An All-In-One Approach to Differential Cryptanalysis for Small Block Ciphers

Martin R. Albrecht, Gregor Leander

An All-In-One Approach to Differential Cryptanalysis for Small Block Ciphers*

Martin R. Albrecht[1] and Gregor Leander[2]

[1] INRIA, Paris-Rocquencourt Center, POLSYS Project
UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France
[2] DTU Mathematics, Department of Mathematics, Technical University of Denmark,
2800 Kgs. Lyngby, Denmark
malb@lip6.fr, G.Leander@mat.dtu.dk

Abstract. We present a framework that unifies several standard differential techniques. This unified view allows us to consider many, potentially all, output differences for a given input difference and to combine the information derived from them in an optimal way. We then propose a new attack that implicitly mounts several standard, truncated, impossible, improbable and possible future variants of differential attacks in parallel and hence allows to significantly improve upon known differential attacks using the same input difference. To demonstrate the viability of our techniques, we apply them to KATAN-32. In particular, our attack allows us to break 115 rounds of KATAN-32, which is 37 rounds more than previous work. For this, our attack exploits the non-uniformity of the difference distribution after 91 rounds which is 20 rounds more than the previously best known differential characteristic. Since our results still cover less than 1/2 of the cipher, they further strengthen our confidence in KATAN-32's resistance against differential attacks.

Keywords. symmetric cryptography, block cipher, differential attack

1 Introduction

Block ciphers are fundamental building blocks of modern cryptography and some of the best understood objects in the area of symmetric cryptography. Compared to, say, stream ciphers and hash functions, the design of a secure block cipher can rely on many established design principles to achieve security against all known attacks; most prominently linear and differential attacks.

* An extended abstract of this work will appear in the proceedings of SAC 2012

However, designing a secure block cipher that, at the same time, is very efficient (in hardware) is still challenging. In particular, lightweight cryptography which recently received considerable attention from the cryptographic community calls for block ciphers that can be efficiently implemented even in very resource constrained devices. Designing secure ciphers for such tiny devices – e.g., RFID tags or sensor networks – requires, on the one hand, innovative design strategies and, on the other hand, perhaps compromises in the security level. One such constraint is the block size used in block ciphers. As the block size, along with the key size, greatly influences the required circuit size, block ciphers tailored to be implemented in small devices have a strong tendency to feature smaller block sizes compared to modern block ciphers mainly focusing on software such as the AES. While modern block ciphers focusing on software usually have a block size of no less than 128 bits, most ciphers designed for efficient implementations in hardware have block sizes of 64 bits or less (see for example PRESENT [7] or HIGHT [11]). A block cipher with a particular small block size of 32-bit is KATAN-32 [9] presented at CHES 2009.

Block ciphers with very small block sizes have some interesting characteristics. From the point of view of the attacker, when using the block cipher in counter mode, it is possible to distinguish the output from a random sequences faster. Similarly, an attacker can build a complete code book faster and time-memory tradeoffs are a greater concern. From the perspective of the designer, most statistical attacks like differential or linear cryptanalysis seem at first glance to become more difficult as the amount of data available to the attacker is much more restricted.

Finally, from a theoretical point of view, small block sizes provide the opportunity to understand well-established attacks better since computations involving the entire code-book are feasible. In particular, for differential cryptanalysis, it becomes feasible to compute the exact expected probabilities for many (sometimes all) differentials. This data then allows to study the behaviour of (classical) differential cryptanalysis and related techniques more precisely.

Yet, it is not obvious a priori how to provide an optimal unified view on these differentials even if this data is available. To provide an answer to this question, this work investigates the probability distribution of output differences under one (or many) input difference and provides an optimal way to use the non-uniform distribution of differences in an attack.

1.1 Prior Work

Differential cryptanalysis was first proposed by Biham and Shamir [3] and since became one of the most prominent tools in the analysis of block ciphers. Many improvements and extensions have been proposed in the past, we mention some of the most influential ones. Knudsen [14] and later Biham, Biryukov and Shamir [2] proposed to use differentials with zero probability, that is *impossible differential* attacks. Based on the work of Lai [16] *High-order differentials* were introduced in [15] and are most effective against ciphers where the algebraic degree can be limited. *Truncated differentials*, first mentioned in [15] can be seen as a collection of differentials and in some cases allow to push differential attacks one or two rounds further. *Boomerang attacks* can be viewed as special cases of second order differentials and are most efficient when the probability of any differential drops rapidly with an increasing number of rounds. Recently, *improbable differentials* have been suggested [21] as a natural extension of impossible differentials and have been successfully applied to the block cipher CLEFIA. Also recently, differential cryptanalysis was extended to *multi-differential cryptanalysis* in [5]. Finally, our application of the log-likelihood can be seen in the framework of [20].

1.2 Our Contribution

Abstractly, differential cryptanalysis exposes a non-uniform distribution of output differences given one (or several) input differences. This is also the point of view from which our investigation sets out. Phrased in these terms, recovering key information using differential techniques becomes the task of distinguishing between distributions, one for the right key and one for the wrong keys. However, usually the attacker does not have access to a full description of these distributions. In standard differential cryptanalysis only one output difference is considered and usually the probability of the best differential characteristic is considered in place of the probability of the output differential. Furthermore, for wrong keys it is assumed that the distribution is uniform.

In comparison the advantage of an attacker when dealing with small block-size ciphers become apparent. The attacker has, under mild assumptions, the ability to compute the parameters of those distributions precisely. Thus, the task is no longer to distinguish (essentially) unknown

distributions, but distributions which are known completely. In particular, the usual hypotheses that wrong keys result in random permutations can be lifted. To this end, we first introduce a model to study and distinguish these distributions. As an important side effect, our framework unifies and generalises standard differential attacks, impossible differentials, improbable differentials and truncated differentials into one attack framework. Since our framework considers the distribution of all output differences it captures all techniques which exploit statistically significant subspaces of the output space.

