Multiple Differential Cryptanalysis using LLR and χ^2 Statistics

Céline Blondeau[1⋆], Benoît Gérard[2⋆⋆], and Kaisa Nyberg[1]

[1] Aalto University School of Science, Department of Information and Computer Science
[2] UCL Crypto Group, Université catholique de Louvain, ICTEAM Institute

Abstract. Recent block ciphers have been designed to be resistant against differential cryptanalysis. Nevertheless it has been shown that such resistance claims may not be as tight as wished due to recent advances in this field. One of the main improvements to differential cryptanalysis is the use of many differentials to reduce the data complexity. In this paper we propose a general model for understanding multiple differential cryptanalysis and propose new attacks based on tools used in multidimensional linear cryptanalysis (namely LLR and χ^2 statistical tests). Practical cases are considered on a reduced version of the cipher PRESENT to evaluate different approaches for selecting and combining the differentials considered. We also consider the tightness of the theoretical estimates corresponding to these attacks.
Keywords: block cipher, multiple differential cryptanalysis, statistical test, data complexity.

1 Introduction

Differential cryptanalysis has been introduced in 1990 by Biham and Shamir [2, 3] in order to break the *Data Encryption Standard* block cipher. This statistical cryptanalysis exploits the existence of a *differential*, *i.e.*, 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 attacks, such as truncated differential cryptanalysis or impossible differential cryptanalysis, for instance.

In the original version of differential cryptanalysis [2], a unique differential is exploited. Then, Biham and Shamir improved their attack by considering several differentials having the same output difference [3]. Truncated differential cryptanalysis introduced by Knudsen [18] uses differentials with many output differences that are structured as a linear space. A theoretical framework have recently been proposed to analyze attacks using multiple differentials by summing the corresponding counters [8].

The motivation of this work is to investigate different other techniques for combining information from multiple differentials. As shown in the case of linear cryptanalysis, different approaches may be used depending on the context. In 2004, Biryukov *et al.* proposed a multiple linear cryptanalysis under the assumption that linear approximations

⋆ The research described in this paper by this author has been funded by the Academy of Finland under project 122736 and was partly supported by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II.
⋆⋆ Postdoctoral researcher supported by Walloon region MIPSs project.

are statistically independent[3] [4]. Later Hermelin *et al.* introduced the multidimensional linear cryptanalysis [13, 14]. Contrary to previous attacks, the multidimensional technique focuses on the distribution of the vector of parity bits obtained when applying approximations to a single plaintext/ciphertext pair instead of considering the vector of empirical biases. In that case, the independence assumption is removed but some heuristic might be used when theoretically analyzing the attack. For both approaches, classical statistical tools are used to discriminate the statistic corresponding to the correct key guess from wrong ones. Again, the choice of the tool may depend on the context. For instance, in [9], because of the hardness of profiling the distribution corresponding to the correct key, the attack on PRESENT shows better results using χ^2 than using `LLR` statistic.

Our contributions. Our contributions are threefold. First, we introduce a general way for formalizing differential attacks defining the notion of *partition* functions (this corresponds to the way counters corresponding to output differences are gathered). Second, we consider the χ^2 and the `LLR` statistical tests used in multidimensional linear cryptanalysis as tools for combining information from the groups of differentials determined by the *partition* function. We derive estimates for the data complexities of the corresponding differential attacks. Finally, we present a set of experiments that aim at (i) evaluating the tightness of the estimates derived, (ii) comparing χ^2 and `LLR` combining tools and (iii) comparing different *partition* functions.

The paper is organized as follows. In Section 2 we define the notations and recall some results from ordered statistics that will be used to derive data complexity estimates. Later in Section 3 we present a general model for multiple differential cryptanalysis and introduce the notion of *partition* function. We also link this notion of *partition* function with already published differential attacks. Then, in Section 4, we present two tools for combining information based on the `LLR` and the χ^2 statistical tests. We derive estimates for the corresponding data complexities and also discuss the way of choosing *partition* functions. Finally, Section 5 contains the experiments that have been performed to compare the different methods.

2 Theoretical Background

2.1 Differential Cryptanalysis Against SPN Ciphers

In this paper we consider SPN ciphers that are a subclass of iterated block ciphers. Let m be the block size of the considered cipher E and K the key used for enciphering samples: $E : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2^m (x \mapsto E_K(x))$. Then, since E is an iterated block cipher, it can be expressed as $E_K(x) = F_{K_r} \circ \cdots \circ F_{K_1}(x)$, where F is the round function parameterized by round sub-keys $K_1, \ldots, K_r$.

[3] While not abusive for the DES cipher, this assumption is misleading for new ciphers.

The attack we are interested in is a member of the so-called last-round attacks themselves constituting the major part of statistical cryptanalyses. These last-round[4] attacks use a particular behavior of $F_{K_{r-1}} \circ \cdots \circ F_{K_1}$ (that is often referred as *statistical characteristic*) to partially recover the value of K_r. In the following we will use the compact notation $F_K^{r'} \stackrel{\text{def}}{=} F_{K_{r'}} \circ \cdots \circ F_{K_1}$. The idea is to partially decipher ciphertexts using different values for part of K_r that we name *candidates* (and that we denote by k). In the case of an incorrect guess we obtain outputs corresponding to $F_k^{-1} \circ F_K^r$ while for the correct key guess k_0 the outputs correspond to F_K^{r-1} and thus the statistical characteristic should be observed if enough samples are available. Such attack relies on the assumption that F_K^{r-1} can be distinguished from the set of functions $F_k^{-1} \circ F_K^r$. In practical situations, these last functions actually behave as randomly chosen permutations as stated by the following Wrong Key Randomization Hypothesis.

Hypothesis 1 *(Wrong Key Randomization) Functions $F_k^{-1} \circ F_K^r$ for wrong key candidates k are indistinguishable from randomly chosen permutations.*

Assuming that this hypothesis does not hold would mean that $r+1$ rounds of the cipher are distinguishable hence the attacker should attack more rounds. As a consequence, this hypothesis is quite reasonable as soon as the attacker targets the largest number of rounds he is able to attack (what is typically the case). The resulting attack consists in the three following steps.

1 **Distillation.** For each key candidate ciphertexts are partially deciphered. The number of occurrences of the characteristic is stored for each candidate.
2 **Analysis.** Key candidates are ranked according to the counters computed in the Distillation step.
3 **Search.** Finally, all master keys corresponding to the most likely key candidate are exhaustively tested. If the correct master key is not found then the search step is performed again using the second most likely candidate and so on ...

