed r.v. and $\mathbf{Y} = f(\mathbf{X})$, where $(\mathbf{X}, \mathbf{Y})$ is a plaintext-ciphertext pair. We consider the r.v.

$$U\mathbf{X} \oplus W\mathbf{Y} \oplus VK, \tag{7}$$

with a fixed unknown key K, and use p to denote the r.v.'s p.d. Here $U = (u_1, \ldots, u_m)$, $W = (w_1, \ldots, w_m)$ and $V = (v_1, \ldots, v_m)$ are some maskmatrices. This approximation can be generated from linearly independent one-dimensional approximations with correlations $\rho_1, \ldots, \rho_m$ using Corollary 1 (assuming that we are also given the combined correlations). The linear mapping V divides the key space to equivalence classes $k = VK \in \mathcal{Z}$.

The bits $k_i = v_i \cdot K$ are called the parity bits. For each k the expected p.d. p^k of $\mathbf{Z}^k = U\mathbf{X} \oplus W\mathbf{Y}$ for the distribution originating from the empirical data will be some permutation of p determined by the key (class) k. For the purposes of this study, we assume that all the keys give distinct permutations such that $p^k \neq p^j$, if $k \neq j$.

Biryukov's attack introduced in [4] uses $m' \geq m$ linear approximations to select the correct key class from $\mathcal{Z}$. It has three phases: distillation, analysis and search phases. They can be described as follows:

1. **Distillation phase** Obtain N plaintext-ciphertext pairs (x_t, y_t) and calculate the empirical correlation vector $\hat{\mathbf{c}} = (\hat{\rho_1}, \ldots, \hat{\rho_{m'}})$.
2. **Analysis phase** For each key class k, give the key a rank d_k and make a sorted list of the keys with smallest d_k at the top of the list.
3. **Search phase** Run through the list and try all keys contained in the equivalence classes until the correct key is found.

The statistic used is $d_k = \|\hat{\mathbf{c}} - \mathbf{c}_k\|_2$, where $\mathbf{c}_k = ((-1)^{k_1}\rho_1, \ldots, (-1)^{k_{m'}}\rho_{m'})$, a vector consisting of the theoretical correlations and the parity bits of k. In addition a measure "gain" was defined to analyze the success of the method taking into account the time complexity of the search phase.

The purpose of our multidimensional approach is to improve the distillation phase in theory and in practice. In order to compare the distillation phase of Biryukov's and our multidimensional method, we discuss a plain multiple linear cryptanalysis method (the plain method), which is similar to the Biryukov's method but without the grading of the key candidates. We measure the success of the plain method and our method using the probability P_{OK}, which is the probability that the right key is at the top of the list. We assume that the plain method uses m linearly and statistical independent linear approximations and recovers m bits of the key based on the deviations d_k. Let q be the experimental p.d. constructed from the data. Our method uses the m base approximations, $2^m - m - 1$ combined approximations and the Kullback-Leibler distance between q and p^k. The Kullback-Leibler distance is used in measuring the difference between p.d.'s. It can be seen to be related to the LLR:

Definition 2. *The relative entropy or the Kullback-Leibler distance between two distributions $p = (p_0, \ldots, p_M)$ and $q = (q_0, \ldots, q_M)$ is defined as*

$$D(q\|p) = \sum_{\eta=0}^{M} q_\eta \log \frac{q_\eta}{p_\eta}. \tag{8}$$

Then, in the analysis phase, instead of a grading problem we face the following multiple hypothesis testing problem.

