MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics

Ahmed Abdelkhalek[1], Yu Sasaki[2], Yosuke Todo[2], Mohamed Tolba[1] and Amr M. Youssef[1]

[1] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada {abdelk_a,m_tolba}@encs.concordia.ca,youssef@ciise.concordia.ca
[2] NTT Secure Platform Laboratories {sasaki.yu,todo.yosuke}@lab.ntt.co.jp

Abstract. Current Mixed Integer Linear Programming (MILP)-based search against symmetric-key primitives with 8-bit S-boxes can only build word-wise model to search for truncated differential characteristics. In such a model, the properties of the Differential Distribution Table (DDT) are not considered. To take these properties into account, a bit-wise model is necessary, which can be generated by the H-representation of the convex hull or the logical condition modeling. However, the complexity of both approaches becomes impractical when the size of the S-box exceeds 5 bits. In this paper, we propose a new modeling for large (8-bit or more) S-boxes. In particular, we first propose an algorithm to generate a bit-wise model of the DDT for large S-boxes. We observe that the problem of generating constraints in logical condition modeling can be converted into the problem of minimizing the product-of-sum of Boolean functions, which is a well-studied problem. Hence, classical off-the-shelf solutions such as the Quine-McCluskey algorithm or the Espresso algorithm can be utilized, which makes building a bit-wise model, for 8-bit or larger S-boxes, practical. Then this model is further extended to search for the best differential characteristic by considering the probabilities of each propagation in the DDT, which is a much harder problem than searching for the lower bound on the number of active S-boxes. Our idea is to separate the DDT into multiple tables for each probability and add conditional constraints to control the behavior of these multiple tables. The proposed modeling is first applied to `SKINNY-128` to find that there is no differential characteristic having probability higher than 2^{-128} for 14 rounds, while the designers originally expected that 15 rounds were required. We also applied the proposed modeling to two, arbitrarily selected, constructions of the seven AES round function based constructions proposed in FSE 2016 and managed to improve the lower bound on the number of the active S-boxes in one construction and the upper bound on the differential characteristic for the other.

Keywords: Mixed Integer Linear Programming (MILP); Automated Cryptanalysis; Differential Characteristic; SKINNY-128; AES-based constructions

1 Introduction

The use of Mixed Integer Linear Programming (MILP) as a supporting tool in symmetric-key cryptography has started by Mouha et al. [MWGP11] and Wu and Wang [WW11]. They have proposed two slightly different approaches to model the problem of finding a lower bound on the number of active S-boxes for both differential and linear cryptanalysis as an MILP problem that can be solved by any MILP solver such as Gurobi Optimizer [Inc15], SCIP [GFG+16] and CPLEX Optimization Studio [ILO16]. Such a lower bound and the maximum differential (linear) probability of the S-box derive an upper bound on

IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2017, No. 4, pp. 99–129
DOI:10.13154/tosc.v2017.i4.99-129

the probability of the best differential characteristic (linear approximation). This helps designers of symmetric-key primitives to prove their resistance against differential (linear) cryptanalysis after a given number of rounds.

For Substitution Permutation Network (SPN) block ciphers, it is possible to evaluate the lower bound on the number of active S-boxes efficiently by searching for truncated differential characteristics. However, the model of truncated differential characteristic search is not applicable to bit-oriented ciphers, e.g. PRESENT [BKL+07]. Moreover, the discovered truncated differential characteristic might not always be valid and may contain contradiction when propagation of actual differences is considered. To solve these issues, Sun et al. proposed an MILP modeling for bit-oriented SPN block ciphers [SHW+14b]. In the bit-wise model, it is hard to represent the valid differential propagations through an S-box. The possible propagation patterns (non-zero entries) and impossible propagation patterns (zero entries) of the Differential Distribution Table (DDT) of an S-box can be represented by linear constraints. Because the goal is to exclude impossible patterns, probabilities of possible transitions are out of concern. Hence, a truncated version of the DDT, where all the non-zero entries of the DDT are replaced by 1, is analyzed. For convenience, this table is called "$*$-DDT" in this paper.

Sun et al. proposed two methods to represent the valid solution range of a $*$-DDT: *H-representation of the convex hull* and *logical condition modeling* [SHW+14b]. The technical details of these two methods are explained in Sect 2.2. Here, we explain that both approaches have important limitations with respect to the following two points.

