Multiple Differential Cryptanalysis: Theory and Practice (Corrected)

Céline Blondeau and Benoît Gérard

SECRET Project-Team - INRIA Paris-Rocquencourt
Domaine de Voluceau - B.P. 105 - 78153 Le Chesnay Cedex - France ,
{celine.blondeau,benoit.gerard}@inria.fr

Abstract. Differential cryptanalysis is a well-known statistical attack on block ciphers. We present here a generalisation of this attack called multiple differential cryptanalysis. We study the data complexity, the time complexity and the success probability of such an attack and we experimentally validate our formulas on a reduced version of PRESENT. Finally, we propose a multiple differential cryptanalysis on 18-round PRESENT for both 80-bit and 128-bit master keys.
Keywords: iterative block cipher, multiple differential cryptanalysis, PRESENT, data complexity, success probability, time complexity.

1 Introduction

Differential cryptanalysis has been introduced in 1990 by Biham and Shamir [1, 2] in order to break the *Data Encryption Standard* block cipher. This statistical cryptanalysis exploits the existence of a *differential*, *i.e.*, of a pair (α, β) of differences such that for a given input difference α, the output difference after encryption equals β with a high probability. This attack has been successfully applied to many ciphers and has been extended to various different attacks, such as truncated differential cryptanalysis, impossible differential cryptanalysis...

In the original version of differential cryptanalysis [1], a unique differential is exploited. Then, Biham and Shamir have improved their attack by considering together several differentials having the same output difference [2]. Truncated differential cryptanalysis introduced by Knudsen [3] uses differentials with many output differences that are structured as a linear space.

Here, we consider what we name *multiple differential cryptanalysis*. Similarly to multiple linear cryptanalysis, multiple differential cryptanalysis is the general case where the set of considered differentials has no particular structure, *i.e.*, several input differences are considered together and the corresponding output differences can be different from an input difference to another.

The problem of estimating the data complexity, time complexity and success probability of a differential cryptanalysis is far from being simple. Since 1991, it is widely accepted that the data complexity of a differential cryptanalysis is of order p_*^{-1}, where p_* denotes the probability of the involved differential [2]. Theoretical studies based on hypothesis testing theory [4–6] confirm this statement and give more specific results. Concerning the success probability, a formula has been recently established by Selçuk in [7]. This formula, which is used in many recent papers on differential cryptanalysis, is derived from a Gaussian approximation of the binomial distribution. However, as already explained by Selçuk, the Gaussian approximation is not good in the setting of differential cryptanalysis. This was the motivation of the general framework presented in [8], that studies the complexity of any statistical cryptanalysis based on counters that follow a binomial distribution. But, this work does not apply to multiple differential cryptanalysis since the involved counters do not follow a binomial distribution in case.

Our contribution. The main purpose of this paper is to provide a detailed analysis of the complexity of any multiple differential attack. It is worth noticing that it includes the variants of differential attacks such as classical differential cryptanalysis or truncated differential cryptanalysis... In Section 2, we introduce multiple differential cryptanalysis and study the complexity of this attack. We mainly provide formulas for the data complexity and the success probability of a multiple differential cryptanalysis. Then, in Section 3, we validate this theoretical framework by many experiments on a reduced version of the cipher PRESENT, namely SMALLPRESENT-[8]. Then, Section 4 focuses on the general problem of computing the involved probabilities. This problem arises in any statistical attack and is not that is not directly related to the use of several differentials. Finally, to conclude this work, we propose a multiple differential cryptanalysis of 18-round PRESENT. This attack is not the best known attack on PRESENT since Cho has presented some attacks up to 26 rounds [9]. Nevertheless, it improves the best previously known differential cryptanalysis on 16 rounds due to Wang [10].

2 Theoretical framework

In this first section, we propose a framework for analysing multiple differential cryptanalyses. More precisely we provide estimates for the data complexity and the success probability of such differential attacks that use any number of differentials. The time and memory complexities of these attacks are also discussed.

2.1 Presentation and notation

Let us start with some notation that will be used all along this paper. We consider an iterative block cipher parametrised by a key K.

$$\begin{aligned} E_K : \mathbb{F}_2^m &\to \mathbb{F}_2^m \\ x &\mapsto y = E_K(x), \end{aligned}$$

where m is the block size. We denote by F the round function of this iterative cipher: $F_k(x)$ is the result of 1-round encryption of x using the subkey k. A multiple differential cryptanalysis aims at recovering the key K_* used to encipher the available sample. We consider here a last-round differential cryptanalysis on an iterative block cipher that recovers n_k bits of the last-round subkey that we will denote by k_* (this subkey is derived from the master key K_*). Such an attack belongs to the class of statistical cryptanalyses and thus follows the three following steps.

- *Distillation phase:* Extract the information on k_* obtained from the N available plaintext/ciphertext pairs.
- *Analysis phase:* From this information, compute the likelihoods of the candidates for the value of k_* and generate the list $\mathcal{L}$ of the best ℓ candidates.
- *Search phase:* Look down the list of candidates and test all the corresponding master keys until the good one is found.

Now, let us introduce the notation used for the differentials.

Definition 1. *[11] An r-round differential for a block cipher is a couple of differences $(\delta_0, \delta_r) \in \mathbb{F}_2^m \times \mathbb{F}_2^m$. The probability of the differential is defined by*

$$\Pr\left[\delta_0 \to \delta_r\right] \stackrel{\text{def}}{=} \Pr_{\mathbf{X},\mathbf{K}}\left[E_{\mathbf{K}}(\mathbf{X}) \oplus E_{\mathbf{K}}(\mathbf{X} \oplus \delta_0) = \delta_r\right].$$

In the setting of multiple differential cryptanalysis, the attacker exploits a collection Δ of differentials. The natural way of ordering these differentials is to gather the differentials with the same input difference. We denote by Δ_0 the set of all input differences involved in the set Δ

$$\Delta_0 \stackrel{\text{def}}{=} \{\delta_0,\ \exists \delta_r, (\delta_0, \delta_r) \in \Delta\}.$$

We number the input differences in Δ_0 so that $\Delta_0 = \{\delta_0^{(1)}, \ldots, \delta_0^{(|\Delta_0|)}\}$. Hence, for a fixed input difference $\delta_0^{(i)} \in \Delta_0$, we obtain a set $\Delta_r^{(i)}$ of the corresponding output differences:

$$\Delta_r^{(i)} \stackrel{\text{def}}{=} \{\delta_r \mid (\delta_0^{(i)}, \delta_r) \in \Delta\}.$$

Therefore, the set of differentials Δ can be expressed as

$$\Delta = \left\{ \left(\delta_0^{(i)}, \delta_r^{(i,j)}\right) \;\middle|\; i = 1 \ldots |\Delta_0| \text{ and } j = 1 \ldots |\Delta_r^{(i)}| \right\}.$$

It is worth noticing that this definition is more general than truncated differential cryptanalysis since the set of output differences can be different from an input difference to another.

As in differential cryptanalysis, the algorithm used in multiple differential cryptanalysis consists in partially deciphering the N ciphertexts using all possible values for the last-round subkey and in counting the number of occurrences of the differentials in Δ. In other words, we count the number of plaintext pairs with a difference $\delta_0^{(i)}$ in Δ_0 that lead to an output difference in $\Delta_r^{(i)}$ after r rounds. However, this attack (as it is) may not work because the cost of the partial decryption is prohibitive (there are too many pairs of ciphertexts and too many possible values for the subkey). In order to decrease this cost, a *sieving phase* is used[1] to discard some pairs for which we already know that the difference after r rounds cannot be in $\Delta_r^{(i)}$. This phase consists in precomputing the sets $\Delta_{r+1}^{(i)}$ of all δ_{r+1} in $\mathbb{F}_2^m$ such that there exists a j for which $\Pr\left[\delta_r^{(i,j)} \rightarrow \delta_{r+1}\right] \neq 0$ and in discarding every pair with an output difference not in $\Delta_{r+1}^{(i)}$. This set of differences is named a *sieve*. The multiple differential attack is summarized in Algorithm 1.

Algorithm 1: Multiple differential cryptanalysis

Input: N chosen plaintext/ciphertext pairs (x_i, y_i) with $y_i = E_{K_*}(x_i)$)
Output: The key K_* used to encipher the samples

Initialise a table D of 2^{n_k} counters to 0.
foreach $\delta_0^{(i)} \in \Delta_0$ **do**
  foreach *plaintext pair* (x_a, x_b) *such that* $x_b = x_a \oplus \delta_0^{(i)}$ **do**
    if $y_a \oplus y_b \in \Delta_{r+1}^{(i)}$ **then**
      foreach candidate k **do**
        Compute $\delta = F_k^{-1}(y_a) \oplus F_k^{-1}(y_b)$;
        if $\delta \in \Delta_r^{(i)}$ **then** $D[k] \leftarrow D[k] + 1$;
Generate a list $\mathcal{L}$ of the ℓ candidates with the highest values of $D[k]$;
foreach $k \in \mathcal{L}$ **do**
  foreach possible master key K corresponding to k **do**
    if $E_K(x) = y = E_{K_*}(x)$ **then return** K;

Such attacks are successful when the correct subkey is in the list $\mathcal{L}$ of candidates. Four important quantities have to be taken into consideration when quantifying the efficiency of

[1] This is widely used in differential cryptanalysis.

a statistical cryptanalysis. The *success probability* P_S that is the probability of the correct subkey to be in the list of the best candidates,

$$P_S \stackrel{\text{def}}{=} \Pr\left[k_* \in \mathcal{L}\right],$$

the *data complexity* N that is the number of plaintext/ciphertext pairs used for the attack, the *time complexity* that heavily depends on the size ℓ of the list $\mathcal{L}$ and the *memory complexity*. The first three quantities are closely related since increasing N will increase P_S and increasing ℓ will also increase P_S together with the time complexity. We now study the time and memory complexities, while formulas for the data complexity and the success probability are provided in Section 2.4.

Remark. In a multiple differential attack, the *number of chosen plaintexts* N and the *number of samples* N_s are different quantities. The number of samples corresponds to the number of pairs with a difference in Δ_0 that we can form with N plaintexts. In an attack with $|\Delta_0|$ input differences, we can choose the plaintexts such that the number of samples is $N_s = \frac{|\Delta_0|\,N}{2}$. This is done by choosing the plaintext set of the form $\bigcup_x \{x \oplus \delta, \delta \in Vect(\Delta_0)\}$ where $Vect(\Delta_0)$ is the linear space spanned by the elements of Δ_0. Such sets are classically named *structures*.