Theorem 2. *Let us have an $|\mathcal{Z}|$-ary hypothesis problem, with $|\mathcal{Z}|$ hypotheses H_k stating that the data originates from p^k, where $k \in \mathcal{Z}$ corresponds to the key. The hypothesis for which the Kullback-Leibler distance $D(q\|p^k)$ is smallest is selected. Given some success probability P_{OK}, the lower bound N_{key} for the amount of data needed to give the smallest value of the statistic when the correct key is used, is given by*

$$N_{key} \approx \frac{4 \log_2 |\mathcal{Z}|}{\min_{j \neq 0} C(p^0, p^j)}. \tag{9}$$

Proof. For each key k we must distinguish p^k from p^j, for all $j \neq k$. Using Proposition 3 in [5], the probability that we choose j when k is true is

$$\Pr(H_j|H_k) = \Phi\left(-\sqrt{N_{kj}C(p^k, p^j)/2}\right),$$

where Φ is the distribution function of the normed normal distribution. Let the probability of successfully distinguishing H_k from all the other hypotheses be P_{OK}. Then $P_{\mathrm{OK}} = \prod_{j\neq i}(1 - \Pr(H_j|H_k))$. Assume $N_{kj}C(p^k, p^j) \gg 1$ for all $j \neq k$. Then

$$P_{\mathrm{OK}} \approx \exp\left(-\frac{1}{\sqrt{2\pi}}\sum_{j\neq k} e^{-N_{kj}C(p^k,p^j)/4}\right). \tag{10}$$

Let $N_k = \max_j N_{kj}$. Since we have to collect the amount of N_k for at least one test with k we can use the same amount for all the tests. On the other hand, let us define $c_k = \min_j C(p^k, p^j)$. Replacing the capacities with c_k, N_k must be increased to get the required success probability. We get a lower bound for N_k by solving N_k from (10)

$$N_k \approx \frac{4\log_2|\mathcal{Z}| - 4\ln(\sqrt{2\pi}\ln P_{\mathrm{OK}})}{c_k}.$$

Since we do not know which k is the right key, we have to choose $N = \max_k N_k$ to be able to find the right key. Since p^j's are each others' permutations, we have $C(p^k, p^j) = C(p^0, p^{k+j})$. But then $c_k = \min_{s\neq 0} C(p^0, p^s) = c_0$ which is independent of k and (9) follows. □

Note that we need the assumption that $p^i \neq p^j$ to ensure that $\min_j C(p^0, p^j) \neq 0$. In [5] a similar formula was derived for the purposes of Algorithm 2 to distinguish the distribution related to the correct key from the, presumably uniform, distribution related to a wrong key. Formula (9) gives an estimate how much data is needed to reliably determine which of the $|\mathcal{Z}|$ distributions gives the best fit with the empirical data. Exactly the same calculations can be done to the Biryukov's statistic with the help of proof of Theorem 1 in [4]. Then the data complexity of the plain attack is proportional to N_{plain} which is given by the formula

$$N_{\mathrm{plain}} = \frac{8\log_2|\mathcal{Z}|}{\min_{j,k j\neq k}\left\|\mathbf{c}_k - \mathbf{c}_j\right\|_2} = \frac{2\log_2|\mathcal{Z}|}{\min_j \rho_j^2}.$$

Since the denominator in N_{key} is usually much larger than in N_{plain}, we have $N_{\mathrm{plain}} > N_{\mathrm{key}}$. Especially, if the combined correlations are large, the advantage is significant.

The data, time and memory complexities of distillation and analysis phases have been given in Table 1. The main difference in the complexities between our method and the plain method is due to the fact that our method uses the full m-dimensional distributions and needs to compute 2^m empirical values from the data, while the plain method determines only the m entries of the empirical correlation vector $\hat{c}$.

The main improvements introduced by Biryukov, et al., in [4] is the implementation of the key ranking procedure and its statistical treatment using the concepts of capacity

Table 1. Complexities of Algorithm 1 for plain, Biryukov's and our multidimensional method

