# Automatic Search for a Maximum Probability Differential Characteristic in a Substitution-Permutation Network*

Arnaud Bannier, Nicolas Bodin, and Eric Filiol

ESIEA, $(C + V)^O$ Lab, Laval, France,
{bannier, bodin, filiol}@esiea.fr

**Abstract.** The algorithm presented in this paper computes a maximum probability differential characteristic in a Substitution-Permutation Network (or SPN). Such characteristics can be used to prove that a cipher is practically secure against differential cryptanalysis or on the contrary to build the most effective possible attack. Running in just a few second on 64 or 128-bit SPN, our algorithm is an important tool for both cryptanalists and designers of SPN.

## 1 Introduction

### 1.1 Motivation

Modern block ciphers are mainly divided in two categories: Feistel ciphers and Substitution-Permutation Networks, or SPN for short. The encryption process in a Feistel cipher or in a SPN consists in applying a simple operation called round function to the plaintext several times. A different round key is used for each iteration of the round function. These round keys are extracted from a master key using an algorithm called key schedule. Such ciphers are called iterated block ciphers. In a SPN, the round function is made of three distinct stages: a key addition, a substitution layer and a permutation layer.

Differential [3] and linear [12] cryptanalysis are considered as the most important attacks against block ciphers [9]. Therefore, all current ciphers have to resist them. Lai, Massey and Murphy [10] have proposed a formalization for differential cryptanalysis. They clearly exposed the hypothesis made for the attack and introduced Markov ciphers. Since SPN are Markov ciphers, our presentation is based on their work.

A differential predicts that if two plaintexts have a given difference $\alpha$, then the corresponding ciphertexts have a given difference $\beta$ with a certain probability. A differential characteristic is more precise since it gives the difference of intermediate messages for each round. To built a differential crytanalysis, we usually use a differential that ends just before the last round of the cipher and some ciphertext pairs for which we know that the corresponding plaintexts have the required difference.

For each possible value of the last round key, we decrypt the last round of the ciphertext pairs. If the proportion of obtained pairs satisfying the predicted difference is close to the expected probability, then the chosen round key is probably the right one. Once the last round key is found, it is generally not difficult to recover the entire master key. Therefore, finding an effective differential is the most important part in differential cryptanalysis.

The actual cipher security against differential cryptanalysis is evaluated with the differential probabilities. As these probabilities are difficult to compute, four measures of security have been proposed [8]. They can be split in two categories according to the security they imply.

* This article is a minor revision of the version that appears in HICSS-48

– The *provable security* is evaluated by two measures called *precise* and *theoretical*. The precise measure gives the maximum differential probability whereas the theoretical measure upper-bounds it.
– The *practical security* is assessed by two measures called *heuristic* and *practical*. The heuristic measure gives the maximum differential characteristic probability while the practical measure upper-bounds it.

The number of chosen plaintexts and the differential cryptanalysis complexity is inversely proportional to these probabilities [3]. A block cipher is said to have *provable* or *practical security* whenever these measures are lower than a threshold depending on its features.

It should be emphasized that differential characteristic probabilities are computed assuming that the subkeys are independent and uniformly distributed. Although the subkeys are fixed in a classical differential attack, this assumption provides a good approximation of the true probability. This hypothesis, called *stochastic equivalence*, seems to hold for almost all secure ciphers. Furthermore, to the authors' knowledge, there is no practical way to compute the exact probability of a differential.

### 1.2 Previous Works

Under the previous hypothesis, computing a characteristic probability is simple. However, practical security is assessed by the maximum differential characteristic probability and the number of differential characteristic is such that an exhaustive search is intractable.

In [13], Matsui presented an algorithm that find a maximum probability characteristic in a Feistel cipher. Such an algorithm computes the cipher heuristic measure and enables the practical security evaluation. Running it several times on DES, Matsui found a permutation of the S-boxes making the DES stronger against both differential and linear cryptanalysis. While its complexity remains high for the cipher FEAL, two successive improvements have been proposed in [14] then [2].

An adaptation of Matsui's algorithm is possible for SPN. However the block size (64 or 128 bits) of modern ciphers makes the calculations intractable. This fact was also highlighted by Collard et al. [7] who then proposed a few improvements to use this algorithm on the cipher Serpent. Another variation is exposed by Ali and Heys [1]. They gave up finding a maximum probability characteristic to reduce the complexity. On the other side, their algorithm cannot prove cipher practical security, but may still help the cryptanalyst to build an attack.

### 1.3 Contributions

This article presents a search algorithm for a maximal probability differential characteristic in a SPN. Due to the duality between differential and linear cryptanalysis [5], all the results of this article can be adapted to linear cryptanalysis.

The aim of our work is to adapt Matsui's algorithm for SPN but especially to reduce its complexity greatly. Indeed, spending three months in computing the practical security of a known cipher is not a problem. However, the designer has to repeat several times this search in order to optimize the choice of its cipher components (S-boxes, permutation) or the number of rounds.

In the last few years, many lightweight ciphers have been suggested [4,6,15]. They are designed to be implemented in restricted environments such as RFID tags. Consequently, their permutation layers are often bit permutation for efficiency purposes. We have focused our attention on this case and our algorithm allows to analyze practical security of a few cipher systems in just a few seconds.

The security analysis of PRESENT [4], PUFFIN [6] and ICEBERG [15] was performed with the practical measure. Their authors have upper-bounded the probability for a small number of rounds (form 1 to 5) and have then deduced an upper-bound for the full cipher. Our algorithm allows to assess their security more precisely by computing maximum probabilities characteristics.

The following section gives the definitions and notation used in this paper. Section 3 presents a simple adaptation of Matsui's algorithm [13]. Our optimizations are exposed in Sections 4 and 5. Finally, Section 6 describes our results.

## 2 Definitions

A $S$-bit *substitution box* (or *S-box*) is a permutation over $\mathbb{F}_2^S$. A S-Box can be seen as a look-up table. The set of integers from $a$ to $b$ included is denoted $[\![a, b]\!]$. Let $0_n = (0, \ldots, 0)$ denote the identity element of $\mathbb{F}_2^n$.

**Definition 1 (SPN).** *Let $S$ and $N$ be positive integers and $\sigma_1, \ldots, \sigma_N$ be $S$-bit S-boxes. Let us define the following function*

$$\begin{aligned} \sigma : (\mathbb{F}_2^S)^N &\longrightarrow (\mathbb{F}_2^S)^N \\ x = (x_1, \ldots, x_N) &\longmapsto (\sigma_1(x_1), \ldots, \sigma_N(x_N)) \ . \end{aligned}$$

*Let $\pi$ be a bijective $\mathbb{F}_2$-linear mapping from $\mathbb{F}_2^{SN}$ to $\mathbb{F}_2^{SN}$. Let us define the round-function $F$ by*

$$F(k, x) = (\pi \circ \sigma)(x \oplus k) \ ,$$

*for any round key $k$ in $\mathbb{F}_2^{SN}$ and for any message $x$ in $\mathbb{F}_2^{SN}$. The* key addition *is the operation $x \mapsto x \oplus k$ which consists of an exclusive OR of the message $x$ with the round key $k$. The functions $\sigma$ and $\pi$ are respectively called the* substitution layer *and the* permutation layer *of the round function $F$. An iterated cipher having $F$ as round-function is called a* Substitution-Permutation Network *or* SPN *for short.*

*Remark 2.* The last round of a SPN is usually different from the previous ones. Since a differential characterisic ends just before the last round, this article remains relevant.

**Definition 3 (bit permutation).** *A linear mapping $\pi : \mathbb{F}_2^{SN} \to \mathbb{F}_2^{SN}$ is called* bit permutation *if there exists a permutation $\phi$ of $[\![1, SN]\!]$ such that*

$$\pi(x_1, \ldots, x_{SN}) = (x_{\phi^{-1}(1)}, \ldots, x_{\phi^{-1}(SN)}) \ .$$

Throughout the article, we consider a generic given SPN. The basic aim of differential cryptanalysis [3] is to study the propagation of a difference between two plaintexts $x_1$ and $x_1^*$ through the SPN rounds. Let $(k_1, \ldots, k_R)$ denote fixed round keys used for encryption. For each $1 \leq r \leq R$, let us define $x_{r+1} = F(k_r, x_r)$ and $x_{r+1}^* = F(k_r, x_r^*)$. The difference $\alpha_r = x_r \oplus x_r^*$ between $x_r$ and $x_r^*$ is fixed by the round key addition since $(x_r \oplus k_r) \oplus (x_r^* \oplus k_r) = x_r \oplus x_r^* = \alpha_r$. Let $y_r = \sigma(x_r \oplus k_r)$ and $y_r^* = \sigma(x_r^* \oplus k_r)$ denote the outputs of the substitution layer and $\beta_r = y_r \oplus y_r^*$ denote their difference. Note that $\beta_r$ is $k_r$-dependant. The linearity of $\pi$ implies that $\alpha_{r+1} = x_{r+1} \oplus x_{r+1}^* = \pi(y_r) \oplus \pi(y_r^*) = \pi(y_r \oplus y_r^*) = \pi(\beta_r)$. Thus, the input difference of the round $r + 1$ depends only on the output of the round $r$.

