From Simple Mathematical Upper Bounds to Tight Tool-based Bounds for the Minimum Number of Active S-boxes in LS-Designs

Jules Baudrin, Rachelle Heim Boissier, Mahnoor Naseer and François-Xavier Standaert

Abstract. LS-designs, proposed at FSE 2014, are a conceptually simple family of block ciphers, where a bitslice S-box and a linear L-box are independently applied to the state. They are known for enabling efficiently masked implementations against side-channel attacks. In this paper, we consolidate the understanding of their security against linear and differential cryptanalysis. For this purpose, we first show that the structure of LS-designs enables us to derive non-trivial mathematical upper bounds on the minimum number of active S-boxes in their linear and differential trails. These bounds are a convenient addition to existing lower bounds based on the L-box branch number, as they provide a direct indication on the security claims' tightness. We then show how such mathematical bounds can be used in complement with tool-based bounds obtained from SAT solvers. Applied to the (128-bit, non-involutive) cipher Fantomas, this allows us to provide tight results on the number of active S-boxes for up to 16 rounds. We also extend existing searches that were limited to 16-bit L-boxes to (practically-relevant) 32-bit L-boxes. We hope these results can facilitate the design space exploration of new LS-designs, possibly operating on larger states.

Keywords: LS-designs · Bitsliced cipher · Linear layer · Branch number · Differential cryptanalysis · Linear cryptanalysis · Automated cryptanalysis · SAT

1 Introduction

Resistance against differential [BS91] and linear [Mat93] cryptanalysis are fundamental design criteria for symmetric cryptographic primitives. They are often guaranteed by exhibiting lower bounds on the number of active S-boxes in differential and linear trails. Together with the differential uniformity (resp., linearity) of the S-box, this allows designers to upper bound the probability of differential (or correlation of linear) trails under classical assumptions. Counting the minimum number of active S-boxes over a given number of rounds is thus widely recognized as an important step in security evaluations [DR20, SMMR17, PL23, WJ19]. Two main approaches are generally considered for this.

The first one, which we will denote as *tool-based*, leverages combinatorial optimization methods. Matsui's branch-and-bound algorithm is a seminal example [MT99]. Other solutions include modeling the search as a Mixed Integer Linear Programming (MILP) problem [MWGP11, WW11, SHS+13, SHW+14, TJTS21], as a satisfiability (SAT) problem [SWW18, LW19, LLL+21, SWW21] or with constraint programming [GMS16].

The second one is to leverage the properties of a cipher's components to obtain what we will denote as *mathematical* bounds. This is for example made easy by the wide-trail strategy [DR20]. In a nutshell, it combines S-boxes with good linear/differential properties and a linear layer with a high *branch number* $\mathcal{B}$ (i.e., a high minimum number of active S-boxes before and after this linear layer). Since the linear layer activates at least $\mathcal{B}$ S-boxes over two rounds, the number of active S-boxes over r rounds is at least $\mathcal{B} \cdot \lfloor \frac{r}{2} \rfloor$.

These approaches are essentially complementary. Mathematical bounds lead to easily explainable guarantees but are rarely tight. Tool-based bounds generally lead to tighter security guaranttes, but can be computationally intensive to obtain. Some works therefore adopt hybrid approaches (which we will also use), combining mathematical bounds with tool-based search in order to evaluate the number of active S-boxes [DFJL19].

As a result, new block ciphers have to come with one, the other or both types of arguments to establish their security against linear and differential cryptanalysis. For example, the LS-designs, introduced at FSE 2014, are a conceptually simple family of ciphers relying on the wide-trail strategy [GLSV14]. They combine the application of a bitslice S-box independently to each column of the state with an $\mathbb{F}_2$-linear layer applied independently to each row, the L-box. It has been shown that this conceptual simplicity can lead to efficient masked implementations to mitigate side-channel attacks [JSV17, BBB+20]. Grosso et al. proposed two initial instances of LS-designs: Robin, which uses an involutive L-box, and Fantomas which uses a non-involutive one. Their initial analysis illustrates the complementary nature of the tool-based and mathematical approaches. It combines good diffusion properties guaranteed by a branch number with tool-based bounds relying on a *truncated trail search*. In a nutshell, this search relies on column-wise activity patterns in which a bit variable indicates whether a column is active or not. The truncated search then enumerates the possible truncated state transitions across rounds, and derives a minimum over the number of active S-boxes in the corresponding truncated trails. However, it does not consider fully instantiated bit-level differences and, therefore, may not be tight.

Contributions. Based on this state of the art, a natural consolidating question is to find out whether this tightness question can be efficiently clarified for LS-designs.

Our contributions in this respect are twofold:

First, we derive a simple mathematical upper bound on the minimum number of active S-boxes in LS-designs. For this purpose, we leverage the observation that the S-box and L-box layers of such ciphers act independently [GLSV14], which we combine with standard results from the theory of linear codes. While designers are more interested in lower bounds (e.g., as provided by the branch number), this upper bound provides a convenient indication on the interval in which the minimum number of active S-boxes lies.

Second, we show that this upper bound enables an efficient and exact automated search for the minimum number of active S-boxes in a fully-instantiated bit-level differential and linear setting. For this purpose, we model the search as a *bounded feasibility* Boolean SAT problem, where each instance tests the existence of a trail with at most a given number of active S-boxes. Despite it could be heuristically performed without upper (and lower) bounds, such bounds allow us to restrict the search space and simplify the use of off-the-shelf SAT solvers for computing exact bounds. We apply this methodology to the LS-design cipher Fantomas and obtain exact bounds on the minimum number of active S-boxes for up to 16 cipher rounds, confirming the tightness of the FSE 2014 truncated search. We also apply it to an exemplary 128-bit cipher with 32-bit L-box, for which the truncated search is not applicable: a practically-relevant improvement given the nice match between such L-box sizes and current embedded (ARM Cortex-like) devices.

Related works. Shen et al. [SLLQ18] constructed 4-round impossible differentials and propose a 6-round impossible differential attack on Robin and Fantomas. Dwivedi et al. [DDSS19] extended linear cryptanalysis by constructing linear approximations for 5 round of Fantomas and 7 rounds of Robin, leading to key recovery attacks on reduced-round instances. They state that they have not used any standard automated tool to construct these trails. Leander et al. [LMR15] spot an invariant subspace weakness in Robin.

Paper outline. In Section 2, we recall the necessary background. In Section 3, we describe our first contribution, namely a simple mathematical upper bound leveraging

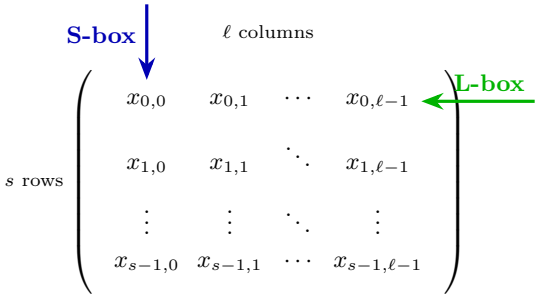

Figure 1: $s \times \ell$ state of an LS-design.

coding theory for the minimum number of active S-boxes in LS-designs. In Section 4, we describe a framework to convert an LS-design into CNF constraints for SAT solvers, and elaborate on the solving strategy adopted to take advantage of our bounds. In Section 5, we combine the proven upper and lower bounds to define a search interval and perform a bounded SAT feasibility search to compute exact bounds on the minimum number of active S-boxes for Fantomas-128. In Section 6, we further apply our hybrid analysis approach to 32-bit L-boxes. We finally discuss the implications of our results and how they may help the design and cryptanalysis of future LS-design ciphers in Section 7.

2 Preliminaries

In this first section, we provide the necessary background to understand the paper, covering LS-designs, differential cryptanalysis, linear cryptanalysis, and (mathematical and automated) methods to evaluate the number of active S-boxes in a block cipher.

