# Improved Modeling for Substitution Boxes with Negative Samples and Beyond

Debranjan Pal[1], Anubhab Baksi[2], Surajit Mandal[3], and Santanu Sarkar[3]

[1] Indian Institute of Technology Kanpur, Kanpur, India

[2] Lund University, Lund, Sweden

[3] Indian Institute of Technology Madras, Chennai, India

debranjan.crl@gmail.com, anubhab.baksi@eit.lth.se,
surajit1810@gmail.com, sarkar.santanu.bir1@gmail.com

**Abstract.** It is a common practice for symmetric-key ciphers is to encode a cryptanalysis problem as an instance of the Mixed Integer Linear Programming (MILP) and then run the instance with an efficient solver. For this purpose, it is essential to model the components in a way that is compatible with the MILP formulation while preserving the characteristics of the cipher. In this work, we look at the problem of efficiently encoding a substitution box (SBox for short). More specifically, we take the evaluation tables, namely, the Difference Distribution Table (DDT), the Linear Approximation Table (LAT), the Division Property Table (DPT) and the Boomerang Connectivity Table (BCT).
In the current stage, the only model proposed/used in the literature is to look for the positive samples (i.e., non-zero) entries in the evaluation table and encode for that (this ultimately leads to the minimum number of active SBoxes for the target cipher). In our first contribution, we experiment with the complementary concept of using the negative samples (i.e., treating the zero entries as non-zero entries and vice versa), finally a proper recovery procedure is performed to get back to the original problem.
In our second contribution, we observe that the top row and column of each table (which are always taken into consideration by the past researchers) are actually redundant, as those are inherently taken care of through additional constraints at another level. We thus propose a simplified model that removes those row and column.
We show the efficacy of our proposals by furnishing results on the evaluation tables on multiple SBoxes. Our proposals often reduce the number of constraints for certain SBox evaluation tables (depending on the properties of the table) from the state-of-the-art results.

**Keywords:** Block Cipher · Substitution Box · Substitution Box Evaluation Table · Mixed Integer Linear Programming

## 1 Introduction

Over the past few years, multiple tools to find the security bounds for ciphers have emerged. These tools take the problem description of the classical attacks, such as the differential or linear attacks, as a constraint satisfaction or objective optimization problem. Not only does it take away the tedious manual effort, but it is also capable of fine-tuning the bounds (thereby finding a more optimal bound). Naturally, there has been a great deal of interest in this research area, as evidenced by the recent publications. Generally, two types of solvers are used: Satisfiability (SAT) solvers like Cryptominisat and Mixed Integer Linear Programming (MILP) solvers like Gurobi or CPLEX are used, though Constraint Programming (CP), Quadratically Constrained Linear Programming, or Simple Theorem Prover (STP) [25] are occasionally employed as the tool of choice.

In fact, it is highly important to evaluate the security of classical attacks. This is typically done by the cipher designers (in the paper proposing a new cipher) and independent researchers interested in that cipher.

While automated solvers undeniably reduce the workload for the cipher designers and analyzers alike, there is the problem of formatting the problem (related to the bounds of the cipher) to a form compatible with the solver. When designing new ciphers, it is standard practice to prioritize resistance against classical

cryptanalysis. One crucial aspect of assessing a cipher's security against classical attacks is determining the minimum number of active SBoxes (Section 2.1) within the cipher[4]. Consequently, calculating the smallest number of active SBoxes for block ciphers [27, 36] has garnered significant attention in cryptographic research.

### 1.1 Overview of Prior Works

In this part, we briefly describe the major works in this field. The basic problem is to take an SBox evaluation table (*Difference Distribution Table* (DDT), *Linear Approximation Table* (LAT), *Division Property Table* (DPT), and *Boomerang Connectivity Table* (BCT)) and convert the information in the said table to a system of linear constraints (so that the constraints can be used in an MILP problem formulation). On top of that, a heuristic idea is to minimize the number of constraints [1, 2, 17, 24, 28, 33] (it is believed that the lower number of constraints will lead to faster runtime of the overall MILP instance), and this poses an extra challenge. Therefore, the fundamental challenges in this context are:

1. Construct an appropriate set of linear constraints that describes the given SBox evaluation table.
2. Identify a minimal subset of the said constraints that still preserves the SBox evaluation table.

To address these challenges, researchers have proposed various techniques. Sun et al. [33, 34] propose a method to convert from the SBox evaluation tables to linear constraints. Sun et al. [34] introduce a greedy method to eliminate redundant inequalities. In 2017, Sasaki and Todo [30] introduce an innovative algorithm designed to automatically reduce the size of descriptive models. In 2020, Boura and Coggia [18] develop a technique for generating new inequalities based on existing ones. Their experimental results showcase the ability to reduce inequalities compared to previous approaches. In 2017, Abdelkhalek et al. [1] present the first MILP model tailored for 8-bit SBoxes. They focus on the product-of-sum representation and employ the Espresso algorithm to generate the associated inequalities.

### 1.2 Contribution

Despite recent progress, it is observed that further optimization of results from previous works remains possible. In this paper, two novel approaches are presented to further optimize the state-of-the-art methods. In summary, improved models for SBoxes are proposed that are suitable for MILP-aided cryptanalysis to find the minimum number of active SBoxes.The approaches are general and can be applied to any SBox evaluation table including DDT, LAT, DPT and BCT. This paper restricts the analysis to bijective SBoxes for simplicity, though further generalization for non-bijective SBoxes is possible.

The key contributions can be summarized as:

- **Switching to negative samples.** A reformulation of the SBox constraint system is proposed to model the absence of specific transitions or correlations (i.e., the negative samples in the SBox evaluation tables). In this approach, the complement model (i.e., the maximum number of inactive SBoxes) is found, from which conversion to the actual problem (i.e., the minimum number of active SBoxes) is performed. If the solution from the MILP solver, say $I$, represents the maximum number of SBoxes that can be inactive, then to determine the minimal number of active SBoxes, this value $I$ is subtracted from the total number of SBoxes in the cipher.
- **Exclusion of $0^{\text{th}}$ row/column.** Through a survey of literature, it is observed that all existing works include the redundant top row and column (that correspond to the zero-to-zero transitions) in the SBox evaluation tables. However, this can be modeled through other means (the constraints which encode whether or not the SBox is active) automatically detect such conditions.

### 1.3 Organization

- In Section 2 the necessary prerequisites are provided, including definitions of the SBox and the cryptographic property tables used in the analysis, such as the *Difference Distribution Table* (DDT), *Linear Approximation Table* (LAT), *Division Property Table* (DPT), and *Boomerang Connectivity Table* (BCT).

[4] This method does not work on ciphers with linear structures, such as `DEFAULT` [5, Chapter 8] or `BAKSHEESH` [6].

- Section 3 reviews related work in the field, discussing previous methods for generating and reducing inequalities for SBox modeling.
- In Section 4 the first proposed approach is introduced, the concept of complementary variable encoding by focusing on modeling negative transitions (impossible propagation) to simplify the MILP model.
- The second proposed approach is presented in Section 5, which involves excluding redundant zero-to-zero transitions (the first row and column) in the SBox evaluation tables to further optimize the modeling process.
- Section 6 presents our experimental results, where we compare our proposed methods to traditional approaches by looking at the number of inequalities they produce across different SBoxes and property tables.
- Section 7 concludes the paper by summarizing our findings and discussing their implications for the automated security evaluation of ciphers.
- Additionally, Appendix A provides some relevant examples. These include description of our proposed methods applied to specific SBoxes and property tables. Indices to the SBoxes used/analyzed in our experiments are provided in Appendix B.

## 2 Prerequisites

### 2.1 Substitution Box (SBox)

A substitution box, (SBox for short), denoted $S$, is a mapping represented by $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$. Any operation on an SBox can be depicted as a transformation from an input $x$ to an output $y$, where both $x$ and $y$ belong to the finite field $\mathbb{F}_2^n$.

Consider the $2n$-bit vector $(x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{n-1})$ over $\mathbb{F}_2^{2n}$ where $x$ indicates the input characteristic to the SBox and $y$ indicates the output characteristic from the SBox. Depending on the property of interest, the input-output characteristics can be differential propagation (in the context of differential attack) or linear masking (in the context of linear attack) and so on.

### 2.2 Difference Distribution Table (DDT)

The differential behavior of an SBox is usually characterized by its *Difference Distribution Table* (DDT). For an $n$-bit input and an $m$-bit output SBox $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the DDT is a $2^n \times 2^m$ matrix whose entry at position $(\Delta x, \Delta y)$ is defined as $\mathrm{DDT}[\Delta x, \Delta y] = \#\{x \in \mathbb{F}_2^n \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\}$. Here, $\Delta x \in \mathbb{F}_2^n$ denotes an input difference and $\Delta y \in \mathbb{F}_2^m$ an output difference, while $\oplus$ represents bitwise XOR. Each row of the DDT corresponds to a fixed input difference and sums to $2^n$, reflecting that every input $x$ contributes exactly one output difference. The first row (for $\Delta x = 0$) is trivial, with $\mathrm{DDT}[0, 0] = 2^n$ and all other entries equal to zero. The maximum value among the nontrivial entries, called the *differential uniformity* of the SBox, determines its resistance against differential cryptanalysis [12, 14, 23].

### 2.3 Linear Approximation Table (LAT)

A *Linear Approximation Table* (LAT) is constructed to measure the biases of linear approximations between the input and output bits of a cryptographic function, such as an SBox. For an $n$-input, $m$-output function $F : \{0,1\}^n \to \{0,1\}^m$, we are interested in linear combinations of its inputs and outputs. Let $a \in \{0,1\}^n$ be a binary vector representing a linear combination of input bits and $b \in \{0,1\}^m$ is also a binary vector representing a linear combination of output bits. The linear approximation is expressed as: $a \cdot x \oplus b \cdot F(x) = 0$, where: $a \cdot x = \bigoplus_{i=1}^{n} a_i x_i$ (XOR of the selected input bits), $b \cdot F(x) = \bigoplus_{j=1}^{m} b_j F(x_j)$ (XOR of the selected output bits). Now, the bias measures how often this approximation holds true compared to random guessing. We define: $P(a, b) = \Pr(a \cdot x \oplus b \cdot F(x) = 0)$, where $x$ is chosen uniformly at random from $\{0,1\}^n$. The bias $\epsilon(a, b)$ is, $\epsilon(a, b) = P(a, b) - \frac{1}{2}$.

The entry $\mathrm{LAT}[a, b]$ in the table is computed as, $\mathrm{LAT}[a, b] = N \cdot \epsilon(a, b)$, where $N = 2^n$ is the total number of possible inputs. Alternatively, $\mathrm{LAT}[a, b] = \#\{x \in \{0,1\}^n : a \cdot x \oplus b \cdot F(x) = 0\} - \frac{N}{2}$.

### 2.4 Division Property Table (DPT)

Let $d$ represent the algebraic degree of the SBox. It is possible to select $2^{d+1}$ inputs to the SBox such that the XOR sum of the corresponding SBox outputs equals 0.

Let $\pi_u : \mathbb{F}_2^n \to \mathbb{F}_2$ be a bit-product function defined for any $u \in \mathbb{F}_2^n$ and $x \in \mathbb{F}_2^n$, where the $i$-th bits of $u$ and $x$ are denoted as $u_i$ and $x_i$, respectively. The function $\pi_u(x)$ is defined as: $\pi_u(x) := \prod_{i=1}^{n} x_i^{u_i}$.

Let $X$ be a multiset whose elements are values in $\mathbb{F}_2^n$, and $k$ be an integer between 0 and $n$. If the multiset $X$ satisfies the division property $D_n^k$, the following two conditions hold, the parity of $\pi_u(x)$ is always even for all $x \in X$ and for any $u$ with a Hamming weight less than $k$. Also, the parity of $\pi_u(x)$ becomes indeterminate for any $u$ with a Hamming weight greater than or equal to $k$. One approach is to examine the presence of monomials in the algebraic normal forms (ANFs) of $x \mapsto \pi_v(S(x))$ for $v \in \mathbb{F}_2^n$. This is represented by the set $P_S(u)$, defined as $\bigcup_{w \in \mathrm{Succ}(u)} Q_S(w)$, where $Q_S(w) = \{v \in \mathbb{F}_2^n : \pi_v(S(x)) \text{ contains } \pi_w(x)\}$, and $\mathrm{Succ}(u) = \{x \in \mathbb{F}_2^n : u \preceq x\}$, which forms an affine subspace of dimension $n - \mathrm{wt}(u)$, with $\mathrm{wt}(u)$ being the Hamming weight of $u$. The table representation of $P_S(u)$ for all $u$ provides valuable insight into the resistance of the function against division-property-based attacks.