*Notation.* Input and output differences of the substitution layer for the round $r$ are respectively referred as

$$\alpha_r = (a_1^r, \ldots, a_N^r) \quad \text{and} \quad \beta_r = (b_1^r, \ldots, b_N^r) \ .$$

These belong to $(\mathbb{F}_2^S)^N$. Whenever an arbitrary round is considered, the index $r$ is omitted and we simply write

$$\alpha = (a_1, \ldots, a_N) \quad \text{and} \quad \beta = (b_1, \ldots, b_N) \ .$$

A difference that can be both in input or output of the substitution layer, is denoted $\gamma = (c_1, \ldots, c_N)$.

In the rest of the paper, the subkeys are assumed to be independent and uniformly distributed. The probability that a difference $a \in \mathbb{F}_2^S$ produces $b \in \mathbb{F}_2^S$ by the $i$-th S-box is given by

$$\mathbb{P}_i(a \to b) = \frac{\#\{x \in \mathbb{F}_2^S \mid \sigma_i(x) \oplus \sigma_i(x \oplus a) = b\}}{2^S} \ . \tag{1}$$

The $2^S \times 2^S$ matrix formed by these probabilities is called the *differential table* of the $i$-th S-box. It should be stressed that $\mathbb{P}_i(0 \to 0) = 1$. The probability that a difference $\alpha \in (\mathbb{F}_2^S)^N$ produces $\beta$ by the substitution layer is

$$\mathbb{P}(\alpha \to \beta) = \prod_{i=1}^{N} \mathbb{P}_i(a_i \to b_i) \ , \tag{2}$$

since the S-boxes are assumed to be independent [10].

**Definition 4 (active S-box).** *Let $\gamma$ be a difference. The $i$-th S-box is* activated by $\gamma$ *if $c_i \neq 0$. Let us define the application*

$$\begin{aligned} \#\mathrm{SB} : (\mathbb{F}_2^S)^N &\longrightarrow [\![1, N]\!] \\ \gamma = (c_1, \ldots, c_N) &\longmapsto \#\{i \in [\![1, N]\!] \mid c_i \neq 0\} \ , \end{aligned}$$

*that relates a difference to the number of S-Boxes it activates.*

**Definition 5 (candidate).** *A* candidate *for an input difference $\alpha$ is an output difference $\beta$ such that $\mathbb{P}(\alpha \to \beta) \neq 0$.*

The following lemma links both previous definitions.

**Lemma 6.** *If $\beta$ is a candidate for $\alpha$, then $b_i = 0_S \Leftrightarrow a_i = 0_S$ for each $i$ such that $1 \leq i \leq N$, that is, they activate the same S-boxes. In this case,*

$$\mathbb{P}(\alpha \to \beta) = \prod_{i, a_i \neq 0_S} \mathbb{P}_i(a_i \to b_i) \ .$$

*Proof.* Assume that $\beta$ is a candidate for $\alpha$. Let $i$ such that $1 \leq i \leq N$. As $\mathbb{P}(\alpha \to \beta) \neq 0$, we have $\mathbb{P}_i(a_i \to b_i) \neq 0$. As S-boxes are one-to-one, the probability $\mathbb{P}_i(a_i \to 0_S)$ is non-zero if and only if $a_i = 0_S$. Further, $\mathbb{P}_i(0_S \to b_i)$ is non-zero only if $b_i = 0_S$. The result follows. □

**Definition 7 (characteristic).** *Let $R$ be a non-negative integer. A $R$-*round differential characteristic *is an element*

$$\mathcal{T} = ((\alpha_1, \beta_1), \ldots, (\alpha_R, \beta_R))$$

*of $((\mathbb{F}_2^{SN})^2)^R$ satisfying $\alpha_{r+1} = \pi(\beta_r)$ for all $1 \leq r < R$. For each $0 \leq i \leq j \leq R$, let $\mathcal{T}_{[i,j]}$ denote the sub-characteristic $((\alpha_i, \beta_i), \ldots, (\alpha_j, \beta_j))$.*

As we have seen, a difference is fixed by the subkey addition and is mapped by the permutation layer almost surely. The subkeys being independently and uniformly distributed, a $R$-round characteristic probability is computed by

$$\mathbb{P}(\mathcal{T}) = \prod_{r=1}^{R} \mathbb{P}(\alpha_r \to \beta_r) = \prod_{r=1}^{R} \prod_{i=1}^{N} \mathbb{P}_i(a_i^r \to b_i^r) \ . \tag{3}$$

**Definition 8 (optimal characteristic).** *A $R$-round characteristic with maximum probability among all the $R$-round characteristics is said* optimal[1]*. In this case, its probability is denoted* $p_{\mathrm{Best}(R)}$.

**Definition 9 (extension).** *Let $r$ and $r'$ be integers such that $0 \leq r \leq r'$. Let $\mathcal{T}$ and $\mathcal{T}'$ be $r$ and $r'$-round characteristics respectively. The characteristic $\mathcal{T}'$* extends *$\mathcal{T}$ if the $r$ first round input and output differences of $\mathcal{T}$ and $\mathcal{T}'$ are equal, that is, $\mathcal{T}'_{[1,r]} = \mathcal{T}$. In this case, $\mathcal{T}' = \mathcal{T} \parallel \mathcal{T}'_{[r+1,r']}$.*

*Example 10.* Consider the SPN SmallPresent(4) [11]. Its parameters are $S = 4$ and $N = 4$. The hexadecimal notation is used for the elements of $\mathbb{F}_2^S = \mathbb{F}_2^4$. For instance, `A` denotes the vector $(1, 0, 1, 0)$. The four S-boxes are defined by

| $a$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\sigma_i(a)$ | C | 5 | 6 | B | 9 | 0 | A | D | 3 | E | F | 8 | 4 | 7 | 1 | 2 |

for each $1 \leq i \leq 4$. Since they are equal, $\mathbb{P}_i(a \to b) = \mathbb{P}_1(a \to b)$ holds for each $i$. Let $\phi$ denote the permutation given by

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\phi(k)$ | 1 | 5 | 9 | 13 | 2 | 6 | 8 | 12 | 3 | 7 | 11 | 15 | 4 | 8 | 12 | 16 |

The linear layer $\pi$ is the bit permutation associated with $\phi$.

Now we turn our attention to the following pairs of input/output differences

$$\begin{array}{lll} \alpha_1 = (0, \mathtt{F}, 0, 0) & \alpha_2 = (0, 0, 0, \mathtt{4}) & \alpha_3 = (0, \mathtt{1}, 0, \mathtt{1}) \\ \beta_1 = (0, \mathtt{1}, 0, 0) & \beta_2 = (0, 0, 0, \mathtt{5}) & \beta_3 = (0, \mathtt{3}, 0, \mathtt{3}) \ . \end{array}$$

Since $\alpha_2 = \pi(\beta_1)$ and $\alpha_3 = \pi(\beta_2)$, they can be concatenated to form longer characteristics. Three rounds of the cipher are represented in Figure 1. The S-boxes activated by $\alpha_1$, $\alpha_2$ and $\alpha_3$ are grayed. It is not hard to check that

$$\mathbb{P}_i(\mathtt{F} \to \mathtt{1}) = \mathbb{P}_i(\mathtt{4} \to \mathtt{5}) = \mathbb{P}_i(\mathtt{1} \to \mathtt{3}) = 2^{-2} \text{ and } \mathbb{P}_i(\mathtt{F} \to \mathtt{2}) = 0$$

with Equation (1). Thus, $\beta_1$ is a candidate for $\alpha_1$ while $(\mathtt{1}, \mathtt{0}, \mathtt{0}, \mathtt{0})$ and $(\mathtt{0}, \mathtt{2}, \mathtt{0}, \mathtt{0})$ are not. Let $\mathcal{T}$ denote the 2-round characteristic $((\alpha_1, \beta_1), (\alpha_2, \beta_2))$. Using Equation (2), we have

$$\mathbb{P}(\alpha_1 \to \beta_1) = \textstyle\prod_{i=1}^{4} \mathbb{P}_i(a_i^1 \to b_i^1) = \mathbb{P}_2(a_2^1 \to b_2^1) = \mathbb{P}_2(\mathtt{F} \to \mathtt{1}) = 2^{-2} \ .$$