We then propose a new attack based on this model that implicitly mounts several standard, truncated, impossible, improbable and possible future variants of differential attacks in parallel and hence allows to significantly improve upon known differential attacks using the same input difference. We stress that these "parallel applications" of various differential attacks are such that they are strictly better than those attacks considered independently. To demonstrate the viability of our model and attack, we apply our attack to two ciphers with small block sizes: the toy-cipher SmallPresent[4] and KATAN-32. For KATAN-32 we present the best known differential attack.[1] In particular, our attack allows us to break 115 rounds of KATAN-32, which is 37 rounds more than previous work [13]. For this, our attack exploits the non-uniformity of the difference distribution after 91 rounds which is 20 rounds more than the previously best known differential characteristic. Since our results takes into account several standard techniques and still cover less than 1/2 of the cipher, they further strengthen our confidence in KATAN-32's resistance against differential attacks. For completeness, we also like to mention a recent preprint [12] using a meet-in-the-middle variant to recover the key for KATAN (slightly) faster than exhaustive search.

Furthermore, our model allows to combine many *input-* and output-differences which allows to reduce the data complexity compared to previous works significantly. This is mainly due to the fact that our approach almost naturally provides the optimal way of combining information from several input and output differences. This is the major difference between our work and [5]. Finally, we also discuss variants and possible extensions to ciphers with larger block sizes.

[1] Our attack is also the best known differential attack on SmallPresent[4].

We highlight that similar approaches have been independently developed by Blondeau, Gérard and Nyberg [6] and Murphy [19]. While these approaches also differ in some theoretical respects (such as using the likelihood instead of the likelihood ratio in the latter case), the main difference between these works and ours is that we put our model to practice and use it to improve upon known attacks.

2 Preliminaries and Notation

In this work we focus on block ciphers where the key is XORed to (parts of) the state. Let R_k denote one round function of a block cipher with (round)-key k, where without loss of generality the key is added in last. By R we denote the round function without the final key addition, that is $R_k(x) = R(x) \oplus k$. Moreover let $E_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be the corresponding r round block cipher, where $K = (k_0, k_1, \ldots, k_r)$ consist of all round keys. More precisely

$$E_K(x) = R_{k_r} \circ R_{k_{r-1}} \circ \cdots \circ R_{k_1}(x \oplus k_0).$$

For a function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ given an input difference δ and an output difference γ we denote

$$P_F(\delta, \gamma) := \mathbf{Pr}(F(X) \oplus F(X \oplus \delta) = \gamma)$$

for randomly uniformly chosen X. That is, $P_F(\delta, \gamma)$ is the probability of the differential $\delta \to \gamma$. Using N (unordered) pairs, the number of pairs following the given differential is denoted by

$$D_F^{(N)}(\delta, \gamma).$$

The expected value of $D_F^{(N)}(\delta, \gamma) = NP_F(\delta, \gamma)$ and we discuss below more precisely how $D_F^{(N)}(\delta, \gamma)$ is distributed.

Note that in the following N always denotes the number of (unordered) plaintext/ciphertext pairs used. As we use unordered pairs, using the full code book corresponds to choosing $N = 2^{n-1}$.

We consider the case where E is a Markov cipher. A cipher E is a Markov cipher when the transitional probabilities for the output differences of

round $r+1$ only depend on the output difference of round r. More precisely the round function has to satisfy [17]:

$$\mathbf{Pr}(R_k(X) \oplus R_k(X \oplus \delta) = \gamma \mid X = x_0) = P_{R_k}(\delta, \gamma)$$

for all choices of x_0 and uniformly random chosen subkeys k. If, furthermore, all round keys are independent, then one can compute the average value of $P_{E_K}(\delta, \gamma)$ over all possible keys by adding the probabilities for all differential characteristics included in the differential. This has first been formalised in [17] and is summarised in the next proposition.

Proposition 1. *For a function $E_K : \mathbb{F}_2^n \to \mathbb{F}_2^n = R_{k_r} \circ R_{k_{r-1}} \circ \cdots \circ R_{k_1}(x \oplus k_0)$ with input difference δ, output difference γ and $P_R(\gamma', \delta')$ the probability of the differential $\gamma' \to \delta'$ for the function R we have*

$$\tilde{P}_E(\delta, \gamma) := \frac{1}{\sharp K} \sum_K P_{E_K}(\delta, \gamma) = \sum_{\gamma_1, \ldots, \gamma_{r-1}} P_R(\delta, \gamma_1) \left(\prod P_R(\gamma_i, \gamma_{i+1}) \right) P_R(\gamma_r, \gamma) \quad (1)$$

The *hypothesis of stochastic equivalence* states (cf. [17]) that for almost all keys we expect $P_{E_K}(\delta, \gamma) \approx \tilde{P}_E(\delta, \gamma)$ which implies that $D_K^{(N)}(\delta, \gamma) \approx N\tilde{P}_E(\delta, \gamma)$ for almost all keys.

This approximation has to be understood as expected value taken over all expanded keys. However, for our purpose, we are not only interested in the expected value of the counter $D_{E_K}^{(N)}(\delta, \gamma)$ but moreover how these values are distributed. This was analysed in [10] and more recently in [4]. It turns out, considering $D_{E_K}^{(N)}(\delta, \gamma)$ as the results of N independent Bernoulli trials with success probability $\tilde{P}_E(\delta, \gamma)$ leads to a precise model of the actual distribution. More precisely, denoting by $\mathcal{B}(n, p)$ the Binomial distribution with n tries and success probability p, the following is a reasonable approximation for the distribution of $D_{E_K}^{(N)}(\delta, \gamma)$.

Assumption 1 (cf. Theorem 14 in [10]) *The counter $D_{E_K}^{(N)}(\delta, \gamma)$ is distributed according to the Binomial distribution $\mathcal{B}(N, \tilde{P}_E(\delta, \gamma))$, that is*

$$\mathbf{Pr}(D_{E_K}^{(N)}(\delta, \gamma) = c) = \binom{N}{c} \tilde{P}_E(\delta, \gamma)^c (1 - \tilde{P}_E(\delta, \gamma))^{N-c}$$

where the probability is taken over random keys K.

We note that we experimentally validated this assumption for all ciphers considered in this work.

2.1 $\tilde{P}_{E_K}(\delta, \gamma)$ in Differential Cryptanalysis

In standard differential cryptanalysis the attacker attempts to find an input difference and an output difference such that $\tilde{P}_{E_K}(\delta, \gamma)$ is "sufficiently high", i.e., bounded away from uniform. In this case we can expect that, with high probability, for each key K there exist sufficiently many right pairs to mount an attack, i.e., to detect the bias of $\tilde{P}_{E_K}(\delta, \gamma)$. Traditionally, in a 1R attack on the cipher $R_{k_{r+1}} \circ E_K$ one (partially) decrypts the last round with all possible (partial) round keys and increases a counter for the current round key guess iff the computed difference fits the expected output difference γ of round r. Afterwards, the keys are ranked according to their counters, that is, the attacker first tries the key with the highest counter, than the one with the second highest counter, etc.