### 2.5 Boomerang Connectivity Table (BCT)

The *Boomerang Connectivity Table* (BCT) is a cryptanalysis tool used to evaluate the probability of successfully connecting two differential characteristics in a boomerang-style attack, particularly when the middle part of the cipher consists of a single SBox layer. In a boomerang attack, the cipher is often considered a composition of sub-ciphers. The attack's success depends on how the differential characteristics of these parts interact at their boundary. The BCT provides a systematic way to measure this interaction.

It analyzes the formation of a boomerang quartet, a set of four values $(x_1, x_2, x_3, x_4)$ at the input of the SBox layer. These values must satisfy two conditions simultaneously:

1. The input difference from the first part of the cipher is $\Delta_i$, such that $x_1 \oplus x_2 = \Delta_i$, and we check if $x_3 \oplus x_4 = \Delta_i$.
2. The output difference required by the second part of the cipher is $\nabla_o$, such that $S(x_1) \oplus S(x_3) = \nabla_o$ and $S(x_2) \oplus S(x_4) = \nabla_o$.

The BCT calculates the probability of these conditions holding true for all possible pairs of $(\Delta_i, \nabla_o)$.

For an $n$-bit invertible SBox, the BCT is a $2^n \times 2^n$ table. Each entry in the table, for a given input difference $\Delta_i$ and output difference $\nabla_o$, stores a count. It represents the number of initial values $x$ that will result in a valid boomerang quartet.

Each entry of $\mathcal{T}_{BCT}(\Delta_i, \nabla_o)$, is calculated by the following formula, which counts the number of successful inputs: $\mathcal{T}_{BCT}(\Delta_i, \nabla_o) = \#\{x \in \{0,1\}^n | S^{-1}(S(x) \oplus \nabla_o) \oplus S^{-1}(S(x \oplus \Delta_i) \oplus \nabla_o) = \Delta_i\}$. To find an entry, one must iterate through every possible input $x$ (from 0 to $2^n - 1$) and check if the condition in the formula is met. The final value is the total count. The probability of forming a right quartet is this count divided by $2^n$.

### 2.6 Classification of Samples

Initially, to construct a model for an SBox, we adopt a binary classification perspective of its evaluation tables. In this viewpoint, each entry in the SBox evaluation table is treated as a data point that is be assigned to one of two classes (though, one may note that the DPT is already a binary table). Entries corresponding to positive values are categorized into positive classes, indicating feasible transitions contributing to potential differential or linear characteristics. Conversely, entries that take the value zero are placed into the negative class, since they represent impossible transitions that must be excluded from any valid characteristic. This classification provides a clean separation between feasible and infeasible behaviors of the SBox, enabling systematic design of linear constraints that capture its propagation properties within a MILP framework.

*Example 1.* The two encodings are illustrated using the DDT of the chosen 3-bit SBox $S$, `05276134`. The DDT (which is given in Table 1(a)) is first converted into a binary table as shown in Table 1(b). In this representation, '+' denotes positive samples and '−' denotes negative samples.

□

Table 1: `ABC` SBox (conversion from DDT to binary DDT)

(a) DDT

| $\Delta x$ \ $\Delta y$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 |
| 2 | 0 | 0 | 4 | 0 | 0 | 4 | 0 | 0 |
| 3 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 4 |
| 4 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |
| 5 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |
| 6 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |
| 7 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |

(b) Binary DDT

| $\Delta x$ \ $\Delta y$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | + | − | − | − | − | − | − | − |
| 1 | − | − | − | − | + | − | + | − |
| 2 | − | − | + | − | − | + | − | − |
| 3 | − | − | + | − | − | − | − | + |
| 4 | − | + | − | + | + | − | + | − |
| 5 | − | + | − | + | + | − | + | − |
| 6 | − | + | − | + | + | − | + | − |
| 7 | − | + | − | + | + | − | + | − |

### 2.7 Hierarchy of Structures

The modeling of an SBox is classified into a hierarchy of structures based on the depth of integration of its evaluation tables. This approach allows systematic construction of MILP models that capture the cryptographic properties of the SBox at varying levels of detail and complexity. Within the context, only the *primary* and *secondary* structures are discussed. However, further hierarchical levels are needed to formulate for the entire MILP model that encodes for multiple rounds of a cipher where each round consists of multiple SBoxes in parallel (*tertiary*) and a makes use of MILP solver friendly optimization in the final hierarchy (*quaternary*), both of which are kept out of scope. Refer to Appendix A for examples of primary and secondary structures.

**Primary Structure** In the primary structure, the focus is on directly modeling the propagation properties of an SBox using its classical cryptographic evaluation tables; such as the DDT, (absolute) LAT, DPT and BCT (and possibly more). These tables capture the cryptographic behavior of the SBox under various cryptanalytic methods (differential, linear, algebraic and boomerang analyses, respectively).

Here, every entry of the DDT, LAT, DPT or BCT is classified into one of two categories, namely positive (i.e., non-zero transition) class and negative (i.e., zero transition) class. Traditionally, the feasible transitions (positive samples) solely are used in the primary structure considers to construct the corresponding MILP model. By treating the modeling task as a binary classification problem, constraints are generated to separately capture the conditions for positive transitions and negative transitions. The primary structure, therefore, enriches the MILP framework with discriminative power, as it systematically distinguishes between valid and impossible propagations (i.e., negative samples).

**Secondary Structure** Moving to the secondary structure, the objective shifts from the binary classification viewpoint to integration for an SBox. Auxiliary variables are introduced to represent the state of each SBox, indicating whether it is active (participating in propagation) or inactive (not participating). Thus, the secondary structure encapsulates the primary structure's classification into a slightly more generalized framework that tracks the activity status of each SBox.

## 3 Related Works

In this section, we assume that the cryptographic table from the SBox is provided (the table elucidates a binary classification problem). The objective is to formulate that table into a collection of linear constraints so that it can be used in a MILP formulation.

### 3.1 Representation of an SBox using Convex Hull (CH)

The *Convex Hull* (CH) of a set $P$ of points is the smallest convex set that contains all the points in $P$. The process involves calculating the H-representation, which is achieved by determining the cutting planes

that collectively form the convex hull. Specifically, we apply the implementation from Sage[5] to extract the H-representation. Consequently, the H-representation yields a system of $w$ linear constraints, taking the form, $A \cdot [x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}]^\top \leq b^\top$. Here, $A$ is the $w \times 2n$ matrix and $b = (b_0, b_1, b_2, \ldots, b_{2n-1})$. Due to the innate implementation in Sage, both matrix $A$ and vector $b$ consist solely of integers. Each linear constraint serves to invalidate certain points, associated with unattainable differential propagation.

### 3.2 Modeling using Conditional Differential Behavior

The logical relationship $(x_0, \ldots, x_{m-1}) = (\delta_0, \ldots, \delta_{m-1}) \in \{0,1\}^m \subseteq \mathbb{Z}^m$, which implies $y = \delta \in \{0,1\} \subseteq \mathbb{Z}$, can be effectively captured through the following linear constraint: $\sum_{i=0}^{m-1}(-1)^{\delta_i} x_i + (-1)^{\delta+1} y - \delta + \sum_{i=0}^{m-1} \delta_i \geq 0$

Here, we consider $(x_0, x_1, x_2, x_3)$ and $(y_0, y1, y_2, y_3)$ as the variables for input and output differences of a 4-bit SBox. Suppose there is an impossible propagation in the DDT of the SBox: $(1010) \rightarrow (0111)$, indicating that the input difference (1010) does not propagate to (0111). Utilizing relationship noted earlier, the linear constraint that asserts the impossible point (i.e., negative sample) $(1001, 0111)$ is expressed as: $-x_0 + x_1 - x_2 + x_3 + y_0 - y_1 - y_2 - y_3 + 4 \geq 0$.

For an SBox, if there are $n$ impossible paths evident in a DDT, at most $n$ linear constraints are necessary to accurately model the DDT. However, it is possible to reduce the value of $n$ by merging existing constraints, thereby generating new ones.

As an illustration, consider two impossible propagations (i.e., negative samples): $(1010) \rightarrow (0111)$ and $(1010) \rightarrow (0110)$. A combined linear constraint: $-x_0 + x_1 - x_2 + x_3 + y_0 - y_1 - y_2 \geq -3$.

effectively eliminates both of these impossible points simultaneously. This consolidation of constraints helps streamline the representation of impossible propagations and enhances the efficiency of the system.

### 3.3 Selecting Best Constraints from Convex Hull

When applying an MILP model, obtaining a feasible solution does not guarantee a valid differential path. The objective is to reduce the number of active SBoxes over a broader range, with the optimal value being less than or equal to the actual minimum count of active SBoxes. To achieve this goal, linear constraints are sought that can effectively prune parts of the MILP model while preserving the valid differential characteristics within the region. Various algorithms are proposed for reducing the number of constraints in an SBox representation.

**Algorithm 1:** Greedy Approach [34]

Input:
$I_{Hull}$: Constraints from convex hull of an SBox.
$P_{IT}$: Impossible transitions (i.e., negative samples) from an evaluation table of an SBox
Output:
$I_R$: Reduced set of constraints

1: $I_R \leftarrow \emptyset$
2: **while** $P_{IT} \neq \emptyset$ **do**
3: $\quad I_Q \leftarrow$ Possible constraints in $I_{Hull}$ which maximizes number of removed impossible transitions from $P_{IT}$.
4: $\quad P_{IT} \leftarrow P_{IT} - \{$Removed impossible transitions using $I_Q\}$
5: $\quad I_{Hull} \leftarrow I_{Hull} - \{I_Q\}$
6: $\quad I_R \leftarrow I_R \cup \{I_Q\}$
7: **return** $I_R$

*Greedy algorithm based approach.* The discrete points within the H-Representation (convex hull) often generate a substantial number of constraints. An effective strategy involves identifying the best valid constraints that maximize the elimination of impossible differential patterns within the feasible region of the convex hull. Algorithm 1, as proposed by Sun et al., presents this greedy approach.

[5] https://www.sagemath.org/

**Algorithm 2:** Modeling by selecting random set of constraints [18]

Inputs:
$P_{VT}$: Input set corresponding to possible transitions in an SBox.
$k$: Number of constraints to be counted together.
Output:
$I_R$: Reduced set of constraints

$I_{Hull} \leftarrow ConvHull(P_{VT})$
$I_R \leftarrow H$
**for all** $p \in P_{IT}$ **do**
    Find all the $k$ subsets of constraints from $I_{Hull}$
    Choose a subset such that $p$ belongs to the hyperplanes of $I_{Q_1}, I_{Q_2}, \cdots, I_{Q_k}$
    $I_{Q_{new}} = I_{Q_1} + \ldots + I_{Q_k}$
    **if** $I_{Q_{new}}$ invalidates new impossible points other than that removes by the original constraints
    $I_R \leftarrow I_R \cup \{I_{Q_{new}}\}$
    **end if**
**return** $I_R$

*Modeling via selection of random subset of constraints.* The generation of an extensive set of new constraints by randomly adding $k$ constraints often results in redundancy, as they are likely to cover the entire space $\{0,1\}^m$. However, when $k$ hyperplanes from the H-representation share a vertex within the $\{0,1\}^m$ cube, combining the corresponding constraints can potentially yield a new constraint $I_{Q_{new}}$. Importantly, this new hyperplane represented by $I_{Q_{new}}$ should intersect with the cube at least once. In such cases, $I_{Q_{new}}$ has the potential to invalidate a distinct set of impossible points within the H-representation compared to the older constraints. The complete procedure is summarized in Algorithm 2.