Differential cryptanalysis. Here we consider the basic differential cryptanalysis which is a last-round attack where the statistical characteristic is an $(r-1)$-round differential. It is a pair of input/output differences (δ_0, δ_{r-1}) and the corresponding probability $p(\delta_0 \rightarrow \delta_{r-1})$,

$$p(\delta_0 \rightarrow \delta_{r-1}) \stackrel{\text{def}}{=} \Pr_{\mathbf{X},\mathbf{K}}\left[F_{\mathbf{K_r}}^{-1}(E_{\mathbf{K}}(\mathbf{X})) \oplus F_{\mathbf{K_r}}^{-1}(E_{\mathbf{K}}(\mathbf{X} \oplus \delta_0)) = \delta_{r-1}\right].$$

Usually, it is assumed that for an incorrect key candidate the probability of observing the differential is $\frac{1}{2^m-1}$. Nevertheless, it has been recalled in [11] that considering that $F_k^{-1} \circ F_K^r$ acts as a random permutation, the distribution of this probability is known to be a Poisson distribution with parameter $\frac{1}{2^m-1}$.

[4] Notice that the attacker may be able to consider less rounds than $r-1$ but for the sake of simplicity we detail the attack assuming one round only is considered.

Using more than one characteristic. Using many characteristics allows the attacker to extract more information from available samples what is of interest as soon as the induced overhead (in both distillation and analysis steps) is negligible compared to the gain in the final search step induced by the additional information obtained (due to the better ranking of the correct key). Premise of this approach have already been proposed in some papers by independently considering different differentials [3] (different analysis phases for different characterics) or by summing the information coming from the different characterics to perform all in one step. In the context of linear cryptanalysis, the method known as *multiple linear cryptanalysis* [17, 4, 12] considers each characteristic independently and proposes to analyze the vectors of information for each key candidate. While the question of characteristics combination have been deeply studied for linear cryptanalysis [17, 4, 12–14], the lack of a comprehensive study on this topic in the context of differential cryptanalysis motivates the present paper. In the following, and after presenting the required background, we propose a general framework and instantiate it with statistical tools already shown to be useful for linear cryptanalysis. Later on, we present experiments we ran to determine what seems to be the best combining technique in practice.

2.2 Order statistics for Gaussian variables

We propose here to recall a result on order statistics for normally distributed random variables that have been used by Selçuk to derive estimates of the data complexity for single linear[5] cryptanalysis [21]. Let us model the attack as follows. We will see later that, due to the tools used, scores obtained will fit into this model.

Model 1 *Let $S(k)$ be the score/statistic obtained for a key candidate k. Then,*

$$S(k) \sim \begin{cases} \mathcal{N}(\mu_R, \sigma_R^2) & \text{if } k = k_0, \\ \mathcal{N}(\mu_W, \sigma_W^2) & \text{otherwise.} \end{cases}$$

Assuming that this model holds then the distributions of ordered wrong-key scores are also normally distributed. This allows expressing the number of required samples for the attack as a function of the minimum rank wished for the correct key and the probability of this rank to be reached. Works have shown that the data complexity of an attack is not influenced by n but by its advantage a [21, 7] that we define now.

Definition 1. *Let 2^n be the number of possible key candidates and ℓ the maximum number of candidates that will be considered in the final search step. Then, the advantage of such attack over exhaustive search is defined as:*

$$a \stackrel{\text{def}}{=} n - \log_2(\ell).$$

The success probability of an attack P_s is the probability that the correct key candidate is ranked among the ℓ first candidates at the output of analysis step.

[5] While single differential cryptanalysis has also been studied in the mentioned paper, results are far from being satisfying as admitted by the author. In that case Poisson distribution is more accurate [11, 6].

The following result expresses the success probability of an attack in the Model 1 as a function of the parameters μ_R, σ_R, μ_W and σ_W. This result is the cornerstone of further data complexity estimate derivations.

Lemma 1. *Let a be the advantage of an attack and N_s be the number of available samples, then, the success probability of the attack P_S can be approximated by:*

$$P_S \approx \Phi_{0,1}\left(\frac{\mu_R - \mu_a}{\sqrt{\sigma_a^2 + \sigma_R^2}}\right),$$

where $\mu_a = \mu_W + \sigma_W \Phi_{0,1}^{-1}(1 - 2^{-a})$, and, $\sigma_a^2 \approx \frac{\sigma_W^2 \, 2^{-(n+a)}}{\varphi_{0,1}^2(\Phi_{0,1}^{-1}(1-2^{-a}))}$.

Proof. The proof follows the one of Theorem 1 in [21]. □

Remark. For the different applications considered in this paper, σ_a turns out to be negligible compared to σ_r and hence we will consider that $\sigma_a^2 + \sigma_R^2 = \sigma_R^2$. Indeed, it can be proved[6] that $\frac{2^{-(n+a)}}{\varphi_{0,1}^2(\Phi_{0,1}^{-1}(1-2^{-a}))} \approx \frac{2^{-n}}{\sqrt{2\pi}}$. In typical cases, n will be large enough for σ_a^2 to be small compared to σ_W^2. Since in the worst observed case, $\sigma_R^2 \approx \sigma_W^2$, then σ_a^2 will also be negligible compared to σ_R^2. Hence, we will use the following approximation for P_s:

$$P_S = \Phi_{0,1}\left(\frac{\mu_R - \mu_W - \sigma_W \Phi_{0,1}^{-1}(1 - 2^{-a})}{\sigma_R}\right). \tag{1}$$

We will discuss this last point later on in the respective sections providing observed values.

3 General Model for Multiple Differential Cryptanalysis

In simple differential cryptanalysis, and contrarily to linear cryptanalysis, one sample is composed of a pair of plaintexts $(x, x \oplus \delta_0)$ and the corresponding ciphertexts $(y = E_K(x), y' = E_k(x \oplus \delta_0))$. Eventually, multiple input differences may be used to perform an attack and then structures should be use to generate more samples from less plaintexts. In the following we will study the complexities of different attacks in terms of the number N_s of required samples to avoid ambiguities. In the case where a single input difference is used then the corresponding data complexity N will be $N = 2N_s$. If more than one input difference is used, then plaintexts should be grouped into structures and then the coefficient 2 in the data complexity may differ.

[6] This result can be derived from the Taylor series of the error function.

3.1 Partition in differential cryptanalysis

In this section we propose a general model for multiple differential cryptanalysis. The aim of such a model is to provide a common language to express various notions of differential cryptanalysis (multiple, improbable, impossible, ...) in such a way that the same analysis tools should be used to evaluate performance of the attacks. This model will also help in the investigation for new techniques that handle multiple characteristics.

From a very abstract point of view, a differential cryptanalysis is composed of two functions.