2.2 Time and memory complexities

In this section we discuss the details of Algorithm 1 in order to compute the time and the memory complexities of the multiple differential cryptanalysis defined in Algorithm 1.

In order to analyse the time complexity of this attack we introduce some notation. Let $S_r \stackrel{\text{def}}{=} \max_i\{|\Delta_r^{(i)}|\}$ and $S_{r+1} \stackrel{\text{def}}{=} \max_i\{|\Delta_{r+1}^{(i)}|\}$. We denote by p_{sieve} the maximum over all input differences in Δ_0 of the probability to pass the sieve *i.e.* $p_{sieve} = 2^{-m} S_{r+1}$.

When performing a multiple differential cryptanalysis, one needs to check many times if some difference belongs to a particular set A of differences. This step of the algorithm can be done with a time complexity logarithmic in $|A|$. On the other hand, this requires the use of $|A|$ memory blocks. Now let us consider each important step of the algorithm.

The total number of pairs to test is $N_s = |\Delta_0|N/2$. For each pair we have to check if it passes the sieve. Thus the time complexity of this step is $N_s \log(S_{r+1})$. Nevertheless, one can decrease this complexity using the following simple trick. If there exists a set of positions in $\{1 \cdots m\}$ on which all elements in Δ_{r+1} vanish, then the plaintext/ciphertext pairs can be gathered depending on the values of the ciphertexts on these bits. Pairs formed by ciphertexts belonging to two different groups will not pass the sieve and thus only the pairs formed by ciphertexts in the same group must be considered. Using this trick together with plaintexts chosen to form structures, this step can take negligible time regarding the rest of the attack. Since the proposed cryptanalysis is a last-round attack, a partial inversion of the round function has to be performed for each pair that passes the sieve and for each last-round subkey. Therefore this step has a complexity of about $2^{n_k}\, N_s p_{sieve}$. Extracting the likeliest ℓ subkeys can be handled in linear time (regarding the number of candidates 2^{n_k}). The last part of the algorithm corresponds to an exhaustive search for the remaining bits of the master key. This step requires $\ell \cdot 2^{n_K - n_k}$ encryptions where n_K is the size of the master key.

Table 1 summarises the time complexities. The terms corresponding to steps with a small time complexity are neglected here, and it is assumed that the generation of the pairs has been done using the aforementioned trick.

Table 1. Time complexity of a multiple differential cryptanalysis where S_r (resp. S_{r+1}) denote the maximal number of output differences for a given input difference in Δ_0 after r-rounds (resp. $(r+1)$-rounds).

Encryptions	Partial decryptions	Comparisons
$O\left(\ell 2^{n_K - n_k}\right)$	$O\left(2^{n_k} N_s 2^{-m} S_{r+1}\right)$	$O\left(2^{n_k} N_s 2^{-m} S_{r+1} \log(1+S_r)\right)$

The partial decryption cost can be seen as a the $1/(r+1)$-th of the cost of an encryption for an $(r+1)$-round cipher. The memory complexity of the attack is essentially due to the storage of the counters, of the plaintext/ciphertext pairs and of the sieves.

2.3 Theoretical framework

In this subsection we develop the theoretical framework used to analyse multiple differential cryptanalysis. In our context, the attacker obtains the ciphertexts corresponding to a set of N chosen plaintexts generated using structures.

The determination of the data complexity and the success probability of a multiple differential cryptanalysis requires the knowledge of the distribution of the counters used in Algorithm 1 and particularly the distribution of $D(k)$.

Definition 2. *Let $D_x^{(i)}(k)$ be the basic counter corresponding to the set of differentials with $\delta_0^{(i)}$ as input difference and with output difference in $\Delta_r^{(i)}$. For a given plaintext and a given candidate k, $D_x^{(i)}(k)$ is defined as*

$$D_x^{(i)}(k) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } F_k^{-1}\big(E_{K_*}(x)\big) \oplus F_k^{-1}\big(E_{K_*}(x \oplus \delta_0^{(i)})\big) \in \Delta_r^{(i)}, \\ 0 & \text{otherwise.} \end{cases}$$

The counters $D_x^{(i)}(k)$ follow a Bernoulli distribution since, for a fixed input difference and a fixed plaintext, only one output difference can occur. For $k = k_*$, the value of $F_k^{-1}(x)$ corresponds to the value obtained after r rounds of the cipher and thus the distribution of $D_x^{(i)}(k_*)$ depends on the probability of the corresponding differential. On the other hand, for $k \neq k_*$, it is usually assumed that the value $F_k^{-1}(x)$ is uniformly distributed among all the possible values. This assumption is known as the *Wrong Key Randomisation Hypothesis* [12]. Most notably the distribution of the $D_x^{(i)}(k)$'s is the same for all wrong candidates k.

Hypothesis 1. (Wrong-Key Randomisation Hypothesis in the differential cryptanalysis setting).

$$\Pr_{\mathbf{X}}\left[F_k^{-1}(E_{K_*}(X)) \oplus F_k^{-1}(E_{K_*}(X \oplus \delta_0^{(i)})) = \delta_r^{(i,j)}\right] = \begin{cases} p_*^{(i,j)} & \text{if } k = k_*, \\ p^{(i,j)} = \frac{1}{2^m - 1} & \text{for } k \neq k_*. \end{cases}$$

In the following of this paper we will take the value 2^{-m} instead of $\frac{1}{2^m-1}$ for $p^{(i,j)}$. Then, using this hypothesis, we obtain that $D_x^{(i)}(k)$ follows a Bernoulli distribution with parameter $p_*^{(i)} \stackrel{\text{def}}{=} \sum_{j=1}^{|\Delta_r^{(i)}|} p_*^{(i,j)}$ if $k = k_*$ and $p^{(i)} \stackrel{\text{def}}{=} \sum_{j=1}^{|\Delta_r^{(i)}|} p^{(i,j)} \approx |\Delta_r^{(i)}|\, 2^{-m}$ otherwise. Then we can defined the sum of this basics counters.

Definition 3. *Let $D_x^{(i)}(k)$ be the basic counters defined in Definition 2. We define the sums of the basic counters the set of all differentials and the counter we are interested in, that is the mark obtained by a subkey during the attack:*

$$D_x(k) \stackrel{\text{def}}{=} \sum_{i=1}^{|\Delta_0|} D_x^{(i)}(k) \quad \text{and} \quad D(k) \stackrel{\text{def}}{=} \frac{1}{2} \sum_x D_x(k).$$

The factor $1/2$ in the sum come from the fact that for any i, x and any key k, the counters $D_x^{(i)}(k)$ and $D_{x\oplus\delta_0^{(i)}}^{(i)}(k)$ are equal. Hence, each statistical phenomenon is counted twice when summing over all possible values for x. Instead of putting such a factor $1/2$ it may be possible to sum over one half of the whole set of x in a way that each pair of plaintexts will be counted only once. For a fixed i, we consider only one input difference $\delta_0^{(i)}$ hence it is easy to split the set of plaintext in two. The problem is not so easy when we have to consider all input differences (*i.e.* for the sum $\sum_x D_x(k)$). Indeed, it may not be possible to find a set $\mathcal{X}$ containing $N/2$ plaintexts such that all pairs are counted once and only once in other words, a set such that $\sum_{x\in\mathcal{X}} D_x(k) = \frac{1}{2}\sum_x D_x(k)$. The existence of such a set $\mathcal{X}$ depends on the structure of the set of input differences Δ_0.

Definition 4. *The set of input differences Δ_0 is* admissible *if there exists a set $\mathcal{X}$ of $N/2$ plaintexts that fulfils the condition*

$$\forall \delta_0^{(i)} \in \Delta_0, \forall x \in \mathcal{X}, x \oplus \delta_0^{(i)} \notin \mathcal{X}. \tag{1}$$

An efficient way to test if a set Δ_0 is admissible is provided in Appendix A.1. From now, we consider that the set Δ_0 has been chosen to be admissible. Hence, each pair is only counted once, but some dependencies between counters still remain. Deriving a general formula for the distribution of a sum of dependent variables is not so easy. Moreover, the variables we consider have really small dependencies and hence, we will assume that they are independent.

Hypothesis 2. *For any subkey k (including k_*) and a set $\mathcal{X}$ that fulfils (1),*

- *For any x, the variables $(D_x^{(i)}(k))_{1\le i\le|\Delta_0|}$ are independent.*
- *The variables $(D_x(k))_{x\in\mathcal{X}}$ are independent.*

This hypothesis is not so far to being true. The same kind of hypothesis is done in differential cryptanalysis. Indeed, in the differential setting, the random variables $D_x(k)$ follow a Bernoulli distribution of parameters p_* or p and the same kind of independence assumption is used in order to say that the counters $D(k)$ follow a binomial distribution.

Assuming Hypothesis 2, the end of this section is now dedicated to the problem of finding good estimates for the distribution of the sum of M independent variables that follow Bernoulli distributions with different parameters. Actually, we aim at applying this estimate to the determination of the distributions of $D(k)$ and $D(k_*)$. In the following, we use $D(k)$ to instantiate some results but the results obviously hold for $D(k_*)$, when p is replaced by p_*.

The first technique to find a good estimate of the distribution of $D(k)$ is to use the following theorem which states that the distribution of the counters $D_i(k)$ is close to a Poisson distribution.

Theorem 1. *[13] Let $D_x^{(i)}(k)$ be M independent Bernoulli random variables with parameters $p^{(i)}$. Let $D_x(k) \stackrel{\text{def}}{=} \sum_{i=1}^M D_x^{(i)}(k)$ and $\lambda = \sum_{i=1}^M p^{(i)}$. Then, for all $A \subset \{0,1,\ldots,M\}$, we have*

$$\left|\Pr\left[D_x(k) \in A\right] - \sum_{a\in A} \frac{\lambda^a e^{-\lambda}}{a!}\right| < \sum_{i=1}^{M} \left(p^{(i)}\right)^2.$$

Hence, the distribution of $D_x(k)$ is close to a Poisson distribution of parameter $\sum_{i=0}^{|\Delta_0|} p^{(i)}$. Then, using the stability of the Poisson distribution under addition, we conclude that

$\sum_{x\in\mathcal{X}} D_x(k)$ follows a Poisson distribution with parameter $\frac{N}{2}\cdot\sum_{i=0}^{|\Delta_0|} p^{(i)}$.We then introduce the following quantities that play a particular role in the analysis of the multiple differential cryptanalysis.

$$p_* \stackrel{\text{def}}{=} \frac{\sum_i p_*^{(i)}}{|\Delta_0|} = \frac{\sum_{i,j} p_*^{(i,j)}}{|\Delta_0|} \quad \text{and} \quad p \stackrel{\text{def}}{=} \frac{\sum_i p^{(i)}}{|\Delta_0|} = \frac{\sum_{i,j} p^{(i,j)}}{|\Delta_0|} \approx \frac{|\Delta|\cdot 2^{-m}}{|\Delta_0|}.$$

The bound on the error due to the use of the Poisson approximation is relatively small regarding probabilities of order 10^{-1} but it is not clear that this approximation is still accurate when considering tails of the distribution. Indeed, we have checked with some experimental results that the cumulative function of the Poisson distribution is not a good estimate of the tails of the cumulative distribution function of the counters $D(k)$. For this reason, we have to use another result from large deviations theory to obtain a better estimate for the tails of the distribution of the $D(k)$'s.