The success probability of a differential attack is usually computed under the *Wrong-Key Randomization Hypotheses*. The Wrong-Key Randomization Hypotheses (see for example [17]) states that for a wrong key guess the corresponding counter is distributed as for a random permutation. Using the notation established above the Wrong-Key Randomization Hypotheses can be stated as follows

Assumption 2 (Wrong-Key Randomization Hypotheses, cf. [17]))

$$D^{(N)}_{R^{-1}_{k'} \circ R_{k_{r+1}} \circ E_K}(\delta, \gamma) \sim \mathcal{B}(N, 2^{-n})$$

for all $k' \neq k_{r+1}$.

2.2 Distinguishing Distributions

Following the above discussion on the distribution of counter values, it is natural to view a differential attack as a technique to find the value k_{r+1} which maximises the likelihood function corresponding to the right-key distribution (Maximum Likelihood Estimation). This estimation may take two distributions into account. For the right key guess, according to Assumption 1 the counter is distributed according to $\mathcal{B}(N, \tilde{P}_E(\delta, \gamma))$ while the counter of a wrong key guess is assumed (cf. Assumption 2) to be distributed accordingly to $\mathcal{B}(N, 2^{-n})$.

In this setting, the maximum likelihood estimation is equivalent to maximising the log-likelihood ratio of the two distributions under consideration. Indeed, by the Neyman-Pearson Lemma the log-likelihood ratio is the most poweful test to determine whether a sample comes from one of two distributions. Denoting $p = \tilde{P}_E(\delta, \gamma)$ and $q = 2^{-n}$, if a key K resulted in a counter value c one computes

$$\begin{aligned} l_k(c) &:= \log\left(\frac{\binom{N}{c}p^c(1-p)^{N-c}}{\binom{N}{c}q^c(1-q)^{N-c}}\right) \\ &= c\log\left(\left(\frac{p(1-q)}{q(1-p)}\right)\right) + N\cdot\log\left(\frac{1-p}{1-q}\right). \end{aligned} \tag{2}$$

The key guesses are ranked according to their $l_k(c)$ values, that is, the key with highest $l_k(c)$ value is tested first. To simplify the computation one can equivalently rank the keys according to

$$l'_k(c) = c \cdot w \text{ where } w = \log\left(\frac{p(1-q)}{q(1-p)}\right),$$

as we are only interested in the relative value of $l_k(c)$.

We may write $l_{k'}$ and $l'_{k'}$ for $l_{k'}(c)$ and $l'_{k'}(c)$ respectively if it is clear from the context which c we are referring to.

Now, observe that $l'_k(c)$ is monotone increasing iff $p > q$ (as in this case $w > 0$). Thus, if the expected counter for the right key is higher than for wrong key guesses then l'_k has the same ranking and the rankings accordingly to $l'_k(c)$ and c is the same. However, if $p < q$ the function is monotone decreasing (as $w < 0$) and the ranks get reversed. This corresponds to *improbable differentials* as defined in [21]. The special case where $p = 0$ corresponds to *impossible differentials* (as introduced in [14] and later used in [2]), as in this case for each counter $c \neq 0$ the value $l_k(c)$ is formally minus infinity. In the latter case we use the convention $w = -\infty$ and $0 \cdot w = 0$. To conclude, we state the following observation.

Observation 1 *Ranking keys according to their maximum likelihood estimation as defined in Equation (2) unifies in a natural way standard differential attacks, impossible differentials and improbable differentials.*

As explained in the next section, it is this unified view that allows for a generalised attack that considers many (in principle all) counters $D^{(N)}_{E_K}(\delta, \gamma)$ simultaneously.

3 The Attack Model

In this section, we present our attack in detail and provide formulas for computing the gain of our attack. In summary, we use many (or even all) counters $D_{E_K}^{(N)}(\delta, \gamma)$ for different δ and γ values simultaneously. We view those counters as samples from one out of two possible (this time multi-dimensional) distributions. One distribution corresponds to the correct round-key guess and the other to the wrong key guesses. Using many counters at the same time allows us to significantly improve the success probability (or – equivalently – reduce the data complexity) compared to standard differential attacks. Informally, and this is the major difference and biggest improvement over a related approach performed in [5], this allows us to perform several standard differential attacks and impossible (or more generally improbable) differential attacks at the same time. In our attacks these simultaneous differential attacks are weighted appropriately ensuring that we do not lose information compared to standard attacks. That is, considering more information never reduces the success probability but strictly improves it.

3.1 Multi-dimensional Distribution of $D_{E_K}^{(N)}(\delta, \gamma)$

While in general any subset of pairs of input/output differences could be considered, here we focus on the case where one input difference is fixed and we consider all possible output differences. In this case, we denote by

$$\mathcal{D}_{E_K}^{(N)}(\delta) = \left(D_{E_K}^{(N)}(\delta, 1), D_{E_K}^{(N)}(\delta, 2), \ldots, D_{E_K}^{(N)}(\delta, 2^n - 1)\right)$$

the vector of all corresponding counters. As discussed in Section 2, each individual counter is distributed according to a binomial distribution $\mathcal{B}(N, \tilde{P}_E(\delta, \gamma))$. As each pair of the N pairs with input difference δ results in exactly one output difference, we have that

$$\sum_{\gamma} D_{E_K}^{(N)}(\delta, \gamma) = \mathcal{N}\,.$$

Thus, assuming that this is the only dependency between the counter values, the vector $\mathcal{D}_{E_K}^{(N)}(\delta)$ follows a multinomial distribution with parameters N and

$$\tilde{P}_{E_K}(\delta) := \left(\tilde{P}_{E_K}(\delta, 1), \ldots, \tilde{P}_{E_K}(\delta, 2^n - 1)\right),$$

denoted by

$$D^{(N)}_{E_K}(\delta,\gamma) \sim \text{Multi}(N, \tilde{P}_{E_K}(\delta)).$$

Later in this work we present experimental evidence comparing the empirical and theoretical gain of the attack to justify this assumption for the ciphers considered in this work. We summarise our assumption on the behaviour below.