2.1 LS-designs

LS-designs are a family of bit-oriented block ciphers introduced at FSE 2014 [GLSV14]. Instances from this family include the 128-bit (involutive) Robin and (non-involutive) Fantomas. One of their main design goal is to exploit bitslice S-boxes with minimum AND complexity, so that their implementations can be more efficiently masked against side-channel attacks. They rely on the wide-trail design strategy to achieve good security against linear and differential cryptanalysis. Concretely, LS-designs operate on a state of size $n = s \cdot \ell$ bits represented as an $s \times \ell$ matrix with coefficients in $\mathbb{F}_2$. They are built by iteratively applying a round function r times. Besides the round key and round constant additions, and as illustrated in Figure 1, a round consists of two main operations:

- **S-box layer**: A non-linear permutation $S\colon \mathbb{F}_2^s \to \mathbb{F}_2^s$, applied in parallel to each column of the state. It is typically implemented in a bitslice manner.
- **L-box layer**: An invertible linear layer $L\colon \mathbb{F}_2^\ell \to \mathbb{F}_2^\ell$ applied to state rows.

For the linear layer, which will be our main concern in this paper, the authors used $[n, k, d]$-binary linear codes (of length $n = 2\ell$ and dimension $k = \ell$), where the minimum distance d equals the branch number $\mathcal{B}(L)$ of the linear layer. This choice ensures at least $\mathcal{B}(L) = 8$ active S-boxes over 2 rounds of Robin and Fantomas.

2.2 Differential cryptanalysis

Differential cryptanalysis, introduced by Biham and Shamir in 1990 [BS91], analyzes and exploits the propagation of differences in symmetric primitives. We next describe it for a block cipher E parametrised by a key $K \in \mathbb{F}_2^\kappa$: $E_K : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$.

For a key alternating cipher, we denote by $R_{K_i} : \mathbb{F}_2^n \to \mathbb{F}_2^n$, a keyed transformation corresponding to round i that includes the S-box layer, linear layer, and the addition of round constants and subkeys. Thus, an r-round iterative cipher is built as:

$$E_K = R_{K_{r-1}} \circ \cdots \circ R_{K_0}.$$

In a nutshell, differential cryptanalysis studies the probability that a non-zero input difference $\Delta_{in} \in \mathbb{F}_2^n$ is mapped to $\Delta_{out} \in \mathbb{F}_2^n$ through several rounds of (for example) a block cipher. An ordered pair of input/output difference $(\Delta_{in}, \Delta_{out})$ is called a *differential*. The *differential probability* of $(\Delta_{in}, \Delta_{out})$ over a function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is defined as:

$$\mathrm{DP}_F(\Delta_{in}, \Delta_{out}) \coloneqq \frac{|\{x \mid F(x) + F(x + \Delta_{in}) = \Delta_{out}\}|}{2^n}.$$

A classical assumption is that the fixed-key probability $DP_{E_K}(\Delta_{in}, \Delta_{out})$ does not deviate significantly from the key-averaged probability, or *expected differential probability*, $\mathrm{EDP}_E(\Delta_{in}, \Delta_{out})$ [LMM91]. However, in general, computing the expected differential probability of a differential is a computationally hard problem. Hence, a classical approach in the design of symmetric ciphers is to exploit their iterative structure to search for the best *differential trails* or *characteristics*, as we describe next.

Definition 1 (Differential trail)**.** A differential trail over r rounds is an ordered $(r+1)$-uplet of differences $\Delta = (\Delta_0, \ldots, \Delta_r)$ where for any round $i \in [\![0, r-1]\!]$, Δ_i denotes the input difference and Δ_{i+1} the output difference.

For a *fixed key*, the differential probability of the trail $(\Delta_0, \ldots, \Delta_r)$ is defined similarly to the probability of a differential – as the cardinal of the set of plaintexts x that *verify*[1] the trail divided by 2^n. The probability of the differential $(\Delta_{in}, \Delta_{out})$, namely $DP_{E_K}(\Delta_{in}, \Delta_{out})$, is then exactly the sum over all trails satisfying $\Delta_0 = \Delta_{in}$ and $\Delta_r = \Delta_{out}$, that is,

$$DP_{E_K}(\Delta_{in}, \Delta_{out}) = \sum_{(\Delta_{in}, \Delta_1, \ldots, \Delta_{r-1}, \Delta_{out})} \mathrm{DP}_{E_K}(\Delta_{in}, \Delta_1, \ldots, \Delta_{r-1}, \Delta_{out}).$$

Note that, by definition, this fixed-key differential probability (as well as all terms in the previous sum) depends on the key and may vary according to it. On the other hand, the average over all keys, namely the expected differentially probability $\mathrm{EDP}_E(\Delta_0, \Delta_r)$, is by construction independent of the choice of a specific key and, on top of it, easier to study. Indeed, under the classical assumption [LMM91] that the round keys are independent and uniformly distributed, the expected differential probability of any key-alternating cipher (which is a subset of Markov ciphers [DR07]) can be expressed as follows:

$$\mathrm{EDP}_E(\Delta_0, \Delta_r) = \sum_{(\Delta_0, \ldots, \Delta_r)} \prod_{i=0}^{r-1} \mathrm{DP}_R(\Delta_i, \Delta_{i+1}),$$

where $\mathrm{DP}_R(\Delta_i, \Delta_{i+1})$ is the (key-independent) probability over one round. However, recent works [AK18, BR22, BDG25] show that the gap between the fixed-key differential probability and the EDP can be significant, and should be taken into account when evaluating a cipher.

These assumptions allow reducing the study of differential properties to the problem of finding trails and computing the probability over each round. To compute one-round

[1] A plaintext x verifies the trail $(\Delta_0, \ldots, \Delta_r)$ if the intermediate values of the encryptions of x and $x \oplus \Delta_0$, for the same key, differ from Δ_i at round i for all $i \in \{1, \ldots, r\}$.

probabilities, cryptanalysts study the differential properties of the non-linear layer. More precisely, they rely on the *difference distribution table* of the S-box, which is a $2^s \times 2^s$-table such that the entry at position $(\delta_{in}, \delta_{out}) \in \mathbb{F}_{2^s} \times \mathbb{F}_{2^s}$ is $2^s DP_S(\delta_{in}, \delta_{out})$. Understanding the propagation through the other layers is trivial since they are affine. For the linear layer L, this propagation is fully deterministic: an input difference Δ maps to $L(\Delta)$ at the output with probability 1, since $L(x \oplus \Delta) \oplus L(x) = L(\Delta)$ for any x.

Since the number of trails is exponential in the state size n and the number of rounds r [LMM91], assessing the security of a block cipher with respect to differential cryptanalysis remains a non-trivial problem even under such assumptions. Therefore, designers typically upper bound the probability of the best differential trail and rely on the assumption that this is a good estimate of the probability of the best differential. In other words, the probability of a trail only lower bounds the probability of its associated differential. Yet, given the computational difficulty of computing differentials, this approach is frequently used as a heuristic way to assess (practical) security against differential cryptanalysis.

To compute such an upper bound, a classical approach is the following:

1. Compute the maximum probability of a differential transition over an S-box [NK92]:
$$p_{\max} = \frac{1}{2^s} \max_{(\delta_{in}, \delta_{out}) \neq (0,0)} \mathrm{DDT}(\delta_{in}, \delta_{out}) \,.$$
2. Estimate the minimum number N_r of *active* S-boxes (meaning the S-boxes with a non-zero input difference) in any r-round differential trail of the cipher.

The probability of the best trail over r rounds is then upper bounded by $(p_{\max})^{N_r}$ [DR20]. In the rest of the paper, we study the second step of this analysis.

2.3 Linear cryptanalysis

Since our following investigations apply almost identically to linear and differential cryptanalysis, we provide a brief description of linear cryptanalysis for completeness, and refer to [Mat93, Hey02] for the details. Linear cryptanalysis studies the probability that a random plaintext x and its associated ciphertext y satisfy $\langle \alpha, x \rangle = \langle \beta, y \rangle$, where $\alpha, \beta \in \mathbb{F}_2^n$ are called *linear masks*. To measure the distance of this probability to the random case, cryptanalysts use the correlation of the linear approximation as a metric. For a function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, it is defined as:

$$C^F(\alpha, \beta) = 2^{-n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x \rangle \oplus \langle \beta, F(x) \rangle} \,.$$

The correlation of a given pair of masks is hard to compute since it is exponential in the state size n. Therefore, and just like in differential cryptanalysis, a usual approach is to rely on *linear trails* (i.e., successive linear approximations over one round) and one-round correlations to estimate this value. To study the propagation through the S-box, one thus relies on the linear equivalent of a DDT, namely the $\mathbb{F}_2^s \times \mathbb{F}_2^s$ Linear Approximation Table (LAT), which has value $2^s C^F(\alpha, \beta)$ at entry (α, β). It also holds for linear cryptanalysis that the propagation of correlations through the linear layer is trivial. For the linear layer L, the linear approximation $\langle \alpha, x \rangle = \langle \beta, L(x) \rangle$ holds for all x if $\alpha = L^\top \beta$ since $\langle \beta, L(x) \rangle = \langle L^\top \beta, x \rangle$. When this holds, the correlation is ± 1, otherwise it is 0.