**Algorithm 3:** MILP based reduction [30]

Inputs:
$P_{IP}$: Impossible patterns (i.e., negative samples) from evaluation table of an SBox.
$P_{VT}$: Possible transitions (i.e., positive samples) from evaluation table of an SBox.
Output:
$I_R$: Reduced set of constraints

$I_{Hull} \leftarrow ConvHull(P_{VT})$
$I_R \leftarrow I_{Hull}$
Create a table $T$ of size $|I_R| \times |P_{IP}|$ where $T_{i,j} = 1$ if constraint $I_{R_i}$ removes pattern $P_{IP_j}$, else set $T_{i,j} = 0$
Set $m = |I_R|$
Generate $m$ binary variables $b_1, b_2, \ldots, b_m$ such that $b_i = 1$ means constraint $i$ is included in the solution space else $b_i = 0$.
*constraintSet* $\leftarrow \emptyset$
**for** point $\in P_{IP}$ **do**
    *constraintSet* $\leftarrow$ *constraintSet* $\cup$ {Construct a constraint $b_k, \ldots, b_l >= 1$ such that $p$ is removed by at least one constraint applying table $T$} ▷ Generate constraints
Create an objective function $\sum_{i=1}^{m} b_i$ with constraints *constraintSet* and solve to get best constraints ($I_R$) that generate a stricter feasible region after maximizing the removed impossible differential patterns.
**return** $I_R$

*MILP-based reduction algorithm.* Sasaki and Todo [30] introduce a method that employs a standard MILP solver to derive a minimized set of constraints. Initially, they identify all the impossible differential points within the DDT of an SBox. These points serve as the foundation for generating impossible patterns. For each impossible pattern generated, the algorithm assesses which subset of constraints can effectively render the pattern invalid. This step pinpoints the crucial constraints required for pattern invalidation. Subsequently, they formulate a constraint specifying that each impossible pattern must be made infeasible within the valid region by at least one constraint. An MILP problem is then constructed to minimize the

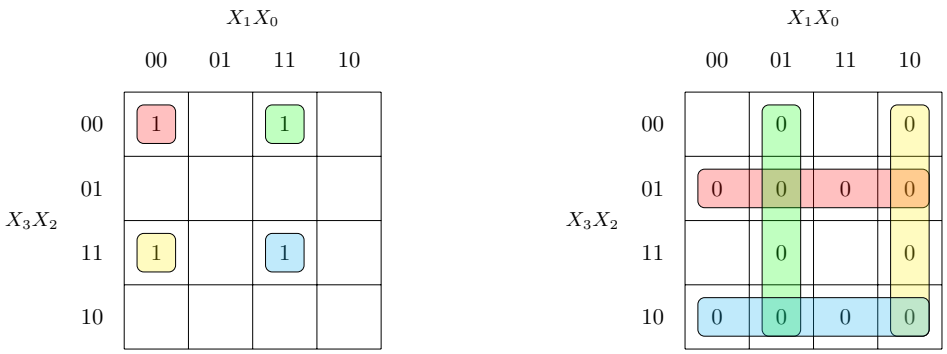
overall set of constraints while adhering to the constraints established in the prior steps. The formulated MILP problem is solved, yielding a minimized set of constraints that effectively reduces the number of impossible patterns. The method is summarized in Algorithm 3.

## 4 Our First Idea: Switching Positive and Negative Samples

In this section, the first proposed approach is described, wherein the roles of the positive and negative samples from the given SBox evaluation table are swapped. The fundamental concept involves treating the positive samples as negative and vice versa. Thus, the complementary problem is modeled, followed by the construction of the overarching MILP problem, and from its solution, the approach reverts to the original problem (by taking the complement again). It is demonstrated that, for certain SBox evaluation tables, this method is beneficial in reducing the number of linear constraints.

To demonstrate the basic concept of this, we present a suitable example. Let $f_1(X_3, X_2, X_1, X_0)$ be a set of 4 elements in $\mathbb{F}_2^4$, defined as: $f_1 = \{0000, 0011, 1100, 1111\}$ that represents the non-zero elements.

The goal is to find a minimal representation of the set $f_1$. Figure 1(a) shows the set $f_1$ represented in a Karnaugh map. The minimized Boolean expression after applying the Karnaugh map method (over 1's), $f_1^1(X_3, X_2, X_1, X_0) = X_3'X_2'X_1'X_0' + X_3'X_2'X_1X_0 + X_3X_2X_1'X_0' + X_3X_2X_1X_0$ (where, each $X_i^{'}$ denotes the complement of binary value $X_i$). Alternatively, the 0's may be applied to find the minimized expression (see Figure 1(b)). Thus, the minimized Boolean expression applying the Karnaugh map method (over 0's) is $f_1^0(X_3, X_2, X_1, X_0) = (X_3 + X_2')(X_3' + X_2)(X_1 + X_0')(X_1' + X_0)$. It can be observed that when using the 0's (i.e., negative samples) map, the resulting product of sum terms is simpler and contains fewer literals than the 1's (i.e., positive samples) map of product of sum.

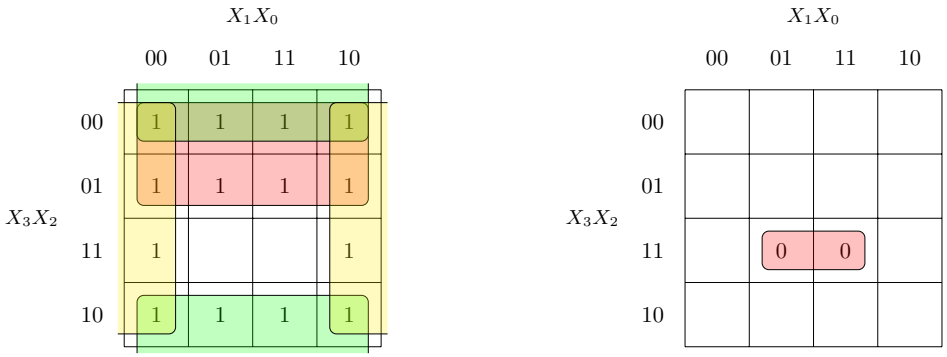



(a) Non-zero elements (b) Zero elements

Fig. 1: K-Map for $f_1$

Suppose $f_2(X_3, X_2, X_1, X_0)$ be a set of 14 elements in $\mathbb{F}_2^4$ in another example, where $f_2 = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1110\}$. Figure 2(a) presents the representation of the set $f_2$ using a Karnaugh map. By applying the Karnaugh map minimization procedure over the 1's, the simplified Boolean expression is, $f_2^1(X_3, X_2, X_1, X_0) = X_0' + X_2' + X_3'$. Alternatively, minimization can also be performed using the 0's (see Figure 2(b)), yielding, $f_2^0(X_3, X_2, X_1, X_0) = X_0' + X_2' + X_3'$. It is worth noting that minimization with respect to the 0's leads directly to a product-of-sums form, which contains the same number of literals as the product-of-sums expression derived from the 1's map. However, obtaining the minimal expression is generally more straightforward when employing the 0's map.

This approach can be extended to the modeling of SBoxes in cryptanalysis-based MILP formulations by altering the constraint-generation process derived from the LAT, DDT, DPT and BCT. Typically, in




(a) Non-zero elements

(b) Zero elements

Fig. 2: K-Map for $f_2$

classical SBox modeling, the binary variables representing input - output patterns are assigned a value of 1 to indicate the presence of an active bit or a valid approximation/differential, and the resulting constraints are constructed to enforce the propagation constraints. However, in certain scenarios, such as when searching for impossible differentials, zero-correlation linear approximations, or other structural properties, it is beneficial to invert this encoding, assign 0 to indicate an active or valid position, and 1 to indicate inactivity or invalid propagation. Through this approach, the constraint system is reformulated so that the constraints capture the absence of specific transitions or correlations rather than their presence. This "complementary variable encoding" can simplify the MILP model when the cryptanalytic objective is to identify patterns of non-propagation or to detect impossible transitions.

This inversion of bit semantics allows the evaluation table to derive constraints to be directly reused by simply flipping the binary mapping while preserving their structural relationship to the underlying combinatorial properties of the SBox. This approach is particularly advantageous when integrating SBox constraints into multi-round block cipher models, where representing zero-propagation conditions efficiently can reduce solver complexity and improve runtime performance.

**Switching Back to Positive Samples: Modeling for Exact Problem**

One may notice that the model we discuss here actually finds the formulation of the complementary problem, that is, it looks for the inactive SBoxes (as it treats the negative samples of an evaluation table as positive samples, and vice-versa). However, since our objective is to find the minimum number of active SBoxes, we need to adjust the MILP problem formulation. One idea is as follows.

After we model the negative samples as the positive samples (and vice-versa), we make the MILP problem as "maximize number of inactive SBoxes". Then we get the maximum possible number of SBoxes that can be inactive (say, the result from the MILP solver is $I$). Then, we subtract that from the total number of SBoxes. Say, for a cipher with 32 SBoxes at one round and tested for 10 rounds, we have in total $32 \times 10 = 320$ SBoxes. Therefore, the minimal number of active SBoxes for our case will be $320 - I$.

Aside from that, another idea is to change the indicator variable that captures if the SBox is active or not, to reflect the complementary nature. For example, if we have an indicator variable $S$ that is 1 when the SBox is active and 0 otherwise, we can replace it as $1 - S$. The objective function will remain the same, i.e., "minimize number of active SBoxes". However, since we have changed the indicator variable, the MILP solver will now find the minimum number of active SBoxes directly.

Table 2: Minimum number of linear constraints for DDT

(a) Including $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4, 28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 157, 99 | 73, 237 | 34, 24 | 31, 22 | 30, 21 | 21, 17 | 21, 17 | 21, 17 |
| KLEIN | | 150, 106 | 158, 311 | 33, 22 | 30, 22 | 28, 21 | 23, 17 | 23, 17 | 23, 17 |
| LILLIPUT | | 150, 106 | 183, 324 | 31, 27 | 30, 26 | 29, 23 | 26, 19 | 26, 20 | 25, 19 |
| MIBS | | 150, 106 | 173, 378 | 32, 27 | 30, 24 | 29, 23 | 26, 20 | 26, 20 | 26, 20 |
| MIDORI S0 | | 159, 97 | 142, 239 | 32, 23 | 30, 25 | 27, 21 | 26, 16 | 26, 17 | 26, 16 |
| MIDORI S1 | | 150, 106 | 166, 367 | 29, 26 | 28, 24 | 28, 22 | 26, 20 | 26, 20 | 26, 20 |
| MINALPHER | | 150, 106 | 140, 338 | 26, 24 | 26, 24 | 26, 22 | 21, 19 | 21, 19 | 21, 18 |
| PICCOLO | | 159, 97 | 72, 202 | 27, 23 | 26, 24 | 21, 21 | 21, 16 | 21, 16 | 21, 16 |
| PRESENT | | 159, 97 | 115, 237 | 35, 22 | 32, 22 | 30, 21 | 21, 17 | 21, 17 | 20, 17 |
| PRIDE | | 159, 97 | 74, 194 | 30, 22 | 28, 22 | 26, 21 | 21, 16 | 21, 17 | 21, 17 |
| PRINCE | | 150, 106 | 151, 300 | 31, 26 | 31, 26 | 29, 22 | 25, 19 | 25, 19 | 25, 18 |
| RECTANGLE | | 159, 97 | 70, 267 | 25, 25 | 24, 23 | 24, 21 | 19, 17 | 20, 17 | 20, 16 |
| SKINNY | | 159, 97 | 72, 202 | 28, 24 | 27, 24 | 26, 21 | 21, 16 | 21, 16 | 21, 16 |
| TWINE | | 150, 106 | 183, 324 | 33, 23 | 30, 25 | 29, 23 | 26, 19 | 26, 20 | 25, 19 |
| ASCON | 5 | 707, 317 | 216, 2415 | 66, 51 | 64, 48 | 63, 40 | 48, 32 | 48, 31 | 47, 31 |
| FIDES-5 | | 527, 497 | 768, 910 | 101, 96 | 96, 90 | 84, 79 | 64, 61 | 64, 64 | 63, 62 |
| KECCAK | | 707, 314 | 182, 316 | 56, 50 | 52, 47 | 52, 46 | 37, 34 | 37, 36 | 32, 36 |
| ICEPOLE | | 687, - | 347, - | 80, - | 77, - | 72, - | 57, - | 59, - | 57, - |
| SC2000-5 | | 527, 497 | 800, 908 | 98, 99 | 95, 92 | 84, 82 | 66, 64 | 66, 65 | 65, 63 |
| FIDES-6 | 6 | -, 2017 | -, 223 | -, - | -, - | -, - | -, 180 | -, 184 | -, - |
| SC2000-6 | | -, 1954 | -, 241 | -, - | -, - | -, - | -, 214 | -, 89 | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