Theorem 2. *[14, chapter 5.4] Let $D(k) = \sum_x D_x(k)$ be a sum of M discrete, independent and identically distributed random variables. Let $\mu(s)$ be the semi-invariant moment generating function of each of the $D_x(k)$. Then, for $s > 0$,*

$$\Pr\left[D(k) \geq M\mu'(s)\right] = e^{M[\mu(s)-s\mu'(s)]}\left[\frac{1}{|s|\sqrt{\pi 2M\mu''(s)}} + o\left(\frac{1}{\sqrt{M}}\right)\right].$$

where μ' and μ'' denote the first and second-order derivatives of μ.

From this theorem, we can compute accurate formulas for the tail of the distribution of $D(k)$ by computing the semi-invariant moment generating function in the special case where all $D_x^{(i)}(k)$ follow a Bernoulli distribution. This computation is detailed in Appendix A.3 and leads us to Theorem 3. The result is expressed using the Kullback-Leibler divergence.

Definition 5. *Let $0 < x < 1$ and $0 < y < 1$ be two real numbers, the Kullback-Leibler divergence is defined by:*

$$D(x||y) \stackrel{\text{def}}{=} x\ln\left(\frac{x}{y}\right) + (1-x)\ln\left(\frac{1-x}{1-y}\right).$$

Before giving the result obtained (Theorem 3), let us recall that $N_s = \frac{|\Delta_0|N}{2}$. This quantity appears naturally in the expression of the distribution tails.

Theorem 3. *Let $D(k)$ be a counter as defined in Definition 3 ($D(k)$ is a sum of $N/2$ independent and identically distributed variables and takes values in $\{0,1,\ldots,N_s\}$). We define two functions of τ and q real numbers in $[0,1]$ with $\tau \neq q$:*

$$G_-(\tau,q) \stackrel{\text{def}}{=} e^{-N_s D(\tau||q)}\cdot\left[\frac{q\sqrt{(1-\tau)}}{(q-\tau)\sqrt{2\pi\tau N_s}} + \frac{1}{\sqrt{8\pi\tau N_s}}\right], \tag{2}$$

$$G_+(\tau,q) \stackrel{\text{def}}{=} e^{-N_s D(\tau||q)}\cdot\left[\frac{(1-q)\sqrt{\tau}}{(\tau-q)\sqrt{2\pi N_s(1-\tau)}} + \frac{1}{\sqrt{8\pi\tau N_s}}\right]. \tag{3}$$

Then, the tails of the cumulative distribution function of $D(k)$ can be approximated by:

$$\Pr\left[D(k) \leq \tau N_s\right] = G_-(\tau,p)\left[1 + O\left(\frac{p-\tau}{p}\right)\right],$$

$$\Pr\left[D(k) \geq \tau N_s\right] = G_+(\tau,p)\left[1 + O\left(\frac{p-\tau}{p}\right)\right].$$

By combining the results of Theorem 1 and Theorem 3, we define the following estimate for the cumulative distribution function of the counters $D(k)$.

Proposition 1. *Let $G_{\mathcal{P}}(\tau, q)$ be the cumulative distribution function of the Poisson distribution with parameter qN_s. Let $G_-(\tau, q)$ and $G_+(\tau, q)$ as defined in Theorem 3. We define $G(\tau, q)$ as*

$$G(\tau, q) \stackrel{\text{def}}{=} \begin{cases} G_-(\tau, q) & \text{if } \tau < q - 3 \cdot \sqrt{q/N_s}, \\ 1 - G_+(\tau, q) & \text{if } \tau > q + 3 \cdot \sqrt{q/N_s}, \\ G_{\mathcal{P}}(\tau, q) & \text{otherwise.} \end{cases}$$

The cumulative distribution functions of the counters $D(k)$ and $D(k_*)$ can be approximated by G and G_*, where

$$G_*(\tau) \stackrel{\text{def}}{=} G(\tau, p_*) \text{ and } G(\tau) \stackrel{\text{def}}{=} G(\tau, p),$$

with $p_* = \dfrac{\sum_{i,j} p_*^{(i,j)}}{|\Delta_0|}$ and $p = \dfrac{\sum_{i,j} p^{(i,j)}}{|\Delta_0|} \approx \dfrac{|\Delta|}{2^m |\Delta_0|}$ from the wrong-key randomisation hypothesis.

2.4 Data complexity and success probability

For a set Δ_0 that is admissible and if Hypothesis 2 holds, the distributions of the counters are tightly estimated by Proposition 1 and are similar to the distributions involved in [8]. Therefore we can use the same framework to estimate the data complexity and the success probability of a multiple differential cryptanalysis. The results obtained are given in Corollary 1 and Corollary 2.

Corollary 1. *Using notation defined in Section 2.1, the data complexity of a multiple differential cryptanalysis with success probability close to 0.5 is*

$$N = -2 \cdot \frac{\ln(2\sqrt{\pi}\ell\, 2^{-n_k})}{|\Delta_0| D(p_*||p)},$$

where ℓ is the size of the list of the remaining candidates and n_k is the number of bits of the key we want to recover.

Proof. In [8], the authors approximate the tails of the binomial cumulative distribution function by $e^{-N_s \cdot D(\tau||p)} \frac{(1-p)\sqrt{\tau}}{(\tau-p)\sqrt{2\pi N(1-\tau)}}$ to obtain an estimate of the number of samples required to perform a statistical cryptanalysis. Here, the tails of the cumulative distribution function of the counters $D(k)$ are similar (see the definitions of G_- and G_+ given in Theorem 3). Therefore, we can use the same method to derive the required number of samples. We fix the relative threshold τ to p_* which corresponds to a success probability close to 0.5. Then, N_s is found by solving equation $1 - G(p_*) = \frac{\ell}{2^{n_k}}$ (recall that G depends on N_s). In differential cryptanalysis, p_* is quite larger than p hence, $G(p_*) = 1 - G_+(p_*, p)$. Therefore a good estimate of N_s can be found using a fixed point method for solving equation $G_+(p_*, p) = \dfrac{\ell}{2^{n_k}}$. As in [8], we here obtain that N_s is close to $-\frac{1}{D(p_*||p)} \left[\ln\left(\frac{\nu \ell 2^{-n_k}}{\sqrt{D(p_*||p)}} \right) + 0.5 \ln(-\ln(\nu \ell 2^{-n_k})) \right]$ where $\nu \stackrel{\text{def}}{=} \frac{(p_*-p)\sqrt{8\pi(1-p_*)p_*}}{2p_*(1-p)+(p_*-p)\sqrt{1-p_*}}$. As proposed in [8], $\ln(2\sqrt{\pi D(p_*||p)})$ can be used as a good estimate of $\ln(\nu)$, implying that the number of samples N_s is close to $-\frac{\ln(2\sqrt{\pi}\ell\, 2^{-n_k})}{D(p_*||p)}$. The result finally follows from the fact that the number of plaintexts is $N = \frac{2N_s}{|\Delta_0|}$.

$\Diamond$

In [8] it is also conjectured that for a value N_s of the form $N_s = -c \cdot \frac{\ln(2\sqrt{\pi}\ell\, 2^{-n_k})}{|\Delta_0| D(p_*||p)}$, the success probability essentially depends on the value of the constant c.

In Corollary 2, we provide an estimate for the success probability of a multiple differential cryptanalysis. This corollary can be proved using arguments similar to the one exposed in the proof of Theorem 3 in [8].

Corollary 2. *Let $G_*(x)$ (resp. $G(x)$) be the estimate of the cumulative distribution function of the counter $D(k_*)$ (resp. of $D(k)$) defined in Proposition 1. The success probability, P_S, of a multiple differential cryptanalysis is given by*

$$P_S \approx 1 - G_* \left[G^{-1} \left(1 - \frac{\ell - 1}{2^{n_k} - 2} \right) - 1 \right] \tag{4}$$

where the pseudo-inverse of G is defined by $G^{-1}(y) = \min\{x | G(x) \geq y\}$.

2.5 Application to known differential cryptanalyses

Intuitively speaking, exploiting more differentials should decrease the cost of the attack since we extract more information on the same key. Nevertheless, this intuition is not always true. Let n_k be the number of key-bits to recover and let us fix the size of the list to ℓ. Then, for a fixed c, taking N_s of the form $N_s = -c \cdot \frac{\ln(2\sqrt{\pi}\ell\, 2^{-n_k})}{|\Delta_0| D(p_*||p)}$, leads to the same success probability whatever is the set of differentials considered (and hence whatever are the values of $|\Delta_0|, p_*$ and p). That means that the greater the value $|\Delta_0| D(p_*||p)$ is, the more information we extract from the samples. This neither takes into account the time complexity for extracting information nor the time complexity for analysing it. More details on these complexities have been given in Section 2.2. We now focus on finding the set of differentials that provides the more information to the attacker.

A general statement on the best way to choose differentials is not so easy to make. Therefore, we will take a look at two particular cases.

Multiple inputs, single output. In [2], Biham and Shamir have exploited several differentials to mount their attack on the DES. The differentials they use have all the same output difference but different input differences. In this case, we have several differences $(\delta_0^{(i)}, \delta_r)$ with probabilities $p_*^{(i)}$ and a corresponding random probability $p^{(i)} \approx 2^{-m}$ when a wrong candidate is used for deciphering. We also assume that differentials are sorted such that the $p_*^{(i)}$ are in decreasing order. The goal is to find a criterion to determine whether adding the best of the remaining differentials decreases the data complexity or not. For a fixed success probability and a fixed size of list, the data complexity decreases if and only if

$$|\Delta_0| D\left(\frac{\sum_{i=1}^{|\Delta_0|} p_*^{(i)}}{|\Delta_0|} \middle|\middle| 2^{-m} \right) \leq (|\Delta_0| + 1) D\left(\frac{\sum_{i=1}^{|\Delta_0|+1} p_*^{(i)}}{|\Delta_0| + 1} \middle|\middle| 2^{-m} \right). \tag{5}$$

This implies for instance that, if we have a set of differentials with several input differences and with the same probability, exploiting them will decrease the data complexity by a factor $|\Delta_0|$ compared to a simple differential attack that uses only one of them.