Assumption 3 *$\mathcal{D}^{(N)}_{E_K}(\delta)$ follows a multinomial distribution where each component is distributed according to Assumption 1 and*

$$\sum_{\gamma} D^{(N)}_{E_K}(\delta,\gamma) = N.$$

In contrast to previous works, we do not rely on the Wrong-Key Randomization Hypotheses (Assumption 2) for our attack. Before mounting our attack, the attacker has to compute the expected probability (or the expected counter value) for all possible output differences (cf Section 3.3). If the attacker is able to do this, he is usually also able to compute the expected probability for the wrong keys, that is compute the distribution of $D^{(N)}_{R^{-1}_{k'}\circ R_{k_{r+1}}\circ E_K}(\delta,\gamma)$ as this is essentially computing two more rounds.

3.2 The Attack Algorithm

First, recall that the attack uses N plaintext/ciphertext pairs, to recover the secret key.

Following the previous section, we assume that the attacker has – in an offline phase – computed the parameters of two distributions.[2] Namely, vectors of parameters p and q such that

$$p_i = \tilde{P}_E(\delta, i) \tag{3}$$
$$q_i = \tilde{P}_{R^{-1}\circ R\circ E}(\delta, i). \tag{4}$$

That is, for a right key the vector of counters is a sample from the distribution

$$\text{Dist}_1 = \text{Multi}(N, p)$$

[2] We discuss how efficienty compute this data in Section 3.3.

and for the wrong keys sampled from the distribution

$$\mathrm{Dist}_2 = \mathrm{Multi}(N, q).$$

After this pre-computation phase, the attack proceeds as follows (a brief overview is given in Algorithm 1).

For all possible last round keys k', the attacker first computes the vector of difference counters $\mathcal{D}^{(N)}_{R^{-1}_{k'} \circ R_{k_{r+1}} \circ E_K}(\delta)$. That is, given the guess for the last round key, the attacker partially decrypts every ciphertext and for all output differences γ computes the number of pairs fulfilling the differential $\delta \to \gamma$. Next, the attacker estimates the likelihood that the vector was sampled from Dist_1. In our case, this is equivalent to computing the difference of the log-likelihood of the vector with respect to Dist_1 and with respect to Dist_2, i.e., to compute the log-likelihood-ratio.

Given that for a random variable X following a multinomial distribution $X \sim \mathrm{Multi}(M, p)$ it holds that

$$\mathbf{Pr}(X_1 = x_1 \text{ and } X_2 = x_2 \ldots \text{ and } X_n = x_n) = \begin{cases} \frac{n!}{x_1! x_2! \ldots x_n!} p_1^{x_1} \ldots p_n^{x_n} & \text{if } \sum x_i = M \\ 0 & \text{else} \end{cases},$$

the log-likelihood-ratio is given by

$$l_{k'} = \sum_i D^{(N)}_{R^{-1}_{k'} \circ R_{k_{r+1}} \circ E_K}(\delta, i) \log\left(\frac{p_i}{q_i}\right)$$

Thus, denoting

$$w_i = \log\left(\frac{p_i}{q_i}\right)$$

one computes

$$l_{k'} = \sum_i w_i \cdot D^{(N)}_{R^{-1}_{k'} \circ R_{k_{r+1}} \circ E_K}(\delta, i). \tag{5}$$

This is a weighted extension of the case where one considers only one counter. As before these weights naturally capture various types of differential attacks, i.e., in each component one considers a standard differential, improbable or impossible differential attack. Furthermore, truncated differentials are captured in this model since these correspond to a subvector of $\mathcal{D}^{(N)}_{E_K}(\delta)$.

The time complexity of Algorithm 1 is $|K'| \cdot N$ where N is the number of pairs considered and $|K'|$ is the number of all last-round subkeys.

Input: δ an input difference
Output: A ranking of all possible last-round-keys according to their log-likelihood ratio

begin
  $K' \leftarrow$ the set of all last-round subkeys;
  for $k' \in K'$ **do**
    compute the vector of counters $D^{(N)}_{R^{-1}_{k'} \circ R_{k_{r+1}} \circ E_K}(\delta)$;
    `// compute the log-likelihood ratio`
    $l_{k'} \leftarrow \sum_i w_i D^{(N)}_{R^{-1}_{k'} \circ R_{k_{r+1}} \circ E_K}(\delta, i)$;
  $R \leftarrow$ sorted key candidates k' according to $l'_{k'}$;
  return R

Algorithm 1: Pseudo-code of the attack

3.3 Computing the Markov Model

Computing the vectors p and q is equivalent to repeated (black-box) matrix-vector products $A \times v$ where A is a $2^n \times 2^n$ matrix over $\mathbb{Q}$ and $v \in \mathbb{Q}^{2^n}$ holds the input probabilities for each difference. If our cipher is an SP-network, we have that $A = PS$ where S is a $2^n \times 2^n$ block diagonal matrix over $\mathbb{Q}$ with difference distribution matrices along the main diagonal and P is a permutation matrix . Thus, the matrix A can be represented using much less than $\mathcal{O}(2^{n^2})$ storage. More generally, the matrix A is efficiently representable because block ciphers (especially small-scale block ciphers) are designed to have small circuits. However, the size of each component $v_i \in \mathbb{Q}$ for $1 \leq i < 2^n$ will grow during the computation. More precisely, if we need b bits to represent the entries of the S-box difference distribution matrix of an s-bit S-box, then we need $r \cdot s \cdot b$ bits per entry to represent entries of the vector p and $(r+2) \cdot s \cdot b$ bits to represent entries of the vector q. Thus, the total storage requirement for p and q is $2^n(2r+2) \cdot s \cdot b$ bits. For ciphers with a 32-bit block size this quickly exhausts available RAM. Compression and approximation may delay the growth but cannot necessarily guarantee sufficient precision. However, given sufficient harddisk space, a standard approach in efficient linear algebra is to use multimodular techniques (and the Chinese remainder theorem) to compute the output vector v as a series of representations modulo different primes v_{i,p_j}, where p_j is a word sized prime. Then, when the value v_i is needed, it is computed on the fly from v_{i,p_j} by applying the Chinese remainder theorem. This is the approach we also used in all our experiments. For example, our attacks on KATAN-32 (see below) require ≈100GB of harddisk space and six 32-bit primes to represent p.