- First a *sampling function* processes, for each key candidate k, the N_s available samples $(s_i)_{1\leq i\leq N_s}$ and extracts the corresponding difference distributions q^k by normalizing the counters. This function corresponds to the distillation step.

$$\begin{array}{rl} \eta: & \mathbb{F}_{2^{2m}}^{N_s} \times \mathcal{K} \to [0,1]^{2^m} \\ & \big(\{s_1,\ldots,s_{N_s}\},k\big) \mapsto q^k = (q_\delta^k)_{\delta\in\mathbb{F}_{2^m}} \end{array}$$

  where $q_\delta^k = \frac{1}{N_s}\#\big\{s_i = (y_i, y_i'),\ F_k^{-1}(y_i) \oplus F_k^{-1}(y_i') = \delta\big\}$.
- Second, a *scoring function* extract a score for the candidate k from the empirical distribution q^k of observed differences. This function corresponds to the first part of the analysis step (then candidates are ordered from the most likely to the least one).

$$\begin{array}{rl} \psi: [0,1]^{2^m} & \to \mathbb{R} \\ q^k & \mapsto \psi(q^k) \end{array}$$

Since in actual ciphers $m \geq 64$, the storage of distributions q^k is not possible. The solution is to consider smaller distributions. From a general point of view this can be done by projecting the observed differences on a set of smaller cardinality by partitioning the space of output differences. We will show later how known attacks translate into this model. We denote by π such partition function from $\mathbb{F}_{2^m}$ to a set V (we assume that $V = \mathcal{I}m(\pi)$). We can generalize the sampling and scoring functions by considering the partition function π

Model 2 *In differential cryptanalysis, the score of a key candidate is obtained composing the following two functions defined for a given mapping π from $\mathbb{F}_{2^m}$ to a set V.*

$$\begin{array}{rl} \eta_\pi: & \mathbb{F}_{2^{2m}}^{N_s} \times \mathcal{K} \to [0,1]^{|V|} \\ & \big(\{s_1,\ldots,s_{N_s}\},k\big) \mapsto q^k = (q_v^k)_{v\in V} \end{array}$$

where

$$q_v^k \stackrel{\text{def}}{=} \frac{1}{N_s}\#\left\{s_i = (y_i, y_i'),\ \pi\left(F_k^{-1}(y_i) \oplus F_k^{-1}(y_i')\right) = v\right\},$$

and

$$\begin{array}{rl} \psi_\pi: [0,1]^{|V|} & \to \mathbb{R} \\ q^k & \mapsto \psi_\pi(q^k) \end{array}$$

Scoring functions and difference distributions. Later in Section 4 we will instantiate different scoring functions ψ_π. Some of them are based on the knowledge of the theoretical behavior of difference distributions q^k. This behavior obviously depends on the fact that k correspond to the correct key or not. If yes, the distribution q^k will be determined by differential probabilities, while if it is not the case, Hypothesis 1 implies that q^k follows a distribution corresponding to what would be obtained when considering the output of a random permutation. Hence, we place ourselves in the following model

Model 3 *Let k be a sub-key candidate and q^k the corresponding difference distribution obtained by a sampling function η_π. Then,*

$$\Pr\left[q_v^k = x\right] = \begin{cases} \Pr\left[p_v = x\right] \textit{ if } k = k_0, \\ \Pr\left[\theta_v = x\right] \textit{ otherwise.} \end{cases}$$

where distributions p and θ are defined as

$$p_v = \sum_{d \in \pi^{-1}(v)} p(\delta_0 \to d) \quad \textit{and} \quad \theta_v = \frac{1}{\#\pi^{-1}(v)}.$$

Remark. An attack based on partitioning input and output spaces was proposed by Harpes and Cramer in [16]. We would like to stress that such attack uses a partition of the plaintext (resp. ciphertext) space while we consider in this paper partitions of input (resp. output) difference space.

3.2 Partitions and Actual Attacks

Simple/impossible/improbable Differential Attacks. In these attacks, one considers a single differential (δ_0, δ_{r-1}) having an unexpected behavior (eg. a too small probability of occurring). Such cryptanalyses can be represented in our model using the following function identifying differences to the set indexed by $V = \{0, 1\}$.

$$\pi(d) = \begin{cases} 1 \text{ if } d = \delta_{r-1}, \\ 0 \text{ otherwise.} \end{cases}$$

The corresponding scoring function is determined by the number of times the characteristic occurred hence only takes into consideration the value q_1 of the projected distribution.

Truncated Differential Attacks. Truncated differential cryptanalysis [18] is similar to differential cryptanalysis in the sense that usually only one truncated differential characteristic (Δ_0, Δ_{r-1}) is used. Such attacks can be represented in our model in the same way that the previous ones *i.e.* using the projected space $V = \{0, 1\}$ and a similar partition function

$$\pi(d) = \begin{cases} 1 \text{ if } d \in \Delta_{r-1}, \\ 0 \text{ otherwise.} \end{cases}$$

Again, the corresponding scoring function only takes into consideration the value q_1 of the projected distribution.

Multiple Differential Attacks. To improve the performances of differential attacks, information coming from different differentials may be combined. We consider here attacks such that differentials used have the same input difference. We discuss at the end of Section 5 how our model can be extended to the use of multiple input differences. Assuming that the collection of differential $\left(\delta_0, \delta_{r-1}^{(i)}\right)_{i=1,\ldots,A}$ is used, we model the attack with projected space $V = \{0, 1, \ldots, A\}$ and partition function:

$$\pi(d) = \begin{cases} i \text{ if } d = \delta_{r-1}^{(i)}, \\ 0 \text{ otherwise.} \end{cases}$$

4 Instantiations and Complexity Estimates

In this section, we provide instantiations of *scoring functions* and the corresponding estimates for data complexities. Later, in Section 5 we experiment these *scoring functions* using different *partition functions* by attacking a reduced version of PRESENT [5, 19] and discuss the corresponding time and memory complexities.

4.1 The Sum-of-counters Scoring Function

This technique consists in summing counters corresponding to considered differentials. Theoretical analysis of this method is done in [8]. Taking notations of the previous section, the scoring function is determined by $\sum_{i=1}^{A} q_i$ or equivalently by the value $1 - q_0$. In this setting the scores cannot be approximated by a Gaussian distribution and even Poisson approximation leads to pessimistic results. This has been explained in [8] where a formula is given to obtain a better estimate than using Poisson distribution. For more details please refer to [8].

4.2 The `LLR` Scoring Function

The Neyman-Pearson lemma [20] gives the optimal form of the acceptance region on which is derived the `LLR` method. The optimality requires that both p and θ distributions are known (or at least the values p_v/θ_v).

Definition 2. *Let $p = [p_v]_{v \in V}$ be the expected probability distribution vector, θ the uniform one and q^k the observed one for a key candidate k. For a given number of sample N_s, the optimal statistical test consists in comparing the following statistic to a fixed threshold.*

$$\texttt{LLR}(q^k, p, \theta) \stackrel{\text{def}}{=} N_s \sum_{v \in V} q_v \log\left(\frac{p_v}{\theta_v}\right).$$

An important remark here is that, similarly to the case presented in Section 4.1, the `LLR` statistic can be computed with a memory complexity of one floating-point counter per candidate. Indeed, this statistic is a weighted sum of counters for which weights are known before attacking. This test has been applied in [1] by Baignères et al. in the

case of linear cryptanalysis. Applying the law of large numbers, they shown that the LLR statistic tends toward a Gaussian distribution with different means and variances according to the distribution q is extracted from. These means are expressed in terms of relative entropy.