Single input, multiple outputs. Some truncated differential attacks [3] can be seen as multiple differential cryptanalyses with a single input and multiple outputs. Here we assume that we exploit several differentials $(\delta_0, \delta_r^{(j)})$ with probability $p_*^{(j)}$ for the correct

subkey. We assume that the $p_*^{(j)}$ are sorted in decreasing order. Adding one more differential with the same input decreases the data complexity until

$$D\left(\sum_{j=1}^{|\Delta_r|} p_*^{(j)} \middle|\middle| 2^{-m}|\Delta_r|\right) \leq D\left(\sum_{j=1}^{|\Delta_r|+1} p_*^{(j)} \middle|\middle| 2^{-m}(|\Delta_r|+1)\right). \tag{6}$$

Moreover, by studying the derivative of the Kullback-Leibler divergence one can obtain that, if $a > b$ and $0 < \lambda \leq a^{-1}$, $D(\lambda a||\lambda b) > \lambda D(a||b)$. Therefore, if we have $|\Delta_r|$ differences with the same input difference and the same probabilities, taking this set of differentials decreases the data complexity by a factor greater than $|\Delta_r|$ compared to a simple differential.

Multiple inputs, multiple outputs. Both previous cases are particular cases of the general situation where the differentials are taken with several input differences and several output differences. Determining the optimal set of differentials that must be chosen to obtain the smallest data complexity is difficult. The reasons are that the differentials do not have the same probabilities and both previously defined criteria use the Kullback-Leibler divergence which is not so easy to study. For all attacks presented in the following sections, we have decided to first determine the optimal set of output differences for each input difference we consider. This has been done using the criterion defined in (6). Then, we have constructed the final set using (5) once the $p_*^{(i)}$'s have been obtained. We do not claim that the resulting set of differentials is optimal but it is an efficient way for choosing the differentials that provides good sets. Finding an algorithm to find the optimal set of differentials (in the sense that provides the more information to the attacker) is an interesting open problem.

3 Experimental validation

In this section we experimentally validate the theoretical framework presented in Section 2. To confirm the tightness of the formulas for the data complexity and the success probability given by Corollary 1 and Corollary 2, we have mounted a multiple differential cryptanalysis on a reduced version of PRESENT namely SMALLPRESENT-[8].

3.1 Description of PRESENT and SMALLPRESENT-[s]

PRESENT is a 64-bit lightweight block cipher proposed at CHES 2007 [15]. It is a Substitution Permutation Network with 16 identical 4-bit S-boxes (see Fig. 4 in Appendix A.2). PRESENT is composed of 31 rounds and is parametrised by a 80-bit or a 128-bit key. The round function is depicted in Fig. 3 in Appendix A.2.

SMALLPRESENT-[s]. For relevant experiments, we need to be able to exhaustively compute the ciphertexts corresponding to all possible plaintexts and for all possible keys. Therefore, we chose to work on a reduced version of PRESENT named SMALLPRESENT-[s] [16]. The family SMALLPRESENT-[s] has been designed to be used for such experiments. Parameter s corresponds to the number of S-boxes per round. The block size is then $4s$. Here, we present the results obtained on SMALLPRESENT-[8] *i.e.* on the version with 8 S-boxes and block size 32 bits. One round of SMALLPRESENT-[8] is depicted in Fig. 4 in Appendix A.2.

Adapting the key-schedule. In the reduced cipher presented in [16], the key-schedule is the same as for the full cipher PRESENT (*i.e.* with a 80-bit master key). But in the original PRESENT, most of the bits of a subkey are directly reused in the next-round subkey, while this is not the case with SMALLPRESENT-[8] since the number of key bits is still 80 but each subkey only uses 32 bits. Then, we decided to modify the key-schedule for our experiments on SMALLPRESENT-[8]. This new key-schedule uses a 40-bit master key and is similar to the one of the full version.

The master key is represented as $K = k_{39}k_{38}\dots k_0$. At round i, the 32-bit round subkey $K_i = k_{39}k_{38}\dots k_8$ consists of the 32 leftmost bits of the current content of the register. After extracting the round key K_i, the key register is updated as follows: the key is rotated by 29 bit positions to the left, the leftmost four bits are passed through the PRESENT S-box, and the *roundcounter* value is XORed with bits $k_{11}k_{10}k_9k_8k_7$.

3.2 Experimental validation of the obtained formulas

To validate the formulas for the data complexity and the success probability given in Corollary 1 and Corollary 2, we have mounted a toy attack on SMALLPRESENT-[8] using both the 40-bit and the 80-bit key-schedules. This attack uses differentials on 9 rounds and aims at recovering some bits of the last-two-round subkeys, *i.e.* it corresponds to an attack on 11 rounds of the cipher.

Design of the toy cryptanalysis. To empirically estimate the success probability of the attack, we have to experiment this multiple differential attack a large number of times. This implies that the number of key bits to recover has to be small enough (*i.e.* not more than 32). We took differentials with output differences of the form `0x????0000`. This structure enable us to recover 16 bits of both last two subkeys. The set of input differences is $\Delta_0 = \{\texttt{0x3}, \texttt{0x5}, \texttt{0x7}, \texttt{0xB}, \texttt{0xD}, \texttt{0xF}\}$. This set is admissible since we can split the set of plaintexts into two parts the even plaintexts and the odd plaintexts. This attack uses 55 differentials over 9 rounds of SMALLPRESENT-[8]. The probability of each differential for both key-schedule (40-bit and 80-bit) has been estimated by a mean over 200 keys. These 55 differentials are given in Appendix A.4 with the estimation of their probabilities. The attack computes the list $\mathcal{L}$ of size $\ell = 2^{12}$ of the likeliest candidates for the last two round subkeys.

Validation of the formula given in Corollary 2. The theoretical success probability of the attack is $P_S = 1 - G_*\left[G^{-1}\left(1 - \frac{\ell-1}{2^{n_k}-2}\right) - 1\right]$, where $G_*(x)$ and $G(x)$ are estimates of the cumulative distribution function of the counter $D(k_*)$ or of $D(k)$. In Fig. 1, we compare the experimental success probability with the theoretical success probabilities obtained using the Gaussian approximation [7], using a Poisson estimation of the distribution of the counters and using the hybrid cumulative function defined in Proposition 1. For both key-schedules, 250 cryptanalyses have been performed to obtain the empirical success rate. The curves obtained for 150, 200 and 250 experiments are quite similar thus we expect that 250 experiments is enough for estimating the success probability. It is worth noticing that the theoretical results in both figures use empirical estimates for the probabilities of the differentials. It is clear from Fig. 1 that the Gaussian approximation used up to now to analyse the complexity of differential cryptanalysis is not the most relevant, as already explained in [7]. Using the Poisson distribution (that provides good results in the case of simple differential cryptanalysis) is not here as good as using the hybrid cumulative function which results from large deviations theory to estimate the tails of the distributions. Since $\frac{\ell-1}{2^{n_k}-2}$ is small, the tightness of the estimate for $G^{-1}\left(1 - \frac{\ell-1}{2^{n_k}-2}\right)$

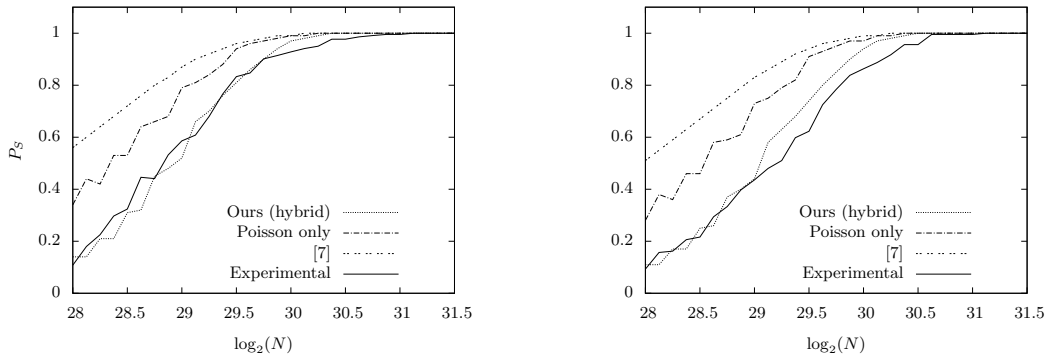


Fig. 1. Comparison of success probabilities for the 40-bit (left) and 80-bit (right) key-schedule.

heavily depends on the accuracy of the tail estimate and thus the hybrid approach is the most relevant one. This result shows that the formula for the success probability given in Corollary 2 is a good approximation of the success probability of a multiple differential cryptanalysis.

Validation of the formula given in Corollary 1. Using the same experiments, we can also confirm the relevance of Corollary 1. It is conjectured in [8] that taking N of the form $N = -2 \cdot c \cdot \frac{\ln(2\sqrt{\pi}\ell\, 2^{-n_k})}{|\Delta_0| D(p_*||p)}$ should lead to a success probability of about 50% for $c = 1$, 80% for $c = 1.5$ and 90% for $c = 2$. In Table 2 we give the empirical success rates corresponding to these three values of N for both attacks on the 40-bit and 80-bit versions of SMALLPRESENT-[8].

Table 2. Empirical success probabilities corresponding to values of N given by Corollary 1.

	$c = 1.0$		$c = 1.5$		$c = 2.0$	
Key-schedule	40-bit	80-bit	40-bit	80-bit	40-bit	80-bit
N	$2^{28.92}$	$2^{29.06}$	$2^{29.50}$	$2^{29.65}$	$2^{29.92}$	$2^{30.06}$
P_S	0.55	0.47	0.83	0.75	0.92	0.88

4 On the estimations of the probabilities p and p_*

We have shown that the formulas given by Corollary 1 and Corollary 2 are well-suited for multiple differential cryptanalysis. But all simulations have been performed on a toy example for which we were able to obtain good estimates of the probabilities of the differentials. However, one of the main difficulties in statistical attacks is the estimation of the underlying probabilities $p_*^{(i,j)}$.

Differential probabilities and trails probabilities. Computing the probability of a differential is, in general, intractable. Indeed, for an r-round differential (δ_0, δ_r), there exist many differential trails that have to be taken into account when computing the probability of this differential.