3.4 Computing the Gain of the Attack

What remains to be discussed is the efficiency of this attack. The key observation (cf. also [1]) is that the distribution of $l_{k'}$ can be well approximated by a normal distribution in the case where all values w_i are relatively close together. The case where all w_i are close to uniform is the most interesting case for our attack, as otherwise standard differential techniques, considering only one counter are sufficient to break the cipher. Recall that there are two distributions to be considered. First, there is a random variable (and a corresponding distribution) for the log-likelihood-ratio of the right key. We denote this random variable by $\mathcal{R}$ and it is defined as

$$\mathcal{R} = \sum_i w_i D^{(N)}_{E_K}(\delta, i).$$

By Assumption 3 we expect $\mathcal{D}^{(N)}_{E_K}(\delta)$ to be multinomial distributed with parameters N and $(p_i)_i$, with p_i defined in Equation (3). Hence the expected value of $\mathcal{R}$ is given by

$$E(\mathcal{R}) = N \sum w_i p_i.$$

Using that the pairwise covariances for a multinomial distribution is known, the variance of $\mathcal{R}$ can be computed (cf. Appendix A) to be

$$\mathrm{Var}(\mathcal{R}) = N \left(\left(\sum_i w_i^2 p_i \right) - \left(\sum_i w_i p_i \right)^2 \right).$$

Therefore, denoting by $\mathcal{N}(E, V)$ the normal distribution with expected value E and variance V, we will use the following approximation

$$\mathcal{R} \sim \mathcal{N} \left(N \sum w_i p_i, N \left(\left(\sum_i w_i^2 p_i \right) - \left(\sum_i w_i p_i \right)^2 \right) \right)$$

which we will justify with experimental results later in this work.

For the wrong keys, we introduce a random variable $\mathcal{W}$ and, following the same lines of argumentation, we approximate the distribution of $\mathcal{W}$ with a normal distribution, as follows

$$\mathcal{W} \sim \mathcal{N} \left(N \sum w_i q_i, N \left(\left(\sum_i w_i^2 q_i \right) - \left(\sum_i w_i q_i \right)^2 \right) \right)$$

with q_i as defined in Equation (4). This enables to estimate the gain of the attack. For this, we assume that the right key value is sampled according to $\mathcal{R}$. As the normal distribution is symmetric, with a probability of $1/2$, the result is larger or equal to $E(\mathcal{R})$. For the wrong keys values are sampled from $\mathcal{W}$. For 50% percent of the keys, computing the gain is now reduced to computing the probability that $\mathcal{W} \geq E(\mathcal{R})$, as this corresponds to an upper bound on to the probability that a wrong key is ranked above the right key. Using the density function of $\mathcal{W}$, defined as

$$f_W(x) = \frac{1}{\sqrt{2\pi \operatorname{Var}(\mathcal{W})}} e^{-\frac{1}{2\operatorname{Var}(\mathcal{W})}(x-E(\mathcal{W}))^2}$$

this probability of a wrong key being ranked higher than the right key is given by

$$\mathbf{Pr}(\mathcal{W} \geq E(\mathcal{R})) = \int_{E(\mathcal{R})}^{\infty} f_W(x).$$

Using the relation of the standard Normal distribution and the Gaussian error function, this can be rewritten as

$$\mathbf{Pr}(\mathcal{W} \geq E(\mathcal{R})) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{E(\mathcal{R}) - E(\mathcal{W})}{\sqrt{2\operatorname{Var}(\mathcal{W})}}\right)\right). \tag{6}$$

The probability $\mathbf{Pr}(\mathcal{W} \geq E(\mathcal{R}))$ is visualised as the filled, and hence darker, area in Figure 1.

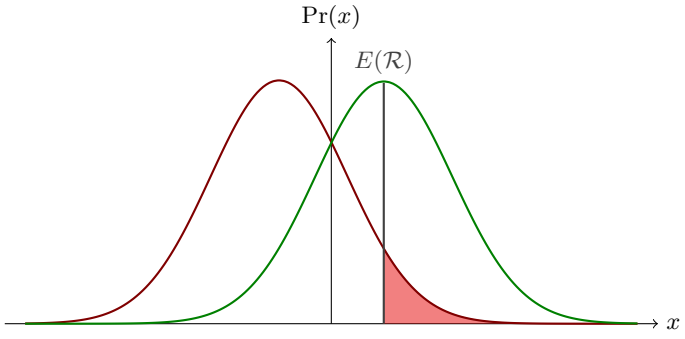

Fig. 1. Computation of the gain.

Note that this approach allows to easily modify the computation in the case where the attacker wants to obtain an error probability for the 10

percent weakest keys (or 90 percent of the keys). In general, to study a ratio of s of all keys one replaces $E(\mathcal{R})$ in Equation (6) by the value t such that

$$\int_{-\infty}^{t} \frac{1}{\sqrt{2\pi \operatorname{Var}(\mathcal{R})}} e^{-\frac{1}{2\operatorname{Var}(\mathcal{R})}(x-E(\mathcal{R}))^2} = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{t - E(\mathcal{R})}{\sqrt{2\operatorname{Var}(\mathcal{R})}}\right)\right) = s.$$

Moreover, this statement of gain relative to the fraction of the keys to which it applies provides more meaningful information than the average gain for all keys, as the average is, in principle, much more sensitive to outliers. That is to say that even if the average gain is very small, there might still be a non-negligible fraction of keys for which the gain is relatively high.

3.5 Some Variants and Improvements

More Input Differences A straight-forward extension which does not require any change to the analysis above is to use a different subset of input- and output-differences. In particular, the attack might benefit from not only using one vector $\mathcal{D}_{E_K}^{(N)}(\delta)$ but several such vectors for several choices of δ. We followed this approach in our experiments against SmallPRESENT-[4].

Considering Subvectors In order to reduce the computational and memory cost of the online phase of the attack, it might beneficial to only consider the most significant components (bounded away from uniform) of the $\mathcal{D}_{E_K}^{(N)}(\delta)$ vector. Again, this can also be done for more than one input difference.

No Key Guessing We may improve the computational and storage requirements of the online phase of the attack by avoiding guessing key bits in the last rounds. That is, we may recover key information from the first rounds. Instead of computing two distributions (one for the right and one for the wrong key) under one input difference δ for the first round, we compute s distributions, one for each possible output difference of the first round. We then encrypt N plaintext pairs following the δ and perform a Maximum Likelihood Estimation for s distributions and the key counter

array as data. That is, we estimate which of the s distributions matches our observed data best. If $\mathrm{Dist}_j(\delta_j, \gamma_i)$ fits the observed data best and δ_j occurs with probability 2^{-b}, then this recovers b bits of information about the key if successful.