Definition 3. *Let p and p' be two probability distribution vectors over V. The* relative entropy *(aka.* Kullback-Leibler divergence*) between p and p' is*

$$D\left(p||p'\right) \stackrel{\text{def}}{=} \sum_{v \in V} p_v \log\left(\frac{p_v}{p'_v}\right).$$

We also define the following metrics

$$D_2\left(p||p'\right) \stackrel{\text{def}}{=} \sum_{v \in V} p_v \log^2\left(\frac{p_v}{p'_v}\right), \quad \textit{and} \quad \Delta D\left(p||p'\right) \stackrel{\text{def}}{=} D_2\left(p||p'\right) - D\left(p||p'\right)^2.$$

Lemma 2. *(Proposition 3 in [1]) The distributions of $\texttt{LLR}(q^k, p, \theta)$ asymptotically tend toward a Gaussian distribution as the number of samples N_s increases. If samples are obtained from distribution p (resp. θ), the* LLR *statistic tends toward $\mathcal{N}(\mu_R, \sigma_R^2)$ (resp. $\mathcal{N}(\mu_W, \sigma_W^2)$) where,*

$$\begin{aligned} \mu_R &= N_s\, D\left(p||\theta\right) \quad , \quad \mu_W = -N_s\, D\left(\theta||p\right), \\ \sigma_R^2 &= N_s\, \Delta D\left(p||\theta\right) \quad , \quad \sigma_W^2 = N_s\, \Delta D\left(\theta||p\right). \end{aligned}$$

Then, we can use Lemma 1 to obtain the following result.

Theorem 1. *Let a be the advantage of an attack then the number N_s of samples required to reach success probability P_S is*

$$N_s = \frac{\left[\sqrt{\Delta D\left(p||\theta\right)}\,\Phi_{0,1}^{-1}(P_S) + \sqrt{\Delta D\left(\theta||p\right)}\,\Phi_{0,1}^{-1}\left(1-2^{-a}\right)\right]^2}{\left[D\left(p||\theta\right) + D\left(\theta||p\right)\right]^2}. \tag{2}$$

Proof. The proof is based on Lemma 1 and can be found in Appendix A.1. □

4.3 The χ^2 Scoring Function

The aforementioned LLR test is optimal when both distributions are known. In our context, the knowledge of θ relies on Hypothesis 1 and the knowledge of p is based on the possibility of the attacker to theoretically compute differential probabilities. Hence, the use of an alternative statistic may be of interest when one of these two distributions is unknown to the attacker. The χ^2 method has already shown its interest particularly in the context of linear cryptanalysis where the so-called *hull effect* prevents the attacker from estimating the correct-guess distribution [9]. As for linear cryptanalysis, the hardest estimation in the differential case is the correct-guess distribution one, the idea is then to compare the empirical distribution to the incorrect-guess distribution: the vector corresponding to the correct key-guess should end up with one of the largest scores (*i.e.* the smallest probability of being drawn from θ).

Definition 4. *Let q^k be an empirical distribution vector. The χ^2 statistic used to determine the probability of the vector to correspond to a realization from distribution θ is*

$$\chi^2(q^k, \theta) = N_s \sum_{v \in V} \frac{(q_v^k - \theta_v)^2}{\theta_v}.$$

Notice that using χ^2 method, all the counters should be stored since it is not possible to compute the statistic on-the-fly as it was the case when summing counters or for `LLR`. This results in an increased memory cost when using this technique. The following quantity appears when considering the parameters of the χ^2 score distributions.

Definition 5. *Let p be a probability distribution vector over V. The capacity of this vector is defined by*

$$C(p) \stackrel{\text{def}}{=} \sum_{v \in V} \frac{(p_v - \theta_v)^2}{\theta_v}.$$

Lemma 3. *[15] The distribution of $\chi^2(q^k, \theta)$ asymptotically tends toward a Gaussian distribution as the number N_s of samples increases. If samples are obtained from distribution p (resp. θ), the χ^2 statistic tends toward $\mathcal{N}(\mu_R, \sigma_R^2)$ (resp. $\mathcal{N}(\mu_W, \sigma_W^2)$) where,*

$$\begin{aligned} \mu_R &= |V| + N_s C(p) \quad , \quad \mu_W = |V|, \\ \sigma_R^2 &= 2|V| + 4N_s C(p) \quad , \quad \sigma_W^2 = 2|V|. \end{aligned}$$

In [15], Hermelin *et al.* proposed an approximation of the data complexity of a χ^2 statistical test. It turns out that, at least in the present context, the estimate proposed in the following theorem is tighter hence we will use this one.

Theorem 2. *Let $C(p)$ be the capacity of the correct-candidate probability vector p. Then, the number N_s of samples of the corresponding attack with success probability P_S and advantage a can be estimated by*

$$N_s = \frac{\sqrt{2|V|}b + 2t^2 + t(\sqrt{2|V|} + 2b)\sqrt{1 + 4\frac{t^2 - b^2}{(\sqrt{2|V|}+2b)^2}}}{C(p)}, \tag{3}$$

where $b = \Phi_{0,1}^{-1}(1 - 2^{-a})$ and $t = \Phi_{0,1}^{-1}(P_S)$. Fixing the success probability to 0.5, we obtain the following estimate for the number of samples:

$$N_s = \frac{\sqrt{2|V|}\Phi_{0,1}^{-1}(1 - 2^{-a})}{C(p)}. \tag{4}$$

Proof. The proof is based on Lemma 1 and can be found in Appendix A.2. □

4.4 Different Partition Functions

We present here two different types of partition functions. The first one encompass all previously proposed attacks by projecting some considered differences to corresponding elements of V and all others to 0. The second family of partition functions induce a balanced partitioning of the difference space (the sets of differences that are projected to elements of V are all of equal cardinality). This last type of partitioning has (to our knowledge) never been investigated and seems to be the most promising one regarding the motivation of this paper. We will now refer to these two techniques for building partition functions as respectively *balanced* and *unbalanced partitioning*.

Let us recall that we consider that differentials used all have the same input difference, we will explain later how different input differences can be handled.

Unbalanced Partitioning When the attacker knows the probability of some differentials $(\delta_0, \delta_{r-1}^{(i)})_{1\leq i\leq A}$, then the natural way of partitioning is to allocate a counter to each of these differentials. A "trash" counter will gather all other output differences.

$$\pi_{unbal}(d) = \begin{cases} i \text{ if } d = \delta_i, \\ 0 \text{ otherwise.} \end{cases} \tag{5}$$

Let us denote by Δ_{r-1} the set of output differences $\Delta_{r-1} \stackrel{\text{def}}{=} (\delta_{r-1}^{(i)})_{1\leq i\leq A}$. It is likely that this set allows the early discarding of the so-called *wrong pairs i.e.* pairs (y, y') such that, for all candidates k, $F_k^{-1}(y) \oplus F_k^{-1}(y') \notin \Delta_{r-1}$. Using such sieving process allows to decrease the number of partial decryptions in the attack and typically consists in considering active bits in the difference $y \oplus y'$. In our model such wrong pairs will only account for the counters q_0^k. As $\sum_{v=0}^{|V|-1} q_v^k = 1$ ($|V| = A + 1$), for each candidate k, q_0^k can be derived from the other values. The theoretical probability θ_0 is equal to $\theta_0 = 1 - \sum_{v=1}^{|V|-1} \theta_v = 1 - \frac{|V|-1}{2^m-1}$

Balanced Partitioning The alternative we propose results in a balanced partitioning of the space of differences and hence the sieving process will not be as effective as in the case of unbalanced partitioning (if needed at all). Balanced partition functions consider in the experiments have a particular structure linked to truncated differentials. A support s to indicate the set of targeted difference bits, $s \subset \{0, m-1\}$, is determined ($|V| = 2^{|s|}$) and the partition function consists in only considering bits belonging to this support:

$$\pi_{bal}(d) = d|_s = \sum_{i=0}^{|s|} 2^i \cdot d_{s(i)}, \tag{6}$$

where $s(i)$ denote the i-th bit that belong to the support.