2.4 Estimating the minimum number of active S-boxes

With the branch number. The branch number of a linear layer allows computing a simple mathematical lower bound on the minimum number of active S-boxes over r rounds,

denoted as N_r. Let $L : \mathbb{F}_2^\ell \mapsto \mathbb{F}_2^\ell$ be an invertible $\mathbb{F}_2$-linear function. We denote with the same notation the $2^\ell \times 2^\ell$ matrix corresponding to this mapping.

The differential (and linear) branch numbers are then defined as follows.

Definition 2. The $\mathbb{F}_2$-differential branch number of a linear mapping L is:

$$\mathcal{B}(L) = \min_{x \in \mathbb{F}_2^\ell \setminus \{0\}} \mathrm{wt}(x) + \mathrm{wt}(Lx)\,,$$

where wt denotes the Hamming weight function (here of a vector of bits).

Similarly, the $\mathbb{F}_2$-linear branch number of a linear layer L is $\mathcal{B}(L^\top)$.

Remark 1. It directly follows that the differential and linear branch numbers are equal when L is symmetric: $L = L^\top$, or orthogonal: $LL^\top = I$ [ARSÖ17].

In the case of LS-designs considered in this work, where L is applied row-wise and S column-wise, the minimum number of active S-boxes before and after the state-wise linear layer is given by $\mathcal{B}(L)$. The minimum number of differentially (resp., linearly) active S-boxes in two consecutive rounds is thus $\mathcal{B}(L)$ (resp., $\mathcal{B}(L^\top)$), and the minimum number of active S-boxes over r rounds is lower bounded by $\lfloor r/2 \rfloor \mathcal{B}(L)$ (resp., $\lfloor r/2 \rfloor \mathcal{B}(L^\top)$).

With other techniques. Beyond branch-number-based bounds, several other algorithms to automatically count/prove the minimum number of active S-boxes have been proposed. Mixed Integer Linear Programming (MILP) is one such technique that helps deriving exact bounds by modeling S-boxes and linear diffusion layers in form of linear constraints. This technique was first used by Mouha *et al.* [MWGP11] on byte-oriented ciphers such as the AES. Sun et al. then extended this work from byte-oriented to bit-oriented ciphers in a series of works that evolved from first modeling the bit-level differences of S-boxes as binary variables [SHS+13] and then incorporating the exact DDT and LAT tables of S-boxes in order to track the true propagation of differences and linear masks across r-round differentials and linear characteristics [SHW+14]. Several works followed these seminal contributions and suggested different ways of modeling both the non-linear and linear layers in block ciphers [ST17, AST+17, BC20, TJTS21, IS24]. Alternatively, the problem of counting the minimum number of active S-boxes has been converted into a Boolean Satifiability problem and using SAT (or SMT) solvers to prove exact bounds [SWW18, LW19, LLL+21, SWW21]. Some proposals also involve the use of Constraint Programming (CP) [GMS16, GLMS20, DFJL19] and neural networks [SGBP20, ITYY21].

Application to LS-designs. Robin and Fantomas have a 8×16-bit state. For Robin, the 8-bit S-box and 16-bit L-box are involutive. The L-box is generated with a Reed Muller code with parameters $[32, 16, 8]$ and has branch number 8. For Fantomas, the S-box and L-box are non-involutive. The L-box is obtained by applying random row and column permutations to the Robin L-box. These operations preserve the branch number.

The involutive nature of Robin's round function allows to easily find trails with a number of active S-boxes that matches the wide-trail lower bound. By contrast, for Fantomas, the non-involutive components were expected to offer better results than guaranteed by this branch number bound. In order to validate this expectation, the authors performed a truncated state search in order to count the minimum number of active S-boxes. Their method represents each S-box activity by a single bit mask (active = 1, inactive = 0). The $s \times \ell$ state is thus truncated to a $1 \times \ell$ bit mask, where each bit represents the corresponding S-box column. This method enables them to provide lower bounds on the minimum number of active S-boxes for Fantomas-128, which we next denote as lower *Truncated Search Bounds* (TSB). As already mentioned in introduction, the bounds obtained from such a truncated state analysis may not always be tight since

they do not instantiate exact bit-level differences (resp., linear masks). It is therefore an open question to find out whether they are tight, which we discuss next.

3 Simple mathematical upper bound on the minimum number of active S-boxes for LS-Designs

In this section, we leverage the simplicity of LS-designs, and in particular the fact that the same S-box (resp., L-box) is applied independently to each column (resp., row), in order to provide a simple (mathematical) upper bound on the minimum number of active S-boxes in an r-round trail. While designers are admittedly more interested in lower bounds, we believe such an upper bound brings a convenient complement, since it allows containing the search for the minimum number of active S-boxes over r rounds within a *tighter* (bounded) interval. While this interval may be found by other (heuristic and less efficient) means, it comes for free in the case of LS-designs, which in turns can be exploited by tool-based searches which benefit from a clear termination criteria. This will be illustrated with an automated search using SAT solvers in the last sections of the paper.

In this context, we recall that the state is an $s \times \ell$ binary matrix. For a differential trail $\Delta = (\Delta_0, \ldots, \Delta_r)$, we let $\delta_{i,j}$ denote the j-th column of Δ_i, that is, the input difference of the jth S-box at round i, where r is the number of rounds and $0 \leq i \leq r-1$. We let $\delta'_{i,j}$ denote the output difference of this S-box and we denote by $\mathrm{AS}(\Delta)$ the number of active S-boxes in the trail Δ:

$$\mathrm{AS}(\Delta) \coloneqq |\{ \ (i,j) \text{ such that } \delta_{i,j} \neq 0\}| \ .$$

Our proof of the upper bound relies on showing that in an LS-design, a set of *compatible* trails has an easy-to-compute number of active S-boxes that depends *only* on the linear layer. We define the subset of compatible trails as:

$$\mathcal{E} = \left\{\Delta \ \text{ such that } \forall i,j, \ \mathrm{DDT}(\delta_{i,j}, \delta'_{i,j}) > 0\right\} \ .$$

Hence, it holds that the minimum number N_r of active S-boxes satisfies:

$$N_r = \min_{\Delta \in \mathcal{E}} \mathrm{AS}\left(\Delta\right) \ .$$

By definition, it thus holds that for any $\Delta \in \mathcal{E}$, $N_r \leq \mathrm{AS}(\Delta)$.

Our linear-layer dependent upper bound is then equal to the *minimum distance* of the *concatenated linear code* associated to L, which is defined as follows:

Definition 3. The r-round concatenated linear code $\mathcal{C}_L^r$ associated to L is the linear code defined by the following generating matrix:

$$\mathrm{G}_L^r = \left(I \parallel L \parallel L^2 \parallel \cdots \parallel L^{r-1}\right) \in \mathbb{F}_2^{\ell \times r\ell}, \text{ with } I \text{ the identity matrix.}$$

We recall, that the minimum distance of the linear code $\mathcal{C}_L^r$ is defined by:

$$d_{\min}\left(\mathcal{C}_L^r\right) = \min_{x \in \mathbb{F}_2^\ell \setminus \{0\}} \mathrm{wt}\left(\mathrm{G}_L^r x\right) \ .$$

Our goal to is prove the following theorem:

Theorem 1. *There exists a differential trail $\Delta^\star \in \mathcal{E}$ such that:*

$$\mathrm{AS}(\Delta^\star) = d_{\min}(\mathcal{C}_L^r) \, . \tag{1}$$

Since $N_r = \min_{\Delta \in \mathcal{E}} \mathrm{AS}(\Delta) \leq \mathrm{AS}(\Delta^\star) = d_{\min}(\mathcal{C}_L^r)$, a direct corollary is:

$$N_r \leq d_{\min}(\mathcal{C}_L^r)\,.$$

To prove Theorem 1, we start by establishing the following lemma, where we denote the zero vector of length s (resp., ℓ) by 0^s (resp., 0^ℓ).