Definition 6. *A differential trail β on r rounds of a cipher is a $(r+1)$-tuple $(\beta_0, \ldots, \beta_r)$ of elements of $\mathbb{F}_2^n$. Its probability is the probability that a plaintext pair with difference β_0 follows the difference path β when being encrypted: $p_\beta \stackrel{\text{def}}{=} \Pr_{\mathbf{X},\mathbf{K}}\left[\forall i, F^i_{\mathbf{K}}(\mathbf{X}) \oplus F^i_{\mathbf{K}}(\mathbf{X} \oplus \beta_0) = \beta_i\right]$.*

The probability of a differential (δ_0, δ_r) can be computed by summing all the differential trails probabilities with input differences δ_0 and output difference δ_r. For actual ciphers, for a fixed differential, there is a lot of differential trails. This is the reason why, for most ciphers, it is impossible to estimate the exact probability of a differential. Using a branch & bound algorithm similar to the one used in linear cryptanalysis, it is possible to find all possible trails with given input and output differences up to a fixed probability. Summing the corresponding trail probabilities then provides a lower bound on the probability of the differential and thus on the efficiency of the attack.

Key dependence of the probabilities of the differentials. For Markov ciphers, introduced in [11], the classical way of estimating the probability of a differential trail is to use the following theorem.

Theorem 4. *[11] If an r-round iterated cipher is a Markov cipher and the r round keys are independent and uniformly random, then the probability of a differential trail $\beta = (\beta_0, \beta_1, \ldots, \beta_r)$ is*

$$p_\beta = \prod_{i=1}^{r} \Pr_{\mathbf{X},\mathbf{K}} \left[F_{\mathbf{K}}(\mathbf{X}) \oplus F_{\mathbf{K}}(\mathbf{X}') = \beta_i | \mathbf{X} \oplus \mathbf{X}' = \beta_{i-1} \right].$$

The point is that while many recent ciphers are Markov ciphers, their master key is not large enough to lead to independent and uniformly distributed round subkeys and thus, this theorem cannot be applied. Nevertheless, the independence of the round subkeys is generally assumed to obtain an estimate of a differential trail probability.

Hypothesis 3. (Round subkeys independence).
The round subkeys of the cipher E are independent and uniformly random.

Using Theorem 4, we define the theoretical probability of a differential trail $\beta = (\beta_0, \beta_1, \ldots, \beta_r)$ as $p_\beta^t \stackrel{\text{def}}{=} \prod_{i=1}^{r} \Pr_{\mathbf{X},\mathbf{K}} \left[F_{\mathbf{K}}(\mathbf{X}) \oplus F_{\mathbf{K}}(\mathbf{X}') = \beta_i | \mathbf{X} \oplus \mathbf{X}' = \beta_{i-1} \right]$. Hence, one may be able to estimate the probability $\Pr_{\mathbf{X},\mathbf{K}} [\delta_0 \to \delta_r]$ of a differential $\delta = (\delta_0, \delta_r)$ by summing the theoretical probabilities of the trails that compose it: $p_\delta^t \stackrel{\text{def}}{=} \sum_{\beta=(\delta_0,\beta_1\ldots,\beta_{r-1},\delta_r)} p_\beta^t$.

Now, another problem arises: the problem of fixed-key dependence. Theorem 4 can be used to estimate the probability of a differential $\delta = (\delta_0, \delta_r)$ but in an attack, the key is fixed and thus we are interested in the probabilities $p_\delta^K \stackrel{\text{def}}{=} \Pr_{\mathbf{X}} [E_K(\mathbf{X}) \oplus E_K(\mathbf{X} \oplus \delta_0) = \delta_r]$. Most of the analyses assume that this probability does not depend on the key *i.e.*, for two keys K and K', $p_\delta^K = p_\delta^{K'} = p_\delta^t$. This hypothesis is known as the *stochastic independence hypothesis*. It is actually far from being true since evidences show that the values of $2^{m-1} p_\delta^{\mathbf{K}}$ are binomially distributed around $2^{m-1} p_\delta^t$ [17, 18]. Nevertheless, in the setting of multiple differential cryptanalysis, this phenomenon seems to fade. The hypothesis we are using is then the following.

Hypothesis 4. (Stochastic equivalence in the multiple differential setting).
For any key K and for a set Δ of differences large enough, $\sum_{\delta \in \Delta} p_\delta^K = \sum_{\delta \in \Delta} p_\delta^t$.

Impact of the estimation of the probabilities of the differentials on the success probabilities. We have pointed out the problems related to the estimation of the probabilities of the differentials. They come from the large number of trails composing the differential and the fact that their probabilities depend on the key. In our attack on SMALLPRESENT-[8] with the 40-bit key-schedule, we have computed the success probability of the attack based on experimental values for the differential probabilities. We have

also computed the theoretical values of the differential probabilities using trails up to probability 2^{-48}. The theoretical probabilities of the differentials are given in Appendix A.4. We observe that these values always underestimate the probability of the differentials. Using this estimation of the probability we have plot the success rate of the attack (Fig. 2) and we show how this underestimation of the probabilities of the differentials affects the estimation of the success probability of the attack

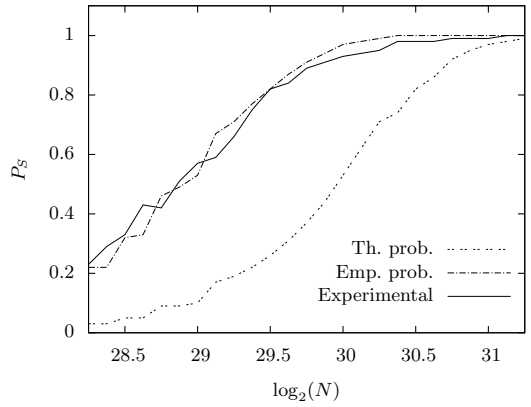


Fig. 2. Success probability of an attack on SMALLPRESENT-[8] with the 40-bit key-schedule

Estimation of p. In the analysis of the distribution of the counters, we have assumed that the $p^{(i,j)}$ were close to 2^{-m} (Hypothesis 1). The probability $p^{(i)}$ of a wrong-key counter to be incremented by a plaintext pair with difference $\delta_0^{(i)}$ has then been estimated by $|\Delta_r^{(i)}|\,2^{-m}$. Thus, p that is the mean of the $p^{(i)}$'s has been estimated by $\frac{|\Delta|}{|\Delta_0|}\,2^{-m}$. We use the results of the experiments on SMALLPRESENT-[8] (Section 3) to show that it is a good estimate for p. Let us recall that we took $|\Delta| = 55$ differentials with $|\Delta_0| = 7$ different input differences. Using the whole codebook we obtain $2^{31} \cdot |\Delta_0|$ samples and thus the expected value of the counters corresponding to wrong subkeys is $2^{31} \cdot |\Delta_0| \left(\frac{55}{|\Delta_0|}\right) 2^{-m} = 27.5$. The mean over the counters corresponding to wrong candidates has been computed for every attack performed and the results are in the range $[27.14; 28.15]$ (the mean value is 27.68). This confirms the relevance of the estimation $p \approx \frac{|\Delta|}{|\Delta_0|}\,2^{-m}$.

5 Application to PRESENT

There exists a lot of attacks on reduced versions of PRESENT. These attacks are summarized in Table 3. The best differential attack on PRESENT is due to Wang [10]. This attack, using 24 14-round differentials having the same output difference, breaks 16 rounds of PRESENT.

We saw in Section 3 that experiments on SMALLPRESENT-[8] corroborate theoretical expectations. Assuming that this holds for the full cipher PRESENT too, we propose a multiple differential cryptanalysis for 18 rounds of PRESENT that improves the attack by Wang. This attack on 18 rounds uses 23 16-round differentials with 16 different input differences. The differentials used are given in Table 4.

This set Δ_0 is admissible (this can be check using the method given in Appendix A.1). For a fixed input the maximum size of the set of output differences is 5:

$$\max_i |\Delta_r^{(i)}| = 5$$

Table 3. Summary of the attacks on PRESENT.

#rounds	version	type of attack	data	time	memory	reference
8	128	integral	$2^{24.3}$	$2^{100.1}$	$2^{77.0}$	[19]
16	80	differential	$2^{64.0}$	$2^{64.0}$	$2^{32.0}$	[10]
17	128	related keys	2^{63}	$2^{104.0}$	$2^{53.0}$	[20]
19	128	algebraic diff.	$2^{62.0}$	$2^{113.0}$	n/r	[21]
24	80	linear	$2^{63.5}$	$2^{40.0}$	$2^{40.0}$	[22]
24	80	statistical sat.	$2^{57.0}$	$2^{57.0}$	$2^{32.0}$	[23]
25	128	linear	$2^{64.0}$	$2^{96.7}$	$2^{40.0}$	[24]
26	80	multiple linear	$2^{64.0}$	$2^{72.0}$	$2^{32.0}$	[9]

δ_0	δ_r	$\log_2(\Pr[\delta_0 \to \delta_r])$
0x1001	0x4040400400000000	-62.21
0x1001	0x40400000000	-62.58
0x1001	0x4000400400000000	-62.84
0x1001	0x4040400000000	-62.84
0x100100000000	0x4040400400000000	-62.97
0x4004	0x4040400400000000	-62.99
0x10010000	0x4040400400000000	-63.13
0x400c	0x4040400400000000	-63.16
0xc004	0x4040400400000000	-63.16
0xc00c	0x4040400400000000	-63.16
0x2002	0x4040400400000000	-63.17
0x1008	0x4040400400000000	-63.21
0x100e	0x4040400400000000	-63.21
0x101	0x4040400400000000	-63.29
0x11	0x4040400400000000	-63.29
0x100100000000	0x40400000000	-63.35
0x200a	0x4040400400000000	-63.37
0xa002	0x4040400400000000	-63.37
0xa00a	0x4040400400000000	-63.37
0x4004	0x40400000000	-63.39
0x1001	0x400400000000	-63.40
0x2004	0x4040400400000000	-63.45
0x4002	0x4040400400000000	-63.45

Table 4. Differentials used in the attack on PRESENT

and each output difference is of the form $\Delta_r \in \{\texttt{0x0?0?0?0?00000000}\}$. The sieves obtained after 18 rounds are similar for each input and of size $|\Delta_{r+2}^{(i)}| \approx 2^{32}$. The differential probabilities have been estimated by summing trails with probability up to 2^{-90} for each differential. The estimates obtained on the involved probabilities are

$$p_* = 2^{-62.59} \quad \text{and} \quad p = 2^{-63.47}.$$

The number of active S-boxes is 4 for the round 17 and 8 for the round 18, implying that the number of bits we recover is 48. In the case of the 80-bit key-schedule, there are 6 bits shared by both two-last-round subkeys and thus we actually recover $n_k = 42$ bits. Moreover, we can use the trick of decomposing the two rounds of the partial deciphering (see [10]). The sieves $\Delta_{r+1}^{(i)}$, that are the sets of possible differences after $r+1$ rounds, are of size at most 2^{12}. We give in Table 5 the complexities of the attack for different values of the data complexity, depending on the size of the list of remaining candidates.