Similarly, $\mathbb{P}(\alpha_2 \to \beta_2) = 2^{-2}$ and $\mathbb{P}(\alpha_3 \to \beta_3) = 2^{-4}$. Then $\mathbb{P}(\mathcal{T}) = \mathbb{P}(\alpha_1 \to \beta_1)\mathbb{P}(\alpha_2 \to \beta_2) = 2^{-4}$ according to Equation (3). Once the differential table is computed, it is easy to check that $\mathcal{T}$ is optimal as it activates the minimum number of S-boxes with maximum probabilities. There are several optimal characteristics, for instance

$$\big(((0, 0, 0, \mathtt{F}), (0, 0, 0, \mathtt{1})), ((0, 0, 0, \mathtt{1}), (0, 0, 0, \mathtt{3}))\big)$$

is also optimal. By definition, $\mathcal{T}' = \mathcal{T} \parallel (\alpha_3, \beta_3)$ extends $\mathcal{T}$. It can be shown that $\mathcal{T}'$ is optimal by running the algorithm presented in this paper. In contrast, $\mathcal{T}'_{[2,3]}$ is not optimal as $\mathbb{P}(\mathcal{T}'_{[2,3]}) = 2^{-6} < 2^{-4}$.

[1] Let us mention that there may exists more than one optimal characteristic.

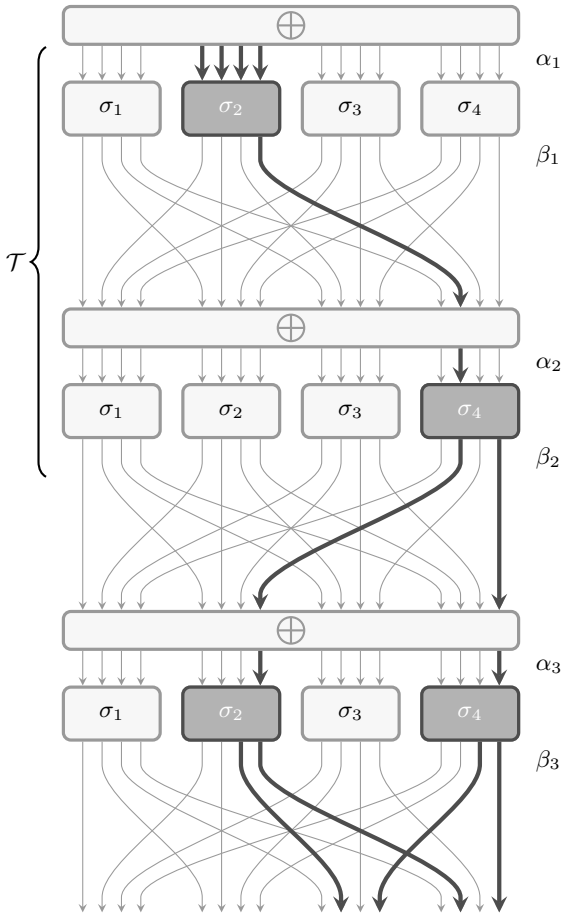



**Fig. 1.** Three rounds of SmallPresent(4)

## 3 Search for an Optimal Characteristic

### 3.1 General Principle

Let us denote $\tilde{R}$ the actual number of rounds. The algorithm presented here computes an optimal $\widetilde{R}$-round characteristic without requiring any a priori knowledge. It is based on the algorithm `OptTrailEst`. The latter accepts an integer $R \geq 2$, the probabilities $(p_{\mathrm{Best}(r)})_{1\leq r<R}$ and an estimation $p_{\mathrm{Estim}}$ of $p_{\mathrm{Best}(R)}$ as arguments and returns an optimal $R$-round characteristic with its probability $p_{\mathrm{Best}(R)}$. The knowledge of $(p_{\mathrm{Best}(r)})_{1\leq r<R}$ and $p_{\mathrm{Estim}}$ speeds up the search. Next, an automatic management of the estimation $p_{\mathrm{Estim}}$ is proposed which gives rise to the algorithm `OptTrail`. Thus, `OptTrail` takes $R$ and $(p_{\mathrm{Best}(r)})_{1\leq r<R}$ as inputs only, and still outputs an optimal $R$-round characteristic.

It should be stressed that $p_{\mathrm{Best}(1)}$ can be easily computed (cf Remark 15). Then, compute

$$p_{\mathrm{Best}(R)} = \texttt{OptTrail}\left(R, (p_{\mathrm{Best}(r)})_{1\leq r<R}\right)$$

for $R$ from 2 to $\widetilde{R}$. The latter computation gives the desired result.

In the rest of this section we explain the principle of `OptTrailEst`. The next section describes several optimizations. Finally, the section 5 presents the automatic management of the estimation $p_{\text{Estim}}$.

**Algorithm** `OptTrailEst`
For each non-zero output difference $\beta_1$,
  Call the search procedure `FirstRound()`
If a characteristic has been found ($\mathcal{E}$ is not empty),
  Return $\mathcal{E}$ and $p_{\text{Estim}}$ ▷ Here, $p_{\text{Best}(R)} = p_{\text{Estim}}$ and $\mathcal{E}$ is optimal
Else
  Return () ▷ Here, $p_{\text{Best}(R)} < p_{\text{Estim}}$
End of the algorithm

**Function** `FirstRound()`
$p_{\text{Rd}(1)} \leftarrow \max_\alpha \mathbb{P}(\alpha \to \beta_1)$
$\alpha_1 \leftarrow \alpha$ such that $\mathbb{P}(\alpha \to \beta_1) = p_{\text{Rd}(1)}$
$\alpha_2 \leftarrow \pi(\beta_1)$
If $R > 2$, then
  Call the search procedure `Round(2)`,
Else
  Call the search procedure `LastRound()`
End of the function ▷ We continue the main loop

**Function** `Round(`$r$`)` ($2 \le r < R$)
For each candidate $\beta_r$ for $\alpha_r$,
  $p_{\text{Rd}(r)} \leftarrow \mathbb{P}(\alpha_r \to \beta_r)$
  If $\prod_{i=1}^{r} p_{\text{Rd}(i)}$ is not lower than the rank-$r$ bound, then
    $\alpha_{r+1} \leftarrow \pi(\beta_r)$
    If $r + 1 < R$, then
      Call the search procedure `Round(`$r+1$`)`
    Else,
      Call the search procedure `LastRound()`
End of the function ▷ We continue `Round(`$r-1$`)` or `FirstRound()`

**Function** `LastRound()`
$p_{\text{Rd}(R)} \leftarrow \max_\beta \mathbb{P}(\alpha_R \to \beta)$
$\beta_R \leftarrow \beta$ such $\mathbb{P}(\alpha_R \to \beta) = p_{\text{Rd}(R)}$
If $\prod_{i=1}^{R} p_{\text{Rd}(i)} \ge p_{\text{Estim}}$, then
  $\mathcal{E} \leftarrow ((\alpha_1, \beta_1), \ldots, (\alpha_R, \beta_R))$ ▷ The current characteristic is saved
  $p_{\text{Estim}} \leftarrow \prod_{i=1}^{R} p_{\text{Rd}(i)} = \mathbb{P}(\mathcal{E})$.
End of the function ▷ We continue `Round(`$R-1$`)` or `FirstRound()`

**Fig. 2.** Search algorithm for an optimal characteristic

Figure 2 describes the algorithm `OptTrailEst`. First, suppose that the condition on the bound in procedure `Round` is true. Under this assumption, the algorithm runs implicitly through the tree of all $R$-round characteristics and saves one which has a maximum probability in $\mathcal{E}$. Observe that the first and last rounds have a special treatment that speeds up the search. When the

program reaches the function `Round(r)`, the current characteristic is

$$\mathcal{T} = ((\alpha_1, \beta_1), \ldots, (\alpha_{r-1}, \beta_{r-1})) \quad \text{and} \quad \mathbb{P}(\mathcal{T}) = \prod_{i=1}^{r-1} \mathbb{P}(\alpha_i \to \beta_i) = \prod_{i=1}^{r-1} p_{\mathrm{Rd}(i)} \ .$$

The input difference $\alpha_r$ for this round equals $\pi(\beta_{r-1})$. Then, for each candidates $\beta_r$ for $\alpha_r$, $\mathcal{T}$ is extended by $(\alpha_r, \beta_r)$ and the search for the next round is called. Therefore, the program performs a depth-first search. When the program reaches the function `LastRound()`, it is not hard to compute the output $\beta_R$ that maximizes the probability of the last round. The characteristic is then saved only if it is better than $\mathcal{E}$. Let us now explain the condition on the bound.