Lemma 1. *Let $D \in \mathbb{F}_2^{s\times\ell} \setminus \{0^{s\times\ell}\}$ be any state difference. Denote by ρ_i, $i \in [\![0, s-1]\!]$ its rows and by c_j, $j \in [\![0, \ell-1]\!]$ its columns. The following statements are equivalent:*

(i) There exists $\delta^\rho \in \mathbb{F}_2^\ell \setminus \{0^\ell\}$ such that $\rho_i \in \{0, \delta^\rho\}$ $\forall i \in [\![0, s-1]\!]$.

(ii) There exists $\delta^c \in \mathbb{F}_2^s \setminus \{0^s\}$ such that $c_j \in \{0, \delta^c\}$ $\forall j \in [\![0, \ell-1]\!]$.

Proof. We prove **(ii)** $\Rightarrow$ **(i)**. Let $\delta^c \in \mathbb{F}_2^s \setminus \{0^s\}$ and let us assume that each column is either 0^s or equal to δ^c. Let us now define $\delta^\rho \in \mathbb{F}_2^\ell$ by:

$$\delta^\rho_j \coloneqq \begin{cases} 0 & \text{if } \; c_j = 0^s, \\ \delta^c_i & \text{if } \; c_j = \delta^c, \end{cases}$$

for all $j \in [\![0, \ell-1]\!]$. It follows that for any $i \in [\![0, s-1]\!]$,

$$\rho_i = \begin{cases} 0^\ell & \text{if } \; \delta^c_i = 0, \\ \delta^\rho & \text{otherwise.} \end{cases}$$

This proves the announced implication. The implication **(i)** $\Rightarrow$ **(ii)** can be proven similarly, or by observing that if D satisfies **(i)**, then $D^\top$ satisfies **(ii)** and therefore satisfies **(i)** by the proof above. Hence, D satisfies **(ii)**. □

Equipped with this Lemma, we now prove Theorem 1:

Proof of Theorem 1. We show that there exists $\Delta^\star \in \mathcal{E}$ such that $\mathrm{AS}(\Delta^\star) = d_{\min}(\mathcal{C}_L^r)$. By definition of the minimum distance, there exists a non-zero vector $x_0 \in \mathbb{F}_2^\ell$ such that:

$$\mathrm{wt}(\mathrm{G}_L^r x_0) = d_{\min}(\mathcal{C}_L^r)\ .$$

For any $i \in [\![0, r-1]\!]$, we let $x_i \coloneqq L^i x_0 \in \mathbb{F}_2^\ell$. It comes that:

$$\mathrm{G}_L^r x_0 = x_0 || x_1 || \ldots || x_{r-1} \in \mathbb{F}_2^{r\ell}\,.$$

Equality 1 can be shown by induction. We consider a non-zero initial state difference $\Delta_0 \in \mathbb{F}_2^{s\times\ell} \setminus \{0^{s\times\ell}\}$ such that any of its rows is either 0^ℓ or x_0. In particular, this state satisfies condition (ii) of Lemma 1, and we let δ^c be the constant value such that each column is either 0^s or δ^c. There exists $\delta' \in \mathbb{F}_2^s$ such that $\mathrm{DDT}_S(\delta^c, \delta') > 0$. Let Δ_0' be the state such that for any $i \in [\![0, s-1]\!]$:

- if the column of index i of Δ_0 is zero, it is set to zero in Δ_0';
- otherwise, this column is set to δ'.

Since $\mathrm{DDT}_S(\delta^c, \delta') > 0$, $\Delta_0 \to \Delta_0'$ is a valid transition through the S-box layer. Furthermore, all the columns of Δ_0' are either 0^s or δ'. So by Lemma 1, all its rows are either 0^ℓ or some constant. Since the column activity pattern before and after the S-box layer is constant, it is straightforward that this constant is x_0. Finally, since the linear layer consists in the parallel application of the mapping L on each row, we set Δ_1 to be the state such that for any $j \in [\![0, \ell-1]\!]$:

- if the row of index j of Δ_0' is zero, it is set to zero in Δ_1;

- otherwise, this row is set to $L(x_0) = x_1$.

Using the same reasoning, we can show that a non-zero difference Δ_i such that all its rows are equal to 0^ℓ or x_i has a valid differential transition through a round to a state Δ_{i+1} such that all its rows are equal to 0^ℓ or x_{i+1}. The differential $\Delta^\star = (\Delta_0, \ldots, \Delta_r)$ satisfies the desired property, and the theorem is proved. □

This simple bound is easy to compute. Indeed, the problem of determining the minimum distance of a linear code is well-studied. While computing the minimum-weight vector deterministically is exponential in the dimension of the code, Prange's algorithm [Pra62] allows to efficiently compute such a vector with overwhelming probability. For example, one can rely on Pernot and Leurent's implementation of this algorithm [LP24]. Additionally, computing a minimum-weight vector for G_L^r for $r \in [\![1, d-1]\!]$, with d the order of the matrix, fully determines the minimum-weight vector for an arbitrary large number of rounds.

We will give an application of this bound to concrete ciphers (including Fantomas) in Sections 5 and 6. Beforehand, we discuss the SAT modeling that we will use in combination with mathematical lower and upper bounds to obtain tight security claims.

4 SAT modeling of LS-designs

In this section, we describe how to model the round function of an LS-design cipher as a Boolean satisfiability problem (SAT) in order to perform an automated search for the minimum number of active S-boxes. We start with the encoding of the S-box, follow with the encoding of the L-box and conclude with the objective function where we take leverage of the mathematical upper bound from Section 3 along with the branch number lower bounds to perform a bounded search for the minimum. We focus our descriptions on differential trails. The modeling of linear trails is similar, replacing the DDT by LAT.

4.1 Encoding the S-box layer

As per Section 2.1, the S-box layer consists in the application of the same s-bit S-box to each column of the $s \times \ell$ state matrix. To search for differential trails, we need to model which S-boxes are active and ensure that the differential transitions are valid. To do so, we combine the generic modeling approach for bit-oriented ciphers in [SHS+13] with the approach in [SHW+14] to model the S-box specific differential propagation through the S-box with the help of DDT. We then convert the modeling constraints into SAT readable Conjunctive Normal Form (CNF). For any $0 \le i \le s-1$, $0 \le j \le \ell - 1$, $0 \le u \le r-1$, we let $x_{i,j}^u$ be the bit at coordinates (i, j) of the difference at the input of the S-box layer at round u, and $y_{i,j}^u$ be the corresponding bit of the difference at the output of the S-box layer. For any u, j, we let S_j^u be the activity variable associated to the S-box at column j in round u. Modeling the S-box layer requires the following set of clauses:

S-box activity constraints. The S-box is active if and only if its input difference is non-zero, i.e.,

$$S_j^u \iff \bigvee_{i=0}^{s-1} x_{i,j}^u.$$

This bi-conditional is encoded with the help of two clauses. First, if any of the $x_{i,j}^u$ is non-zero, then S_j^u must be active:

$$\left(\neg x_{i,j}^u \vee S_j^u\right) \qquad \forall i \in \{0, \ldots, s-1\}.$$

Second, if S_j^u is active, then at least one of the bits $x_{i,j}^u$ is non-zero:

$$\left(\neg S_j^u \vee x_{0,j}^u \vee x_{1,j}^u \vee \cdots \vee x_{s-1,j}^u\right).$$

Bijective S-box constraints. Since we deal with a bijective S-box, an input difference is non-zero if and only if the output difference is non-zero. This yields the following clauses, for each of the necessary and sufficient conditions:

$$(\neg x^u_{i,j} \vee y^u_{0,j} \vee y^u_{1,j} \vee \cdots \vee y^u_{s-1,j}), \qquad \forall i \in \{0, \ldots, s-1\},$$

and

$$(\neg y^u_{i,j} \vee x^u_{0,j} \vee x^u_{1,j} \vee \cdots \vee x^u_{s-1,j}), \qquad \forall i \in \{0, \ldots, s-1\}.$$