Table 5. Different attacks on PRESENT with memory complexity 2^{32}

80-bit	N	ℓ	P_S	Time complexity
	2^{62}	$\ell = 2^{41}$	73%	$2^{79.00}$
	2^{64}	$\ell = 2^{39}$	77%	$2^{76.00}$
	2^{64}	$\ell = 2^{41}$	98%	$2^{79.00}$

128-bit	N	ℓ	P_S	Time complexity
	2^{62}	$\ell = 2^{47}$	73%	$2^{127.00}$
	2^{64}	$\ell = 2^{44}$	77%	$2^{124.00}$
	2^{64}	$\ell = 2^{47}$	98%	$2^{127.00}$

6 Conclusions

In this paper, we propose a general framework for analysing the complexity of multiple differential cryptanalysis. By studying the distributions of the counters involved in the attack, we give formulas for the data complexity, the time complexity and the success probability of such attacks. We have validated these theoretical results by mounting an attack on SMALLPRESENT-[8]. Using this framework we propose an attack on 18 rounds on PRESENT. This is not the best known attack on PRESENT since linear cryptanalysis seems to perform better on this cipher, but it improves the best previously known differential cryptanalysis of PRESENT [10].

References

1. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In Menezes, A., Vanstone, S.A., eds.: Advances in Cryptology - CRYPTO 1990. Volume 537 of LNCS., Springer (1991) 2–21
2. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology **4** (1991) 3–72
3. Knudsen, L.R.: Truncated and higher order differentials. In Preneel, ed.: Fast Software Encryption - FSE 1994. Volume 1008 of LNCS., Springer (1995) 196–211
4. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In Lee, P.J., ed.: Advances in Cryptology - ASIACRYPT 2004. Volume 3329 of LNCS., Springer (2004) 432–450
5. Baignères, T., Vaudenay, S.: The complexity of distinguishing distributions. In Safavi-Naini, R., ed.: Information Theoretic Security International Conference - ICITS 2008. Volume 5155 of LNCS., Springer (2008) 210–222
6. Blondeau, C., Gérard, B.: On the data complexity of statistical attacks against block ciphers. In Kholosha, A., Rosnes, E., Parker, M.G., eds.: Workshop on Coding and Cryptography - WCC 2009. (2009) 469–488
7. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. Journal of Cryptology **21** (2008) 131–147
8. Blondeau, C., Gérard, B., Tillich, J.P.: Accurate estimates of the data complexity and success probability for various cryptanalyses. DCC special issue on Coding and Cryptography (2010) appear online, DOI: 10.1007/s10623-010-9452-2
9. Cho, J.Y.: Linear cryptanalysis of reduced-round PRESENT. In Pieprzyk, J., ed.: Topics in Cryptology - CT-RSA 2010. Volume 5985 of LNCS., Springer (2010) 302–317
10. Wang, M.: Differential cryptanalysis of reduced-round PRESENT. In Vaudenay, S., ed.: Progress in Cryptology - AFRICACRYPT 2008. Volume 5023 of LNCS., Springer (2008) 40–49
11. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In Davies, D.W., ed.: Advances in Cryptology - EUROCRYPT 1991. Volume 547 of LNCS., Springer (1991) 17–38
12. 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.: Advances in Cryptology - EUROCRYPT 1995. Volume 921 of LNCS., Springer (1995)
13. Le Cam: An approximation theorem for the poisson binomial distribution. In: Pacific Journal of Mathematics. Volume 10. (1960) 1181–1197
14. Gallager, R.G.: Information Theory and Reliable Communication. John Wiley and Sons (1968)
15. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In Paillier, P., Verbauwhede, I., eds.: Cryptographic Hardware and Embedded Systems - CHES 2007. Volume 4727 of LNCS., Springer (2007) 450–466
16. Leander, G.: Small scale variants of the block cipher PRESENT. Cryptology ePrint Archive, Report 2010/143 (2010) http://eprint.iacr.org/2010/143.

17. Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. Journal of Mathematical Cryptology **1** (2007) 12–35
18. Blondeau, C., Gérard, B.: Links between theoretical and effective differential probabilities: Experiments on present. In: TOOLS'10. (2010) http://eprint.iacr.org/2010/261.
19. Z'aba, M., Raddum, H., Henricksen, M., Dawson, E.: Bit-pattern based integral attack. In Nyberg, K., ed.: Fast Software Encryption - FSE 2008. Volume 5086 of LNCS., Springer (2008) 363–381
20. Özen, O., Varici, K., Tezcan, C., Kocair, Ç.: Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT. In Boyd, C., Nieto, J.M.G., eds.: Information Security and Privacy - ACISP 2009. Volume 5594 of LNCS., Springer (2009) 90–107
21. Albrecht, M., Cid, C.: Algebraic techniques in differential cryptanalysis. In Dunkelman, O., ed.: Fast Software Encryption - FSE 2009. Volume 5665 of LNCS., Springer (2009) 193–208
22. 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 - SAC 2009. Volume 5867 of LNCS., Springer (2009) 249–265
23. Collard, B., Standaert, F.X.: A statistical saturation attack against the block cipher PRESENT. In Fischlin, M., ed.: Topics in Cryptology - CT-RSA 2009. Volume 5473 of LNCS., Springer (2009) 195–210
24. Nakahara, J., Sepehrdad, P., Zhang, B., Wang, M.: Linear (hull) and algebraic cryptanalysis of the block cipher PRESENT. Volume 5888 of LNCS., Springer (2009) 58–75

A Appendix

A.1 Checking if a set Δ_0 is admissible

For a set of input differences Δ_0 we want to determine whether this set is admissible, that mean we want to know if it is possible to obtain the value of the counter $D(k)$ by summing $N/2$ of the $D_x(k)$. This is possible if and only if there exists a set $\mathcal{X}$ containing $N/2$ plaintexts such that $\forall \delta_0^{(i)} \in \Delta_0$, $\forall x \in \mathcal{X}, x \oplus \delta_0^{(i)} \notin \mathcal{X}$. This is the case if $\mathcal{X}$ and its complement form the two parts of a bipartite graph where the edges correspond to the $\delta_0^{(i)}$. The existence of such a graph is equivalent to the non-existence of odd weight cycles (*i.e.* null sums of an odd number of $\delta_0^{(i)}$).

Testing this can be efficiently done if we now look at the problem in terms of coding theory. Let G be the matrix whose columns correspond to the binary decompositions of the differences in Δ_0. Then, saying that every odd combination of the columns is non-zero is equivalent to say that the dual of the code determined by G has only codewords with even Hamming weights. Also, this is equivalent to the fact that the dual of this dual code contains the all-one vector. Since the dual of the dual of a code is the original code, we deduce that the set Δ_0 is admissible if and only if the code determined by G contains the all-one vector. This can be tested in polynomial time using a Gaussian elimination. Indeed, putting the matrix G in the systematic form (i.e. $G' = (I||A)$ where I is the identity matrix), the following equivalence holds.

$$(1 \dots 1) \cdot G' = (1 \dots 1) \iff \Delta_0 \text{ is admissible.}$$

A.2 Specification of PRESENT and SMALLPRESENT

The following figures depict the round functions of PRESENT and SMALLPRESENT-[8]. The S-box used in both ciphers is also given.

A.3 Proof of Theorem 3

In this section we use the notation defined in Section 2.1 and Section 2.3. To use Theorem 2 we need to compute the semi-invariant moment generating function of each $D_x(k)$. Let us recall that $D_x(k) = \sum_{i=1}^{|\Delta_0|} D_x(k)^{(i)}$ where $D_x(k)^{(i)}$ follows a Bernoulli distribution with

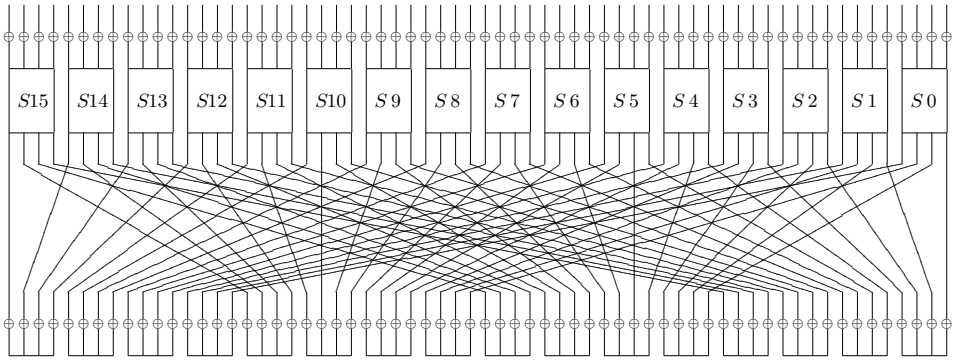
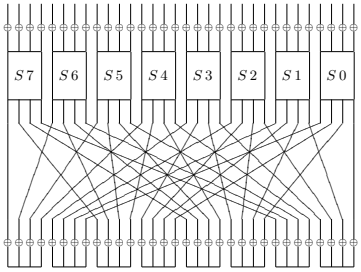


Fig. 3. One round of PRESENT.



x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Fig. 4. One round of SMALLPRESENT-[8] and PRESENT/SMALLPRESENT S-box.

parameter $p^{(i)}$ or $p_*^{(i)}$. For the sake of simplicity, we denote by q_i the parameter of the involved Bernoulli distribution *i.e.* , $q_i = p^{(i)}$ or $q_i = p_*^{(i)}$ and d will denote $\#\Delta_0$. To prove the theorem we need to introduce some notation:

$$\bar{q} \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{d} q_i}{d}, \quad m_2 \stackrel{\text{def}}{=} \frac{\sum_i q_i^2}{d}, \quad s_0 \stackrel{\text{def}}{=} \ln\left(\frac{\tau(1-\bar{q})}{\bar{q}(1-\tau)}\right)$$

The semi-invariant moment generating function [14] of the $D_x(k)$ and its derivatives are

$$\mu(s) = \sum_{i=1}^{d} \ln\left(1 - q_i + q_i e^s\right) \quad , \quad \mu'(s) = \sum_{i=1}^{d} \frac{q_i e^s}{1 - q_i + q_i e^s} \quad \text{and} \quad \mu''(s) = \sum_{i=1}^{d} \frac{q_i e^s (1-q_i)}{(1 - q_i + q_i e^s)^2}.$$

Let s_r be the value such that $\mu'(s_r) = d\tau$. The meaning of Theorem 3 is that substituting s_0 for s_r gives a good estimate of the tails of the distribution. Let f be the function such that $f(s_r) = s_r$:

$$f(s) \stackrel{\text{def}}{=} \ln(d\tau) - \ln\left(\sum_{i=1}^{d} \frac{q_i}{1 - q_i + q_i e^s}\right).$$

We first notice that $\mu''(s) = \mu'(s)(1 - f'(s))$. This, together with the definition of s_r and Theorem 2, leads to the following formula where $M = N/2$.

$$\Pr\left[D(k) \geq d\tau M\right] = e^{M[\mu(s_r) - s_r d\tau]} \left[\frac{1}{|s_r|\sqrt{2\pi d\tau M(1 - f'(s_r))}} + o\left(\frac{1}{\sqrt{M}}\right)\right]. \qquad (7)$$

Now, we are going to quantify the error made substituting s_0 for s_r in (7), but we first need to estimate $f(s_0) - s_0$.