Application to 8-bit S-boxes It has been pointed out by several authors that MILP approaches cannot be applied to 8-bit S-boxes. For example, Sun et al. wrote that "To the best of our knowledge, the MILP approach is unable to search for actual differential characteristics of ciphers with 8-bit S-boxes" [SGL+17]. Sasaki and Todo wrote that "MILP requires too many inequalities to represent the differential propagations in the DDT of 8-bit S-boxes" [ST17b]. In fact, generating the linear inequalities for the H-representation of the convex hull, often by using SageMath, requires an exponential complexity in the number of input and output difference bits. Sasaki and Todo demonstrated an exhaustive list of compact representations in logical condition modeling against a 4-bit S-box, but its applicability to 8-bit S-boxes is questionable [ST16].

Optimizing the probability of differential characteristics To evaluate the security against differential cryptanalysis, ideally, the upper bound on the probability of differential characteristics should be evaluated directly instead of the rough evaluation by raising the maximum differential probability per S-box to the power of the lower bound on the number of active S-boxes. However, most of the previous MILP work focused on modeling the $*$-DDT. Up to the authors knowledge, there exists only one paper in the literature that tackles this problem [SHW+14a]. However, the system of inequalities becomes much heavier than analyzing the $*$-DDT and the authors were only able to evaluate 4 rounds of Serpent [BAK98].

1.1 Our Contributions

In this paper, a new MILP modeling that can be applied to large S-boxes is presented. This modeling can be used not only for minimizing the number of active S-boxes but also for maximizing the probability of differential characteristics. The new modeling contains two algorithmic improvements: the first improvement is for efficiently modeling the $*$-DDT of large S-boxes, and the second one is for efficiently modeling the probabilities within the DDT of large S-boxes. These improvements allow us to utilize the MILP approach on ciphers using 8-bit S-boxes.

It is to be noted that we do not claim that MILP is more advantageous than other approaches such as SAT and constraint programming (CP). In general, MILP/CP/SAT are tools to solve NP-complete problems and we believe that one cannot outperform the other two. On the other hand, there seems to be a previous belief that MILP cannot be used to model the differential propagation through 8-bit S-boxes which we refute in this paper by introducing the following two novel ideas.

The first novel idea is to use the *Quine-McCluskey algorithm* [Qui52, Qui55, McC56] for modeling the $*$-DDT. We observe that the logical condition modeling is highly related to the product-of-sum representation of Boolean functions. The Quine-McCluskey algorithm can derive the minimum product-of-sum representation of a given Boolean function from its truth table. By exploiting this feature, the minimum set of linear inequalities to represent the $*$-DDT in the logical condition model can be obtained.

The second novel idea is to split the DDT into multiple tables for each probability. For example, all the non-trivial entries of the DDT of the AES S-box are either 2 (for 2^{-7}) or 4 (for 2^{-6}), and we prepare two tables, one contains '1' only for the entries of 2^{-7} and the other contains '1' only for the entries of 2^{-6}. We then introduce indicator variables $Q_{2^{-7}}, Q_{2^{-6}} \in \{0, 1\}$ and introduce conditional constraints such that the inequalities to represent the table for 2^{-7} is effective only when $Q_{2^{-7}} = 1$ and the inequalities to represent the table for 2^{-6} is effective only when $Q_{2^{-6}} = 1$. We also introduce another indicator variable Q that takes 1 if the S-box is active and 0 otherwise. We then set $Q_{2^{-7}} + Q_{2^{-6}} = Q$ for each S-box, and the entire probability (or rather the base-2 logarithm of the probability multiplied by -1) can be represented by a sum of $7 \times Q_{2^{-7}} + 6 \times Q_{2^{-6}}$ over all the S-boxes. Moreover, this method can freely handle probabilities whose base-2 logarithm are not integers. For example, the DDT of the `SKINNY-128` S-box has entries with probability $2^{-2.4}$, and this is simply represented as $2.4 \times Q_{2^{-2.4}}$.