DDT constraints. Sun et al. [SHW+14] introduced an approach to encode the DDT constraints of an S-box (size $s \leq 6$). We use its refinement by Abdelkhalek et al. [AST+17] to handle larger S-boxes (size $s \leq 8$). This refinement is referred to as $\star$-DDT (a similar definition of $\star$-LAT exists for the LAT). The $\star$-DDT of S is the $2^s \times 2^s$ binary table obtained from the DDT of S by focusing *solely* on the possibility of differential transitions, and not on their probabilities. More formally, for any $\Delta_{\text{in}}, \Delta_{\text{out}} \in \mathbb{F}_2^s$, the entry $\star\text{-DDT}_S(\Delta_{\text{in}}, \Delta_{\text{out}})$ equals 1 if $\text{DDT}_S(\Delta_{\text{in}}, \Delta_{\text{out}}) > 0$ and 0 otherwise. As per [AST+17], to each non-zero entry $\star\text{-DDT}_S(\Delta_{\text{in}}, \Delta_{\text{out}}) = 1$ we associate the Boolean function $f_{\Delta_{\text{in}},\Delta_{\text{out}}} : \mathbb{F}_2^{2s} \to \mathbb{F}_2$ defined by $f_{\Delta_{\text{in}},\Delta_{\text{out}}}(x, y) = 1$ if and only if $(x, y) = (\Delta_{\text{in}}, \Delta_{\text{out}})$. Enforcing $f_{\Delta_{\text{in}},\Delta_{\text{out}}}(x^u_{0,j}, \ldots, x^u_{s-1,j}, y^u_{0,j}, \ldots, y^u_{s-1,j}) \geq 1$ for every $0 \leq u \leq r-1$ and $0 \leq j \leq \ell - 1$. This constraint thus excludes exactly the differential transitions that are not possible through the S-box. In practice, each $f_{\Delta_{\text{in}},\Delta_{\text{out}}}$ is represented as a CNF, that is, as a product-of-sums. To minimize the number of clauses, we use the SboxAnalyzer [HNE22], which relies on the Espresso logic minimizer [BHMS84].[2]

4.2 Encoding the linear layer

Depending on the SAT solver, linear constraints can be modeled in two ways.

Indirect XOR encoding. Most SAT solvers, such as Glucose [AS18] and CaDiCaL [BFF+24], do not support XOR as a native operation and only deal with CNFs. In some sporadic cases, such as MiniSAT [ES03], linear combinations can be added to the input model but they are automatically translated into multiple CNF clauses. For more flexibility, we choose to handle such translations ourselves before submitting the model to the solver. We rely on the Tseitin transformation [Tse83]: we first decompose any multi-XOR constraint into a sequence of three-variable XORs, that is, constraints of the form $a \oplus b \oplus c = 0$ or $a \oplus b \oplus c = 1$, by introducing sufficiently many auxiliary variables. Each three-variable XOR is modeled by the following four CNF clauses:

$$\begin{aligned}
&(\neg a \vee \neg b \vee \neg c),\\
&(a \vee b \vee \neg c),\\
&(a \vee \neg b \vee c),\\
&(\neg a \vee b \vee c).
\end{aligned}$$

Direct XOR encoding. As an exception, CryptoMiniSat [SNC09] comes with built-in support for the XOR operation. In cryptographic applications, this enables avoiding auxiliary variables and decreasing the number of CNF clauses generated by the translation of linear constraints. The solver also benefits from this handling of XOR operations, allowing strategies such as solving updated linear constraints by Gaussian elimination after each variable assignment [Soo10]. Such strategies sometimes lead to a faster solving time than with the indirect XOR encoding, sometimes at the cost of a higher memory usage.

[2] The $\star$-DDT clauses implicitly encode the bijectivity of the S-box: a zero input difference can only map to a zero output difference and vice versa (so no more bijectivity clauses are required with this encoding).

Combining the encodings of Sections 4.1 and 4.2, gives us a SAT model Φ_r whose assignments encode the bit level differential trails over r-rounds of an LS-design cipher.

4.3 Objective Function

As discussed above, our goal is to estimate the minimum number of active S-boxes N_r in r-round differential trails. In the automated cryptanalysis literature, this estimation is usually represented as an optimization problem that minimizes:

$$\sum_{u=0}^{r-1}\sum_{j=0}^{\ell-1} S_j^u. \tag{2}$$

Optimization objectives are not natively supported by SAT solvers. Therefore, we convert the minimization problem in Eq. (2) to a sequence of feasibility checks by adding a constraint to Φ_r that bounds the sum of active variables to at-most $t \in \mathbb{N}$. We denote the resulting formula as $\Phi_r^{\leq t}$. The at-most cardinality constraint is written as follows:

$$\sum_{u=0}^{r-1}\sum_{j=0}^{\ell-1} S_j^u \;\leq\; t. \tag{3}$$

By construction, the smallest value of t for which $\Phi_r^{\leq t}$ is satisfiable gives us the minimum number of active S-boxes N_r. The cardinality constraint in Eq. (3) is encoded into CNF by using methods such as the sequential counter encoding [Sin05], available via the PySAT toolkit [IMM18], which provides multiple encodings and state-of-the-art SAT solvers.

Linear Search for Minimum N_r. To search for the values of t for which $\Phi_r^{\leq t}$ is satisfiable and identify the minimum value, there are two direct approaches:

- Searching linearly upwards (i.e., testing $t = lb, lb+1, lb+2, \dots$) from the branch number lower bound until the tool concludes that $\Phi_r^{\leq t}$ is SAT.
- Starting from a trivial upper bound, which in the case of LS-designs is $r \times \ell$, and search linearly downwards (i.e., testing $t = ub_{\text{trivial}}, ub_{\text{trivial}}-1, ub_{\text{trivial}}-2, \dots$) until the first UNSAT instance of $\Phi_r^{\leq t}$ is found by the tool.

Naturally, the lack of precise interval may cause the solving process to be slow. We next show that one can take advantage of the upper bound (ub) in Section 3 combined with the branch number lower bound (lb) in order to effectively reduce the search space.

Binary search for the minimum N_r. Since the cardinality is an "at most" constraint, a trail that satisfies for $\leq t$ implies it also satisfies for $\leq t'$ for any $t' \geq t$. Hence, we can perform the binary search starting from the non-trivial bounded interval $[lb, ub]$, where at each step we evaluate a midpoint value $t = \lfloor (lb+ub)/2 \rfloor$. If $\Phi_r^{\leq t}$ is satisfiable, we update the upper edge of the interval to $t-1$; otherwise, we update the lower edge to $t+1$ and look for the next midpoint. The search terminates when the interval contains a single value, which is N_r. It requires a total of at most $\lceil \log_2(ub - lb + 1) \rceil$ calls to the solver, in comparison with the linear search that needs up to $(ub - lb + 1)$ calls over the same bounded interval (and possibly more if the bound interval is unknown). For example, in the case of Fantomas-128 at $r = 12$, we have $lb = 48$ and $ub = 72$. Hence, in the worst case, the linear search requires up to 25 solver calls. The linear search upwards starting from lb finds $N_r = 64$ after 17 solver calls. The linear search downwards from ub takes 9 solver calls to reach the same. Binary search finds this minimum value in 5 solver calls. Admittedly, these comparisons are not strict since various heuristic strategies could be used to estimate

the interval (including using a trivial upper bound). Nevertheless, our mathematical upper bound helps performing principled searches and reducing computational efforts, which can become practically-relevant for large-scale examples. We report more empirical comparisons between the binary and linear searches in Section 5.2.

5 Tight SAT-based bounds

We now show how the mathematical upper bounds from Section 3 and the branch number lower bounds can be combined to perform an automated SAT-based search for counting the minimum number of differentially/linearly active S-boxes. We illustrate this approach with Fantomas-128 for which it is not known whether existing claims are tight.