(b) Excluding $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4, 28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 98, 127 | 119, 141 | 27, 26 | 27, 24 | 25, 22 | 18, 17 | 18, 17 | 18, 17 |
| KLEIN | | 120, 105 | 208, 281 | 29, 23 | 27, 22 | 25, 20 | 21, 18 | 21, 18 | 21, 18 |
| LILLIPUT | | 120, 105 | 225, 297 | 28, 23 | 27, 22 | 24, 21 | 20, 19 | 20, 19 | 20, 19 |
| MIBS | | 120, 105 | 213, 320 | 28, 26 | 27, 25 | 25, 23 | 22, 20 | 22, 20 | 21, 20 |
| MIDORI S0 | | 129, 96 | 164, 217 | 33, 23 | 30, 22 | 27, 20 | 24 ,17 | 24, 17 | 24, 17 |
| MIDORI S1 | | 120, 105 | 209, 334 | 26, 25 | 25, 25 | 23, 22 | 21, 20 | 21, 20 | 21, 20 |
| MINALPHER | | 120, 105 | 214, 320 | 27, 24 | 26, 24 | 25, 22 | 19, 19 | 19, 18 | 19, 18 |
| PICCOLO | | 129, 96 | 106, 186 | 25, 23 | 25, 22 | 24, 20 | 19, 15 | 19, 16 | 19, 15 |
| PRESENT | | 129, 96 | 170, 233 | 32, 21 | 30, 20 | 26, 18 | 18, 17 | 18, 15 | 18, 15 |
| PRIDE | | 129, 96 | 107, 170 | 28, 21 | 27, 21 | 25, 21 | 20, 17 | 20, 19 | 20, 19 |
| PRINCE | | 120, 105 | 202, 270 | 27, 26 | 27, 25 | 25, 22 | 22, 18 | 22, 19 | 22, 18 |
| RECTANGLE | | 129, 96 | 104, 158 | 24, 24 | 22, 24 | 20, 22 | 16, 16 | 16, 17 | 16, 17 |
| SKINNY | | 129, 96 | 106, 186 | 28, 24 | 27, 23 | 24, 20 | 19, 15 | 19, 16 | 19, 15 |
| TWINE | | 120, 105 | 225, 297 | 28, 25 | 27, 24 | 24, 21 | 20, 19 | 20, 19 | 20, 19 |
| ASCON | 5 | 645, 316 | 273, 1739 | 65, 48 | 63, 45 | 59, 40 | 46, - | 46, 31 | 45, - |
| FIDES-5 | | 465, 496 | 1034, 729 | 93, 95 | 84, 87 | 78, 80 | 61, - | 61, - | 59, - |
| KECCAK | | 645, 316 | 229, 308 | 52, 53 | 50, 49 | 47, 45 | 33, - | 32, - | 31, - |
| SC2000-5 | | 465, 496 | 1091, 747 | 92, 95 | 87, 90 | 76, 82 | 61, - | 61, - | 59, - |
| FIDES-6 | 6 | 1953, 2016 | 8200, 7012 | 257, 275 | 248, 263 | -, - | -, - | -, - | -, - |
| SC2000-6 | | 2016, 1953 | 9816, 11238 | 292, 294 | 286, 284 | -, - | -, - | -, - | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

Table 3: Minimum number of linear constraints for (absolute) LAT

(a) Including $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4,28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 123, 133 | 50, 211 | 19, 18 | 19, 18 | 19, 16 | 15, 13 | 15, 14 | 15, 13 |
| KLEIN | | 105, 151 | 285, 133 | 27, 31 | 27, 29 | 24, 27 | 18, 23 | 18, 23 | 18, 23 |
| LILLIPUT | | 105, 151 | 239, 126 | 26, 32 | 26, 32 | 24, 32 | 19, 22 | 19, 22 | 18, 22 |
| MIBS | | 105, 151 | 347, 124 | 28, 36 | 26, 35 | 23, 35 | 19, 26 | 19, 26 | 19, 26 |
| MIDORI S0 | | 123, 133 | 195, 92 | 31, 28 | 29, 28 | 25, 25 | 19, 16 | 19, 18 | 19, 17 |
| MIDORI S1 | | 105, 151 | 338, 104 | 25, 32 | 25, 31 | 23, 29 | 18, 22 | 18, 22 | 18, 22 |
| MINALPHER | | 105, 151 | 304, 143 | 25, 32 | 25, 31 | 21, 28 | 18, 23 | 18, 23 | 18, 23 |
| PICCOLO | | 123, 133 | 64, 76 | 22, 17 | 21, 16 | 20, 16 | 14, 11 | 14, 11 | 14, 11 |
| PRESENT | | 123, 133 | 169, 137 | 27, 29 | 25, 26 | 23, 24 | 17, 13 | 17, 17 | 17, 17 |
| PRIDE | | 123, 133 | 67, 72 | 21, 16 | 21, 16 | 21, 15 | 15, 11 | 15, 12 | 15, 11 |
| PRINCE | | 105, 151 | 298, 114 | 25, 32 | 25, 31 | 23, 29 | 18, 20 | 18, 21 | 18, 20 |
| RECTANGLE | | 123, 133 | 55, 186 | 22, 17 | 22, 16 | 21, 15 | 18, 12 | 19, 12 | 18, 12 |
| SKINNY | | 123, 133 | 64, 76 | 21, 16 | 21, 16 | 20, 16 | 14, 11 | 14, 11 | 14, 11 |
| TWINE | | 105, 151 | 239, 126 | 27, 32 | 26, 32 | 24, 32 | 19, 22 | 19, 22 | 18, 22 |
| ASCON | 5 | 647, 377 | 304, 1613 | 83, 70 | 79, 69 | 75, 58 | -, 44 | 52, 44 | 50, - |
| FIDES-5 | | 527, 497 | 765, 889 | 94, 96 | 94, 90 | 83, 77 | -, 62 | 66, 63 | 64, 62 |
| ICEPOLE | | 497, - | 857, - | 102, - | 102, - | 90, - | 80, 64 | 84, - | 81, - |
| KECCAK | | 647, 377 | 137, 236 | 69, 67 | 67, 66 | 67, 65 | 37, 38 | 42, 40 | 37, 39 |
| SC2000-5 | | 527, 497 | 801, 917 | 100, 93 | 99, 88 | 84, 78 | 68, 63 | 68, 64 | 67, 63 |
| FIDES-6 | 6 | -, 3025 | -, 2699 | -, 185 | -, 177 | -, 159 | -, - | -, 119 | -, - |
| SC2000-6 | | -, 3214 | -, 1053 | -, 320 | -, 315 | -, 307 | -, - | -, 274 | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

(b) Excluding $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4,28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 93, 132 | 74, 116 | 18, 19 | 18, 18 | 17, 15 | 13, 12 | 13, 13 | 13, 12 |
| KLEIN | | 75, 150 | 413, 92 | 23, 28 | 22, 27 | 19, 27 | 15, 23 | -, 23 | -, 23 |
| LILLIPUT | | 75, 150 | 376, 87 | 19, 31 | 19, 31 | 18, 31 | 15, 22 | 15, 21 | 15, 21 |
| MIBS | | 75, 150 | 497, 86 | 23, 35 | 22, 35 | 19, 35 | 16, 26 | 16, 25 | 16, 25 |
| MIDORI S0 | | 93, 132 | 248, 82 | 25, 25 | 23, 25 | 20, 25 | 16, 16 | 17, 16 | 16, 17 |
| MIDORI S1 | | 75, 150 | 449, 76 | 24, 30 | 22, 30 | 19, 29 | 15, 22 | 15, 22 | 15, 22 |
| MINALPHER | | 75, 150 | 438, 100 | 23, 29 | 21, 28 | 19, 27 | 16, 23 | 16, 22 | 16, 22 |
| PICCOLO | | 93, 132 | 103, 62 | 19, 15 | 18, 15 | 17, 15 | 12, 11 | 12, 11 | 11, 10 |
| PRESENT | | 93,132 | 276, 93 | 20, 28 | 19, 24 | 18, 16 | 15, 16 | 15, 17 | 15, 17 |
| PRIDE | | 93, 132 | 110, 59 | 19, 13 | 19, 13 | 18, 13 | 12, 10 | 12, 9 | 11, 9 |
| PRINCE | | 75, 150 | 379, 84 | 22, 30 | 21, 30 | 18, 29 | 15, 19 | 15, 19 | 15, 10 |
| RECTANGLE | | 93, 132 | 84, 90 | 21, 19 | 21, 18 | 20, 16 | 13, 12 | 14, 12 | 13, 11 |
| SKINNY | | 93, 132 | 103, 62 | 19, 15 | 18, 15 | 17, 15 | 12, 11 | 12, 11 | 11, 10 |
| TWINE | | 75, 150 | 376, 86 | 20, 32 | 19, 32 | 18, 31 | 15, 22 | 15, 21 | 15, 21 |
| ASCON | 5 | 585, 376 | 390, 1107 | 76, 68 | 73, 65 | 69, 60 | -, - | 50, 45 | 47, - |
| FIDES-5 | | 465, 495 | 1034, 716 | 90, 91 | 87, 84 | 76, 77 | -, - | 62, - | 61, - |
| KECCAK | | 585, 376 | 164, 228 | 66, 66 | 64, 65 | 62, 64 | -, - | 41, - | 36, - |
| SC2000-5 | | 465, 496 | 1085, 760 | 92, 89 | 87, 87 | 77, 79 | -, - | 63, - | 62, - |
| FIDES-6 | 6 | 945, 3024 | 17147, 2392 | 115, 182 | 110, 172 | -, 159 | -, - | -, - | -, - |
| SC2000-6 | | -, - | -, 942 | -, 307 | -, - | -, - | -, - | -, - | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

## 5 Our Second Idea: Exclusion of Trivial Transitions in SBox Modeling

In the construction of DDT, LAT, DPT and BCT; the trivial zero transitions are included. These entries correspond to cases where at least one of the input and output characteristics os zero. Since the goal of SBox modeling is to capture meaningful propagation patterns that reflect the cryptographic strength or weakness of the design, such trivial transitions can be considered redundant.

By deliberately excluding these trivial transitions, the modeling process becomes more efficient. Specifically, the MILP representation, which relies on generating sets of constraints for feasible and infeasible transitions, can be significantly simplified. The exclusion of trivial cases results in a direct reduction of the number of constraints required, since no modeling effort is spent on transitions that do not affect the differential, linear, division, or boomerang propagation properties of the cipher. This reduction not only decreases computational overhead but also improves solver performance when searching for optimal or impossible trails.

Moreover, this selective elimination aligns with the broader principle of focusing on cryptographically significant events in the analysis. By ignoring non-contributing transitions, the SBox model preserves all essential information needed for security evaluation while discarding unnecessary complexity. The gain here is twofold: a reduction in the mathematical burden of constraint generation and an improvement in the clarity of the cryptographic structure being analyzed. Thus, ignoring trivial transitions is not merely a simplification, but a targeted optimization that enhances both the efficiency and precision of SBox modeling.