As a proof-of-concept, we evaluate the maximum probability of differential characteristics of `SKINNY-128` [BJK+16] and two AES-round based constructions [JN16] with the proposed S-box modeling. For `SKINNY-128`, the designers showed that the minimum number of active S-boxes for 14 rounds is 61 in the single-key setting, which indicates that the upper bound on the characteristic probability for 14 rounds is 2^{-122}. With the proposed method, we show that there does not exist any 14-round differential characteristic with probability higher than 2^{-128}, hence having 14 rounds is sufficient to resist simple differential distinguishers. To reach this conclusion, we propose and utilize two optimization techniques for `SKINNY-128`: cutting-off low probability transitions and equivalence classes. The former reduces the number of inequalities to 17%, and the latter reduces the entire running time to 1/4.

For the AES-round based constructions in [JN16], among the seven constructions (C1 to C7) proposed by the designers, we applied our approach on two arbitrarily selected constructions, i.e., C1 and C5. The designers showed that the minimum number of active S-boxes is 22 for both constructions, which ensures that the upper bound on the probability is smaller than 2^{-128}. For C1, we show that the best probability is slightly smaller, namely 2^{-134}. For C5, we show that no truncated differential characteristic with 22 or 23 active S-boxes can be instantiated. Hence, we improve the lower bound on the number of active S-boxes by 2.

1.2 Related Work

The use of MILP in symmetric-key cryptography has been amplified in the past few years. Some are directly related to our research and others only have weak connection to it. We give below a brief summary of the previous MILP-based work.

In the first work on bit-oriented models [SHW+14b], Sun et al. found the best differential characteristic of a number of bit-oriented SPN block ciphers in the single-key and related-key settings. They have used a heuristic approach and therefore the automatically found

differential characteristic has to be verified manually as it might be invalid. Shortly afterwards, Sun et al. proposed a methodology to exactly represent the differential propagation through an S-box [SHW+14a]. Their methodology was computationally feasible when the size of the S-box is 5 bits or less. Then, Fu et al. proposed an MILP-based method to search for the best differential and linear characteristics in ARX based block ciphers under the assumption of independent inputs to both the modular addition and the different rounds [FWG+16]. In addition, by incorporating linear incompatibility in the MILP model, MILP-based tool was used to greatly enhance the related-tweakey differential bounds of Deoxys and its internal tweakable block ciphers [CHP+17].

Recently, Sasaki and Todo presented an MILP-based tool to automatically search for the longest impossible differential in SPN-based block ciphers [ST17b]. They have pointed out the inability of the current approaches to efficiently represent large S-boxes and suggested the use of what they have called the *arbitrary S-box mode* to represent the differential propagation of 8-bit S-boxes. Independently, Cui et al. proposed a similar tool to search for impossible differentials and zero-correlation linear approximations with emphasis on ARX block ciphers [CJF+16].

MILP usage was not limited to differential and linear cryptanalysis only and was extended to Integral cryptanalysis. Xiang et al. [XZBL16] have proposed an MILP-based method to find integral distinguishers based on the division property [Tod15] and applied it to 6 lightweight block ciphers. Soon after, their approach was extended to primitives with non-bit permutation linear layers and ARX based primitives [SWW16, SWLW16]. However, their solutions were found to encompass some infeasible division trails which could affect the search results and yield shorter integral distinguishers as shown in [ZR17]. Moreover, a new MILP model was developed to consider the effect of the ladder switch technique when combining two short differential trails into boomerang or rectangle attacks [CHP+17].

We emphasize that while we focus our attention on differential cryptanalysis, the proposed method for efficiently modeling large S-boxes can be directly used in other contexts such as linear cryptanalysis, impossible differential attack, zero-correlation attack, and integral attack.

Paper outline. The remainder of this paper is organized as follows. In Sect. 2, we explain technical details of MILP modeling. In Sect. 3, we present a new algorithm to efficiently model large S-boxes. In Sect. 4, we present a new modeling of large S-boxes to efficiently optimize the probability of differential characteristics. The proposed modeling is applied to `SKINNY-128` in Sect. 5 and to two of the AES-round based constructions in Sect. 6. Finally, the paper is concluded in Sect. 7.

2 Details of MILP Modeling

In this section, we explain how to model the problem of finding the best truncated differential characteristic and differential characteristic using MILP. The word-wise modeling is introduced in Sect. 2.1 and then the bit-wise modeling is introduced in Sect. 2.2.