4 Application

In this section, we apply our framework to two blockciphers with very small block sizes. First, we consider SmallPRESENT-[4] to demonstrate the idea and then we consider reduced round variants of KATAN-32 for which we present the currently best known attack.

4.1 Toy Example SmallPRESENT-[4]

SmallPRESENT-[s] [18] is a small-scale (toy) cipher designed to aid the development and verification of cryptanalysis techniques. The cipher is an SP-network with s parallel 4-bit S-box applications. Hence the block size is $4s$. The permutation layer is a simple permutation of wires. We focus on SmallPRESENT-[4], the version with 16 bit block size, as this allows us to derive sufficient experimental data rather quickly. The S-box S is the same as for PRESENT (cf. [7]) itself and the round keys are independent. For more details we refer to [18]. A standard differential attack, with one round of partial decryption, seems feasible for not more than 7 rounds. By looking at all the whole vector of output differences, we are able to break 9 rounds with a significant gain. Moreover, compared to standard differential attacks, the data complexity for 7 rounds is reduced by a factor of 2^5. We summarise our findings for attacking $7, 8$ and 9 rounds below.

Attacking 7 Rounds of SmallPRESENT-[4] The best 6 round differential for one active S-box is $\delta = \texttt{0x0007}, \gamma = \texttt{0x0404}$ where $\tilde{P}_E(\delta, \gamma) = 2^{-13.57}$ which is still sufficient to mount a standard differential attack. Using the full code book, our approach gives $E(\mathcal{R}) = 53.821$, $V(\mathcal{R}) = 124.087$, $E(\mathcal{W}) = -47.289$ and $V(\mathcal{W}) = 84.237$ which implies a gain of $\gg 2^{16}$ for 50% of the keys. Indeed, in 100 experiments we always recovered the correct key (rank=0). Even when using only 2^9 pairs, which for

a standard differential attack would not be sufficient, we expect a gain of more than 3.5.

Attacking 8 Rounds of SmallPRESENT-[4] The best 7 round differential for one active S-box is $\delta = \texttt{0x0007}, \gamma = \texttt{0x0505}$ where $\tilde{P}_E(\delta, \gamma) = 2^{-15.39}$ which is not sufficient to mount a standard differential attack. Our model gives $E(\mathcal{R}) = 2.225, V(\mathcal{R}) = 4.611, E(\mathcal{W}) = -2.149, V(\mathcal{W}) = 4.152$ which implies a gain of $2^{5.97}$ for 50% of the keys. In 100 experiments the median of right key ranks was below $2^{16-6} = 2^{10}$. More precisely, in 100 experiments we got 1 ranks $< 2^0$, 2 ranks $< 2^1$, 3 ranks $< 2^2$, 5 ranks $< 2^3$, 8 ranks $< 2^4$, 11 ranks $< 2^5$, 16 ranks $< 2^6$, 25 ranks $< 2^7$, 31 ranks $< 2^8$, 42 ranks $< 2^9$ and 52 ranks $< 2^{10}$. Using $N = 2^{14}$, $N = 2^{13}$, $N = 2^{12}$ and $N = 2^{11}$ we get a gain of 3.954, 2.821, 2.159 and 1.758 respectively. Using more than one input difference and the full code book, namely `0x0007`, `0x000f`, `0x0700`, `0x0070` and `0x0f00` we expect a gain of $2^{18.03}$ for 50% of the keys. In 100 experiments we got median 0.0. More precisely, we got 57 times rank 0, 65 ranks < 2, 73 ranks $< 2^2$, 79 ranks $< 2^3$, 82 ranks $< 2^4$, 84 ranks $< 2^5$, 88 ranks $< 2^6$, 91 ranks $< 2^7$, 95 ranks $< 2^8$, and 96 ranks $< 2^9$.

Attacking 9 Rounds of SmallPRESENT-[4] The best 8 round differential for one active S-box is $\delta = \texttt{0x0007}, \gamma = \texttt{0x5055}$ where $\tilde{P}_E(\delta, \gamma) = 2^{-15.92}$ which is not sufficient to mount a standard differential attack. Our model gives $E(\mathcal{R}) = 0.057, V(\mathcal{R}) = 0.113, E(\mathcal{W}) = -0.056, V(\mathcal{W}) = 0.112$ which implies a gain of $2^{1.44}$ for 50% of the keys. Indeed, for 100 experiments the median rank was $23578.5 \approx 2^{16-1.44}$. More precisely, we got 1 rank $< 2^5$, 3 ranks $< 2^8$, 4 ranks $< 2^{10}$, 5 ranks $< 2^{11}$, 12 ranks $< 2^{12}$, 20 ranks $< 2^{13}$, 32 ranks $< 2^{14}$ and 52 ranks $< 2^{16-1.44}$. Using all sixty input differences where one S-box is active in round one, we expect a gain of $2^{4.625}$. Which is better than exhaustive key search (over half the key space) by a factor of $2^{3.625}$. 1 sample with rank zero (2.33%), 2 ranks $< 2^4$ (4.65%), 5 ranks $< 2^5$ (11.63%), 6 ranks $< 2^6$ (13.95%), 10 ranks $< 2^7$ (23.26%), 17 ranks $< 2^8$ (39.53%), 20 ranks $< 2^9$ (46.51%), 25 ranks $< 2^{10}$ (58.14%), 27 ranks $< 2^{11}$ (62.79%), 30 ranks $< 2^{12}$ (69.77%), 37 ranks $< 2^{13}$ (86.05%) and 42 ranks $< 2^{14}$ (97.67%). All ranks were $< 2^{15}$. For 50% of the keys we got rank $< 579 \approx 2^{16-6.82}$.

4.2 Application to KATAN32

KATAN-32 is one member of a family of ciphers defined in [9]. It has a block-size of 32 bits, an 80 bit key and 254 rounds. The plaintext is loaded into two registers of length 13 and 19, respectively. In each round, two bits of the registers are updated, involving one key bit each. We refer to [9] for more information. The currently best know differential attack on KATAN-32 is a conditional differential attack that can break up to 78 rounds (see [13]). The best attack overall breaks the full cipher slightly faster than exhaustive key search [12]. Note that, for KTANTAN-32, which differers from KATAN-32 only in the key-scheduling, better attacks are known (see [8,22]) but they do not apply to KATAN-32.