Lemma 1. *Using the previous notation we have*

$$f(s_0) - s_0 = \frac{\tau - \bar{q}}{\bar{q}^2} \cdot (\bar{q}^2 - m_2) + o\left(\frac{\tau - \bar{q}}{\bar{q}^2} \cdot (\bar{q}^2 - m_2)\right).$$

Proof. First, we extract s_0 from the formula.

$$f(s_0) = \ln(d\tau) - \ln\left(\sum_{i=1}^{d} \frac{q_i}{1-q_i+q_i e^{s_0}}\right) = \ln\left(\frac{d\tau}{(1-\tau)\bar{q}}\right) - \ln\left(\sum_{i=1}^{d} \frac{q_i}{(1-q_i)(1-\tau)\bar{q}+q_i(1-\bar{q})\tau}\right)$$
$$= \ln\left(\frac{(1-\bar{q})\tau}{(1-\tau)\bar{q}}\right) - \ln\left(\frac{1-\bar{q}}{d}\sum_{i=1}^{d} \frac{q_i}{(1-q_i)(1-\tau)\bar{q}+q_i(1-\bar{q})\tau}\right)$$
$$= s_0 - \ln\left(\frac{1}{d}\sum_{i=1}^{d} q_i \cdot \frac{1-\bar{q}}{(1-q_i)(1-\tau)\bar{q}+q_i(1-\bar{q})\tau}\right).$$

Then we quantify the difference

$$f(s_0) - s_0 = -\ln\left(\frac{1}{d}\sum_{i=1}^{d} q_i \cdot \frac{1-\bar{q}}{q_i(\tau-\bar{q})+\bar{q}(1-\tau)}\right) = -\ln\left(\frac{1}{d}\sum_{i=1}^{d} \frac{q_i(1-\bar{q})}{\bar{q}(1-\tau)} \cdot \frac{1}{1+\frac{q_i(\tau-\bar{q})}{\bar{q}(1-\tau)}}\right)$$
$$= -\ln\left(\frac{1}{d}\sum_{i=1}^{d} \frac{q_i}{\bar{q}}(1-\bar{q})[1+\tau+o(\tau)]\left[1-\frac{q_i(\tau-\bar{q})}{\bar{q}(1-\tau)}+o\left(\frac{q_i(\tau-\bar{q})}{\bar{q}}\right)\right]\right)$$
$$= -\ln\left(\sum_{i=1}^{d} \frac{q_i}{d\bar{q}} + \sum_{i=1}^{d} \frac{\tau-\bar{q}}{d\bar{q}}\left[q_i - \frac{q_i^2}{\bar{q}}\right] + o\left(\frac{\tau-\bar{q}}{d\bar{q}}\left[q_i - \frac{q_i^2}{\bar{q}}\right]\right)\right)$$
$$= -\ln\left(1+\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)+o\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)\right)$$
$$= \frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)+o\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right).$$

◊

Lemma 2. *Using the previous notation we have*

$$s_r = s_0 + O\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right) \quad \text{and} \quad f'(s_0) = \tau\frac{m_2}{\bar{q}^2} + o\left(\tau\frac{m_2}{\bar{q}^2}\right).$$

Proof. The Taylor expansion of f is $f(s_r) = f(s_0) + (s_r - s_0)f'(s_0) + O\left(f''(s_0)(s_r-s_0)^2\right)$. Thus, since $f(s_r) = s_r$, we get $s_r = s_0 + O\left(\frac{f(s_0)-s_0}{1-f'(s_0)}\right)$.

By definition, $f'(s_0) = \sum_{i=1}^{d} \frac{q_i^2 e^{s_0}}{(1-q_i+q_i e^{s_0})^2} \cdot \left[\sum_{i=1}^{d} \frac{q_i}{1-q_i+q_i e^{s_0}}\right]^{-1}$ and $e^{s_0} = \tau/\bar{q} + o(\tau/\bar{q})$. Therefore,

$$f'(s_0) = \left[\sum_{i=1}^{d} q_i^2 e^{s_0}(1+o(1))\right] \cdot \left[\sum_{i=1}^{d} q_i(1-o(1))\right]^{-1}$$
$$= \left[\frac{d\tau}{\bar{q}}m_2 + o\left(\frac{d\tau}{\bar{q}}m_2\right)\right] \cdot (d\bar{q})^{-1}\Big[1+o(1)\Big],$$

leading to $f'(s_0) = \tau\frac{m_2}{\bar{q}^2} + o\left(\tau\frac{m_2}{\bar{q}^2}\right)$. Then, using the fact that $\frac{1}{1-f'(s_0)} = O(1)$ and Lemma 1, we obtain that

$$s_r = s_0 + O\left(\frac{f(s_0)-s_0}{1-f'(s_0)}\right) = s_0 + O\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right).$$

◊

Lemma 3. *Using the previous notation we obtain*

$$\mu(s_r) = d\ln\left(\frac{1-\bar{q}}{1-\tau}\right) + O\left(d\frac{(\tau-\bar{q})}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\max(\tau-\bar{q},\tau)\right),$$
$$1-f'(s_r) = 1-\tau+O\left(\frac{\max(\tau-\bar{q},\tau)}{\bar{q}^2}(\bar{q}^2-m_2)\right).$$

Proof. Using Lemma 2 we have $e^{s_r} = e^{s_0} \times e^{O\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)} = e^{s_0}\left[1+O\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)\right]$. Therefore,

$$\begin{aligned}\mu(s_r) = \sum_{i=1}^{d} \ln\left(1-q_i+q_i e^{s_r}\right) &= \sum_{i=1}^{d} \ln\left(1-q_i+q_i e^{s_0}\left[1+O\left(\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)\right]\right)\\ &= \sum_{i=1}^{d} \ln\left(\left[1-q_i+q_i e^{s_0}\right]\left[1+O\left(\frac{q_i}{\bar{q}}\frac{\tau}{\bar{q}}\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)\right]\right)\\ &= \sum_{i=1}^{d} \ln\left(1-q_i+q_i e^{s_0}\right) + O\left(d\tau\cdot\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right).\end{aligned}$$

And finally, $\mu(s_r) = \mu(s_0) + O\left(d\tau\cdot\frac{\tau-\bar{q}}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)$. Moreover,

$$\begin{aligned}\mu(s_0) &= \sum_{i=1}^{d} \ln\left(1-q_i+q_i e^{s_0}\right)\\ &= \sum_{i=1}^{d} \ln\left((1-q_i)\bar{q}(1-\tau)+q_i(1-\bar{q})\tau\right) - d\ln\left(\bar{q}(1-\tau)\right)\\ &= \sum_{i=1}^{d} \ln\left(\bar{q}-q_i\bar{q}-\bar{q}\tau+q_i\tau\right) - d\ln\left(\bar{q}(1-\tau)\right)\end{aligned}$$

$$\begin{aligned}\mu(s_0) &= \sum_{i=1}^{d} \ln\left(\frac{\bar{q}(1-\tau)+q_i(\tau-\bar{q})}{(1-\bar{q})\bar{q}}\right) - d\ln\left(\frac{1-\tau}{1-\bar{q}}\right)\\ &= d\ln\left(\frac{1-\bar{q}}{1-\tau}\right) + \sum_{i=1}^{d} \ln\left(1+\frac{(q_i-\bar{q})(\tau-\bar{q})}{(1-\bar{q})\bar{q}}\right)\\ &= d\ln\left(\frac{1-\bar{q}}{1-\tau}\right) + \sum_{i=1}^{d} \frac{(q_i-\bar{q})(\tau-\bar{q})}{(1-\bar{q})\bar{q}} + O\left(d\frac{(\tau-\bar{q})^2}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right)\\ &= d\ln\left(\frac{1-\bar{q}}{1-\tau}\right) + O\left(d\frac{(\tau-\bar{q})^2}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\right).\end{aligned}$$

Therefore $\mu(s_r) = d\ln\left(\frac{1-\bar{q}}{1-\tau}\right) + O\left(d\frac{(\tau-\bar{q})}{\bar{q}^2}\cdot(\bar{q}^2-m_2)\max(\tau-\bar{q},\tau)\right)$. The second part of the lemma is given by the Taylor expansion of $f'(s_r)$:

$$f'(s_r) = f'(s_0) + O\left(s_0-s_r\right) = \tau\frac{m_2}{\bar{q}^2} + O\left(\frac{\tau-\bar{q}}{\bar{q}^2}(\bar{q}^2-m_2)\right).$$

Therefore

$$\begin{aligned}1-f'(s_r) &= (1-\tau)\left[1+O\left(\frac{\tau(\bar{q}^2-m_2)}{\bar{q}^2(1-\tau)}\right)\right] + O\left(\frac{\tau-\bar{q}}{\bar{q}^2}(\bar{q}^2-m_2)\right)\\ &= (1-\tau) + O\left(\frac{\max(\tau-\bar{q},\tau)}{\bar{q}^2}(\bar{q}^2-m_2)\right).\end{aligned}$$

$\Diamond$

Proof of Theorem 3.
We will use (7), Lemma 2 and Lemma 3. In this part we will consider that τ is greater than $\bar{q}$. First we consider the exponential term.