5.1 Computing infrastructure and solvers

The cost of computing the mathematical upper bounds in Section 3 for up to 16 rounds is almost negligible in memory and time. By contrast, the use of automated tools for problems such as finding the minimum number of active S-boxes becomes computationally intensive as the number of cipher rounds grows. All the search results reported next are obtained using a machine with the following specifications:

- Processor: AMD Ryzen Threadripper 3990X 64 core,
- Number of CPU cores: 64 cores / 128 threads available,
- Memory: 218 GB RAM,
- Operating system: Ubuntu 24.04.4 LTS.

As for the solvers, preliminary tests indicated that SAT solvers outperform MILP optimization in our context, likely due to the possibility to leverage bounded feasibility checks for a small interval of solutions. We then compared CaDiCAL v1.9.5 and CryptoMiniSAT with single threads per instance running in parallel across cores. Both could run with limited memory (in the gigabytes range). Somewhat surprisingly, and despite CryptoMiniSAT's specialized support for handling multi-XOR's with direct encoding, it turned out CaDiCAL gave the best (fastest) results for the larger instances (with 16 rounds). We could not make this observation match with a reasoning based on the number of clauses, and therefore posit that it is due to the structure of the problem being better captured by CaDiCAL [CAH23].[3] We ran our SAT-based searches for up to 10 days of computations, which allowed us analyzing up to 16 rounds of the Fantomas-128 cipher.

5.2 Empirical comparison of search strategies

We compared the binary search approach discussed in Section 4.3 against the linear search upwards, starting at the lower bound ($t = lb, lb+1, lb+2, \dots$) until the first SAT instance is found. For each $r \in \{3, \dots, 11\}$, we ran both strategies on Fantomas-128 over the search interval $[lb, ub]$ defined by the branch number lower bound and the upper bound from Theorem 1. For each strategy, we report the number of solver calls, the total (build + solve) time and the peak memory usage across all solver calls. The results are summarized in Table 1 and Fig. 2. As expected, binary search systematically outperforms linear search for all metrics and the gap increases when the search interval $[lb, ub]$ grows.

[3] More precisely, encoding the linear layer for one round of Fantomas with the direct XOR encoding results in 128 clauses vs. a significantly larger 3328 clauses with the indirect XOR encoding.

Table 1: Performances of binary and linear searches on Fantomas-128.

					Binary search			Linear search (upwards)		
r	lb	ub	size	N_r	calls	time (s)	mem (MB)	calls	time (s)	mem (MB)
2	8	8	1	8	1	–	–	1	–	–
3	8	12	5	12	3	23	386	5	29	642
4	16	20	5	20	3	6362	758	5	6976	1048
5	16	24	5	24	4	6103	966	9	12094	2049
6	24	32	9	30	3	23048	956	7	23898	1928
7	24	36	13	34	4	104542	1171	11	73465	2860
8	32	44	13	40	4	155230	1980	9	302658	3935
9	32	48	17	46	4	206780	2102	15	1083394	7275
10	40	56	17	52	4	486865	2642	13	1050673	5936
11	40	62	23	58	4	531591	3373	19	1374511	8133

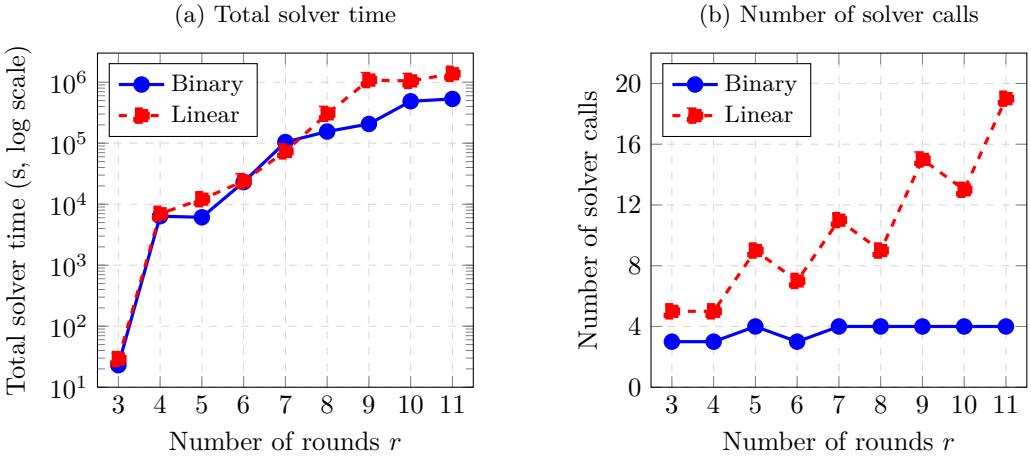

Figure 2: Performances of binary and linear searches on Fantomas-128.

5.3 Empirical results for Fantomas-128

Table 2 reports results on the exact minimum number active S-boxes for Fantomas-128 for up to 16 rounds. The upper part of the table corresponds to previous results that may not be tight. The lower part of the table reports the mathematical upper bound of Section 3 and the outcomes of our (exact) SAT-based searches with the computing infrastructure and solvers described above. Regarding the mathematical upper bound, we observe that it gradually loses tightness as the number of rounds increases (e.g., comparing it with the truncated search from [GLSV14]), justifying the combination with a tool-based search. As for the comparison between the truncated search from [GLSV14], we observe that it matches the results of our SAT-based searches. This is interesting because, as already mentioned, the FSE 2014 truncated search captures diffusion at the level of column activity patterns. Hence, it does not enforce the feasibility of corresponding bit-level differential transitions, and the resulting bounds are not guaranteed to be tight (i.e., may correspond to activity patterns that cannot be realized as valid differential or linear trails). This is in contrast with the SAT-based search where the modeling of Section 4 ensures that any impossible bit-level differences are excluded from the model (i.e., solutions obtained this way are guaranteed to correspond to valid differential or linear trails). Overall, our SAT analysis therefore confirms the bounds reported in [GLSV14]. We note that this is specific to Fantomas-128 and does not extend automatically to any LS-designs.

Table 2: Bounds on the minimum number of active S-boxes for FANTOMAS-128.

Bound Type	**Rounds**														
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BNB	8	9	16	17	24	25	32	33	40	41	48	49	56	57	64
TSB	8	12	20	24	30	34	40	46	52	58	64	68	74	80	86
M-UB	8	12	20	24	32	36	44	48	56	62	72	77	84	90	96
SAT-B ($\star$-DDT)	8	12	20	24	30	34	40	46	52	58	64	68	74	80	86
SAT-B ($\star$-LAT)	8	12	20	24	30	34	40	46	52	58	64	68	74	80	86

Notes. BNB: Branch Number Bound [DR20]. TSB: Truncated Search Bound [GLSV14]. M-UB: Mathematical upper bound (see Section 3). SAT-B: Tight bounds obtained via SAT solvers.

6 Application to a 32-bit L-box

As mentioned by the authors of [GLSV14], the truncated search method does not scale well starting from matrices of size 16-bits, since building all possible truncated transitions for even a single 16-bit L-box is computationally intensive. More recently, Leurent and Pernot noted that the truncated state search does not support 32-bit words [LP24]. Interestingly the SAT-based method in Section 4 models bit-level differential (or linear) transitions directly via the $\star$-DDT (or $\star$-LAT) clauses. Hence, this method scales better to larger sizes. The cost of solving the model naturally grows with the round count and the state size, but the construction step itself does not run into a similar bottleneck as the truncated state search. In this section, we therefore illustrate the applicability of the SAT-based approach on an exemplary $s \times \ell = 4 \times 32 = 128$-bit instance of an LS-design cipher.

For the linear layer, we use a 32-bit right-circulant L-box `0xcf3000a4` with a branch number $B(L) = 12$. For the S-box layer, we do not commit to a specific 4-bit S-box and instead rely on the generic modeling steps for a bijective S-box as discussed in Section 4.1. This corresponds to studying the minimum number of S-boxes that must be active in any LS-design built around this L-box, regardless of the choice of an S-box. The L-box generation, branch number and upper bound computation via Prange's algorithm are all steps carried out using Leurent and Pernot's implementation discussed in [LP24].

Table 3: Bounds on the minimum number of active S-boxes for a 4×32-bit LS-design.