Our contributions with respect to KATAN-32 are twofold. One the one hand, we present an attack on 115 rounds of the cipher, which is the best known cryptanalytic result so far. On the other hand, we computed the complete distribution of output differences – and hence the probabilities for all output differences – for various numbers of rounds[3]. Compared to the probability of the best differential characteristic, these probabilities are (as expected) higher, however, our results indicate that there is no significant clustering of trails for KATAN-32 that could be used in a standard differential attack. Thus, KATAN-32 still provided a sufficiently high security margin and our results strengthen the confidence in the security of the cipher.

Below, we always assume $\delta = \texttt{0x1006a880}$ which is the input difference for the best known differential characteristic which holds with probability 2^{-31} after 71 rounds disregarding any dependencies. Note, however that the special structure of KATAN-32 means that in fact the first probabilistic difference only depends on the plaintext values and not on the key values.

We consider a $\ell = 24$R attack below to maximise the number of rounds. This implies a computational cost of $2^{32} \cdot 2^{2\ell} = 2^{32+48} = 2^{80}$ partial decryptions in the online phase of the attack. Exhaustive search over half the key space would have to perform 2^{79} full encryptions where one full encryption costs roughly 4 partial decryptions. Hence, our attacks are

[3] Computing this requires roughly one day of computation time on a 2.6 Ghz PC given 32 GB of RAM with our implementation.

twice as fast as exhaustive search. However, we emphasise that compared to exhaustive search the gain in our attack is significantly smaller.

71 + 24 Rounds of KATAN-32 The best output difference $\gamma =$ `0x00000008` has probability $\tilde{P}_E(\delta, \gamma) \approx 2^{-29.52}$, the output difference with the lowest probability is $\tilde{\gamma} =$ `0x00000080` with $\tilde{P}_E(\delta, \gamma) \approx 2^{-32.10}$. We get $E(\mathcal{R}) \approx 2505.2110272$, $V(\mathcal{R}) \approx 5096.6607713$, $E(\mathcal{W}) \approx -2467.4478539$, $V(\mathcal{W}) \approx 4868.2802123$. Which gives an expected gain of > 50 for 50% of the keys. In Figure 2 we plot the expected gain for increasing numbers of considered output differences.

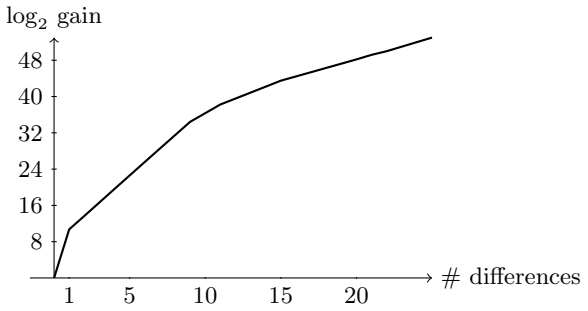

Fig. 2. Expected gain for increasing number of output differences for 71 rounds of KTANTAN-32.

We verified this estimate by considering the 16 differences with the highest probability. We compared randomly chosen right keys with randomly chosen wrong keys and always recovered the right key as the key with the highest rank.

88 + 24 Rounds of KATAN-32 The best output difference $\gamma =$ `0x02000004` has probability $\tilde{P}_E(\delta, \gamma) \approx 2^{-31.97}$, the output difference with the lowest probability is $\tilde{\gamma} =$ `0x00200000` with $\tilde{P}_E(\delta, \tilde{\gamma}) \approx 2^{-32.01}$. We get $E(\mathcal{R}) \approx 0.7203661$, $V(\mathcal{R}) \approx 1.4407603$, $E(\mathcal{W}) \approx -0.7203574$, $V(\mathcal{W}) \approx 1.4406867$. Which gives an expected gain of 3.1201916 for 50% of the keys.

90 + 24 Rounds of KATAN-32 The best output difference $\gamma =$ `0x08080015` has probability $\tilde{P}_E(\delta, \gamma) \approx 2^{-31.99}$, the output difference with

the lowest probability is $\tilde{\gamma} = \texttt{0x08000000}$ with $\tilde{P}_E(\delta, \tilde{\gamma}) \approx 2^{-32.01}$. We get $E(\mathcal{R}) \approx 0.3750060$, $V(\mathcal{R}) \approx 0.7500168$, $E(\mathcal{W}) \approx -0.3750042$, $V(\mathcal{W}) \approx 0.7500035$. Which gives an expected gain of 2.3715683 for 50% of the keys.

91 + 24 Rounds of KATAN-32 The best output difference $\gamma = \texttt{0x00400000}$ has probability $\tilde{P}_E(\delta, \gamma) \approx 2^{-31.98}$, the output difference with the lowest probability is $\tilde{\gamma} = \texttt{0x02000000}$ with $\tilde{P}_E(\delta, \tilde{\gamma}) \approx 2^{-32.00}$. We get $E(\mathcal{R}) \approx 0.3695390$, $V(\mathcal{R}) \approx 0.7390803$, $E(\mathcal{W}) \approx -0.3695384$, $V(\mathcal{W}) \approx 0.7390745$. Which gives an expected gain of 2.3586180 for 50% of the keys.

The expected gain for 50% of the keys for 92 and 94 rounds is 1.9220367 and 1.2306869 respectively.

5 Conclusions and Further Work

In this work we presented a unifying framework for several standard differential attacks. This unified view allows to naturally consider multiple differentials and by that improving upon known results. Our framework always provides better success probabilities than any of the combined differential attacks alone; although at the potential cost of increased computation time and memory. We demonstrated the viability of our approach by extending the the best differential for SmallPRESENT-[4] by two rounds and the best known differential for KATAN-32 by 20 rounds.

However, for many ciphers computing the distribution of counter values, i.e., $\mathcal{D}^{(N)}_{E_K}(\delta)$, is prohibitively expensive. Yet, starting from one difference computing one or two rounds is usually feasible since only few output differences are possible after such a small number of rounds. It is thus possible to extend a standard differential attack using techniques discussed in this. Instead of considering $\mathcal{D}^{(N)}_{E_K}(\delta)$ the attacker would consider $\mathcal{D}^{(N)}_{R_{k_r}}(\delta)$. Then, in the online phase of the attack counters are weighted accordingly to their distribution. We leave the details of such an approach open for further investigation.