What may be considered as an advantage is that such partition functions consider all pairs of plaintexts. Hence more information may be available (at the potential cost of higher time or memory requirements). In this *balanced* model the distribution θ of the wrong key is uniform. That means, taking notation of Model 3, that the quantity θ_v is

equal for all $v \in V$: $\theta_v = \frac{2^m}{|V|}$.
The main drawback of this model is that the differentials are grouped and depending on the way they are gathered the attack may be more or less efficient.

5 Experiments

In this section we experiment different combinations of *partition* and *scoring* functions on 9 rounds of SMALLPRESENT-[8][7] a reduced-version of PRESENT presented in [19]. The goal is to investigate the potential improvements mentioned in Section 4 and to test their robustness in a real attack context (that is with potentially badly estimated distributions). Details about the choices made for experiment parameters not discussed in the following can be found in Appendix A.3.

5.1 On the Choice of Partition Functions

Depending on the targeted cipher the structure of the possible partition functions may differ a lot. Nevertheless using both a *balanced* and an *unbalanced* partitioning (*see.* Equations (6) and (5)) we expect covering a large spectrum of the possibilities in the context of SPN ciphers.

About π_{unbal}*.* Such unbalanced partition function is generally chosen in such a way that an efficient sieve can be performed to discard wrong pairs. In our settings (*see.* Equation (5)), these discarded pairs correspond to the ones that increment counter q_0 for all key candidates. The use of such sieving process leads to an important gain in the time complexity of the partial decryption phase.

The weakness of this kind of partition function is that only few pairs are really useful to the attack (non-discarded pairs). More precisely, for N_s samples and a given index value $v \neq 0$,

$$\#\left\{(y, y') | \pi_{unbal}\left(F_k^{-1}(y) \oplus F_k^{-1}(y')\right) = v\right\} = \mathcal{O}\left(\tfrac{N_s}{2^m}\right), \text{ where } \tfrac{N_s}{2^m} \leq 1.$$

In the context of classical simple differential cryptanalysis this phenomenon is related to the thresholds that can be observed on curves representing success rate or advantage as a function of the number of available samples. When using scoring techniques as the one proposed in this paper, this may explain part of the discrepancies between theoretical and empirical results (particularly in the χ^2 context).

About π_{bal}*.* In the case of *balanced* partition functions, the aforementioned behavior is not observed since all pairs are taken into account. Indeed,

$$\#\left\{(y, y') | \pi_{bal}\left(F_k^{-1}(y) \oplus F_k^{-1}(y')\right) = v\right\} = \mathcal{O}\left(N_s \cdot \theta_v\right), \text{ while } \theta_v = \frac{1}{|V|}.$$

[7] It is an SPN cipher that processes 32-bit blocs using a 40-bit master key. One round is composed of a key addition, a non-lienar layer of 4-bit S-boxes and a bit permutation.

That means that for *balanced* partition functions, if N_s is larger than $|V|$, the noise is reduced[8]. Nevertheless, in such context we generally cannot use an efficient sieving process hence the time complexity of the resulting attack is more important: for each sample a partial decryption of the last round has to be performed. Part of this drawback is removed due to the smaller data complexity but yet both approaches may be of interest depending on the context.

5.2 Experimental Results

The present work proposes to model multiple differential cryptanalysis as the combination of a partition and a scoring function. We derived estimates for the data complexity corresponding to different scoring functions and introduced two families of partition functions. Hence, there are many things that experiments may tell us about the relevance of these tools. We will first discuss the tightness of the estimates for the data complexity we derived. Then, we will focus on the scoring functions and their robustness regarding badly estimated distributions. Thus, we ran experiments in two different contexts:

(i) using "actual" correct-key distribution: this distribution has been obtained by experimentally computing differential probabilities for fixed key then averaging over 200 different keys[9];
(ii) using estimated correct-key distribution: we model the fact that an attacker may only have access to estimates of the differential probabilities by degrading the actual correct-key distribution for a given error rate.

All experiments have been performed targeting 9 rounds of the cipher. The main reason is that the corresponding data complexities are high enough for the attack to make sense and small enough for us to perform enough experiments. For the same reason, we choose size of output spaces $|V|$ in such a way that the counter storage of the resulting attacks can be handled in RAM and that the number of key candidates is at most 2^{16}.

Tightness of the data complexity estimates. The tightness of the data complexity estimates presented in Theorem 1 and Theorem 2 depends on different parameters (the size of the output space, the partition function and so on). It also strongly depends on the correctness of estimates used for the distributions. In order to focus on the validity of provided formulas we ran experiments in the setting (i)[10]: correct-key distribution is the mean obtained from 200 keys. We observe that for both χ^2 (Figure 1) and LLR (Figure 2), formulas provided by Theorem 2 and Theorem 1 give rather good estimates for the data complexity. Thus any observed deviation should not be attributed to an incorrect estimate of the differential probabilities.

Comparison of scoring functions (known distributions). We now consider Figure 1 and Figure 2 in a different way, since we aim at comparing both χ^2 and LLR scoring functions. Obviously the LLR scoring function has much smaller data requirements. For

[8] Intuitively: for a fixed value of $|V|$, the noise is decreasing as the number of sample is increasing
[9] This technique has been shown to provide good results in [6].
[10] Notice that for the χ^2 scoring function, we first computed capacities for different fixed keys and then averaged obtained values.

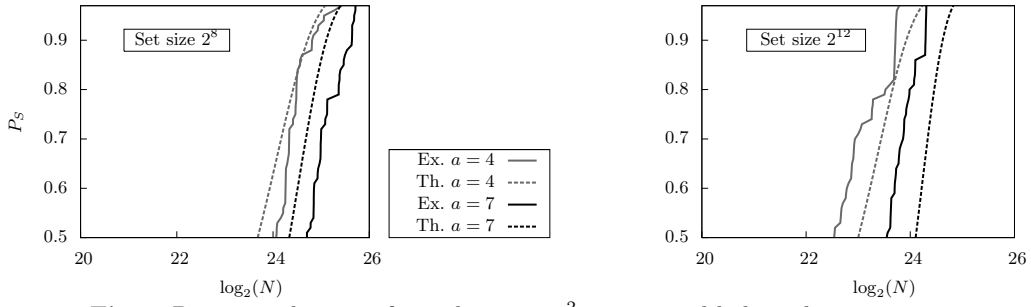

Fig. 1. Data complexities of attacks using χ^2 scoring and balanced partitioning.