## 6 Results

We have the approach to all the earlier approaches for SBox modeling. To make the explanation clear, we consider both types of constraints generated from zero and non-zero transitions in LAT/DDT/DPT/BCT. Our results section has mainly two parts, each having four tables. The first part (Table 2(a), 3(a), 4(a), 5(a)) provides the number of constraints for different SBox evaluation tables. The second part (Table 2(b), 3(b), 4(b), 5(b)) explains the number of constraints for different SBox evaluation tables excluding the first row and first column, that is, ignoring the trivial transitions. Each resultant table describes the number of constraints required for modeling different 4, 5 and 6-bit SBoxes applying different SBox modeling approaches. Each cell of any table has one pair of values, namely the number of constraints required for applying negative sampling and positive sampling (in this order). For those we do not get a result, we mark by the '-' symbol.

Our analysis yields superior results for a wide range of ciphers by utilizing the zero transition property detailed in the DDT Table 2(a). Specifically, applying this to the Sage approach led to improvements for almost all SBoxes tested. Additionally, we recorded advancements for `KECCAK` using the subset addition method [29] and for `SC2000-5` in relation to the Sun et al.'s approach [34].

*DDT excluding trivial transitions.* Table 2(b) compares the constraints generated from zero and non-zero transitions (i.e., positive and negative samples) in DDT for different modeling, ignoring the trivial transitions. This approach leads to significant improvements for almost every existing approach for a wide range of ciphers, including `RECTANGLE`, `FIDES-5`, `KECCAK`, `SC2000-5`, `FIDES-6` and `SC2000-6` when applied to the DDT as given in Table 2(b).

*LAT.* Table 3(a) presents a comparison of different SBox modeling algorithms for LAT under both positive and negative sampling. In this case, we observe improved results for SBoxes used in `KLEIN`, `LILLIPUT`, `MIBS`, `MIDORI S1`, `MINALPHER`, `PRINCE` and `TWINE` across most of the modeling approaches.

*LAT excluding trivial transitions.* Table 3(b) compares the constraints generated from zero and non-zero transitions in LAT for different modeling, ignoring the trivial transitions. Here we obtain improved results for SBoxes like `KLEIN`, `LILLIPUT`, `MIBS`, `MIDORI S1`, `MINALPHER`, `PRESENT`, `PRINCE`, `TWINE` and `FIDES-6` for most of the modeling algorithms.

Table 4: Minimum number of linear constraints for DPT

(a) Including $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4, 28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 208, 48 | 50, 213 | 31, 14 | 31, 14 | 31, 13 | 23, 9 | 23, 9 | 23, 9 |
| KLEIN | | 206, 50 | 50, 117 | 21, 11 | 20, 10 | 20, 9 | 15, 7 | 15, 7 | 13, 6 |
| LILLIPUT | | 210, 46 | 49, 122 | 20, 11 | 20, 11 | 20, 9 | 14, 7 | 14, 7 | 12, 7 |
| MIBS | | 204, 52 | 51, 84 | 20, 9 | 19, 9 | 19, 8 | 14, 7 | 14, 7 | 12, 7 |
| MIDORI S0 | | 209, 47 | 50, 45 | 19, 6 | 19, 6 | 19, 6 | 14, 5 | 14, 5 | 12, 5 |
| MIDORI S1 | | 208, 48 | 47, 75 | 20, 10 | 19, 9 | 19, 9 | 13, 7 | 13, 7 | 12, 7 |
| MINALPHER | | 206, 50 | 50, 134 | 21, 11 | 20, 10 | 20, 9 | 15, 7 | 15, 7 | 14, 7 |
| PICCOLO | | 211, 45 | 47, 98 | 27, 13 | 26, 13 | 26, 12 | 20, 8 | 20, 8 | 20, 7 |
| PRESENT | | 210, 46 | 48, 103 | 22, 10 | 20, 10 | 20, 9 | 15, 7 | 15, 7 | 13, 7 |
| PRIDE | | 213, 43 | 47, 108 | 25, 11 | 24, 11 | 24, 10 | 18, 7 | 18, 7 | 17, 6 |
| PRINCE | | 207, 49 | 48, 86 | 20, 12 | 20, 11 | 20, 10 | 14, 8 | 14, 9 | 12, 8 |
| RECTANGLE | | 208, 48 | 52, 197 | 31, 15 | 31, 14 | 31, 13 | 24, 9 | 24, 9 | 23, 9 |
| SKINNY | | 211, 45 | 47, 98 | 27, 13 | 26, 13 | 26, 12 | 20, 8 | 20, 8 | 20, 7 |
| TWINE | | 210, 46 | 49, 122 | 21, 11 | 20, 10 | 20, 9 | 14, 7 | 14, 7 | 12, 7 |
| ASCON | 5 | 835, 189 | 158, 1393 | 99, 22 | 99, 22 | 99, 20 | 72, 13 | 72, 15 | 67, 15 |
| FIDES-5 | | 825, 199 | 173, 2012 | 87, 23 | 83, 23 | 81, 20 | 56, - | 56, - | 54, - |
| KECCAK | | 888, 136 | 111, 412 | 91, 21 | 91, 19 | 91, 17 | 61, - | 61, 15 | 61, 15 |
| ICEPOLE | | 873, 151 | 121, 18 | 31, 8 | 31, 8 | 26, 8 | 16, - | 16, 7 | 16, 6 |
| SC2000-5 | | 883, 141 | 174, 1719 | 68, 24 | 67, 24 | 65, 19 | 60, - | 60, - | 55, - |
| FIDES-6 | 6 | 3678,- | 403, - | 178, - | 176, - | 173, - | -, - | 152, - | 149, - |
| SC2000-6 | | 3734, - | 204, - | 51, - | 49, - | 45, - | -, - | 33, - | 28, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

(b) Excluding $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4, 28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 177, 48 | 49, 213 | 21, 14 | 21, 14 | 21, 12 | 16, 8 | 16, 8 | 17, 8 |
| KLEIN | | 175, 50 | 68, 117 | 8, 9 | 8, 8 | 7, 7 | 6, 6 | 6, 6 | 6, 5 |
| LILLIPUT | | 179, 46 | 54, 122 | 11, 10 | 10, 9 | 9, 8 | 7, 6 | 7, 6 | 7, 6 |
| MIBS | | 173, 52 | 62, 84 | 8, 7 | 8, 7 | 7, 7 | 6, 6 | 6, 6 | 5, 6 |
| MIDORI S0 | | 178, 47 | 49, 45 | 9, 5 | 9, 5 | 8, 5 | 7, 4 | 7, 4 | 7, 4 |
| MIDORI S1 | | 177, 48 | 51, 75 | 8, 9 | 8, - | 7, 8 | 6, 6 | 6, 6 | 6, 6 |
| MINALPHER | | 175, 50 | 56, 134 | 10, 9 | 10, 9 | 9, 8 | 7, 6 | 7, 6 | 6, 6 |
| PICCOLO | | 180, 45 | 69, 98 | 17, 12 | 17, 12 | 16, 11 | 14, 7 | 14, 7 | 13, 7 |
| PRESENT | | 179, 46 | 48, 103 | 9, 9 | 9, 9 | 9, 8 | 7, 6 | 7, 6 | 6, 6 |
| PRIDE | | 182, 43 | 77, 108 | 13, 10 | 13, 10 | 13, 9 | 10, 7 | 10, 7 | 10, 6 |
| PRINCE | | 176, 49 | 50, 86 | 9, 10 | 9, 10 | 8, 9 | 7, 7 | 7, 7 | 7, 7 |
| RECTANGLE | | 177, 48 | 52, 197 | 21, 14 | 21, 13 | 21, 11 | 16, 8 | 16, 9 | 16, 8 |
| SKINNY | | 180, 45 | 69, 98 | 17, 13 | 17, 12 | 16, 11 | 14, 7 | 14, 7 | 13, 7 |
| TWINE | | 179, 46 | 54, 122 | 11, 10 | 11, 9 | 9, 8 | 7, 6 | 7, 6 | 7, 6 |
| ASCON | 5 | 772, 189 | 178, 1393 | 86, 22 | 82, 20 | 81, 18 | -, - | 58, 15 | 53, - |
| FIDES-5 | | 762, 199 | 171, 2012 | 73, 21 | 69, 22 | 65, 18 | -, - | 47, - | 45, - |
| KECCAK | | 825, 136 | 128, 412 | 81, 19 | 81, 18 | 81, 15 | -, - | 56, - | 56, - |
| SC2000-5 | | 820, 141 | 217, 1719 | 47, 23 | 47, 23 | 46, 18 | -, - | 45, - | 45, - |
| FIDES-6 | 6 | 3551, 418 | 763, - | 148, - | 144, - | 144, - | -, - | 132, - | 132, - |
| SC2000-6 | | 3607, 362 | 174, - | 17, - | 17, - | 16, - | -, - | -, - | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

Table 5: Minimum number of linear constraints for BCT

(a) Including $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4, 28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 102, 154 | 173, 104 | 25, 26 | 24, 26 | 23, 25 | 17, 19 | 17, 17 | 16, 17 |
| KLEIN | | 122, 144 | 246, 164 | 30, 31 | 28, 27 | 24, 25 | 20, 23 | 20, 19 | 20, 19 |
| LILLIPUT | | 120, 136 | 225, 210 | 29, 29 | 28, 28 | 26, 26 | 21, 22 | 21, 19 | 21, 19 |
| MIBS | | 120, 136 | 213, 220 | 29, 30 | 29, 28 | 26, 26 | 22, 23 | 22, 20 | 21, 20 |
| MIDORI S0 | | 107, 149 | 127, 146 | 26, 25 | 24, 24 | 23, 22 | 18, 18 | 18, 16 | 17, 15 |
| MIDORI S1 | | 105, 151 | 242, 159 | 25, 33 | 24, 32 | 22, 20 | 18, 28 | 20, 24 | 18, 24 |
| MINALPHER | | 122, 144 | 215, 182 | 30, 30 | 29, 30 | 27, 20 | 21, 26 | 21, 22 | 21, 22 |
| PICCOLO | | 103, 153 | 139, 83 | 24, 23 | 23, 22 | 21, 22 | 16, 16 | 16, 14 | 15, 14 |
| PRESENT | | 107, 149 | 239, 147 | 26, 30 | 26, 28 | 23, 28 | 18, 24 | 18, 21 | 17, 20 |
| PRIDE | | 103, 153 | 115, 70 | 25, 26 | 23, 25 | 22, 24 | 16, 16 | 16, 16 | 16, 16 |
| PRINCE | | 105, 151 | 269, 159 | 27, 31 | 25, 31 | 23, 29 | 19, 23 | 19, 21 | 19, 20 |
| RECTANGLE | | 107, 149 | 162, 103 | 26, 25 | 25, 23 | 21, 22 | 15, 18 | 15, 16 | 15, 16 |
| SKINNY | | 103, 153 | 139, 83 | 23, 22 | 23, 22 | 21, 22 | 16, 16 | 16, 14 | -, 14 |
| TWINE | | 120, 136 | 225, 210 | 30, 29 | 28, 27 | 26, 26 | 21, 23 | 21, 19 | 21, 19 |
| ASCON | 5 | 445, 579 | 1154, 666 | 57, 64 | 54, 60 | 47, 54 | -, 42 | 38, 40 | -, 38 |
| FIDES-5 | | 465, 559 | 1034, 815 | 94, 97 | 93, 93 | 78, 83 | 61, 66 | -, 63 | -, 62 |
| KECCAK | | 445, 579 | 1230, 435 | 40, 58 | 41, 56 | 36, 55 | -, 41 | -, 45 | -, 43 |
| SC2000-5 | | 465, 559 | 1091, 831 | 94, 100 | 92, 99 | 77, 84 | -, 70 | -, 65 | -, 64 |
| FIDES-6 | 6 | 1953, 2143 | 8200, 6290 | 260, 286 | -, 278 | -, - | -, - | -, - | -, - |
| SC2000-6 | | 1980, 2206 | 12384, 7158 | 278, 314 | -, 305 | -, - | -, - | -, - | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