Acknowledgements

We would like to thank Sean Murphy for helpful discussions on the log-likelihood ratio and its relation to the maximum likelihood estimation. We would also like to thank Céline Blondeau, Benoit Gérard and Kaisa Nyberg for helpful discussions on an earlier draft of this work. Furthermore, we would like to thank William Stein for allowing us to use his computers purchased under National Science Foundation Grant No. DMS-0821725. The second author gratefully acknowledges the support from the Danish National Research Foundation and the National Science Foundation of China(Grant No.11061130539) for the Danish-Chinese Center for Applications of Algebraic Geometry in Coding Theory and Cryptography. Finally, we would like to thank Michael Hortmann for introducing the authors to cryptography.

References

1. Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond Linear Cryptanalysis? In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450, Berlin, Heidelberg, New York, 2004. Springer Verlag.
2. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using Impossible Differentials. In *Advances in Cryptography - EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23, Berlin, Heidelberg, New York, 1999. Springer Verlag.
3. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptography - CRYPTO 1990*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21, Berlin, Heidelberg, New York, 1991. Springer Verlag.
4. Celine Blondeau and Benoit Gérard. Links between theoretical and effective differential probabilities: Experiments on PRESENT. ECrypt II Workshop on Tools for Cryptanalysis, 2010.
5. Celine Blondeau and Benoit Gérard. Multiple Differential Cryptanalysis: Theory and practice. Lecture Notes in Computer Science, Berlin, Heidelberg, New York, 2011. Springer Verlag.
6. Celine Blondeau, Benoıt Gérard, and Kaisa Nyberg. LLR and Differential Cryptanalysis. preprint, 2011.
7. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J˙ B. Robshaw, Yannick Seurin, and Charlotte Vikkelsø. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466, Berlin, Heidelberg, New York, 2007. Springer.

8. Andrey Bogdanov and Christian Rechberger. A 3-subset Meet-in-the-Middle attack: Cryptanalysis of the lightweight block cipher KTANTAN. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - SAC 2010*, volume 6544, pages 229–240, Berlin, Heidelberg, New York, 2010. Springer Verlag.
9. Christophe De Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer Verlag, 2009.
10. Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. Cryptology ePrint Archive, Report 2005/212, 2005. `http://eprint.iacr.org/`.
11. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jaesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A new block cipher suitable for low-resource device. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59, Berlin, Heidelberg, New York, 2006. Springer Verlag.
12. Simon Knellwolf. Meet-in-the-Middle cryptanalysis of KATAN. ECRYPT Workshop on Lightweight Cryptography (to appear), 2011.
13. Simon Knellwolf, Willi Meier, and María Naya-Plasencia. Conditional Differential Cryptanalysis of NLFSR-based cryptosystems. In Masayuki Abe, editor, *Advances in Cryptography - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 130–145, Berlin, Heidelberg, New York, 2010. Springer Verlag.
14. Lars Knudsen. DEAL – a 128-bit block cipher. Technical report, Department of Informatics, University of Bergen, Norway, 1998.
15. Lars R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption 1995*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211, Berlin, Heidelberg, New York, 1995. Springer Verlag.
16. X. Lai. Higher order derivatives and differential cryptanalysis. communications and cryptography, 1994.
17. Xuejia Lai and James L. Massey. Markov ciphers and Differential Cryptanalysis. In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38, 1991.
18. Gregor Leander. Small scale variants of the block cipher PRESENT. Cryptology ePrint Archive, Report 2010/143, 2010. `http://eprint.iacr.org/`.
19. Sean Murphy. The analysis of simultaneous differences in Differential Cryptanalysis. `http://www.isg.rhul.ac.uk/~sean/SimDiffA.pdf`, 2011.
20. Sean Murphy, Fred Piper, Michael Walker, and Peter Wild. Likelihood estimation for block cipher keys. `http://www.isg.rhul.ac.uk/~sean/maxlik.pdf`, 1995.
21. Cihangir Tezcan. The improbable differential attack: Cryptanalysis of reduced round CLEFIA. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010*, volume 6498 of *Lecture Notes in Computer Science*, pages 197–209, Berlin, Heidelberg, New York, 2010. Springer Verlag.
22. Lei Wei, Christian Rechberger, Jian Guo, Hongjun Wu, Huaxiong Wang, and San Ling. Improved Meet-in-the-Middle cryptanalysis of KTANTAN. Cryptology ePrint Archive, Report 2011/201, 2011. `http://eprint.iacr.org/`.

A Computation of the Variance of $\mathcal{R}$

Assume that we are given a random variable X following a multinomial distribution with parameters M and p and constants w_i. We explain (cf. also [19]), for completeness, how to compute the variance of

$$Y := \sum w_i X_i.$$

Note that if X is multinomial distributed, the covariance between two compontents X_i and X_j is given by

$$\operatorname{Cov}(X_i, X_j) = \begin{cases} -Mp_ip_j & \text{if } i \neq j \\ Mp_i(1-p_i) & \text{if } i = j \end{cases}$$

Therefore, the variance of Y is compute as follows

$$\begin{aligned}
\operatorname{Var}(Y) &= \sum_{i,j} \operatorname{Cov}(w_iX_i, w_jX_j) \\
&= \sum_i \operatorname{Var}(w_iX_i) + 2\sum_{i<j} \operatorname{Cov}(w_iX_i, w_jX_j) \\
&= \sum_i w_i^2 \operatorname{Var}(X_i) + 2\sum_{i<j} w_iw_j \operatorname{Cov}(X_i, X_j) \\
&= \sum_i w_i^2 Mp_i(1-p_i) - 2\sum_{i<j} w_iw_j Mp_ip_j \\
&= M\sum_i w_i^2 p_i - M\left(\sum_i w_i^2 p_i^2 + 2\sum_{i<j} w_iw_j p_ip_j\right) \\
&= M\sum_i w_i^2 p_i - M\left(\sum_i w_ip_i\right)^2 \\
&= M\sum_i w_i^2 p_i - ME(Y/M)^2 \\
&= M\sum_i w_i^2 p_i - \frac{E(Y)^2}{M} \\
&= M\left(\left(\sum_i w_i^2 p_i\right) - \left(\sum_i w_ip_i\right)^2\right),
\end{aligned}$$

which is exactly what we apply in this work.