2.1 Searching for Word-wise Truncated Differential Characteristics

We first explain the MILP model for AES [Nat01], which will be analyzed later in Sect. 6. The state of AES is composed of 16 bytes. To evaluate r rounds, one defines $16r$ binary variables $x_i \in \{0, 1\}$, in which $x_i = 1$ denotes that the ith byte has a non-zero difference (active) and $x_i = 0$ denotes that the ith byte has no difference (inactive). To minimize the number of active S-boxes, the objective function is set to "minimize Σx_i." One then needs to define the solution range of x_i by using linear inequalities to exclude solutions with invalid propagation through the round function. `SubBytes` does not have any impact

in the truncated differential characteristic search and `ShiftRows` can be naturally handled by permuting the bytes of the internal state. The only complex operation is `MixColumns` which has the property that its branch number is 5. Mouha et al. [MWGP11] introduced a dummy variable, $d_j \in \{0,1\}$, for Column j, and expressed the constraints of the branch number with 9 inequalities per column. For example, suppose that the status of the 4 input and output bytes of `MixColumns` is denoted by a_0, a_1, a_2, a_3 and b_0, b_1, b_2, b_3, respectively, where $a_i, b_i \in \{0,1\}$. Valid patterns for a_i, b_i can be expressed as:

$$\Sigma_{i=0}^{3} a_i + \Sigma_{i=0}^{3} b_i \geq 5d_j, \tag{1}$$

$$d_j \geq a_i \text{ for } i \in \{0,1,2,3\}, d_j \geq b_i \text{ for } i \in \{0,1,2,3\}. \tag{2}$$

Indeed, when $d_j = 0$, Eq. (1) has no effect and all a_i, b_i are set to 0 due to Eq. (2). When $d_j = 1$, Eq. (2) has no effect and the sum of the active input and output bytes is ensured to be at least 5 due to Eq. (1).

Another basic operation is the XOR of two variables. Suppose that two bytes whose active status are denoted by v_0 and v_1 are XORed and the active status of the output byte is denoted by v_2 where $v_i \in \{0,1\}$. Then, $(v_0, v_1, v_2) \in \{(0,0,1),(0,1,0),(1,0,0)\}$ are impossible propagation patterns, and thus must be excluded from the solution space. This can be done by including one inequality per pattern as follows:

$$v_0 + v_1 - v_2 \geq 0, \qquad v_0 - v_1 + v_2 \geq 0, \qquad -v_0 + v_1 + v_2 \geq 0. \tag{3}$$

2.2 Searching for Bit-wise Differential Characteristics

In bit-wise models [SHW+14b, SHW+14a], binary variables are assigned to each bit of the state. In this case, the modeling of linear operations stay relatively simple*, while setting the solution range to exclude invalid propagation patterns through S-boxes becomes hard. As explained in Sect. 1, one needs to represent all impossible propagation patterns of the $*$-DDT by linear constraints, which is often done by H-representation of the convex hull or logical condition modeling.

H-representation of the convex hull. Let $(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \in \mathbb{R}^{2n}$ be a $2n$-dimensional vector, where $\mathbb{R}$ is the real number field, and an input-output differential pattern of an S-box is represented as the point $(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1})$. Then, by computing the H-representation of the convex hull of all possible input-output differential patterns of the S-box, we can get w linear inequalities such as $\mathbf{A} \times (x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \leq b$, where $\mathbf{A}$ is a $2n \times w$ matrix whose elements are integer and b is an integer. Every linear inequality removes some points that correspond to impossible differential patterns. However, this representation includes redundant linear inequalities in the context of the MILP-based differential characteristic search because the feasible points are restricted to $\{0,1\}^{2n}$ not $\mathbb{R}^{2n}$, and too many redundant linear inequalities make the solver of the MILP problem slower. In order to reduce the number of inequalities, heuristic methods have been applied, e.g., a greedy algorithm proposed in [SHW+14b].

Logical condition modeling. For simplicity, let us consider the case of a 4-bit S-box. Let (x_0, x_1, x_2, x_3) and (y_0, y_1, y_2, y_3) be MILP variables for the input and output differences, respectively. Assuming that $(1001) \rightarrow (1101)$ is an impossible propagation, which means that the input difference (1001) does not propagate to (1101), the following linear inequality removes only this impossible point.

$$-x_0 + x_1 + x_2 - x_3 - y_0 - y_1 + y_2 - y_3 + 4 \geq 0.$$

*It is worth mentioning that the model of the XOR operation slightly changes. Besides Eq. (3), $(v_0, v_1, v_2) = (1,1,1)$ becomes impossible, which can be excluded by $-v_0 - v_1 - v_2 \geq -2$.