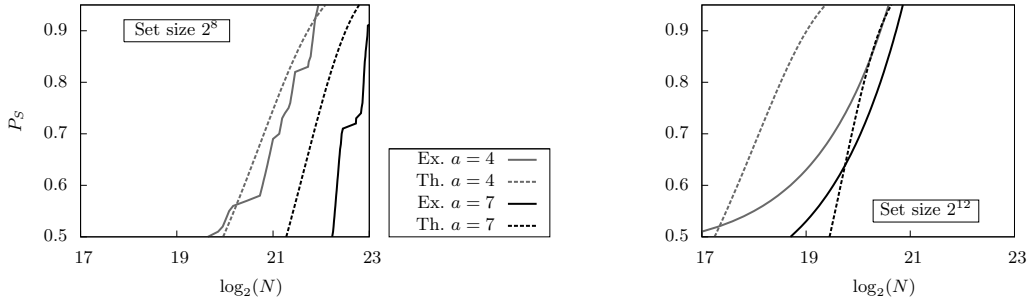

Fig. 2. Data complexities of attacks using `LLR` scoring and balanced partitioning.

instance, for an advantage $a = 7$ and an output space size $|V| = 2^{12}$, it only requires $2^{18.7}$ plaintexts to reach a success probability of one half while $2^{23.55}$ is required using χ^2. This is a natural result since `LLR` attacks are run with actual values of the differential probabilities and hence have more information to process the available data.

Comparison of scoring functions (estimated distributions). In [9], Cho has shown that if the attacker only has a badly estimated correct-key distribution then using the `LLR` statistical test is not relevant anymore. We conducted experiments in that direction assuming that the estimated probability distributions were biased. We emulated this phenomenon by adding some random noise to the distribution estimate (that is $\hat{p_v} = p_v \pm \frac{p_v}{100}$) then normalizing $\hat{p_v}$.

We present in Figure 3 the results of our investigation in the case of a *balanced* partition function with $|V| = 2^8$ (case were the best match is obtained between theory and practice) when the attacker only knows a correct estimate of the distribution. Using both `LLR` or χ^2 scoring functions leads to inaccurate estimations of the data complexity.

It turns out that the noised distribution we obtained can be distinguished from the corresponding uniform distribution θ more easily and hence theoretical expectations are optimistic. For χ^2 method, it can be seen by comparing capacities (the noised distribution has a larger capacity than the actual one) and in the case of `LLR` method this can be seen by looking at the relative entropy between θ and the noised distribution (that is larger). The main information is that this badly estimated distribution does not affect the attack using χ^2 scoring function, what is quite natural since the distribution p is

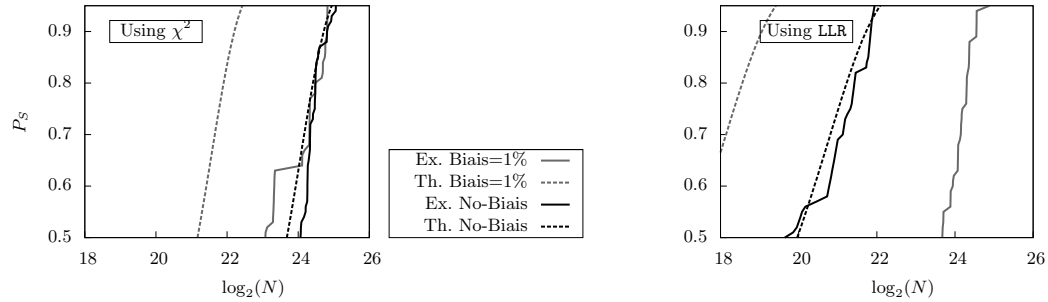

Fig. 3. Data complexities for biased distribution (using balanced partitioning of a set of cardinality $|V| = 2^8$ with advantage $a = 4$).

not involved in the process, while for `LLR` scoring function this induces an overhead in the data complexity. With only a 1% bias, χ^2 scoring function achieve slightly better performance than `LLR` (in terms of data complexity).

Notice that in practice, when instantiating attacks on real ciphers with large state size, it is not so easy to obtain a good estimation of the correct-key distributions. A folklore result is that the differential probability can be underestimated by adding probabilities of corresponding differential trails found using a Branch and Bound algorithm. The main difficulty comes from the choice made by designers known as "wide trail strategy" [10]. Such strategy implies that the number of significant trails in a differential (or linear approximation) exponentially increases with the number of rounds. Experiments made (but not presented in this paper) show that even on SMALLPRESENT-[8] estimating distributions directly using a Branch and Bound algorithm lead to error drastically larger than 1% hence it is likely that in practice an attacker should favor the χ^2 scoring function.

Comparing partition functions. Let us now consider the impact of partition functions used. Figure 1 and Figure 2 are related to experiments that have been performed using the newly introduced *balanced* partitioning. We also ran experiments using the former *unbalanced* partitioning for which an efficient sieving process can be performed (*see.* Figure 4). We chose to perform attacks with an output set of size $|V| = 2^{16}$. The reason is that for smaller sizes corresponding attacks require much more data. Hence, to fairly compare partition functions we used best possible parameters that allow performing enough attacks for plotting results in a given time.

First we observe that due to the use of a sieving process, the theoretical estimates for the data complexity are pretty optimistic (a sketch of explanation is given in Section 5.1). Focusing on experimental curves, we can conclude that from a purely information theoretical point of view, using *balanced partitioning* allows extracting more information from available samples than using *unbalanced* ones. Nevertheless, the cost (in terms of memory and time *see.* Appendix A.4) also has to be considered when comparing both types of partition functions.

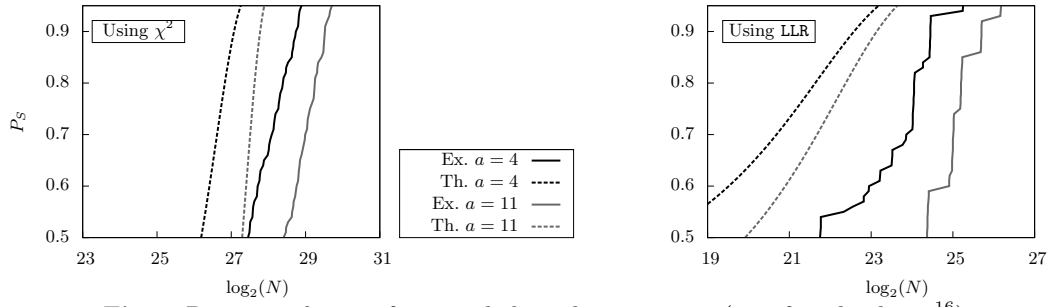

Fig. 4. Data complexities for an unbalanced partitioning (set of cardinality 2^{16}).