**Definition 11 (rank-$r$ bound).** *Let $\mathcal{T}$ be a $r$-round characteristic with $r < R$. Its probability is* lower than the rank-$r$ bound *if*

$$\mathbb{P}(\mathcal{T}) < \frac{p_{\mathrm{Estim}}}{p_{\mathrm{Best}(R-r)}} \ .$$

This condition on the probability of the current characteristic allows to prune the search tree without missing an optimal characteristic. It can be rewritten $\mathbb{P}(\mathcal{T}) \cdot p_{\mathrm{Best}(R-r)} < p_{\mathrm{Estim}}$ and means that even if the characteristic is extended by an optimal $(R-r)$-round characteristic, the probability of the whole characteristic would be lower than $p_{\mathrm{Estim}}$.

The significance of $p_{\mathrm{Estim}}$ is now clear. If $p_{\mathrm{Estim}} > p_{\mathrm{Best}(R)}$, a characteristic expandable in an optimal $R$-round characteristic can be cut. Furthermore, no characteristic will be saved because of the condition in `LastRound()`. On the other hand, the closer $p_{\mathrm{Estim}}$ is from $p_{\mathrm{Best}(R)}$, the stronger is the pruning condition and the lower is the complexity of `OptTrailEst`.

### 3.2 Proof of the Algorithm

We have explained the general principle of the algorithm. Let us now prove the optimality of the characteristic returned.

**Lemma 12.** *Let $r$ be an integer such that $1 \leq r < R$. Let $\mathcal{T}$ be a $r$-round characteristic whose probability is lower than the rank-$r$ bound. Then there exists no $R$-round characteristic extending $\mathcal{T}$ of probability greater than or equal to $p_{\mathrm{Estim}}$.*

*Proof.* By contradiction, assume that $\mathcal{T}'$ is an extension of $\mathcal{T}$ such that $\mathbb{P}(\mathcal{T}') \geq p_{\mathrm{Estim}}$. Then the probability of the $(R-r)$-round characteristic $\mathcal{T}'_{[r+1,R]}$ is

$$\mathbb{P}\big(\mathcal{T}'_{[r+1,R]}\big) = \frac{\mathbb{P}(\mathcal{T})\mathbb{P}(\mathcal{T}'_{[r+1,R]})}{\mathbb{P}(\mathcal{T})} = \frac{\mathbb{P}(\mathcal{T} \parallel \mathcal{T}'_{[r+1,R]})}{\mathbb{P}(\mathcal{T})} = \frac{\mathbb{P}(\mathcal{T}')}{\mathbb{P}(\mathcal{T})} \ .$$

By assumption, $\mathbb{P}(\mathcal{T}) < p_{\mathrm{Estim}} / p_{\mathrm{Best}(R-r)}$ holds. Note that this strict inequality implies $p_{\mathrm{Estim}} > 0$. It follows that

$$\frac{\mathbb{P}(\mathcal{T}')}{\mathbb{P}(\mathcal{T})} \geq \frac{p_{\mathrm{Estim}}}{\mathbb{P}(\mathcal{T})} > \frac{p_{\mathrm{Estim}}}{p_{\mathrm{Estim}} \,/\, p_{\mathrm{Best}(R-r)}} = p_{\mathrm{Best}(R-r)} \ .$$

By definition of $p_{\mathrm{Best}(R-r)}$, this leads to a contradiction and hence the result follows. □

**Theorem 13 (validity of the algorithm).** *The algorithm* `OptTrailEst` *returns a characteristic $\mathcal{E}$ such that $\mathbb{P}(\mathcal{E}) = p_{\mathrm{Best}(R)}$ if there exists a $R$-round characteristic of probability greater than $p_{\mathrm{Estim}}$, in other words, if $p_{\mathrm{Estim}} \leq p_{\mathrm{Best}(R)}$. Otherwise, the algorithm returns the empty characteristic.*

*Proof.* Suppose the condition on the bound removed. If $p_{\mathrm{Estim}}$ is lower than $p_{\mathrm{Best}(R)}$, an optimal characteristic is saved in $\mathcal{E}$, otherwise $\mathcal{E}$ remains empty. The previous Lemma ensures that the pruning condition avoids only characteristic with probability strictly lower than $p_{\mathrm{Estim}}$. The result still holds. □

## 4 Optimizations

The complexity of this version of `OptTrailEst` is too large to be achievable with real-sized SPN. For example, the first step requires to call the procedure `FirstRound` for all non-zero output differences $\beta_1$. Since there are $2^{SN} - 1$ such differences, we can lower-bound its complexity by $2^{64}$ if $(N, S) = (16, 4)$ and by $2^{128}$ if $(N, S) = (16, 8)$. The optimization of the different parts is the focus of this section.

### 4.1 Construction of the First Difference

As we have just seen, the number of calls to the function `FirstRound()` is a problem that we must now solve. To optimize this step, a partition of the set of all non-zero differences is defined. Then, we give an effective way to test whether no difference in one part can be the beginning of an optimal characteristic.

Let $n$ be an integer such that $1 \leq n \leq N$. The maximum probability of the $n$-th S-box is

$$p_{\mathrm{SB}(i)} = \max_{a,b \in \mathbb{F}_2^S} \mathbb{P}_i(a \to b) \ .$$

Let us sort these probabilities in the decreasing order. This is equivalent to define a permutation $\tau$ of $[\![1, N]\!]$ such that $p_{\mathrm{SB}(\tau(i))} \geq p_{\mathrm{SB}(\tau(i+1))}$ for all $1 \leq i < N$. Let $p_{[n]\text{-SB}}$ denote the maximum probability of a one-round characteristic activating $n$ S-boxes. In other words we have

$$p_{[n]\text{-SB}} = \max_{\substack{\alpha,\beta \in (\mathbb{F}_2^S)^N \\ \#\mathrm{SB}(\alpha)=n}} \mathbb{P}(\alpha \to \beta) \ .$$

**Proposition 14.** *Let $n$ be an integer such that $1 \leq n \leq N$. Then,*

$$p_{[n]\text{-SB}} = \prod_{i=1}^{n} p_{\mathrm{SB}(\tau(i))} \ .$$

*Proof.* Let $\alpha$ be an input difference activating $n$ S-boxes and $\beta$ be an output difference. We will prove that $\prod_{i=1}^{n} p_{\mathrm{SB}(\tau(i))} \geq \mathbb{P}(\alpha \to \beta)$. For each $i$ in $[\![1, N]\!]$, define $q_i = \mathbb{P}_i(a_i \to b_i)$. By definition, $\mathbb{P}(\alpha \to \beta) = \prod_{i=1}^{N} \mathbb{P}_i(a_i \to b_i) = \prod_{i=1}^{N} q_i$. Let $\rho$ be a permutation of $[\![1, N]\!]$ such that $q_{\rho(i)} \geq q_{\rho(i+1)}$. Since the input difference $\alpha$ activates $n$ S-boxes, it must be the case that $q_{\rho(i)} = 0$ for each $i > n$. It follows that $\mathbb{P}(\alpha \to \beta) = \prod_{i=1}^{N} q_i = \prod_{i=1}^{N} q_{\rho(i)} = \prod_{i=1}^{n} q_{\rho(i)}$.