	Distillation			Analysis		
	Plain	Biryukov	Our method	Plain	Biryukov	Our method
Data	$O(N_{\text{plain}})$	$O(N_{\text{s.i.}})$	$O(N_{\text{key}})$	-	-	-
Time	$O(mN_{\text{plain}})$	$O(m'N_{\text{s.i.}})$	$O(2^m N_{\text{key}})$	$O(m\|\mathcal{Z}\|)$	$O(m'\|\mathcal{Z}\|)$	$O(2^m\|\mathcal{Z}\|)$
Memory	$O(m)$	$O(m')$	$O(2^m)$	$O(\|\mathcal{Z}\|)$	$O(\|\mathcal{Z}\|)$	$O(\|\mathcal{Z}\|)$

and gain which helps to reduce the lower bound of the data complexity to $N_{s.i.}$. For additional improvement of the practical performance of their method, Biryukov, et al., extend the base set of the m linearly (and presumably also statistically) independent approximations with combined approximations. This extension was justified in [4] by informal arguments and assuming that the linear approximations also in the extended set are statistically independent. Statistical independence of linear approximations is difficult to verify in practice. One method would be to evaluate experimentally the correlations of all linear combinations of the approximations and use Piling-Up Lemma [1] to check for statistical independence. In practical applications of the method of Biryukov, et al., in [4] and [8], statistical independence was not verified. Let us denote by m' the number of approximations used, where $m \leq m' < 2^m$. The resulting complexities are given in Table 1. Selection of m is always a trade-off between complexity and maximising the capacity. Typical values for m and m' are, for example, $m = 10$ and $m' = 86$ in [4] and $m = 10$ and $m' = 64$ in [8]. Also often $|\mathcal{Z}| = 2^m$.

In the next section we will compare Biryukov's method and our method in practice using small experiments on the four-round Serpent. The same "test-bed" was previously used by Collard, et al., in [8] to carry out experiments of Biryukov's method. When comparing our results with their results we can see that similar advantage in practical performance can be achieved using our method and the Biryukov's with $m' > m$, compared to the plain method with just m approximations. In addition, our method has a few important advantages over the Biryukov's. We provide sound theoretical justification for using combined approximations. More importantly, no assumption about statistical independence of the approximations is needed.

4 Multidimensional Linear Attack on 4-Round Serpent

Serpent [12] is one of the block ciphers proposed to the Advanced Encryption Standard (AES) competition. It was selected to be among the five finalists [13]. The best known linear approximation of 9-round Serpent was reported by Biham et al. in FSE 2001 [14]. Recently, experimental results on multiple linear cryptanalysis of 4-round Serpent were presented by Collard, et al., in [8]. In this section, we will apply the multidimensional linear attack to the reduced round Serpent and compare our results to the previous attacks presented in [8].

4.1 Multidimensional linear attack on 4-round Serpent

In [8], authors used maximum $m' = 64$ linear approximations to perform Matsui's Algorithm 1 type -attack on 4-round Serpent. The detailed description of approximations can be found in [15]. Those 64 linear approximations used in the attack are not linearly independent. Hence, strictly speaking, the attack in [8] is not consistent with the technique in [4] which assumes that multiple approximations are statistically independent. On the other hand, our attack does not require such a statistical assumption. One can exploit as many approximations with non-negligible correlations as possible for recovering the targeted key bits without such restriction.

In experiments, we chose a 4-round linear trail (from S_4 to S_7) that was used in [8]. We picked up $m = 10$ linearly independent approximations $L_0, ..., L_9$ which can be used to recover 10 bits of the first round key. [3] The input and output masks of the approximations used in our attack are listed in Table 2. Let us denote L_i as follows:

Table 2. Input and output masks used for the multidimensional linear attack

	index	mask = (MSB, . . . , LSB)
input mask	u_0	(0x70000000, 0x00000000, 0x00000000, 0x07000900)
	u_1	(0x70000000, 0x00000000, 0x00000000, 0x07000B00)
	u_2	(0x70000000, 0x00000000, 0x00000000, 0x0B000900)
	u_3	(0xB0000000, 0x00000000, 0x00000000, 0x07000900)
	u_4	(0x70000000, 0x00000000, 0x00000000, 0x07000500)
	u_5	(0x70000000, 0x00000000, 0x00000000, 0x07000600)
	u_6	(0x70000000, 0x00000000, 0x00000000, 0x07000C00)
	u_7	(0x70000000, 0x00000000, 0x00000000, 0x01000900)
	u_8	(0x70000000, 0x00000000, 0x00000000, 0x0A000900)
	u_9	(0xB0000000, 0x00000000, 0x00000000, 0x03000B00)
output mask	w	(0x00007000, 0x03000000, 0x00000000, 0x00000000)