This inequality is obtained by setting -1 to the coefficients of the variables corresponding to 1 and by setting 0 to the coefficients of the variables corresponding to 0. The constant term is calculated as the total number of coefficients of -1 minus one, and the right-hand side is always ≥ 0. In fact, the left-hand side is minimized to -1 only if $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3) = (10011101)$, and thus only this pattern is excluded from the solution space. Therefore, if there are m impossible propagations in a DDT, m linear inequalities are enough to represent the DDT accurately. However, we can further reduce the number of linear inequalities by combining some of them.For example, assuming that there are two impossible propagations $(1001) \to (1101)$ and $(1001) \to (1100)$, one linear inequality:

$$-x_0 + x_1 + x_2 - x_3 - y_0 - y_1 + y_2 + 3 \geq 0$$

remove both propagations together. Similar to the case of the H-representation, the greedy heuristic algorithm can be used to reduce the number of linear inequalities.

Limitations of previous modeling. As mentioned in Sect. 1, both of the two methods have two important limitations: (i) they cannot be applied to large S-boxes, in particular 8-bit S-boxes, and (ii) it is hard to optimize the probability of differential characteristics instead of the number of active S-boxes. In the following sections, we propose a new method to overcome both limitations.

3 New Algorithms to Model $*$-DDT for Large S-boxes

In this section, we show a unified algorithm to generate the linear constraints for a $*$-DDT based on the logical condition modeling, where the number of linear inequalities is minimized or as small as possible. Here, the minimality is ensured only in the context of the logical condition modeling, that is, the coefficients of each term is limited to $\{-1, 0, 1\}$. When the coefficients can be any integer, the presented algorithm does not ensure the minimality of the number of inequalities. In fact, the H-representation usually yields a system with a fewer number of inequalities, though it cannot be computed for a large S-box. This algorithm is also used as a subroutine to generate the linear constraints for a DDT while considering the probability values. Further details will be presented in Sect. 4.

3.1 Product-of-Sum Representation of Boolean Functions

We revisit the logical condition modeling and show that the product-of-sum representation of Boolean functions is highly related to this modeling. Thanks to this observation, we can notice that the minimum set of linear inequalities to represent a DDT can be computed by applying the *Quine-McCluskey algorithm* [Qui52, Qui55, McC56].

Let $f(\vec{x}, \vec{y})$ be a Boolean function of $2n$-bit inputs, where $\vec{x} = (x_1, x_2, \cdots, x_n)$ and $\vec{y} = (y_1, y_2, \cdots, y_n)$ denote the input and output differences, respectively, and the output of $f(\vec{x}, \vec{y})$ is 1 only if the input-output differential pattern is possible in $*$-DDT[†]. Then, the simple inequality $f(\vec{x}, \vec{y}) \geq 1$ is enough to remove all impossible input-output differential patterns. Of course, we have to convert the inequality $f(\vec{x}, \vec{y}) \geq 1$ into linear inequalities for the use of MILP. In a first step, let us consider the product-of-sum representation of the Boolean function f as

$$f(\vec{x}, \vec{y}) = \bigwedge_{\vec{c} \in \{0,1\}^{2n}} \left(\alpha_{\vec{c}} \vee \bigvee_{i=1}^{n} (x_i \oplus c_i) \vee \bigvee_{i=1}^{n} (y_i \oplus c_{n+i}) \right),$$

[†]This function f is the same as the Boolean function γ_F introduced in [CCZ98].

where $\alpha_{\vec{c}} \in \{0,1\}$, $\alpha_{\vec{c}} = f(\vec{c})$, and $\vec{c} = (c_1, c_2, \cdots, c_n)$[‡]. Here $\wedge$ and $\vee$ denote logical AND and OR operations, respectively. In this representation, the necessary condition for $f(\vec{x},\vec{y}) \geq 1$ is rewritten as $\bigvee_{i=1}^{n}(x_i \oplus c_i) \bigvee_{i=1}^{n}(y_i \oplus c_{n+i}) = 1$ for all $\vec{c} \in \{\{0,1\}^{2n} | \alpha_{\vec{c}} = 0\}$. When at least one of $(x_i \oplus c_i)$ and $(y_i \oplus c_{n+i})$ is 1, the logical OR is 1. Moreover, since a 1-bit XOR $a \oplus b$ can be rewritten as $a + b - 2ab$, the condition can be rewritten as:

$$\sum_{i=1}^{n}(x_i + c_i - 2x_i c_i) + \sum_{i=1}^{n}(y_i + c_{i+n} - 2y_i c_{i+n}) \geq 1$$

for all $\vec{c} \in \{\{0,1\}^{2n} | \alpha_{\vec{c}} = 0\}$. Note that the same conversion can be applied even if some variables in $\vec{x}$ and $\vec{y}$ are not involved in the representation.