On the use of differentials with different input differences. There are two straightforward ways of extending this work to multiple input differences. The first one is to consider the same *partition* function for each input difference so that only one output distribution is considered. The second technique is orthogonal since it consists in considering independently the distributions coming from different input differences. The corresponding scoring functions boils down to summing scores obtained for each distribution.

We ran experiments using both approaches and surprisingly did not obtained radically better results than using a single input difference. Nevertheless, we observed a strong correlation between the distributions obtained that should be exploited. This is a very promising scope for further improvements of this work.

6 Conclusion

This paper builds on the work made on the topic of linear cryptanalysis using multiple approximations. We investigate different statistical tests (namely `LLR` and χ^2) to combine information coming from a large number of differentials while, to our knowledge, only summing counters were considered up to now. To analyze these tools, we introduce a formal way of representing multiple differential cryptanalysis using *partition* functions and present two different families of such functions namely *balanced* and *unbalanced* partitioning (previous attacks being modelled as unbalanced partitioning). Finally, we present experiments performed on a reduced version of PRESENT that confirm the tightness of the data complexity estimates derived in some contexts. These experiments show a relatively good accuracy of the estimates and illustrate the fact that using *balanced* partitioning one is able to take profit of all available pairs.

Further research include exploiting the similarities observed between distributions corresponding to different input differences and solving the challenging problem of estimating correct-key distributions for actual ciphers.

References

1. 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

2. 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
3. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology **4** (1991) 3–72
4. Biryukov, A., Cannière, C., Quisquater, M.: On Multiple Linear Approximations. In: CRYPTO'04. Volume 3152 of *LNCS.*, Springer–Verlag (2004) 1–22
5. 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
6. Blondeau, C., Gérard, B.: Links between theoretical and effective differential probabilities: Experiments on PRESENT. In: TOOLS'10. (2010) http://eprint.iacr.org/2010/261.
7. Blondeau, C., Gérard, B., Tillich, J.-P.: Accurate estimates of the data complexity and success probability for various cryptanalyses. In Charpin, P., Kholosha, S., Rosnes E., and Parker, M.G. eds.: *Designs, Codes and Cryptography.* Volume 59 Numbers 1-3, Springer (2011) 3–34
8. Blondeau, C., Gérard, B.: Multiple Differential Cryptanalysis: Theory and Practice. In Joux, A., ed.: *Fast Software Encryption - FSE 2011.* Volume 6733 of *LNCS.*, Springer (2011) 35 – 54
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. Daemen, J., Rijmen, V.: The Wide Trail Design Strategy. In Honary, B., ed.: *Cryptography and Coding - IMACC 2001.* Volume 2260 of *LNCS.*, Springer (2001) 222–238
11. Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. Journal of Mathematical Cryptology **1** (2007) 12–35
12. Gérard, B., Tillich, J.-P.: On linear cryptanalysis with many linear approximations. In Parker, M. G., ed.: *Cryptography and Coding - IMACC 2009.* Volume 5921 of *LNCS.*, Springer (2009) 112–132
13. Hermelin, M., Cho, J.Y., Nyberg, K.: A new technique for multidimensional linear cryptanalysis with applications on reduced round Serpent. In Pil Joong Lee and Jung Hee Cheon, ed.: *Information Security and Cryptology - ICISC 2008*, volume 5461 of *LNCS.*, Springer (2009) 383–398.
14. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional linear cryptanalysis of reduced round Serpent. In Yi Mu, Willy Susilo, and Jennifer Seberry, ed.: *Information Security and Privacy - ACISP 2008*, volume 5107 of *LNCS.*, Springer (2008) 203–215
15. Hermelin, M., Cho, J.Y., Nyberg.,K.: Multidimensional extension of Matsui's algorithm 2. In Orr Dunkelman, ed.: *Fast Software Encryption - FSE 2009*, volume 5665 of *LNCS.*, Springer (2009) 209–227
16. Harpes, C., Massey, J.L.: Partitioning Cryptanalysis. In Biham, ed.: *Fast Software Encryption - FSE 1997.* Volume 1267 of *LNCS.*, Springer (1997) 13–27
17. Kaliski, B.S., Robshaw, M.J.B.: Linear Cryptanalysis Using Multiple Approximations. In: *Advances in Cryptology - CRYPTO 1994.* Volume 839 of *LNCS.*, Springer–Verlag (1994) 26–39
18. Knudsen, L.R.: Truncated and higher order differentials. In Preneel, ed.: *Fast Software Encryption - FSE 1994.* Volume 1008 of *LNCS.*, Springer (1995) 196–211
19. Leander, G.: Small scale variants of the block cipher PRESENT. Cryptology ePrint Archive, Report 2010/143 (2010) http://eprint.iacr.org/2010/143.
20. Neyman, P., Pearson, E.: On the problem of the most efficient tests of statistical hypotheses. Philosophical Trans. of the Royal Society of London (1933) 289–337
21. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. Journal of Cryptology **21** (2008) 131–147
22. 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

A Proofs

A.1 Proof of Theorem 1

We applied the result of the Lemma 1 to the LLR case. As previously mentioned[11], we consider that $\sigma_R^2 \gg \sigma_a^2$ and hence we neglect the term σ_a^2 to obtain the approximation of P_S given in (1)

$$\begin{aligned} P_S &= \Phi_{0,1}\left(\frac{\mu_R - \mu_W - \sigma_W \Phi_{0,1}^{-1}\left(1-2^{-a}\right)}{\sigma_R}\right), \\ \Phi_{0,1}^{-1}(P_S) &= \frac{N_s\left[D\left(p||\theta\right) + D\left(\theta||p\right)\right] - \sigma_W \Phi_{0,1}^{-1}\left(1-2^{-a}\right)}{\sigma_R}, \\ \sqrt{N_s} &= \frac{\sqrt{\Delta D\left(p||\theta\right)}\,\Phi_{0,1}^{-1}(P_S) + \sqrt{\Delta D\left(\theta||p\right)}\,\Phi_{0,1}^{-1}\left(1-2^{-a}\right)}{D\left(p||\theta\right) + D\left(\theta||p\right)}. \end{aligned}$$

What finally yields (2) and finishes the proof.