As $\prod_{i=1}^{n} p_{\mathrm{SB}(\tau(i))}$ is the product of the $n$ best probabilities, the inequality $\prod_{i=1}^{n} p_{\mathrm{SB}(\tau(i))} \geq \prod_{i=1}^{n} p_{\mathrm{SB}(\rho(i))}$ holds. Next, $\prod_{i=1}^{n} p_{\mathrm{SB}(\rho(i))} \geq \prod_{i=1}^{n} q_{\rho(i)}$ since $p_{\mathrm{SB}(i)} \geq q_i \geq 0$ for each $i$ in $[\![1, N]\!]$. The result hence follows. □

This proposition makes the computation of $p_{[n]\text{-SB}}$ easy as the probabilities $p_{\mathrm{SB}(i)}$ can be read on the differential tables.

*Remark 15.* As a corollary, the inequalities $p_{[1]\text{-SB}} \geq \ldots \geq p_{[N]\text{-SB}}$ hold. Thus, the probability of an optimal one-round characteristic is

$$p_{\mathrm{Best}(1)} = \max_{\alpha,\beta \in (\mathbb{F}_2^S)^N} \mathbb{P}(\alpha \to \beta) = p_{[1]\text{-SB}} = p_{\mathrm{SB}(\tau(1))} \ .$$

Of course, the differential tables and the probabilities $p_{\mathrm{SB}(i)}$ and $p_{[i]\text{-SB}}$ are precomputed to optimize the search.

**Algorithm** `OptTrailEst`
For $n$ from 1 to $N$,
  If $p_{[n]\text{-SB}}$ is lower than the rank-one bound, then
    Exit the loop ▷ See Theorem 16
  Else,
    For each output difference $\beta_1$ activating $n$ S-boxes,
      Call `FirstRound()`
If a characteristic has been found ($\mathcal{E}$ is not empty), then
  Return $\mathcal{E}$ and $p_{\text{Estim}}$
Else
  Return ()

**Fig. 3.** First optimization – construction of the first difference

**Theorem 16.** *Let $n$ and $n'$ be integers such that $1 \leq n \leq n' \leq N$. If $p_{[n]\text{-SB}}$ is lower than the rank-one bound, then there exists no $R$-round characteristic activating $n'$ S-boxes in the first round with probability greater than or equal to $p_{\text{Estim}}$.*

*Proof.* Assume that $p_{[n]\text{-SB}}$ is lower than the rank-one bound. Let $\mathcal{T}$ be a one-round characteristic activating $n'$ S-boxes. By definition, $\mathbb{P}(\mathcal{T}) \leq p_{[n']\text{-SB}}$. The inequality $p_{[n']\text{-SB}} \leq p_{[n]\text{-SB}}$ follows from Proposition 14, therefore $\mathbb{P}(\mathcal{T}) \leq p_{[n]\text{-SB}}$. Hence, $\mathbb{P}(\mathcal{T})$ is lower than the rank-one bound and Lemma 12 ensures that there exists no $R$-round characteristic extending $\mathcal{T}$ with probability greater than or equal to $p_{\text{Estim}}$. This concludes the proof. □

This theorem states that whenever $p_{[n]\text{-SB}}$ is lower than the rank-one bound, we only have to test the output differences $\beta_1$ activating at most $n-1$ S-boxes. There are

$$\sum_{i=1}^{n-1} \binom{N}{i} (2^S - 1)^i$$

such differences, compared with $2^{SN} - 1$ otherwise.

We have run the final algorithm with several SPN having a bit permutation as linear layer. With $N = 16$ and $S = 4$, $p_{[4]\text{-SB}}$ is always lower than the rank-one bound. There are at most $2^{21}$ differences to be tested instead of $2^{64}$. With $N = 16$ and $S = 8$, the gap is even larger since $p_{[3]\text{-SB}}$ is always lower than the bound. Finally, $2^{21}$ differences instead of $2^{128}$ remain to be tested. The algorithm optimized with Theorem 16 is described in Figure 3.

### 4.2 The Round Function

Following Matsui's algorithm [13], the round candidates are constructed recursively. Let $\alpha$ denote the input difference of the current round. According to Lemma 6, a candidate $\beta$ is constructed by selecting an output for each S-box activated by $\alpha$.

Recall that the *support* of $L$ in $\mathbb{F}_2^N$ is the set $\text{supp}(L) = \{i \in [\![1, N]\!] \mid L_i \neq 0\}$ and its *Hamming weight* is $\omega(L) = \#\,\text{supp}(L)$.

The function $\text{list} : (\mathbb{F}_2^S)^N \to \mathbb{F}_2^N$ maps a difference $\gamma$ to the vector $\text{list}(\gamma) = (x_1, \ldots, x_N)$ where $x_i$ equals 1 if and only if the $i$-th S-box is activated by $\gamma$. It is clear that $\#\text{SB}(\gamma) = \omega(\text{list}(\gamma))$. Further, $\beta$ is a candidate for $\alpha$ if and only if $\text{list}(\alpha) = \text{list}(\beta)$.

Let $L \in \mathbb{F}_2^N$ be a compact representation of active S-boxes and define

$$p_{\text{ListSB}(L)} = \prod_{i \,\in\, \text{supp}(L)} p_{\text{SB}(i)} \ .$$

Let $\vee$ denote the bitwise OR and $\wedge$ denote the bitwise AND. Next, the vector of size $n$ in which the first $n$ coordinates are 1 and the last $N-n$ are 0 is denoted $(0_n 1_{N-n})$.

*Example 17.* With the same notations as in the previous example, $\mathrm{list}(\alpha_3) = \mathrm{list}(\beta_3) = (0,1,0,1)$ and $\mathrm{list}(\alpha_3) \wedge (0_2 1_2) = (0,1,0,1) \wedge (0,0,1,1) = (0,0,0,1)$.

**Theorem 18.** *Let $r$ be an integer such that $1 \leq r \leq R$ and $\mathcal{T}$ be a $r$-round characteristic. Denote $m = \#\mathrm{SB}(\alpha_r)$ and let $n$ be an integer satisfying $1 \leq n \leq m$. Define the function $\rho : [\![1, m]\!] \to [\![1, N]\!]$ that maps $i$ to the index of the $i$-th S-box activated by $\alpha_r$. Finally, let us define $L = \mathrm{list}(\alpha_r) \wedge (0_{\rho(n)} 1_{N-\rho(n)})$. If*

$$\mathbb{P}\left(\mathcal{T}_{[1,r-1]}\right)\Big(\prod_{i=1}^{n} \mathbb{P}_{\rho(i)}\big(a^r_{\rho(i)} \to b^r_{\rho(i)}\big)\Big) p_{\mathrm{ListSB}(L)}$$

*is lower than the $r$-rank bound, then for all $\gamma$ satisfying:*

- $c_{\rho(i)} = b^r_{\rho(i)}$ *for each $i < n$, and*
- $\mathbb{P}_{\rho(n)}(a^r_{\rho(n)} \to c_{\rho(n)}) \leq \mathbb{P}_{\rho(n)}(a^r_{\rho(n)} \to b^r_{\rho(n)})$,

*there exists no $R$-round characteristic extending $\mathcal{T}_{[1,r-1]} \parallel (\alpha_r, \gamma)$ with probability greater than or equal to $p_{\mathrm{Estim}}$.*

*Proof.* If $\gamma$ is not a candidate for $\alpha_r$, then $\mathbb{P}(\alpha_r \to \gamma) = 0$ and any characteristic extending $\mathcal{T}_{[1,r-1]} \parallel (\alpha_r, \gamma)$ has also a zero probability. Therefore, we assume that $\gamma$ is a candidate for $\alpha_r$ in the following. Since $\mathrm{supp}(L) = \{\rho(i) \mid n+1 \leq i \leq m\}$, it follows that

$$\begin{aligned}
\mathbb{P}(\alpha_r \to \gamma) &= \textstyle\prod_{i=1}^{m} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to c_{\rho(i)}) \\
&= \textstyle\prod_{i=1}^{n} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to c_{\rho(i)}) \prod_{i=n+1}^{m} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to c_{\rho(i)}) \\
&\leq \textstyle\prod_{i=1}^{n} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to b^r_{\rho(i)}) \prod_{i=n+1}^{m} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to c_{\rho(i)}) \\
&\leq \big(\textstyle\prod_{i=1}^{n} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to b^r_{\rho(i)})\big) p_{\mathrm{ListSB}(L)}
\end{aligned}$$

Next, we have the inequality

$$\begin{aligned}
\mathbb{P}(\mathcal{T}_{[1,r-1]} \parallel (\alpha_r \to \gamma)) &= \mathbb{P}(\mathcal{T}_{[1,r-1]}) \mathbb{P}(\alpha_r \to \gamma) \\
&\leq \mathbb{P}\big(\mathcal{T}_{[1,r-1]}\big)\big(\textstyle\prod_{i=1}^{n} \mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to b^r_{\rho(i)})\big) p_{\mathrm{ListSB}(L)} \ .
\end{aligned}$$

Consequently, the probability of $\mathcal{T}_{[1,r-1]} \parallel (\alpha_r, \gamma)$ is lower than the $r$-rank bound. The result then is a consequence of Lemma 12. □

*Remark 19.* All the probabilities $p_{\mathrm{ListSB}(L)}$, with $L$ in $\mathbb{F}_2^N$ are precomputed. For each $1 \leq i \leq N$ and each input difference $a$, the output differences are sorted in decreasing order according to $\mathbb{P}_i(a \to \cdot)$.