$$\begin{aligned} e^{M[\mu(s_r)-s_r d\tau]} &= \exp\left[M\, d\ln\left(\frac{1-\bar{q}}{1-\tau}\right) - M\,\ln\left(\frac{\tau(1-\bar{q})}{\bar{q}(1-\tau)}\right) d\tau + O\left(M\, d\tau \frac{\tau-\bar{q}}{\bar{q}^2}(\bar{q}^2-m_2)\right)\right] \\ &= e^{-N_s D(\tau||\bar{q})}\left[1+O\left(N_s\tau\frac{\tau-\bar{q}}{\bar{q}^2}(\bar{q}^2-m_2)\right)\right]. \end{aligned}$$

Then, we focus on the polynomial term of the formula.

$$\begin{aligned} \left[|s_r|\sqrt{2\pi\tau N_s(1-f'(s_r))}\right]^{-1} &= \left[|s_0| + O\left(\frac{\tau-\bar{q}}{\bar{q}^2}(\bar{q}^2-m_2)\right)\right]^{-1} \\ &\quad\times\left[\sqrt{2\pi\tau N_s}\sqrt{1-\tau+O\left(\frac{\tau}{\bar{q}^2}(\bar{q}^2-m_2)\right)}\right]^{-1} \\ &= \left[s_0\sqrt{2\pi\tau N_s(1-\tau)}\left[1+O\left(\frac{\tau}{\bar{q}^2}(\bar{q}^2-m_2)\right)\right]\right]^{-1}. \end{aligned}$$

Since we supposed τ greater than $\bar{q}$, s_0 is positive and,

$$s_0 \stackrel{\text{def}}{=} -\ln\left(\frac{\bar{q}(1-\tau)}{\tau(1-\bar{q})}\right) = -\ln\left(1-\frac{\tau-\bar{q}}{\tau(1-\bar{q})}\right) = \frac{\tau-\bar{q}}{\tau(1-\bar{q})}\left[1+\frac{\tau-\bar{q}}{2\tau(1-\bar{q})}+o\left(\frac{\tau-\bar{q}}{\tau}\right)\right].$$

Thus,

$$\begin{aligned} \left[s_r\sqrt{2\pi\tau N_s(1-f'(s_r))}\right]^{-1} &= \frac{1+O\left(\frac{\tau}{\bar{q}^2}(\bar{q}^2-m_2)\right)}{s_0\sqrt{2\pi\tau N_s(1-\tau)}} \\ &= \frac{\tau(1-\bar{q})}{(\tau-\bar{q})\sqrt{2\pi\tau N_s(1-\tau)}}\cdot\left[1+\frac{(\tau-\bar{q})}{2\tau(1-\bar{q})}+o\left(\frac{\tau-\bar{q}}{\tau}\right)\right] \\ &= \frac{\sqrt{\tau}(1-\bar{q})}{(\tau-\bar{q})\sqrt{2\pi N_s(1-\tau)}}+\frac{1}{\sqrt{8\pi T}}+o\left(\frac{1}{\sqrt{T}}\right). \end{aligned}$$

To conclude, we inject those two terms in (7) and get

$$\begin{aligned} \Pr\left[D(k)\geq\tau N_s\right] &= \left[\frac{\sqrt{\tau}(1-\bar{q})}{(\tau-\bar{q})\sqrt{2\pi N_s(1-\tau)}}+\frac{1}{\sqrt{8\pi T}}+o\left(\frac{1}{\sqrt{T}}+\frac{1}{\sqrt{M}}\right)\right] \\ &\quad\times e^{-N_s D(\tau||\bar{q})}\cdot\left[1+O\left(N_s\tau\frac{\tau-\bar{q}}{\bar{q}^2}(\bar{q}^2-m_2)\right)\right] \\ &= e^{-N_s D(\tau||\bar{q})}\left[\frac{\sqrt{\tau}(1-\bar{q})}{(\tau-\bar{q})\sqrt{2\pi N_s(1-\tau)}}+\frac{1}{\sqrt{8\pi T}}+o\left(\frac{1}{\sqrt{T}}\right)\right]. \end{aligned}$$

The formula for $\Pr\left[D(k)\leq\tau N_s\right]$ can be obtained using the same reasoning.

A.4 Differentials used for the toy cryptanalysis

The 55 differentials used for the toy cryptanalysis presented in Section 3.2 are the following.

Table 6. Differentials used in our attack on SMALLPRESENT-[8]. The theoretical probabilities are obtained with trails up to probability 2^{-48}. Probabilities for the 40-bit key-schedule and the 80-bit key-schedule are obtained by a mean over 200 keys.

Differential	Theo.	40-bit	80-bit	Differential	Theo.	40-bit	80-bit
0x3 → 0x40400000	$2^{-30.28}$	$2^{-29.80}$	$2^{-29.85}$	0x5 → 0x40400000	$2^{-30.20}$	$2^{-29.76}$	$2^{-29.80}$
0x3 → 0x04040000	$2^{-30.33}$	$2^{-29.80}$	$2^{-29.84}$	0x5 → 0x04040000	$2^{-30.25}$	$2^{-29.87}$	$2^{-29.73}$
0x3 → 0x50500000	$2^{-30.46}$	$2^{-29.96}$	$2^{-30.07}$	0x5 → 0x50500000	$2^{-30.34}$	$2^{-29.87}$	$2^{-29.76}$
0x3 → 0x05050000	$2^{-30.58}$	$2^{-29.98}$	$2^{-29.99}$	0x5 → 0x10100000	$2^{-30.50}$	$2^{-30.06}$	$2^{-30.28}$
0x3 → 0x10100000	$2^{-30.59}$	$2^{-29.90}$	$2^{-30.10}$	0x5 → 0x05050000	$2^{-30.52}$	$2^{-30.02}$	$2^{-30.06}$
0x3 → 0x01010000	$2^{-30.64}$	$2^{-29.94}$	$2^{-30.45}$	0x5 → 0x01010000	$2^{-30.55}$	$2^{-29.96}$	$2^{-29.94}$
0x3 → 0x80800000	$2^{-30.70}$	$2^{-30.17}$	$2^{-30.24}$	0x5 → 0x08080000	$2^{-30.57}$	$2^{-30.01}$	$2^{-29.97}$
0x3 → 0x08080000	$2^{-30.70}$	$2^{-30.10}$	$2^{-30.01}$	0x5 → 0x80800000	$2^{-30.57}$	$2^{-29.98}$	$2^{-30.04}$
0x3 → 0x0a0a0000	$2^{-30.97}$	$2^{-30.27}$	$2^{-30.32}$	0x5 → 0x0a0a0000	$2^{-30.77}$	$2^{-30.08}$	$2^{-30.04}$
0x7 → 0x40400000	$2^{-29.47}$	$2^{-29.20}$	$2^{-29.21}$	0xB → 0x40400000	$2^{-30.21}$	$2^{-29.60}$	$2^{-29.88}$
0x7 → 0x04040000	$2^{-29.54}$	$2^{-29.23}$	$2^{-23.23}$	0xB → 0x04040000	$2^{-30.26}$	$2^{-29.75}$	$2^{-29.92}$
0x7 → 0x50500000	$2^{-29.59}$	$2^{-29.26}$	$2^{-29.30}$	0xB → 0x50500000	$2^{-30.41}$	$2^{-29.96}$	$2^{-29.99}$
0x7 → 0x10100000	$2^{-29.74}$	$2^{-29.33}$	$2^{-29.70}$	0xB → 0x05050000	$2^{-30.59}$	$2^{-29.97}$	$2^{-30.06}$
0x7 → 0x05050000	$2^{-29.76}$	$2^{-29.37}$	$2^{-29.43}$	0xB → 0x08080000	$2^{-30.64}$	$2^{-29.94}$	$2^{-30.02}$
0x7 → 0x01010000	$2^{-29.86}$	$2^{-29.54}$	$2^{-29.56}$	0xB → 0x80800000	$2^{-30.65}$	$2^{-29.95}$	$2^{-30.06}$
0x7 → 0x0a0a0000	$2^{-30.00}$	$2^{-29.63}$	$2^{-29.65}$	0xB → 0x10100000	$2^{-30.73}$	$2^{-30.13}$	$2^{-30.33}$
0x7 → 0x80800000	$2^{-30.19}$	$2^{-29.61}$	$2^{-29.72}$	0xB → 0x01010000	$2^{-30.81}$	$2^{-30.13}$	$2^{-30.18}$
0x7 → 0x08080000	$2^{-30.21}$	$2^{-29.66}$	$2^{-29.66}$	0xB → 0x0a0a0000	$2^{-30.86}$	$2^{-30.09}$	$2^{-30.10}$
0x7 → 0x40500000	$2^{-30.76}$	$2^{-30.22}$	$2^{-30.09}$	0xF → 0x40400000	$2^{-29.49}$	$2^{-29.26}$	$2^{-29.36}$
0xD → 0x05050000	$2^{-29.81}$	$2^{-29.30}$	$2^{-29.39}$	0xF → 0x04040000	$2^{-29.56}$	$2^{-29.23}$	$2^{-29.31}$
0xD → 0x40400000	$2^{-29.82}$	$2^{-29.42}$	$2^{-29.42}$	0xF → 0x50500000	$2^{-29.80}$	$2^{-29.46}$	$2^{-29.45}$
0xD → 0x04040000	$2^{-29.91}$	$2^{-29.50}$	$2^{-29.46}$	0xF → 0x05050000	$2^{-29.82}$	$2^{-29.39}$	$2^{-29.37}$
0xD → 0x10100000	$2^{-30.01}$	$2^{-29.50}$	$2^{-29.83}$	0xF → 0x80800000	$2^{-29.88}$	$2^{-29.32}$	$2^{-29.37}$
0xD → 0x50500000	$2^{-30.08}$	$2^{-29.60}$	$2^{-29.71}$	0xF → 0x08080000	$2^{-29.88}$	$2^{-29.58}$	$2^{-29.38}$
0xD → 0x01010000	$2^{-30.15}$	$2^{-29.52}$	$2^{-30.14}$	0xF → 0x10100000	$2^{-30.10}$	$2^{-29.69}$	$2^{-29.76}$
0xD → 0x0a0a0000	$2^{-30.25}$	$2^{-29.74}$	$2^{-29.78}$	0xF → 0x01010000	$2^{-30.16}$	$2^{-29.68}$	$2^{-29.94}$
0xD → 0x80800000	$2^{-30.39}$	$2^{-29.82}$	$2^{-29.96}$	0xF → 0x0a0a0000	$2^{-30.22}$	$2^{-29.67}$	$2^{-29.80}$
				0xF → 0x00110000	$2^{-30.60}$	$2^{-29.97}$	$2^{-29.78}$