(b) Excluding $0^{\text{th}}$ row and $0^{\text{th}}$ column

| SBox | Size | Naïve [4] | Full Convex Hull [34] | Chosen Convex Hull (Deterministic) [34] | Chosen Convex Hull (Randomized) [4, 28] | Sasaki, Todo [30] | Boura, Coggia$^\star$ [18] | Subset Addition [29] $k=2$ | Subset Addition [29] $k=3$ |
|---|---|---|---|---|---|---|---|---|---|
| GIFT | 4 | 102, 123 | 173, 143 | 24, 21 | 21, 21 | 21, 21 | 17, 17 | 17, 17 | 16, 17 |
| KLEIN | | 112, 113 | 246, 226 | 28, 25 | 25, 24 | 23, 23 | 20, 19 | 20, 19 | 20, 19 |
| LILLIPUT | | 120, 105 | 225, 297 | 26, 25 | 23, 23 | 21, 21 | 20, 19 | 20, 19 | 20, 19 |
| MIBS | | 120, 105 | 213, 320 | 29, 26 | 26, 25 | 23, 23 | 22, 20 | 22, 20 | 21, 20 |
| MIDORI S0 | | 107, 118 | 127, 162 | 25, 23 | 20, 20 | 20, 20 | 17, 16 | 17, 16 | 17, 15 |
| MIDORI S1 | | 105, 120 | 242, 220 | 22, 29 | 29, 28 | 27, 27 | 17, 24 | 18, 24 | 17, 24 |
| MINALPHER | | 112, 113 | 215, 249 | 29, 26 | 27, 26 | 25, 25 | 20, 22 | 20, 22 | 20, 22 |
| PICCOLO | | 103, 122 | 139, 86 | 22, 24 | 23, 24 | 21, 21 | 15, 14 | 15, 14 | 15, 14 |
| PRESENT | | 107, 118 | 239, 180 | 25, 27 | 28, 27 | 24, 24 | 18, 21 | 18, 21 | 17, 20 |
| PRIDE | | 103, 122 | 115, 88 | 22, 25 | 22, 22 | 22, 22 | 16, 14 | 16, 16 | 15, 16 |
| PRINCE | | 105, 120 | 269, 221 | 25, 29 | 28, 27 | 25, 25 | 19, 21 | 19, 21 | 19, 20 |
| RECTANGLE | | 107, 118 | 162, 126 | 24, 22 | 21, 21 | 19, 19 | 15, 16 | 15, 16 | 15, 16 |
| SKINNY | | 103, 122 | 139, 86 | 22, 24 | 24, 24 | 21, 21 | 15, 14 | 15, 14 | 15, 14 |
| TWINE | | 120, 105 | 225, 297 | 28, 25 | 24, 23 | 21, 21 | 20, 19 | 20, 19 | 20, 19 |
| ASCON | 5 | 445, 516 | 1154, 712 | 54, 60 | 57, 56 | 50, 50 | -, - | 37, 40 | 35, 38 |
| FIDES-5 | | 465, 496 | 1034, 729 | 94, 94 | 87, 89 | 80, 80 | -, - | 61, 63 | 59, 62 |
| KECCAK | | 445, 516 | 1230, 552 | 40, 64 | 65, 64 | 55, 55 | -, - | 32, 45 | -, 43 |
| SC2000-5 | | 465, 496 | 1091, 747 | 93, 96 | 89, 91 | 82, 82 | -, - | -, 65 | -, 64 |
| FIDES-6 | 6 | 1953, 2016 | 8200, 7012 | 256, 269 | 269, - | -, - | -, - | -, - | -, - |
| SC2000-6 | | 1890, 2079 | 12384, 8568 | 275, - | 296, - | -, - | -, - | -, - | -, - |

$\star$: Our implementation of Boura and Coggia's method [18]
Each cell indicates (negative samples, positive samples)

*DPT.* Table 4(a) presents a comparative analysis of different S-box modeling techniques for DPT under both positive and negative sampling strategies. The results indicate that improvements are observed primarily in the Sage generated outcomes, while across all evaluated ciphers, the negative sampling approach consistently demonstrates superior performance.

*DPT excluding trivial transitions.* Table 4(b) compares the constraints derived from zero and non-zero transitions in the DPT across different modeling approaches, excluding the trivial transitions. In this setting, improvements occur only in the Sage-generated results, while for most ciphers the negative sampling strategy yields better performance. In particular, `PRINCE` shows strong results when evaluated with the greedy [34], randomized greedy [29] and MILP-aided [30] approaches.

*BCT.* Table 5(a) describes the comparisons of different SBox modeling algorithms for BCT against positive and negative sampling. Here we obtain improved results for SBoxes like `GIFT`, `KLEIN`, `MIBS`, `MIDORI S0`, `MIDORI S1`, `MINALPHER`, `PRESENT`, `PRINCE`, `PRIDE`, `ASCON`, `FIDES-5` and `KECCAK` for some of the modeling algorithms. Here, all the algorithms provide better results with respect to negative sampling.

*BCT excluding trivial transitions.* Table 5(b) compares the constraints generated from zero and non-zero transitions in BCT for different modeling, ignoring the trivial transitions. Here we obtain improved results for SBoxes like `GIFT`, `KLEIN`, `MIBS`, `MIDORI S0`, `MIDORI S1`, `MINALPHER`, `PRESENT`, `PRINCE`, `PRIDE`, `ASCON`, `FIDES-5` and `KECCAK` for some of the modeling algorithms. Here, greedy [34], Boura and Coggia's method [18], subset addition [29] algorithms have better performance with respect to negative sampling.

*Overall Assessment.* In order to see the impact of our work, one may compare, for instance, the results on the `GIFT` (absolute) LAT (from Table 3(a)) indicate that the best-known result prior to our work was 13 constraints (using [18] and [29, $k = 3$]). In contrast, our approach reduces this to 12 constraints, as can be seen from Table 3(b). Basically, we augment with 3 more options, so that the user can choose the best one out of 4.

## 7 Conclusion

In this paper, two efficient strategies for modeling SBoxes in MILP-based cryptanalysis are introduced. Instead of modeling possible transitions, a model based on the impossible transitions (i.e., negative samples) is created. The approach is further streamlined by excluding the redundant top row and left column (that correspond to trivial zero-to-zero transition).

The experimental results, presented across a wide range of 4, 5, and 6-bit SBoxes and various property tables including the DDT, LAT, DPT and BCT, confirm that these methods successfully produce a smaller set of linear constraints than traditional modeling approaches. By reducing the number of constraints, these models are expected to improve the efficiency of automated cryptanalysis, ultimately providing a more powerful tool for the security evaluation of modern ciphers.

## A Exemplary Constraints for Evaluation Tables

For better clarity, some examples of the generated linear constraints are presented here. All variables are binary, and $x$'s (respectively, $y$'s) denote the input (respectively, output) characteristics.

Notice that Examples 2, 3, 4, 5 and 6 correspond to the primary structure; and Examples 7 and 8 correspond to the secondary structure (see Section 2.7 for the relevant details). The switching of positive and negative samples is handled in the secondary structure setting (it cannot be captured in the primary structure), and similarly the exclusion of zero-to-zero transitions are automatically managed by the auxiliary constraints used in the secondary structure.

*Example 2 (Sun et al.'s [34] ABC SBox (DDT) with positive samples using convex hull (Sage)).*

```
+1x2 +0x1 -2x0 +1y2 +2y1 +1y0 >= 0
+1x2 +2x1 -2x0 +1y2 +0y1 +1y0 >= 0
+1x2 +0x1 +2x0 -1y2 -2y1 -1y0 >= 2
+3x2 +2x1 +4x0 -1y2 -2y1 -1y0 >= 0
+0x2 +0x1 +0x0 +0y2 +1y1 +0y0 >= 0
+0x2 -1x1 +0x0 +0y2 +0y1 +0y0 >= 1
+1x2 +0x1 +0x0 -1y2 +0y1 +1y0 >= 0
-1x2 +0x1 +0x0 -1y2 +0y1 -1y0 >= 2
+0x2 +0x1 +1x0 +0y2 +0y1 +0y0 >= 0
+1x2 -2x1 +0x0 +1y2 +2y1 +1y0 >= 0
```

```
+1x2 +2x1 +2x0 -1y2 +0y1 -1y0 >= 0
+0x2 +0x1 -1x0 +0y2 +0y1 +0y0 >= 1
+3x2 -2x1 -2x0 +1y2 +4y1 +1y0 >= 0
-1x2 +0x1 +0x0 +1y2 +0y1 +1y0 >= 0
+0x2 +1x1 +0x0 +0y2 +0y1 +0y0 >= 0
+1x2 +2x1 +0x0 +1y2 -2y1 +1y0 >= 0
+1x2 +0x1 +0x0 +1y2 +0y1 -1y0 >= 0
+0x2 +0x1 +0x0 +0y2 -1y1 +0y0 >= 1
+1x2 -1x1 -1x0 +0y2 +1y1 +0y0 >= 1
```

□

*Example 3 (Greedy Approach [34] on* `LILLIPUT` *(LAT without $0^{th}$ row and $0^{th}$ column): Negative samples).*

```
+2x3 +1x2 +7x1 -5x0 +3y3 +4y2 -1y1 -7y0 >= 6
+2x3 +4x2 +1x1 +4x0 -2y3 -1y2 -3y1 +1y0 >= 1
-3x3 -1x2 -1x1 -1x0 +3y3 +1y2 +2y1 +2y0 >= 2
+1x3 -2x2 +1x1 +2x0 -3y3 -1y2 +1y1 -1y0 >= 4
-1x3 +2x2 -5x1 +1x0 +2y3 +5y2 -2y1 +3y0 >= 3
+4x3 -1x2 +1x1 -3x0 +1y3 +5y2 +6y1 +3y0 >= -2
-2x3 +1x2 +4x1 +2x0 +3y3 -4y2 -1y1 +1y0 >= 3
-4x3 -1x2 -5x1 -2x0 +4y3 -2y2 +3y1 -1y0 >= 10
+4x3 +1x2 -2x1 +2x0 +3y3 -1y2 -3y1 +1y0 >= 2
+2x3 -1x2 -2x1 -1x0 -2y3 +0y2 +1y1 +1y0 >= 4
```

```
-1x3 +2x2 +1x1 -2x0 -1y3 -2y2 +1y1 +1y0 >= 4
+2x3 +1x2 +4x1 +1x0 -3y3 +3y2 +1y1 -1y0 >= 0
-2x3 -1x2 -1x1 +1x0 -2y3 -1y2 -1y1 +2y0 >= 6
-2x3 -1x2 -2x1 +2x0 +1y3 +3y2 -2y1 +1y0 >= 4
+0x3 -1x2 +1x1 +0x0 -1y3 -1y2 +0y1 -1y0 >= 3
-2x3 +2x2 -1x1 +1x0 -2y3 +1y2 +2y1 -1y0 >= 4
+1x3 +2x2 -2x1 +0x0 -1y3 -1y2 -2y1 -2y0 >= 6
+1x3 -1x2 -1x1 -2x0 +2y3 +1y2 +3y1 -2y0 >= 3
-1x3 -1x2 +1x1 -2x0 -1y3 +1y2 -2y1 -1y0 >= 6
```

□

*Example 4 (Randomized greedy approach [4,29] on* `PRINCE` *(DPT without $0^{th}$ row and $0^{th}$ column): Negative samples).*

```
+1x3 +1x2 +1x1 +2x0 +2y3 +2y2 +2y1 +3y0 >= -5
+1x3 +0x2 +0x1 +0x0 +1y3 +2y2 +1y1 +1y0 >= -2
+0x3 +1x2 +0x1 +0x0 +1y3 +1y2 +2y1 +2y0 >= -2
+1x3 +1x2 +2x1 +1x0 +2y3 +4y2 +4y1 +3y0 >= -6
+0x3 +0x2 +0x1 +1x0 +2y3 +1y2 +1y1 +2y0 >= -2
```

```
-1x3 -1x2 -1x1 +1x0 -1y3 +1y2 +1y1 -1y0 >= 4
-1x3 -1x2 +1x1 -1x0 +1y3 +1y2 -1y1 -1y0 >= 4
-1x3 -1x2 +1x1 -1x0 +1y3 -1y2 +1y1 -1y0 >= 4
-1x3 -1x2 -1x1 -1x0 -1y3 -1y2 -1y1 -1y0 >= 7
```