Bound Type	**Rounds**			
	2	3	4	5
BNB	12	13	24	25
M-UB	12	18	30	37
SAT-B (Bijective S-box)	12	18	30	37

As we can see in Table 3, the results obtained for the minimum number of active S-boxes using the SAT search match the mathematical upper bounds obtained from Theorem 1. Therefore, for the tested numbers of rounds, the mathematical upper bound is tight for this instance of cipher with a 32-bit right circulant L-box. As discussed in [LP24], the linear codes associated to the circulant L-boxes their implementation gives are quasi cyclic (i.e., double-circulant) codes. These codes are known in coding theory to have good properties. Finding whether the tightness of the mathematical upper bounds is directly connected to the quasi-cyclic structure of the linear codes is an interesting scope for further investigation.

7 Conclusion

Analyzing the security of symmetric cryptographic primitives like block ciphers is difficult and error-prone. In this paper, we show that the conceptual simplicity of LS-designs allows deriving mathematical bounds and obtaining efficient tool-based results for the minimum number of active S-boxes in differential and linear trails. This consolidates the understanding of their robustness against linear and differential cryptanalysis and leads to the following two directions for further research. On the tool side, our results for now focus on the minimum number of active S-boxes in linear and differential trails. A natural next step would be to look for trails with high probability, which is expected to be more computationally intensive. Hence, one could use the trails obtained from our analysis in order to try restricting the search space by pruning impossible paths. On the design side, our results could facilitate the investigation of new LS-design instances operating on larger states (e.g., for permutation-based designs). The interleaved L-boxes proposed in [BBB+20] and extended in [LP24] appear as relevant candidates for this purpose.

References

[AK18] Ralph Ankele and Stefan Kölbl. Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis. In Carlos Cid and Michael J. Jacobson Jr., editors, *Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers*, Lecture Notes in Computer Science, pages 163–190. Springer, 2018. doi:10.1007/978-3-030-10970-7_8.

[ARSÖ17] Sedat Akleylek, Vincent Rijmen, Muharrem Tolga Sakalli, and Emir Öztürk. Efficient methods to generate cryptographically significant binary diffusion layers. *IET Inf. Secur.*, 11(4):177–187, 2017. URL: https://doi.org/10.1049/iet-ifs.2016.0085, doi:10.1049/IET-IFS.2016.0085.

[AS18] Gilles Audemard and Laurent Simon. On the glucose SAT solver. *Int. J. Artif. Intell. Tools*, 27(1):1840001:1–1840001:25, 2018. doi:10.1142/S0218213018400018.

[AST+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP modeling for (large) s-boxes to optimize probability of differential characteristics. *IACR Trans. Symmetric Cryptol.*, 2017(4):99–129, 2017. URL: https://doi.org/10.13154/tosc.v2017.i4.99-129, doi:10.13154/TOSC.V2017.I4.99-129.

[BBB+20] Davide Bellizia, Francesco Berti, Olivier Bronchain, Gaëtan Cassiers, Sébastien Duval, Chun Guo, Gregor Leander, Gaëtan Leurent, Itamar Levi, Charles Momin, Olivier Pereira, Thomas Peters, François-Xavier Standaert, Balazs Udvarhelyi, and Friedrich Wiemer. Spook: Sponge-based leakage-resistant authenticated encryption with a masked tweakable block cipher. *IACR Trans. Symmetric Cryptol.*, 2020(S1):295–349, 2020. URL: https://doi.org/10.13154/tosc.v2020.iS1.295-349, doi:10.13154/TOSC.V2020.IS1.295-349.

[BC20] Christina Boura and Daniel Coggia. Efficient MILP modelings for sboxes and linear layers of SPN ciphers. *IACR Trans. Symmetric Cryptol.*, 2020(3):327–361, 2020. URL: https://doi.org/10.13154/tosc.v2020.i3.327-361, doi:10.13154/TOSC.V2020.I3.327-361.

[BDG25] Christina Boura, Patrick Derbez, and Baptiste Germon. Extending the quasidifferential framework: From fixed-key to expected differential probability.

IACR Trans. Symmetric Cryptol., 2025(1):515–541, 2025. URL: https://doi.org/10.46586/tosc.v2025.i1.515-541, doi:10.46586/TOSC.V2025.I1.515-541.

[BFF+24] Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. Cadical 2.0. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part I*, volume 14681 of *Lecture Notes in Computer Science*, pages 133–152. Springer, 2024. doi:10.1007/978-3-031-65627-9_7.

[BHMS84] Robert K. Brayton, Gary D. Hachtel, Curtis T. McMullen, and Alberto L. Sangiovanni-Vincentelli. *Logic Minimization Algorithms for VLSI Synthesis*, volume 2 of *The Kluwer International Series in Engineering and Computer Science*. Springer, 1984. doi:10.1007/978-1-4613-2821-6.

[BR22] Tim Beyne and Vincent Rijmen. Differential cryptanalysis in the fixed-key model. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, Lecture Notes in Computer Science, pages 687–716. Springer, 2022. doi:10.1007/978-3-031-15982-4_23.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *J. Cryptol.*, 4(1):3–72, 1991. doi:10.1007/BF00630563.

[CAH23] Cayden R. Codel, Jeremy Avigad, and Marijn J. H. Heule. Verified encodings for SAT solvers. In Alexander Nadel and Kristin Yvonne Rozier, editors, *Formal Methods in Computer-Aided Design, FMCAD 2023, Ames, IA, USA, October 24-27, 2023*, pages 141–151. IEEE, 2023. URL: https://doi.org/10.34727/2023/isbn.978-3-85448-060-0_22, doi:10.34727/2023/ISBN.978-3-85448-060-0_22.

[DDSS19] Ashutosh Dhar Dwivedi, Shalini Dhar, Gautam Srivastava, and Rajani Singh. Cryptanalysis of round-reduced fantomas, robin and iscream. *Cryptogr.*, 3(1):4, 2019. URL: https://doi.org/10.3390/cryptography3010004, doi:10.3390/CRYPTOGRAPHY3010004.

[DFJL19] Patrick Derbez, Pierre-Alain Fouque, Jérémy Jean, and Baptiste Lambin. Variants of the AES key schedule for better truncated differential bounds. *IACR Cryptol. ePrint Arch.*, page 95, 2019. URL: https://eprint.iacr.org/2019/095.

[DR07] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *J. Math. Cryptol.*, 1(3):221–242, 2007. doi:10.1515/JMC.2007.011.

[DR20] Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition*. Information Security and Cryptography. Springer, 2020. doi:10.1007/978-3-662-60769-5.

[ES03] Niklas Eén and Niklas Sörensson. An extensible sat-solver. In Enrico Giunchiglia and Armando Tacchella, editors, *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 502–518. Springer, 2003. doi:10.1007/978-3-540-24605-3_37.

[GLMS20] David Gérault, Pascal Lafourcade, Marine Minier, and Christine Solnon. Computing AES related-key differential characteristics with constraint programming. *Artif. Intell.*, 278, 2020. URL: https://doi.org/10.1016/j.artint.2019.103183, doi:10.1016/J.ARTINT.2019.103183.

[GLSV14] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici. Ls-designs: Bitslice encryption for efficient masked software implementations. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers*, volume 8540 of *Lecture Notes in Computer Science*, pages 18–37. Springer, 2014. doi:10.1007/978-3-662-46706-0_2.

[GMS16] David Gérault, Marine Minier, and Christine Solnon. Constraint programming models for chosen key differential cryptanalysis. In Michel Rueher, editor, *Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings*, volume 9892 of *Lecture Notes in Computer Science*, pages 584–601. Springer, 2016. doi:10.1007/978-3-319-44953-1_37.

[Hey02] Howard M. Heys. A tutorial on linear and differential cryptanalysis. *Cryptologia*, 26(3):189–221, 2002. doi:10.1080/0161-110291890885.

[HNE22] Hosein Hadipour, Marcel Nageler, and Maria Eichlseder. Throwing boomerangs into feistel structures application to clefia, warp, lblock, lblock-s and TWINE. *IACR Trans. Symmetric Cryptol.*, 2022(3):271–302, 2022. URL: https://doi.org/10.46586/tosc.v2022.i3.271-302, doi:10.46586/TOSC.V2022.I3.271-302.