### 4.3 Active S-Boxes in the Next Round

Throughout this part, the linear layer $\pi$ is assumed to be a bit permutation.

**Definition 20.** *Let $L$ and $L'$ be elements of $\mathbb{F}_2^N$. Define $L \preccurlyeq L'$ if and only if* $\mathrm{supp}(L) \subseteq \mathrm{supp}(L')$. *It is easy to show that $\preccurlyeq$ is partial order. Clearly,*

$$(L \preccurlyeq L') \implies \big(p_{\mathrm{ListSB}(L)} \geq p_{\mathrm{ListSB}(L')}\big) \ .$$

Define the function $\mathrm{D} : [\![1, N]\!] \times \mathbb{F}_2^S \to (\mathbb{F}_2^S)^N$ that maps a pair $(i, c)$ to $\mathrm{D}(i, c) = (c_1, \ldots, c_N)$ where $c_j$ equals $c$ if $i = j$ and $0_S$ otherwise. To simplify, let us denote $\mathrm{D}_i(c) = \mathrm{D}(i, c)$. It is easy to see that $\gamma = \sum_{i=1}^{N} \mathrm{D}_i(c_i) = \sum_{i, c_i \neq 0_S} \mathrm{D}_i(c_i)$ for each difference $\gamma$. Furthermore, we say that two differences $\gamma$ and $\gamma'$ (seen as elements of $\mathbb{F}_2^{SN}$ instead of $(\mathbb{F}_2^S)^N$) are mutually disjoint if for all $1 \leq i \leq SN$, $\gamma_i = \gamma'_i \Rightarrow \gamma_i = \gamma'_i$

*Example 21.* Using again the previous notations,

$$\beta_2 = (\mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{5}) = \mathrm{D}_4(\mathtt{5}) \quad \text{and} \quad \beta_3 = \textstyle\sum_{i, b_i^3 \neq 0_4} \mathrm{D}_i(b_i^3) = \mathrm{D}_2(\mathtt{3}) + \mathrm{D}_4(\mathtt{3}) \ .$$

**Lemma 22.** *Let $\gamma_1, \ldots, \gamma_n$ be $n$ pairwise mutually disjoint differences. Then*

$$\mathrm{list}\Big(\sum_{i=1}^{n} \gamma_i\Big) = \bigvee_{i=1}^{n} \mathrm{list}(\gamma_i) \ .$$

*Proof.* By induction on $n$. The case $n = 1$ being trivial, we assume $n = 2$. Define $L = \mathrm{list}(\gamma_1 + \gamma_2)$, $L^1 = \mathrm{list}(\gamma_1)$ and $L^2 = \mathrm{list}(\gamma_2)$. Let $i$ be an integer such that $1 \leq i \leq N$. Since $\gamma_1$ and $\gamma_2$ are mutually disjoint, the equality $c_i^1 = c_i^2$ implies $c_i^1 = 0_S$ and $c_i^2 = 0_S$. The converse being immediate, the equivalence $(c_i^1 + c_i^2 = 0_S) \Leftrightarrow (c_i^1 = 0_S \text{ and } c_i^2 = 0_S)$ follows, that is $(c_i^1 + c_i^2 \neq 0_S) \Leftrightarrow (c_i^1 \neq 0_S \text{ or } c_i^2 \neq 0_S)$. Next, $L_i = 1 \Leftrightarrow c_i^1 + c_i^2 \neq 0_S \Leftrightarrow (c_i^1 \neq 0_S \text{ or } c_i^2 \neq 0_S) \Leftrightarrow (L_i^1 = 1 \text{ or } L_i^2 = 1)$. Therefore, $L = L^1 \vee L^2$. The result follows by induction on $n$ as $\gamma_n$ and $\sum_{i=1}^{n-1} \gamma_i$ are mutually disjoint. □

**Corollary 23.** *Let $\beta$ be an output difference. Let $m$ be an integer such that $1 \leq m \leq N$ and $\rho : [\![1, m]\!] \to [\![1, N]\!]$ an one-to-one function. Then*

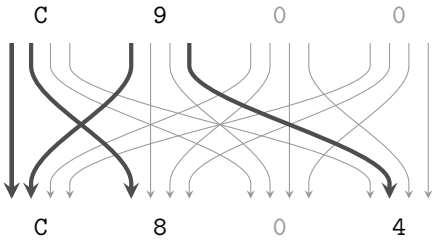
$$\mathrm{list}\Big(\pi\Big(\sum_{i=1}^{m} \mathrm{D}_{\rho(i)}(b_{\rho(i)})\Big)\Big) = \bigvee_{i=1}^{m} \mathrm{list}\left(\pi(\mathrm{D}_{\rho(i)}(b_{\rho(i)})\right) \ .$$

*Proof.* Since $\pi$ is linear, the following equality holds

$$\pi\Big(\sum_{i=1}^{m} \mathrm{D}_{\rho(i)}(b_{\rho(i)})\Big) = \sum_{i=1}^{m} \pi(\mathrm{D}_{\rho(i)}(b_{\rho(i)})) \ .$$

Clearly, the $\mathrm{D}_{\rho(i)}(b_{\rho(i)})$ are mutually disjoint as $\rho$ is one-to-one. Since $\pi$ is a bit permutation, it must be the case that the $\pi(\mathrm{D}_{\rho(i)}(b_{\rho(i)}))$ are also disjoint. From Lemma 22, we have $\mathrm{list}(\sum_{i=1}^{m} \pi(\mathrm{D}_{\rho(i)}(b_{\rho(i)}))) = \bigvee_{i=1}^{m} \mathrm{list}(\pi(\mathrm{D}_{\rho(i)}(b_{\rho(i)})))$. □

*Example 24.* On the one hand, $\mathrm{list}(\pi(\mathrm{D}_1(\mathtt{C}) + \mathrm{D}_2(\mathtt{9}))) = \mathrm{list}(\pi(\mathtt{C}, \mathtt{9}, \mathtt{0}, \mathtt{0})) = \mathrm{list}(\mathtt{C}, \mathtt{8}, \mathtt{0}, \mathtt{4}) = (1, 1, 0, 1)$.



On the other hand,

$$\begin{aligned}&\mathrm{list}(\pi(\mathrm{D}_1(\mathtt{C}))) \vee \mathrm{list}(\pi(\mathrm{D}_2(\mathtt{9})))\\ &\quad = \mathrm{list}(\mathtt{8},\mathtt{8},\mathtt{0},\mathtt{0}) \vee \mathrm{list}(\mathtt{4},\mathtt{0},\mathtt{0},\mathtt{4})\\ &\quad = (1,1,0,0) \vee (1,0,0,1) = (1,1,0,1)\ .\end{aligned}$$

**Theorem 25.** *We use the same notations as in Theorem 18 except that $r \leq R-1$. Define $L' = \bigvee_{i=1}^{n} \mathrm{list}(\pi(\mathrm{D}_{\rho(i)}(b^r_{\rho(i)})))$. If*

$$\Big[\mathbb{P}\left(\mathcal{T}_{[1,r-1]}\right)\Big(\prod_{i=1}^{n}\mathbb{P}_{\rho(i)}\big(a^r_{\rho(i)} \to b^r_{\rho(i)}\big)\Big)p_{\mathrm{ListSB}(L)}\Big] \times p_{\mathrm{ListSB}(L')}$$

*is lower than the rank-$(r+1)$ bound, then for all $\gamma$ such that $c_{\rho(i)} = b^r_{\rho(i)}$ for each $i \leq n$, there exists no $R$-round characteristic extending $\mathcal{T}_{[1,r-1]} \parallel (\alpha_r, \gamma)$ with probability greater than or equal to $p_{\mathrm{Estim}}$.*