A.2 Proof of Theorem 2

From Lemma 1 and assuming[12] $\sigma_a^2 \ll \sigma_R^2$, we can use (1)

$$P_S = \Phi_{0,1}\left(\frac{\mu_R - \mu_W - \sigma_W \Phi_{0,1}^{-1}\left(1-2^{-a}\right)}{\sigma_R}\right).$$

Hence,

$$\Phi^{-1}(P_S) = \frac{N_s C(p) - \sqrt{2|V|}\Phi_{0,1}^{-1}(1-2^{-a})}{\sqrt{2|V| + 4N_s C(p)}}.$$

We observe that the number of samples appears together with the capacity $C(p)$. Let us denote by X the value $N_s C(p)$ and express this equation as a degree-two polynomial then solve it. To lighten notation we denote $\Phi_{0,1}^{-1}(1-2^{-a})$ by b and $\Phi_{0,1}^{-1}(P_S)$ by t:

$$\begin{aligned} \frac{X - \sqrt{2|V|}b}{\sqrt{2|V| + 4X}} &= t, \\ \frac{X^2 - 2X\sqrt{2|V|}b + 2|V|b^2}{2|V| + 4X} &= t^2, \\ X^2 - 2X\sqrt{2|V|}b + 2|V|b^2 - t^2(2|V| + 4X) &= 0, \\ X^2 - 2X(\sqrt{2|V|}b + 2t^2) + 2|V|(b^2 - t^2) &= 0 \end{aligned}$$

[11] In Section 2 we promised figures to illustrate the fact that $\sigma_R^2 \gg \sigma_a^2$. For instance, for $a = 8$ we have $\sigma_R^2/\sigma_a^2 \geq 2^3$ and for $a = 35$ we have $\sigma_R^2/\sigma_a^2 \geq 2^7$.

[12] In the case of χ^2 method σ_W^2 is smaller than σ_R^2 by definition. Hence, regarding the discussion in Section 2, it is obvious that this hypothesis actually holds.

As the data complexity is an increasing function of the success probability, and because $N_s = \frac{X}{C(p)}$, the only meaningful root of this equation is:

$$\begin{aligned} X &= \sqrt{2|V|}b + 2t^2 + \sqrt{(\sqrt{2|V|}b + 2t^2)^2 - 2|V|(b^2 - t^2)}, \\ &= \sqrt{2|V|}b + 2t^2 + t\sqrt{2|V| + 4b\sqrt{2|V|} + 4t^2}, \\ &= \sqrt{2|V|}b + 2t^2 + t\sqrt{(\sqrt{2|V|} + 2b)^2 + 4(t^2 - b^2)}, \\ &= \sqrt{2|V|}b + 2t^2 + t(\sqrt{2|V|} + 2b)\sqrt{1 + 4\frac{t^2 - b^2}{(\sqrt{2|V|} + 2b)^2}}. \end{aligned}$$

In the reasonable case where $P_S = 0.5$, then we obtain a simple formula for N_s that is $N_s = \frac{b\sqrt{2|V|}}{C(p)}$. In the case where b is not too large compared to $|V|$ then, we can use $N_s \approx \frac{(b+t)\cdot(\sqrt{2|V|}+2t)}{C(p)}$. In other situations the square-root term should not be close to 1 and hence should not be neglected.

A.3 Details on experimental parameters

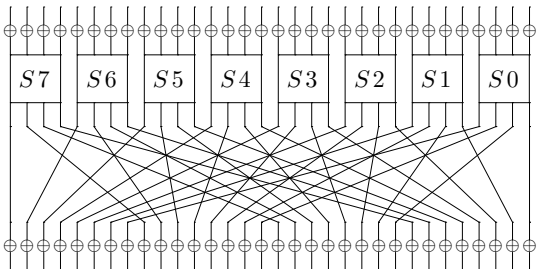

Fig. 5. One round of SMALLPRESENT-[8]

In Section 5, we present different experiments on SMALLPRESENT-[8] (*see.* Figure 5). We provide here explanations and details about the different parameters that were used to conduct these experiments.

Choice of the input difference. All experiments proposed in this paper were performed using 8-round differentials of SMALLPRESENT-[8] all having the same input difference. We ran a Branch and Bound algorithm to find the best 8-round differentials but restricted this search to input differences activating Sboxes in $\{S0, S1, S2, S3\}$ or in $\{S4, S5, S6, S7\}$ (*see.* Figure 5). It turned out that best trails were corresponding to input differences `0x7` and `0xF`. The one of these two differences that was the more promising when considering 8-round distributions was $\delta_0 =$ `0x7` and hence we chose to perform experiments using this difference.

Choice of the output space cardinality. As explain in Section A.4, the time and memory complexities of an attack depend on the partition function used. For instance, for a *balanced* partition function π_{bal}, the time complexity increases with the number of Sboxes that the attacker need to decipher. To conserve a practical time complexity we observed that we can only decipher up to 3 Sboxes for data complexities of order 2^{30}. That is why for the balanced partition functions we propose experiments with vectors of size up to $|V| = 2^{12}$. For the unbalanced case the time complexity of the attack remains practical as soon as $|V| \leq 2^{16}$ and hence we choose to perform tests with $|V| = 2^{16}$.

Choice of S-boxes considered when partitioning. According to the structure of SMALLPRESENT-[8], it seems reasonable to use nibble-oriented partitions. This method allows us to restrict the partial decryption only on the targeted Sboxes and using a subspace of this output differences will only reduce the information that we can collect without modifying a lot the time and memory complexities. As SMALLPRESENT-[8] only have 8 Sboxes, an exhaustive search for the best group of targeted differences was practical[13] and thus has been performed. Among all these combinations, we chose the ones that provided the best expected capacities (hence corresponding to smaller data complexities with the χ^2 scoring function). Summarizing, we chose distributions on 8 rounds to attack 9 rounds of SMALLPRESENT-[8] which correspond to the following targeted Sboxes:

- For the *balanced* partition function π_{bal} with $|V| = 2^8$ the targeted Sboxes are $S7$ and $S6$.
- For the *balanced* partition function π_{bal} with $|V| = 2^{12}$ the targeted Sboxes are $S7$, $S6$ and $S5$.
- For the *unbalanced* partition function π_{unbal} with $|V| = 2^{16}$ the targeted Sboxes are $S7$, $S6$, $S5$ and $S4$.

A.4 Time and memory complexities

When targeting a fixed number n of key-bits and a fixed advantage a, differences in the time and memory complexities between *partition* functions and statistics mainly rely on the score computation. Indeed, the complexities of the Analysis and Search phases are then the same. The aims of this appendix is to give a rough idea of the time and memory complexities of multiple differential attacks using the partition function π_{unbal} or π_{bal} defined in Section 4.4 and using both statistics presented in this paper. Of course this discussion remains very general since the result strongly depends on the block cipher construction.

We assume here, as a reference, that the cost of a partial decryption can be evaluated in terms of $|V|$[14]. Depending of the partition function the number of pairs (y, y') that we partially decipher is different:

[13] This exhaustive search required at most the computation of the expected capacities or the expected distribution vectors for $\binom{8}{4}$ combinations.

[14] In the case of our experiments on SMALLPRESENT-[8], as V corresponds to an union of Sboxes, the cost can be measured in terms of the number of Sbox inversions.

– π_{unbal}: $N_s \cdot \frac{|V|}{2^m}$ partial decryptions are performed.
– π_{bal}: N_s partial decryptions are performed.

For each key candidate a score is computed and the memory complexity of this computation depends on the used statistical: `LLR` or χ^2 in our case. Indeed for the `LLR` test the storage of the vector of counters is not necessary hence for each key a single counter will be used while for χ^2 technique a vector of size $|V|$ will have to be stored for each candidate. To summarize, in the analysis phase $2^n|V|$ counters are required for χ^2 while only 2^n counters are needed for the `LLR`.