$$u_i \cdot P + w \cdot C = v_i \cdot K \quad i = 0, \ldots, 9 \tag{11}$$

where u_i, w and v_i stand for the input mask, output mask and the key mask, respectively and P, C and K represent the plaintext, ciphertext and the key, respectively. Note that the output mask w is identical for all the approximations since this experiment targets the first round key, not the last one.

Let $Q = \text{span}\{L_0, ..., L_9\}$ such that Q is a set of approximations generated by the 10 base approximations L_i. Then, $|Q| = 2^{10} - 1$. Note that the 64 linear approximations used in [8] form a subset of Q.

[3] We can find maximum 12 linear appr. to recover 12 bits of the first round key from this linear trail. However, we targeted only 10 bits of the key for direct comparison of the performance between the Biryukov's attack and multidimensional attack.

Our experiments were performed in two ways: In the first experiment, we used all the linear approximations of the set Q. Among $2^{10} - 1$ linear approximations of the Q, we found that 200 of them held with non-negligible correlations, as listed in Table 3. The correlations of the approximations were calculated by the Piling-up lemma [1]. We

Table 3. Correlations of approximations

correlation	# of approximations	
	64 appr.	10 base appr., 200 non-negligible
2^{-11}	8	8
2^{-12}	56	64
2^{-13}	0	128

note that their real correlations can be different from calculated ones due to the effect of correlations of other linear trails using the same input and output masks. However, we assume that the theoretical correlations of the approximations are close to the calculated correlations.

In the second experiment, we generated from $L_0, ..., L_9$ the 64 linear approximations which were the same as those used in [8] and used them in our method while approximating the rest of the combined correlations to be zero. In this manner we get a rougher approximation of the full 10-dimensional p.d. than with using 200 approximations. The purpose of this experiment was to compare the performance of the Biryukov's attack to that of our attack when the same approximations are exploited in both attacks.

For comparison, we applied both the Biryukov's and our method to the 4-Round Serpent and measured their gains by experiment so that we could compare our method with the results in [8]. It was already noted in [8] that the plain method (using m approximations) gives poorer results than the Biryukov's method (using $m' > m$ approximations). No explanation was given to this heuristics in [4] or [8]. Following the theory of the previous sections this heuristic can be justified: Increasing m' makes the Biryukov's method approximate the real multidimensional method. However, since the LLR is the optimal statistic, the Biryukov's method cannot perform better than our method even when $m' = 2^m - 1$.

According to Lemma 1 in [4], the key class k is determined by searching for the minimum Euclidean distance $||\hat{\mathbf{c}} - \mathbf{c}_k||_2$, where $\hat{\mathbf{c}} = (\hat{\rho_1}, \ldots, \hat{\rho_{10}})$ is the estimated correlation of ten approximations. On the other hand, in our attack, we measure the empirical probability distributions q of multiple approximations and determine the key class k by searching for the minimum Kullback-Leibler distance $D(q||p^k)$, where p^k is some permutation of the theoretical probability distribution p. The p.d. p is computed by Corollary 1 using theoretical correlations of one-dimensional approximations. The p.d. q could be calculated in the same way by using the experimental correlations but in this work it was constructed directly using 2^m counters.

We performed the experiments repeatedly 100 times and obtained the average gain of each method. We used a different 128-bit key that was randomly selected each time.