*Proof.* Following the proof of Theorem 18, we can assume that $\gamma$ is a candidate for $\alpha_r$ and deduce the upper-bound

$$\mathbb{P}(\mathcal{T}_{[1,r-1]} \parallel (\alpha_r,\gamma)) \leq \mathbb{P}(\mathcal{T}_{[1,r-1]})\Big(\prod_{i=1}^{n}\mathbb{P}_{\rho(i)}\big(a^r_{\rho(i)} \to b^r_{\rho(i)}\big)\Big)p_{\mathrm{ListSB}(L)}\ .$$

Define $\alpha_{r+1} = \pi(\gamma)$. Let $\beta_{r+1}$ be an output difference. Similarly, we can assume that $\beta_{r+1}$ is a candidate for $\alpha_{r+1}$. Define,

$$L'' = \mathrm{list}\left(\pi\left(\textstyle\sum_{i=1}^{m}\mathrm{D}_{\rho(i)}\left(c_{\rho(i)}\right)\right)\right) = \mathrm{list}(\pi(\gamma)) = \mathrm{list}(\alpha_{r+1})\ .$$

Since $L'' = \mathrm{list}(\alpha_{r+1})$, it follows that $\mathbb{P}(\alpha_{r+1} \to \beta_{r+1}) \leq p_{\mathrm{ListSB}(L'')}$. According to Corollary 23,

$$L'' = \bigvee_{i=1}^{m}\mathrm{list}\left(\pi\left(\mathrm{D}_{\rho(i)}\left(c_{\rho(i)}\right)\right)\right) \succcurlyeq \bigvee_{i=1}^{n}\mathrm{list}\left(\pi\left(\mathrm{D}_{\rho(i)}\left(c_{\rho(i)}\right)\right)\right) = L'\ .$$

Consequently $p_{\mathrm{ListSB}(L'')} \leq p_{\mathrm{ListSB}(L')}$ and $\mathbb{P}(\alpha_{r+1} \to \beta_{r+1})$ is upper-bounded by $p_{\mathrm{ListSB}(L')}$. Finally,

$$\begin{aligned}&\mathbb{P}(\mathcal{T}_{[1,r-1]} \parallel (\alpha_r,\gamma) \parallel (\alpha_{r+1},\beta_{r+1}))\\ &\quad = \mathbb{P}(\mathcal{T}_{[1,r-1]} \parallel (\alpha_r,\gamma)) \times \mathbb{P}(\alpha_{r+1} \to \beta_{r+1})\\ &\quad \leq \big[\mathbb{P}(\mathcal{T}_{[1,r-1]})\big(\textstyle\prod_{i=1}^{n}\mathbb{P}_{\rho(i)}(a^r_{\rho(i)} \to b^r_{\rho(i)})\big)p_{\mathrm{ListSB}(L)}\big] \times p_{\mathrm{ListSB}(L')}\ .\end{aligned}$$

Thus, the probability of $\mathcal{T}_{[1,r-1]} \parallel (\alpha_r,\gamma) \parallel (\alpha_{r+1},\beta_{r+1})$ is lower than the rank-$(r+1)$ bound and there exists no $R$-round characteristic extending it with probability greater than or equal to $p_{\mathrm{Estim}}$. Using the fact that this property holds for all $\beta_{r+1}$, the desired result is proven. □

The search procedure `Round` optimized with Theorems 18 and 25 is described in Figure 4.

### 4.4 Test on the Bound

The previous results can be preserved while strengthening the condition on the bound. Suppose we have found a characteristic with probability greater than or equal to $p_{\mathrm{Estim}}$. Then, we have $p_{\mathrm{Estim}} = \mathbb{P}(\mathcal{E})$. Now, assume that the probability of the current characteristic $\mathcal{T}$ satisfies $\mathbb{P}(\mathcal{T}) \cdot p_{\mathrm{Best}(R-r)} = p_{\mathrm{Estim}}$. In this case, this probability is not lower than the rank-$r$ bound and the algorithm tries to extend it. However, the previous equality implies that we can optimally find a $R$-round characteristic with probability $p_{\mathrm{Estim}}$. As such a characteristic is already known ($\mathcal{E}$), the extension of $\mathcal{T}$ can be aborted. This discussion leads us to improve Definition 11.

**Function** `Round`$(r)$ $(2 \leq r < R)$
$\beta_r \leftarrow 0_{SN}$
For each $i$ such that $1 \leq i \leq \#\mathrm{SB}(\alpha_r)$
  Let $\rho_r(i)$ denote the position of the $i$-th S-box activated by $\alpha_r$
  Call `SubRound(`$r$`,1)`
End of the function ▷ We continue `Round(`$r-1$`)` or `FirstRound()`

**Function** `SubRound(`$r,n$`)`
If $n > \#\mathrm{SB}(\alpha_r)$, then
  $p_{\mathrm{Rd}(r)} \leftarrow \mathbb{P}(\alpha_r \rightarrow \beta_r) = \prod_{j=1}^{\#\mathrm{SB}(\alpha_r)} p_{\mathrm{Rd}(r,j)}$;
  $\alpha_{r+1} \leftarrow \pi(\beta_{r+1})$
  If $r+1 < R$, then
    Call `Round(`$r+1$`)`
  Else
    Call `LastRound()`
Else
  For each $b^r_{\rho_r(n)}$ sorted in decreasing order according to $\mathbb{P}_{\rho r(n)}(a^r_{\rho_r(n)} \rightarrow \cdot)$
    $L_{r,n} \leftarrow \mathrm{list}(\alpha_r) \wedge (0_{\rho_r(n)} 1_{N-\rho_r(n)})$
    $p_{\mathrm{Rd}(r,n)} \leftarrow \mathbb{P}_{\rho_r(n)}\left(a^r_{\rho_r(n)} \rightarrow b^r_{\rho_r(n)}\right)$
    If $\prod_{i=1}^{r-1} p_{\mathrm{Rd}(i)} \cdot \prod_{j=1}^{n} p_{\mathrm{Rd}(r,j)} \cdot p_{\mathrm{ListSB}(L_{r,n})}$ is lower than the rank-$r$ bound, then
      Exit the loop ▷ See Theorem 18
    If $\pi$ is a bit permutation, then
      $L'_{r,n} \leftarrow \bigvee_{i=1}^{n} \mathrm{list}(\pi(\mathrm{D}_{\rho r(i)}(b^r_{\rho_r(i)})))$;
      If $\prod_{i=1}^{r-1} p_{\mathrm{Rd}(i)} \cdot \prod_{j=1}^{n} p_{\mathrm{Rd}(r,j)} \cdot p_{\mathrm{ListSB}(L_{r,n})} \cdot p_{\mathrm{ListSB}(L'_{r,n})}$ is not lower than the rank-$(r+1)$ bound, then
        Call `SubRound(`$r,n+1$`)` ▷ See Theorem 25
    Else
      Call `SubRound(`$r,n+1$`)`
End of the function ▷ We continue `SubRound(`$r,n-1$`)` or `Round(`$r$`)`

**Fig. 4.** Second optimization – the search function `Round`

**Definition 26 (rank-$r$ bound).** *Let $\mathcal{T}$ be a $r$-round characteristic with $r < R$. Its probability is* lower than the rank-$r$ bound *if*

- *$\mathcal{E}$ is empty and*
- $\mathbb{P}(\mathcal{T}) < p_{\mathrm{Estim}} \,/\, p_{\mathrm{Best}(R-r)}$,

*or if*

- *$\mathcal{E}$ contains a characteristic and*
- $\mathbb{P}(\mathcal{T}) \leq p_{\mathrm{Estim}} \,/\, p_{\mathrm{Best}(R-r)}$.

## 5 Automatic Management of the Estimation

The parameter $p_{\mathrm{Estim}}$ has a high impact on the complexity of `OptTrailEst`. Several methods exist to obtain a good estimation of $p_{\mathrm{Best}(R)}$. For instance, an iterative characteristic can be used. Following an idea of Ohta, Moriai and Aoki [14], we propose the algorithm `OptTrail`. The latter has two main advantages. First, the estimation management is fully automatic – no knowledge is

required on the SPN. Second, its complexity has the same order of magnitude, as `OptTrailEst` runs with $p_{\text{Estim}} = p_{\text{Best}(R)} \,/\, 2$.