□

*Example 5 (MILP [30] on* `RECTANGLE` *(DDT without $0^{th}$ row and $0^{th}$ column): Negative samples).*

```
+1x3 +1x2 +0x1 +2x0 -1y3 -2y2 -1y1 -2y0 >= 4
+0x3 +1x2 +1x1 -1x0 +0y3 -1y2 -1y1 +1y0 >= 2
+1x3 -1x2 +0x1 -1x0 +1y3 -1y2 +1y1 +0y0 >= 2
+1x3 +1x2 +2x1 +1x0 +0y3 +0y2 -1y1 -1y0 >= 0
-2x3 -1x2 -2x1 -1x0 +2y3 +1y2 -1y1 -1y0 >= 6
+1x3 -1x2 -1x1 +1x0 +2y3 +1y2 +1y1 +1y0 >= 0
-1x3 +0x2 +0x1 -1x0 +1y3 +1y2 +1y1 +0y0 >= 1
-1x3 -1x2 +1x1 +0x0 -1y3 +0y2 -1y1 +1y0 >= 3
-1x3 +1x2 +1x1 +0x0 +0y3 +0y2 +1y1 -1y0 >= 1
-1x3 +0x2 -1x1 +1x0 +1y3 -1y2 +0y1 -1y0 >= 3
```

```
+0x3 +1x2 -1x1 +0x0 +1y3 -1y2 -1y1 +1y0 >= 2
+2x3 -1x2 -1x1 +1x0 -2y3 -1y2 -1y1 -2y0 >= 6
+0x3 -1x2 -1x1 +0x0 +2y3 +1y2 +1y1 +2y0 >= 0
-1x3 -1x2 +1x1 +1x0 +1y3 +1y2 +2y1 +2y0 >= 0
-1x3 -1x2 +1x1 -2x0 -1y3 -1y2 +2y1 -2y0 >= 6
+0x3 +2x2 -2x1 +0x0 -1y3 +1y2 +1y1 +2y0 >= 1
+1x3 -2x2 +1x1 -2x0 -1y3 +1y2 +2y1 -1y0 >= 4
+0x3 +0x2 -1x1 +1x0 -1y3 +1y2 +1y1 +1y0 >= 2
-1x3 -1x2 -1x1 +1x0 -1y3 +1y2 +1y1 -1y0 >= 4
+1x3 +1x2 -1x1 -2x0 -1y3 +2y2 -1y1 -1y0 >= 4
```

□

*Example 6 (Subset Addition ($k = 2$) [29] on* `MIDORI S1` *(BCT without $0^{th}$ row and $0^{th}$ column): Negative samples).*

```
-5x3 +3x2 -2x1 -1x0 -5y3 +3y2 -2y1 -1y0 >= 11
-2x3 -1x2 +3x1 -2x0 +3y3 +1y2 -2y1 -3y0 >= 7
+3x3 +3x2 -1x1 -3x0 +2y3 +1y2 -2y1 +2y0 >= 3
+3x3 +3x2 +3x1 +2x0 +2y3 +2y2 +4y1 +3y0 >= -7
-3x3 +2x2 +1x1 -3x0 -3y3 +2y2 +1y1 -3y0 >= 9
+3x3 +1x2 -2x1 -3x0 -2y3 -1y2 +3y1 -2y0 >= 7
-1x3 -2x2 +3x1 -3x0 -3y3 -3y2 -1y1 +2y0 >= 10
+2x3 -3x2 +1x1 -3x0 -4y3 -5y2 -2y1 -1y0 >= 14
+3x3 +2x2 +1x1 -4x0 +5y3 -2y2 -4y1 +4y0 >= 5
```

```
+1x3 +2x2 -1x1 +2x0 -1y3 -1y2 +2y1 +3y0 >= 0
+5x3 -2x2 -4x1 +4x0 +3y3 +2y2 +1y1 -4y0 >= 5
-4x3 +2x2 -3x1 +1x0 -3y3 +3y2 -1y1 -2y0 >= 9
-1x3 -2x2 -2x1 -1x0 +2y3 -3y2 -2y1 -3y0 >= 11
-1x3 -4x2 -2x1 -5x0 -2y3 -3y2 -3y1 -5y0 >= 20
-1x3 -1x2 +2x1 +3x0 +1y3 +2y2 -1y1 +2y0 >= 0
-2x3 -3x2 +1x1 +3x0 -1y3 -3y2 +3y1 -2y0 >= 8
-3x3 -4x2 +1x1 -2x0 +5y3 -1y2 +2y1 -2y0 >= 8
-1x3 -2x2 -2x1 +2x0 -1y3 -2y2 -2y1 +2y0 >= 8
```

□

*Example 7 (Positive sample handling in secondary structure).* In this case, suppose $S$ is a binary variable which indicates if the SBox is active (with respect to positive samples). We present two methods to handle positive samples in secondary structure, those are based on the models from [34] and [4], respectively.

The objective function is to maximize $S$. For constructing the secondary structure, we start from the primary structure constraints, and then modify depending on the chosen method. We start with the primary structure constraints, then modify accordingly to convert to the secondary structure.

Primary structure:

```
-2x3 +1x2 +0x1 -2x0 +2y3 +2y2 +1y1 -1y0 >= 3
+2x3 +0x2 +1x1 -1x0 +2y3 -2y2 +2y1 -2y0 >= 3
+1x3 +3x2 +2x1 -3x0 -2y3 +2y2 +2y1 +3y0 >= 1
+1x3 +3x2 +3x1 +2x0 -1y3 +0y2 -2y1 -2y0 >= 2
-1x3 +2x2 +0x1 -2x0 +2y3 -2y2 -2y1 +1y0 >= 5
-1x3 +2x2 +1x1 -2x0 -2y3 -1y2 +2y1 +1y0 >= 4
+1x3 -2x2 +3x1 -3x0 -3y3 -3y2 -1y1 -2y0 >= 11
+3x3 -1x2 -1x1 -3x0 -2y3 +1y2 +2y1 +2y0 >= 4
+2x3 +2x2 -1x1 -2x0 -2y3 -1y2 -1y1 -1y0 >= 6
+4x3 +2x2 +2x1 +4x0 -1y3 -1y2 +2y1 -2y0 >= 0
+2x3 +2x2 -2x1 +3x0 -3y3 -1y2 -4y1 +4y0 >= 6
+1x3 -1x2 +0x1 +1x0 +1y3 +0y2 -2y1 -1y0 >= 3
+1x3 -3x2 -1x1 -2x0 +2y3 +3y2 +0y1 +3y0 >= 3
-2x3 -2x2 +0x1 +1x0 +1y3 +0y2 +2y1 +2y0 >= 2
-2x3 -2x2 +2x1 +1x0 -1y3 +1y2 -1y1 +2y0 >= 4
-1x3 -2x2 -1x1 -2x0 -2y3 +1y2 -1y1 -2y0 >= 9
-2x3 +0x2 -2x1 +1x0 -1y3 +1y2 +2y1 -2y0 >= 5
-1x3 +2x2 -3x1 -1x0 -1y3 -3y2 -2y1 +3y0 >= 8
```

Secondary structure following Sun et al.'s [34] method (5 auxiliary constraints):

```
x0 + x1 + x2 + x3 - S >= 0

x0 - S <= 0
x1 - S <= 0
x2 - S <= 0
x3 - S <= 0

-2x3 +1x2 +0x1 -2x0 +2y3 +2y2 +1y1 -1y0 >= 3
+2x3 +0x2 +1x1 -1x0 +2y3 -2y2 +2y1 -2y0 >= 3
+1x3 +3x2 +2x1 -3x0 -2y3 +2y2 +2y1 +3y0 >= 1
+1x3 +3x2 +3x1 +2x0 -1y3 +0y2 -2y1 -2y0 >= 2
-1x3 +2x2 +0x1 -2x0 +2y3 -2y2 -2y1 +1y0 >= 5
-1x3 +2x2 +1x1 -2x0 -2y3 -1y2 +2y1 +1y0 >= 4
+1x3 -2x2 +3x1 -3x0 -3y3 -3y2 -1y1 -2y0 >= 11
+3x3 -1x2 -1x1 -3x0 -2y3 +1y2 +2y1 +2y0 >= 4
+2x3 +2x2 -1x1 -2x0 -2y3 -1y2 -1y1 -1y0 >= 6
+4x3 +2x2 +2x1 +4x0 -1y3 -1y2 +2y1 -2y0 >= 0
+2x3 +2x2 -2x1 +3x0 -3y3 -1y2 -4y1 +4y0 >= 6
+1x3 -1x2 +0x1 +1x0 +1y3 +0y2 -2y1 -1y0 >= 3
+1x3 -3x2 -1x1 -2x0 +2y3 +3y2 +0y1 +3y0 >= 3
-2x3 -2x2 +0x1 +1x0 +1y3 +0y2 +2y1 +2y0 >= 2
-2x3 -2x2 +2x1 +1x0 -1y3 +1y2 -1y1 +2y0 >= 4
-1x3 -2x2 -1x1 -2x0 -2y3 +1y2 -1y1 -2y0 >= 9
-2x3 +0x2 -2x1 +1x0 -1y3 +1y2 +2y1 -2y0 >= 5
-1x3 +2x2 -3x1 -1x0 -1y3 -3y2 -2y1 +3y0 >= 8
```

Secondary structure following Baksi's [4] method (1 auxiliary constraint):

```
x0 + x1 + x2 + x3 + y0 + y1 + y2 + y3 - S >= 0

-2x3 +1x2 +0x1 -2x0 +2y3 +2y2 +1y1 -1y0 + 3(1 - S) >= 3
+2x3 +0x2 +1x1 -1x0 +2y3 -2y2 +2y1 -2y0 + 3(1 - S) >= 3
+1x3 +3x2 +2x1 -3x0 -2y3 +2y2 +2y1 +3y0 + 1(1 - S) >= 1
+1x3 +3x2 +3x1 +2x0 -1y3 +0y2 -2y1 -2y0 + 2(1 - S) >= 2
-1x3 +2x2 +0x1 -2x0 +2y3 -2y2 -2y1 +1y0 + 5(1 - S) >= 5
-1x3 +2x2 +1x1 -2x0 -2y3 -1y2 +2y1 +1y0 + 4(1 - S) >= 4
+1x3 -2x2 +3x1 -3x0 -3y3 -3y2 -1y1 -2y0 + 11(1 - S) >= 11
+3x3 -1x2 -1x1 -3x0 -2y3 +1y2 +2y1 +2y0 + 4(1 - S) >= 4
+2x3 +2x2 -1x1 -2x0 -2y3 -1y2 -1y1 -1y0 + 6(1 - S) >= 6
+4x3 +2x2 +2x1 +4x0 -1y3 -1y2 +2y1 -2y0 + 0(1 - S) >= 0
+2x3 +2x2 -2x1 +3x0 -3y3 -1y2 -4y1 +4y0 + 6(1 - S) >= 6
+1x3 -1x2 +0x1 +1x0 +1y3 +0y2 -2y1 -1y0 + 3(1 - S) >= 3
+1x3 -3x2 -1x1 -2x0 +2y3 +3y2 +0y1 +3y0 + 3(1 - S) >= 3
-2x3 -2x2 +0x1 +1x0 +1y3 +0y2 +2y1 +2y0 + 2(1 - S) >= 2
-2x3 -2x2 +2x1 +1x0 -1y3 +1y2 -1y1 +2y0 + 4(1 - S) >= 4
-1x3 -2x2 -1x1 -2x0 -2y3 +1y2 -1y1 -2y0 + 9(1 - S) >= 9
-2x3 +0x2 -2x1 +1x0 -1y3 +1y2 +2y1 -2y0 + 5(1 - S) >= 5
-1x3 +2x2 -3x1 -1x0 -1y3 -3y2 -2y1 +3y0 + 8(1 - S) >= 8
```

The following observations can be made regarding the two methods. Baksi's method [4] takes less number of auxiliary constraints. It requires only 1 auxiliary constraint, even though it modifies all the original constraints by inserting an additional term involving $S$. Sun et al.'s method [34], in contrast, requires more auxiliary constraints ($n+1$ for an $n$-bit SBox).