The results are displayed in Fig. 1. For comparison, the gain γ of the attack was measured using the formula which was introduced in [4] as follows

$$\gamma = -\log_2 \frac{2 \cdot M - 1}{2^{10}}$$

In Fig. 1, the multidimensional attack using 10 linearly independent approximations with full span (200 non-negligible approximations) reaches the full gain at around 2^{22} texts. Compared to this result, Biryukov's attack shows that the gain of the attack is saturated with around 2^{23} texts. Hence, this experiment shows that our method requires less data to get the same accuracy as Biryukov's method. The plain method with $m = 10$ approximations would give even weaker results not reaching the maximum gain until with about 2^{26} texts, see Fig. 5 of [8].

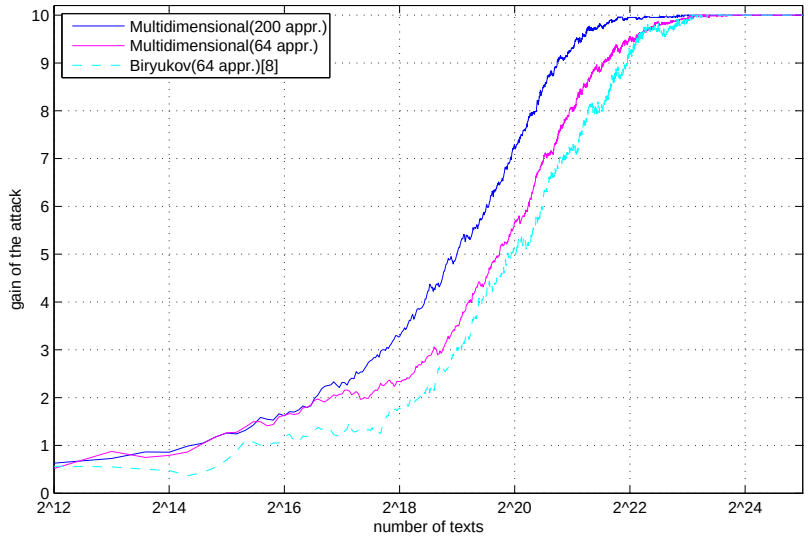

Fig. 1. Comparison of the gain of the different attacks using multiple linear approximations

5 Conclusions

In this paper we investigated a few different approaches presented in recent years on linear cryptanalysis using multiple approximations. We used the statistical theory presented in [5] and developed a new multidimensional cryptanalysis attack. For this purpose, we also showed how to construct multidimensional linear approximations from

one-dimensional approximations. The main advantage of the new method is that the assumption on statistical independence of the linear approximations can be removed.

We also applied our method to the 4-round version of block cipher Serpent that was studied in [8] using Biryukov's method [4]. We studied the cases of 10 linear approximations, showed how to make multidimensional approximations from them and measured the success of recovering 10 key parity bits.

We also saw in Table 3 examples where the combined approximations had correlations of the same magnitude as the base approximations. This demonstrates that the assumption about statistical independence between the base approximations needed in Biryukov's method used in [8] does not hold. The theoretical framework presented in this paper removes the need of this assumption.