The algorithm `OptTrail` is presented in Figure 5. To understand how it works, it is worth recalling that `OptTrailEst` finds no characteristic whenever $p_{\text{Estim}} > p_{\text{Best}(R)}$ (Theorem 13). In this case, $p_{\text{Estim}}$ is not modified by `OptTrailEst`. Since $p_{\text{Best}(R)} \leq p_{\text{Best}(R-1)}$, we begin by running `OptTrailEst` with $p_{\text{Estim}} = p_{\text{Best}(R-1)}/2$. The estimation is then each time divided by two until an optimal characteristic is found. This happens whenever the condition $p_{\text{Estim}} \leq p_{\text{Best}(R)}$ becomes true.

Indeed, the larger is the value of $p_{\text{Estim}}$, the stronger is the pruning condition and the lower is the complexity of the search. The exact nature of this result is still unknown. However, we have observed experimentally that the complexity of `OptTrailEst` running with $p_{\text{Estim}} \geq 2^4 \cdot p_{\text{Best}(R)}$, is negligible compared with the complexity of the same algorithm running with $p_{\text{Estim}} = p_{\text{Best}(R)} \,/\, 2$. The following result comes from this observation: *if* `OptTrailEst` *is computable with* $p_{\text{Estim}} = p_{\text{Best}(R)} \,/\, 2$*, then* `OptTrail` *is also computable.*

**Proposition 27.** *The complexity of* `OptTrailEst` *decreases as the input* $p_{\text{Estim}}$ *increases.*

**Algorithm** `OptTrail`
$p_{\text{Estim}} \leftarrow p_{\text{Best}(R-1)}$
Until no characteristic has been found,
  $p_{\text{Estim}} \leftarrow p_{\text{Estim}} \,/\, 2$
  Call `OptTrailEst` with $p_{\text{Estim}}$ as estimation
Return $\mathcal{E}$ and $p_{\text{Estim}}$

**Fig. 5.** Automatic estimation management

## 6 Results

Experiments and simulations have been performed by a *AMD Phenom II X4 965 Black Edition 3.4 GHz* processor. The running time for a $\tilde{R}$-round cipher includes the precomputations and $\tilde{R} - 1$ calls to `OptTrail`, as explained in Section 3.

To prove the practical security of PRESENT [4] against differential cryptanalysis, the authors have shown that the probability of any 5-round characteristic is upper-bound by $2^{-20}$ and had exhibited a 5-round characteristic of probability $2^{-21}$. The algorithm presented here allows us to prove in 0.3 second that this upper-bound is reached with the following optimal characteristic:

$$\begin{aligned}
\alpha_1 &= (0,0,0,7,0,0,0,0,0,0,0,0,0,0,0,7)\\
\beta_1 &= (0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1)\\
\beta_2 &= (0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,3)\\
\beta_3 &= (0,0,0,0,0,0,0,0,0,0,0,4,0,0,0,4)\\
\beta_4 &= (0,0,0,0,0,0,3,3,0,0,0,0,0,0,0,0)\\
\beta_5 &= (0,0,0,0,0,0,0,0,0,6,0,0,0,6,0,0)\ .
\end{aligned}$$

They have then deduced that any 25-round characteristic probability is upper-bounded by $2^{-100}$. Our algorithm shows that the optimal characteristic probability is $2^{-110}$ in 0.5 second. The

number of rounds is not a problem since an optimal 64-round characteristic is computed in just 2 seconds. Note that PRESENT has 32 rounds.

The permutation used in SMALLPRESENT [11] (and in PRESENT) can be generalized for each integers $N$ and $S$. Define $\phi_{S,N}$ for all $1 \leq k \leq SN$ by

$$\phi_{S,N}(k) = N(k - 1 \bmod S) + \lfloor \tfrac{k-1}{S} \rfloor + 1 \ .$$

It is easy to verify that the permutation $\phi$ used in Example 10 is $\phi_{4,4}$. We have constructed a 128-bit SPN on the same model as PRESENT to test our algorithm efficiency. Define $\pi$ as the bit permutation associated with $\phi_{8,16}$ and the 16 S-boxes as the AES S-box. We have obtained an optimal 13-round characteristic of probability $2^{-89}$ in 7.1 seconds.

To analyze PUFFIN security against differential cryptanalysis, Cheng et al [6] have upper-bounded the probability of an optimal 31-round characteristic by $2^{-62}$. In 0.02 second, we have computed a characteristic that reaches this bound. It is obtained by extending the following iterative characteristic:

$$\begin{aligned}
\alpha_i &= (4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\
\beta_i &= (4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\
\alpha_{i+1} &= (0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\
\beta_{i+1} &= (0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \ .
\end{aligned}$$

Finally, we have tested our algorithm on ICEBERG [15]. Its permutation layer is not a bit permutation but a linear diffusion. The optimization presented in Section 4.3 is thus no longer applicable. The authors have upper-bound the probability of an optimal 16-rounds characteristic by $2^{-160}$. We proved that it is in fact $2^{-171,6}$ in 2.3 seconds. All these results are outlined in Figure 6.

| Cipher | Block size | Round number | Upper-bound | Best probability | Running time |
|---|---|---|---|---|---|
| PRESENT | 64 | 5 | $2^{-20}$ | $2^{-20}$ | 0.3 s |
| PRESENT | 64 | 25 | $2^{-100}$ | $2^{-110}$ | 0.5 s |
| PRESENT-like | 128 | 13 | – | $2^{-89}$ | 7.1 s |
| PUFFIN | 64 | 31 | $2^{-62}$ | $2^{-62}$ | 0.02 s |
| ICEBERG | 64 | 16 | $2^{-160}$ | $2^{-171.6}$ | 2.3 s |

**Fig. 6.** Summary of Results

## Conclusion

In this paper, we have presented a generic algorithm that computes a maximum probability differential characteristic in a SPN. Running this algorithm may allow to prove the practical security of the block cipher. In the opposite case of weak cipher, the returned characteristic allows the cryptanalyst to build an optimal attack.

Especially optimized for SPN using a bit permutation as permutation layer, we are able to find a maximum probability characteristic of PRESENT and PUFFIN within one second. Block cipher designers have then a powerful tool which can be run several times to improve block cipher components.

## References


1. Kashif Ali and Howard M Heys. An Algorithm to Analyze Block Cipher Resistance to Linear and Differential Cryptanalysis. In *Proceedings of Newfoundland Electrical and Computer Engineering Conference (NECEC 2006)*, 2006.
2. Kazumaro Aoki, Kunio Kobayashi, and Shiho Moriai. Best differential characteristic search of FEAL. In *Fast Software Encryption*, pages 41–53. Springer, 1997.
3. Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.
4. Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems-CHES 2007*, pages 450–466. Springer, 2007.
5. Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology—EUROCRYPT'94*, pages 356–365. Springer, 1995.
6. Huiju Cheng, Howard M Heys, and Cheng Wang. Puffin: A novel compact block cipher targeted to embedded digital systems. In *Digital System Design Architectures, Methods and Tools, 2008. DSD'08. 11th EUROMICRO Conference on*, pages 383–390. IEEE, 2008.
7. Baudoin Collard, F-X Standaert, and J-J Quisquater. Improved and multiple linear cryptanalysis of reduced round Serpent. In *Information Security and Cryptology*, pages 51–65. Springer, 2008.
8. Masayuki Kanda, Youichi Takashima, Tsutomu Matsumoto, Kazumaro Aoki, and Kazuo Ohta. A strategy for constructing fast round functions with practical security against differential and linear cryptanalysis. In *Selected Areas in Cryptography*, pages 264–279. Springer, 1999.
9. Lars R Knudsen and Matthew JB Robshaw. *The block cipher companion*. Springer, 2011.
10. Xuejia Lai, James L Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology—EUROCRYPT'91*, pages 17–38. Springer, 1991.
11. Gregor Leander. Small Scale Variants Of The Block Cipher PRESENT. *IACR Cryptology ePrint Archive*, 2010:143, 2010.
12. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology—EUROCRYPT'93*, pages 386–397. Springer, 1994.
13. Mitsuru Matsui. On correlation between the order of S-boxes and the strength of DES. In *Advances in Cryptology—EUROCRYPT'94*, pages 366–375. Springer, 1995.
14. Kazuo Ohta, Shiho Moriai, and Kazumaro Aoki. Improving the search algorithm for the best linear expression. In *Advances in Cryptology—CRYPT0'95*, pages 157–170. Springer, 1995.
15. Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware. In *Fast Software Encryption*, pages 279–298. Springer, 2004.