Further, both methods [4,34] inherently deal with $0^{\text{th}}$ row and $0^{\text{th}}$ column issues. For instances, in Sun et al.'s method; the first auxiliary constraint ensures that when all $x_i$ are zero, $S$ must be set to 0 (inactive). The next $n$ auxiliary constraints ensure that if any $x_i$ is non-zero, $S$ can be set to 1 (active); and for bijective SBoxes the same constraints also ensure that at least one of the $y_i$ is non-zero. □

*Example 8 (Negative sample handling in secondary structure).* $S$ is a binary variable which indicates if the SBox is active (with respect to negative samples), following the secondary structure modeling from [34] (the alternate method from [4] is omitted here for brevity). Similar to Example 7, the objective function remains the same, to maximize $S$.

```
x0 + x1 + x2 + x3 - (1 - S) >= 0

x0 - (1 - S) <= 0
x1 - (1 - S) <= 0
x2 - (1 - S) <= 0
x3 - (1 - S) <= 0

-2x3 +1x2 +0x1 -2x0 +2y3 +2y2 +1y1 -1y0 >= 3
+2x3 +0x2 +1x1 -1x0 +2y3 -2y2 +2y1 -2y0 >= 3
+1x3 +3x2 +2x1 -3x0 -2y3 +2y2 +2y1 +3y0 >= 1
+1x3 +3x2 +3x1 +2x0 -1y3 +0y2 -2y1 -2y0 >= 2
-1x3 +2x2 +0x1 -2x0 +2y3 -2y2 -2y1 +1y0 >= 5
```

```
-1x3 +2x2 +1x1 -2x0 -2y3 -1y2 +2y1 +1y0 >= 4
+1x3 -2x2 +3x1 -3x0 -3y3 -3y2 -1y1 -2y0 >= 11
+3x3 -1x2 -1x1 -3x0 -2y3 +1y2 +2y1 +2y0 >= 4
+2x3 +2x2 -1x1 -2x0 -2y3 -1y2 -1y1 -1y0 >= 6
+4x3 +2x2 +2x1 +4x0 -1y3 -1y2 +2y1 -2y0 >= 0
+2x3 +2x2 -2x1 +3x0 -3y3 -1y2 -4y1 +4y0 >= 6
+1x3 -1x2 +0x1 +1x0 +1y3 +0y2 -2y1 -1y0 >= 3
+1x3 -3x2 -1x1 -2x0 +2y3 +3y2 +0y1 +3y0 >= 3
-2x3 -2x2 +0x1 +1x0 +1y3 +0y2 +2y1 +2y0 >= 2
-2x3 -2x2 +2x1 +1x0 -1y3 +1y2 -1y1 +2y0 >= 4
-1x3 -2x2 -1x1 -2x0 -2y3 +1y2 -1y1 -2y0 >= 9
-2x3 +0x2 -2x1 +1x0 -1y3 +1y2 +2y1 -2y0 >= 5
-1x3 +2x2 -3x1 -1x0 -1y3 -3y2 -2y1 +3y0 >= 8
```

However, an alternate version of this model can be constructed by simply running for the minimum number of active negative samples instead of maximum number of active positive samples. In that case, no auxiliary constraints are needed, and the objective function in that case would be to minimize $S$. Finally, to revert back to the original maximization problem, one has to subtract that from the total number of SBoxes. □

## B Reference to SBoxes

The following SBoxes are used in our experiments:

- ASCON [20]
- FIDES-5 [13]
- GIFT [8]
- ICEPOLE [26]
- KECCAK [11]
- KLEIN [21]
- LILLIPUT [10]
- MIBS [22]
- MIDORI [7]
- MINALPHER [19]
- PICCOLO [31]
- PRESENT [15]
- PRIDE [3]
- PRINCE [16]
- RECTANGLE [37]
- SKINNY [9]
- TWINE [35]
- SC2000 [32]

## References


1. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) s-boxes to optimize probability of differential characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017), https://doi.org/10.13154/tosc.v2017.i4.99-129 2
2. Aitipamula, A., Pal, D., Chowdhury, D.R.: Efficient and optimized modeling of s-boxes. In: Progress in Cryptology - AFRICACRYPT 2025 - 16th International Conference on Cryptology in Africa, Rabat, Morocco, July 21-23, 2025, Proceedings. Lecture Notes in Computer Science, vol. 15651, pp. 441–469. Springer (2025), https://doi.org/10.1007/978-3-031-97260-7_20 2
3. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçin, T.: Block ciphers - focus on the linear layer (feat. PRIDE). In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8616, pp. 57–76. Springer (2014), https://doi.org/10.1007/978-3-662-44371-2_4 18
4. Baksi, A.: New insights on differential and linear bounds using mixed integer linear programming (full version). Cryptology ePrint Archive, Report 2020/1414 (2020), https://eprint.iacr.org/2020/1414 10, 11, 13, 14, 16, 17
5. Baksi, A.: DEFAULT: Cipher-Level Resistance Against Differential Fault Attack. Springer, Singapore (2022), https://link.springer.com/chapter/10.1007/978-981-16-6522-6_8 2
6. Baksi, A., Breier, J., Chattopadhyay, A., Gerlich, T., Guilley, S., Gupta, N., Isobe, T., Jati, A., Jedlicka, P., Kim, H., Liu, F., Martinásek, Z., Sakamoto, K., Seo, H., Shiba, R., Shrivastwa, R.R.: Baksheesh: Similar yet different from gift. Cryptology ePrint Archive, Paper 2023/750 (2023), https://eprint.iacr.org/2023/750 2
7. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015). https://doi.org/10.1007/978-3-662-48800-3_17, https://doi.org/10.1007/978-3-662-48800-3_17 18
8. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: Gift: A small present. Cryptology ePrint Archive, Report 2017/622 (2017), https://eprint.iacr.org/2017/622 18
9. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. IACR Cryptology ePrint Archive **2016**, 660 (2016), http://eprint.iacr.org/2016/660 18

10. Berger, T.P., Francq, J., Minier, M., Thomas, G.: Extended generalized feistel networks using matrix representation to propose a new lightweight block cipher: Lilliput. IEEE Trans. Computers **65**(7), 2074–2089 (2016). https://doi.org/10.1109/TC.2015.2468218, https://doi.org/10.1109/TC.2015.2468218 18
11. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Sponge functions. In: Ecrypt Hash Workshop (May 2007) (2007), https://keccak.team/files/CSF-0.1.pdf 18
12. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer-Verlag, Berlin, Heidelberg (1993) 3
13. Bilgin, B., Bogdanov, A., Knezevic, M., Mendel, F., Wang, Q.: Fides: Lightweight authenticated cipher with side-channel resistance for constrained hardware. In: Bertoni, G., Coron, J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8086, pp. 142–158. Springer (2013) 18
14. Biryukov, A., Kushilevitz, E.: From differential cryptanalysis to ciphertext-only attacks. In: Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. pp. 72–88 (1998), https://doi.org/10.1007/BFb0055721 3
15. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: CHES. vol. 4727, pp. 450–466. Springer (2007) 18
16. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knežević, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: Prince - a low-latency block cipher for pervasive computing applications (full version). Cryptology ePrint Archive, Report 2012/529 (2012), https://ia.cr/2012/529 18
17. Boura, C., Coggia, D.: Efficient milp modelings for sboxes and linear layers of spn ciphers. IACR Transactions on Symmetric Cryptology **2020**(3), 327–361 (Sep 2020), https://tosc.iacr.org/index.php/ToSC/article/view/8705 2
18. Boura, C., Coggia, D.: Efficient MILP modelings for sboxes and linear layers of SPN ciphers. IACR Trans. Symmetric Cryptol. **2020**(3), 327–361 (2020), https://doi.org/10.13154/tosc.v2020.i3.327-361 2, 7, 10, 11, 13, 14, 15
19. Chakraborti, A., Datta, N., Nandi, M.: Practical fault attacks on minalpher: How to recover key with minimum faults? In: Security, Privacy, and Applied Cryptography Engineering - 7th International Conference, SPACE 2017, Goa, India, December 13-17, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10662, pp. 111–132. Springer (2017). https://doi.org/10.1007/978-3-319-71501-8_7, https://doi.org/10.1007/978-3-319-71501-8_7 18
20. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing. J. Cryptol. **34**(3), 33 (2021), https://doi.org/10.1007/s00145-021-09398-9 18
21. Gong, Z., Nikova, S., Law, Y.W.: Klein: A new family of lightweight block ciphers. In: Juels, A., Paar, C. (eds.) RFID. Security and Privacy. pp. 1–18. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 18
22. Izadi, M., Sadeghiyan, B., Sadeghian, S.S., Khanooki, H.A.: MIBS: A new lightweight block cipher. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) Cryptology and Network Security, 8th International Conference, CANS 2009, Kanazawa, Japan, December 12-14, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5888, pp. 334–348. Springer (2009) 18
23. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings. pp. 17–38 (1991), https://doi.org/10.1007/3-540-46416-6_2 3
24. Li, T., Sun, Y.: Superball: A new approach for MILP modelings of boolean functions. IACR Trans. Symmetric Cryptol. **2022**(3), 341–367 (2022) 2
25. Liu, Y., Liang, H., Li, M., Huang, L., Hu, K., Yang, C., Wang, M.: STP models of optimal differential and linear trail for s-box based ciphers. Sci. China Inf. Sci. **64**(5) (2021), https://doi.org/10.1007/s11432-018-9772-01
26. Morawiecki, P., Gaj, K., Homsirikamol, E., Matusiewicz, K., Pieprzyk, J., Rogawski, M., Srebrny, M., Wojcik, M.: ICEPOLE: High-speed, hardware-oriented authenticated encryption. Cryptology ePrint Archive, Paper 2014/266 (2014), https://eprint.iacr.org/2014/266 18
27. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C., Yung, M., Lin, D. (eds.) Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2011), https://doi.org/10.1007/978-3-642-34704-7_52
28. Pal, D., Chandratreya, V.P., Chowdhury, D.R.: Efficient algorithms for modeling sboxes using MILP. CoRR **abs/2306.02642** (2023), https://doi.org/10.48550/arXiv.2306.02642 2, 10, 11, 13, 14
29. Pal, D., Chandratreya, V.P., Chowdhury, D.R.: New techniques for modeling sboxes: An MILP approach. In: Cryptology and Network Security - 22nd International Conference, CANS 2023, Augusta, GA, USA, October

31 - November 2, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14342, pp. 318–340. Springer (2023), https://doi.org/10.1007/978-981-99-7563-1_15 10, 11, 12, 13, 14, 15, 16
30. Sasaki, Y., Todo, Y.: New algorithm for modeling s-box in MILP based differential and division trail search. In: Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8-9, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10543, pp. 150–165. Springer (2017), https://doi.org/10.1007/978-3-319-69284-5_11 2, 7, 10, 11, 13, 14, 15, 16
31. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An Ultra-Lightweight Blockcipher. In: Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. pp. 342–357 (2011). https://doi.org/10.1007/978-3-642-23951-9_23, https://doi.org/10.1007/978-3-642-23951-9_23 18
32. Shimoyama, T., Yanami, H., Yokoyama, K., Takenaka, M., Itoh, K., Yajima, J., Torii, N., Tanaka, H.: The block cipher SC2000. In: Matsui, M. (ed.) Fast Software Encryption, 8th International Workshop, FSE 2001 Yokohama, Japan, April 2-4, 2001, Revised Papers. Lecture Notes in Computer Science, vol. 2355, pp. 312–327. Springer (2001), https://doi.org/10.1007/3-540-45473-X_26 18
33. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Paper 2014/747 (2014), https://eprint.iacr.org/2014/747 2
34. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In: 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. Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer (2014) 2, 6, 10, 11, 12, 13, 14, 15, 16, 17
35. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: A lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) Selected Areas in Cryptography. pp. 339–354. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 18
36. Wu, S., Wang, M.: Security evaluation against differential cryptanalysis for block cipher structures. Cryptology ePrint Archive, Paper 2011/551 (2011), https://eprint.iacr.org/2011/551 2
37. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: RECTANGLE: A bit-slice ultra-lightweight block cipher suitable for multiple platforms. IACR Cryptol. ePrint Arch. p. 84 (2014) 18