References

1. Matsui, M.: Linear cryptanalysis method for DES cipher. In: EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology, Secaucus, NJ, USA, Springer-VerlagNew York, Inc. (1993) 386–397
2. Matsui, M.: The First Experimental Cryptanalysis of the Data Encryption Standard. In: Advances in Cryptology - CRYPTO '94: 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1994. Proceedings. Volume 839 of Lecture Notes in Computer Science. (1994) 1–11
3. Burton S. Kaliski, J., Robshaw, M.J.B.: Linear Cryptanalysis Using Multiple Approximations. In: CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (1994) 26–39
4. Biryukov, A., Cannire, C.D., Quisquater, M.: On Multiple Linear Approximations. In: Advances in Cryptology - CRYPTO 2004. Volume 3152 of Lecture Notes in Computer Science., Springer-Verlag (2004) 1–22
5. Baignères, T., Junod, P., Vaudenay, S.: How Far Can We Go Beyond Linear Cryptanalysis? In: Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security. Volume 3329 of Lecture Notes in Computer Science. (2004) 432–450
6. Wagner, D.: Towards a unifying view of block cipher cryptanalysis. In B. Roy, W.M., ed.: Fast Software Encryption - FSE'04. Volume 3017 of Lecture Notes in Computer Science. (2004) 16–33
7. Englund, H., Maximov, A.: Attack the Dragon. In: Progress in Cryptology - INDOCRYPT 2005. Volume 3797 of Lecture Notes in Computer Science. (2005) 130–142
8. Collard, B., Standaert, F.X., Quisquater, J.J.: Experiments on the Multiple Linear Cryptanalysis of Reduced Round Serpent. In: Proceedings of FSE 2008 (to appear). Lecture Notes in Computer Science (2008)
9. Nyberg, K., Hermelin, M.: Multidimensional Walsh Transform and a Characterization of Bent Functions. In Tor Helleseth, P.V.K., Ytrehus, O., eds.: Proceedings of the 2007 IEEE Information Theory Workshop on Information Theory for Wireless Networks. IEEE (2007) 83–86
10. Cover, T.M., Thomas, J.A.: Elements of Information Theory. 2nd edn. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience (2006)
11. Junod, P.: On the Complexity of Matsui's Attack. Lecture Notes in Computer Science **2259** (2001) 199–211
12. Anderson, R., Biham, E., Knudsen, L.: Serpent: A Proposal for the Advanced Encryption Standard. In: First Advanced Encryption Standard (AES) conference. (1998)

13. NIST: A request for Candidate Algorithm Nominations for the Advanced Encryption Standard AES. http://csrc.nist.gov/archive/aes/index2.html (1997)
14. Biham, E., Dunkelman, O., Keller, N.: Linear Cryptanalysis of Reduced Round Serpent. In: FSE. (2001) 16–27
15. Collard, B., Standaert, F., Quisquater, J. http://www.dice.ucl.ac.be/fstandae/PUBLIS/50b.zip (2008)

A Brief Description of Serpent Algorithm

We use the notation of [12]. Each intermediate value of round i is denoted by $\hat{B}_i$ (a 128-bit value). Each $\hat{B}_i$ is treated as four 32-bit words X_0, X_1, X_2, X_3 where bit j of X_i is bit $4 * i + j$ of the $\hat{B}_i$. Serpent has a set of eight 4-bit to 4-bit S Boxes $S_0, \ldots, S_7$ and a 128-bit to 128-bit linear transformation LT. Each round function R_i uses a single S-box 32 times in parallel.

Serpent ciphering algorithm is formally described as follows.

$$\hat{B_0} = P \quad \hat{B_{i+1}} = R_i(\hat{B_i}) \quad C = B_{32},$$

where

$$R_i(X) = LT(\hat{S}_i(X \oplus \hat{K}_i)), \quad i = 0, \ldots, 30$$
$$R_i(X) = \hat{S}_i(X \oplus \hat{K}_i) \oplus \hat{K}_{32}, \quad i = 31.$$

The linear transformation LT is described as follows.

$$\begin{aligned} X_0, X_1, X_2, X_3 &= S_i(B_i \oplus K_i) \\ X_0 &= X_0 \lll 12 \\ X_2 &= X_2 \lll 3 \\ X_1 &= X_1 \oplus X_0 \oplus X_2 \\ X_3 &= X_3 \oplus X_2 \oplus (X_0 \lll 3) \\ X_1 &= X_1 \lll 1 \\ X_3 &= X_3 \lll 7 \\ X_0 &= X_0 \oplus X_1 \oplus X_3 \\ X_2 &= X_2 \oplus X_3 \oplus (X_1 \lll 7) \\ X_0 &= X_0 \lll 5 \\ X_2 &= X_2 \lll 22 \\ B_{i+1} &= X_0, X_1, X_2, X_3 \end{aligned}$$

The detailed description of Serpent can be found in [12].