[IMM18] Alexey Ignatiev, António Morgado, and João Marques-Silva. Pysat: A python toolkit for prototyping with SAT oracles. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings*, volume 10929 of *Lecture Notes in Computer Science*, pages 428–437. Springer, 2018. doi:10.1007/978-3-319-94144-8_26.

[IS24] Murat Burhan Ilter and Ali Aydin Selçuk. MILP modeling of matrix multiplication: cryptanalysis of KLEIN and PRINCE. *Turkish J. Electr. Eng. Comput. Sci.*, 32(1):183–197, 2024. doi:10.55730/1300-0632.4062.

[ITYY21] Mohamed Fadl Idris, Je Sen Teh, Jasy Liew Suet Yan, and Wei-Zhu Yeoh. A deep learning approach for active s-box prediction of lightweight generalized feistel block ciphers. *IEEE Access*, 9:104205–104216, 2021. doi:10.1109/ACCESS.2021.3099802.

[JSV17] Anthony Journault, François-Xavier Standaert, and Kerem Varici. Improving the security and efficiency of block ciphers based on ls-designs. *Des. Codes Cryptogr.*, 82(1-2):495–509, 2017. URL: https://doi.org/10.1007/s10623-016-0193-8, doi:10.1007/S10623-016-0193-8.

[LLL+21] Yu Liu, Huicong Liang, Muzhou Li, Luning Huang, Kai Hu, Chenhe Yang, and Meiqin Wang. STP models of optimal differential and linear trail for s-box based ciphers. *Sci. China Inf. Sci.*, 64(5), 2021. URL: https://doi.org/10.1007/s11432-018-9772-0, doi:10.1007/S11432-018-9772-0.

[LMM91] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38. Springer, 1991. doi:10.1007/3-540-46416-6_2.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of robin, iscream and zorro. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 254–283. Springer, 2015. doi:10.1007/978-3-662-46800-5_11.

[LP24] Gaëtan Leurent and Clara Pernot. Design of a linear layer optimised for bitsliced 32-bit implementation. *IACR Trans. Symmetric Cryptol.*, 2024(1):441–458, 2024. URL: https://doi.org/10.46586/tosc.v2024.i1.441-458, doi:10.46586/TOSC.V2024.I1.441-458.

[LW19] Huicong Liang and Meiqin Wang. Cryptanalysis of the lightweight block cipher BORON. *Secur. Commun. Networks*, 2019:7862738:1–7862738:12, 2019. doi:10.1155/2019/7862738.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993. doi:10.1007/3-540-48285-7_33.

[MT99] Mitsuru Matsui and Toshio Tokita. Cryptanalysis of a reduced version of the block cipher E2. In Lars R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 71–80. Springer, 1999. doi:10.1007/3-540-48519-8_6.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011. doi:10.1007/978-3-642-34704-7_5.

[NK92] Kaisa Nyberg and Lars R. Knudsen. Provable security against differential cryptanalysis. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 566–574. Springer, 1992. doi:10.1007/3-540-48071-4_41.

[PL23] Truong Minh Phuong and Tran Thi Luong. Evaluating the number of active s-boxes in dynamic aes block ciphers using mds matrices of size 4×4 and 8×8. *TNU Journal of Science and Technology*, 2023. URL: https://api.semanticscholar.org/CorpusID:267038799.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8(5):5–9, 1962. doi:10.1109/TIT.1962.1057777.

[SGBP20] Ling Sun, David Gérault, Adrien Benamira, and Thomas Peyrin. Neurogift: Using a machine learning based sat solver for cryptanalysis. In Shlomi Dolev, Vladimir Kolesnikov, Sachin Lodha, and Gera Weiss, editors, *Cyber Security Cryptography and Machine Learning - Fourth International Symposium, CSCML 2020, Be'er Sheva, Israel, July 2-3, 2020, Proceedings*, volume 12161 of *Lecture Notes in Computer Science*, pages 62–84. Springer, 2020. doi:10.1007/978-3-030-49785-9_5.

[SHS+13] Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang. Automatic security evaluation of block ciphers with s-bp structures against related-key differential attacks. In Dongdai Lin, Shouhuai Xu, and Moti Yung, editors, *Information Security and Cryptology - 9th International Conference, Inscrypt 2013, Guangzhou, China, November 27-30, 2013, Revised Selected Papers*, volume 8567 of *Lecture Notes in Computer Science*, pages 39–51. Springer, 2013. doi:10.1007/978-3-319-12087-4_3.

[SHW+14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 158–178. Springer, 2014. doi:10.1007/978-3-662-45611-8_9.

[Sin05] Carsten Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In Peter van Beek, editor, *Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings*, volume 3709 of *Lecture Notes in Computer Science*, pages 827–831. Springer, 2005. doi:10.1007/11564751_73.

[SLLQ18] Xuan Shen, Guoqiang Liu, Chao Li, and Longjiang Qu. Impossible differential cryptanalysis of fantomas and robin. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 101-A(5):863–866, 2018. URL: https://doi.org/10.1587/transfun.E101.A.863, doi:10.1587/TRANSFUN.E101.A.863.

[SMMR17] Mahdi Sajadieh, Arash Mirzaei, Hamid Mala, and Vincent Rijmen. A new counting method to bound the number of active s-boxes in rijndael and 3d. *Des. Codes Cryptogr.*, 83(2):327–343, 2017. URL: https://doi.org/10.1007/s10623-016-0217-4, doi:10.1007/S10623-016-0217-4.

[SNC09] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings*, volume 5584 of *Lecture Notes in Computer Science*, pages 244–257. Springer, 2009. doi:10.1007/978-3-642-02777-2_24.

[Soo10] Mate Soos. Enhanced gaussian elimination in dpll-based SAT solvers. In Daniel Le Berre, editor, *POS-10. Pragmatics of SAT, Edinburgh, UK, July 10,*

2010, volume 8 of *EPiC Series in Computing*, pages 2–14. EasyChair, 2010. URL: https://doi.org/10.29007/g7ss, doi:10.29007/G7SS.

[ST17] Yu Sasaki and Yosuke Todo. New algorithm for modeling s-box in MILP based differential and division trail search. In Pooya Farshim and Emil Simion, editors, *Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8-9, 2017, Revised Selected Papers*, volume 10543 of *Lecture Notes in Computer Science*, pages 150–165. Springer, 2017. doi:10.1007/978-3-319-69284-5_11.

[SWW18] Ling Sun, Wei Wang, and Meiqin Wang. More accurate differential properties of LED64 and midori64. *IACR Trans. Symmetric Cryptol.*, 2018(3):93–123, 2018. URL: https://doi.org/10.13154/tosc.v2018.i3.93-123, doi:10.13154/TOSC.V2018.I3.93-123.

[SWW21] Ling Sun, Wei Wang, and Meiqin Wang. Accelerating the search of differential and linear characteristics with the SAT method. *IACR Trans. Symmetric Cryptol.*, 2021(1):269–315, 2021. URL: https://doi.org/10.46586/tosc.v2021.i1.269-315, doi:10.46586/TOSC.V2021.I1.269-315.

[TJTS21] Vikas Tiwari, Neelima Jampala, Appala Naidu Tentu, and Ashutosh Saxena. Towards finding active number of s-boxes in block ciphers using mixed integer linear programming. *Informatica (Slovenia)*, 45(6), 2021. URL: https://doi.org/10.31449/inf.v45i6.3427, doi:10.31449/INF.V45I6.3427.

[Tse83] Grigori S Tseitin. On the complexity of derivation in propositional calculus. In *Automation of Reasoning*, pages 466–483. Springer, 1983.

[WJ19] Qian Wang and Chenhui Jin. A method to bound the number of active s-boxes for a kind of aes-like structure. *Comput. J.*, 62(8):1121–1131, 2019. URL: https://doi.org/10.1093/comjnl/bxz006, doi:10.1093/COMJNL/BXZ006.

[WW11] Shengbao Wu and Mingsheng Wang. Security evaluation against differential cryptanalysis for block cipher structures. *IACR Cryptol. ePrint Arch.*, page 551, 2011. URL: http://eprint.iacr.org/2011/551.