Let us revisit the logical condition modeling. If $\vec{c}$ satisfying $\alpha_{\vec{c}} = 0$ represents impossible input-output difference, i.e., $\vec{c} = (\Delta x \| \Delta y)$, the equation above corresponds to the linear inequality of the logical condition modeling required to remove the impossible propagation $(\Delta x \| \Delta y)$.

To better understand this representation, we show a simple example using the 3-bit S-box presented in Table 1 and whose $*$-DDT is given in Table 2.

Table 1: An example of 3-bit S-box

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
$S[x]$	0x5	0x3	0x4	0x6	0x2	0x7	0x0	0x1

Table 2: The $*$-DDT of the 3-bit S-box given in Table 1

Input Difference (Δx)	Output Difference (Δy)							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
0x0	1	0	0	0	0	0	0	0
0x1	0	1	1	0	0	1	1	0
0x2	0	1	1	0	0	1	1	0
0x3	0	0	0	1	0	0	0	1
0x4	0	0	0	0	1	0	0	1
0x5	0	1	1	0	0	1	1	0
0x6	0	1	1	0	0	1	1	0
0x7	0	0	0	1	1	0	0	0

According to the $*$-DDT, we construct a Boolean function from 6 bits to 1 bit, where $\Delta x \| \Delta y$ is the input and the output is 1 only if the input-output differential pattern is possible. Then the Boolean function is represented using the product-of-sum representation as:

$$\begin{aligned} f(\vec{x},\vec{y}) &= \bigwedge_{\vec{c}\in\{0,1\}^{2n}} \left(\bigvee_{i=1}^{n}(x_i \oplus c_i) \vee \bigvee_{i=1}^{n}(y_i \oplus c_{n+i}) \right) \\ &= \left(x_3 \vee x_2 \vee x_1 \vee y_3 \vee y_2 \vee \overline{y_1}\right) \wedge \left(x_3 \vee x_2 \vee x_1 \vee y_3 \vee \overline{y_2} \vee y_1\right) \\ &\quad \wedge \cdots \wedge \left(\overline{x_3} \vee \overline{x_2} \vee \overline{x_1} \vee \overline{y_3} \vee \overline{y_2} \vee \overline{y_1}\right), \end{aligned}$$

where $\overline{x_i}$ and $\overline{y_i}$ denote the negation of x_i and y_i, respectively, i.e., $\overline{x_i} = x_i \oplus 1$ and $\overline{y_i} = y_i \oplus 1$. Every term connected by the logical OR operation corresponds to the impossible propagations of the $*$-DDT, e.g., the first term $\left(x_3 \vee x_2 \vee x_1 \vee y_3 \vee y_2 \vee \overline{y_1}\right)$ corresponds to the impossible propagation 0x0 $\rightarrow$ 0x1. Then, for $f(\vec{x},\vec{y})$ to be 1, $\vec{x}$ and

[‡]This representation is equivalent to the Conjunctive Normal Form (CNF) of Boolean functions.

$\vec{y}$ must avoid making even a single term to be 0, or must avoid hitting any impossible propagation pattern. Therefore the number of terms corresponds to the number of entries with '0' in $*$-DDT.

3.2 Algorithm for Simplification of Product-of-Sum Representation

We now want to compute the minimum set of linear inequalities on the logical condition modeling. As explained above, the modeling corresponds to the product-of-sum representation of Boolean functions, and the number of required inequalities corresponds to the number of terms in the product-of-sum representation. This minimization problem has been well studied in various venues of computer science, and it is well-known that we can solve this problem by the Quine-McCluskey (QM) algorithm [Qui52, Qui55, McC56]. Unfortunately, since this problem is NP-hard, the algorithm requires exponential time to solve it. When we aim to obtain the optimized solution, the complexity of the QM algorithm is $O(3^{2n}\ln(2n))$[§]. More practically, heuristic algorithms like the *Espresso algorithm* [BSVMH84] are useful, and there are off-the-shelf softwares to implement this algorithm, e.g., Logic Friday [Log].

The same Boolean function shown in Sect. 3.1 can be rewritten as:

$$\begin{aligned} f(\vec{x},\vec{y}) = {} & (x_2 \vee x_1 \vee y_2 \vee \overline{y_1}) \wedge (x_2 \vee x_1 \vee \overline{y_2} \vee y_1) \wedge (x_2 \vee \overline{x_1} \vee y_2 \vee y_1) \wedge \\ & (\overline{x_2} \vee x_1 \vee y_2 \vee y_1) \wedge (x_2 \vee \overline{x_1} \vee \overline{y_2} \vee \overline{y_1}) \wedge (\overline{x_2} \vee x_1 \vee \overline{y_2} \vee \overline{y_1}) \wedge \\ & (\overline{x_2} \vee \overline{x_1} \vee y_2 \vee \overline{y_1}) \wedge (\overline{x_2} \vee \overline{x_1} \vee y_3 \vee y_1) \wedge (x_3 \vee x_2 \vee x_1 \vee \overline{y_3}) \wedge \\ & (\overline{x_3} \vee \overline{x_2} \vee \overline{x_1} \vee \overline{y_3} \vee \overline{y_2}) \wedge (x_2 \vee x_1 \vee y_3 \vee \overline{y_1}) \wedge \\ & (x_3 \vee \overline{x_2} \vee \overline{x_1} \vee y_1) \wedge (\overline{x_3} \vee y_3 \vee y_2 \vee y_1), \end{aligned}$$

where the number of terms is minimized by using the QM algorithm. Since every term can be converted into one linear inequality, the minimum number of linear inequalities to represent the $*$-DDT in the logical condition modeling is 13. Since the number of entries with '0' in $*$-DDT is 41, the QM algorithm reduces the number of inequalities from 41 to 13. For example, the first term of the equation above is converted into following linear inequality.

$$x_2 + x_1 + y_2 + (1 - y_1) \geq 1.$$

Table 3: Number of constraints to represent $*$-DDTs of AES and `SKINNY-128` S-boxes.

Structure	# non-zero entries	QM	Espresso
AES S-box	33150	-	8302
`SKINNY-128` S-box	54067	372	376

Table 3 shows the number of linear inequalities to represent $*$-DDT for AES and `SKINNY-128` S-boxes, where we used the Logic Friday for the Espresso algorithm and the QM algorithm is implemented from scratch. The QM algorithm could not bring results for the AES S-box within one day in our implementation. On the other hand, from the observation in [ST17a], minimizing the number of linear inequalities of each S-box does

[§]The QM algorithm consists of two parts: all terms that cannot be merged with other terms, called *prime implicants*, are first computed. Then we pick minimal set of prime implicants to represent the Boolean function accurately. The complexity $O(3^{2n}\ln(2n))$ is the number of prime implicants on the Boolean function from $2n$ bits to 1 bit. Therefore, about $3^8\ln(8) \approx 2^{15}$ and $3^{16}\ln(16) \approx 2^{28}$ prime implicants are required for 4-bit and 8-bit S-boxes, respectively. The time complexity for the second part highly depends on each instance. Also note that the memory size is an issue to apply the QM algorithm for larger S-boxes. $n = 9$ or 10 may be feasible but $n = 16$ is obviously infeasible.

not always make solving the entire problem fast. Therefore, we expect that the Espresso algorithm is enough for the application to the modeling of the MILP problem.

For the readers interested in testing the Espresso algorithm, we explain how to use the Logic Friday software in Appendix B.

4 New MILP Modeling to Optimize Probability of Differential Characteristics

Given a DDT whose entries are represented by binary elements as in a $*$-DDT, the set of linear inequalities is generated as shown in Sect.3. However, we cannot evaluate the exact probability of each active S-box only by such set of linear inequalities because of the binary representation. We propose a new method to model a DDT for the evaluation of the exact probability.

4.1 Modeling a DDT with Probability

In our new model, we separate the entries of a DDT of an S-box according to the values of their probabilities. When the DDT has entries with probability pb, we pick all such entries and construct a new DDT. In other words, we rewrite the DDT as a weighted sum of binary matrices.

Definition 1 (pb-DDT)**.** For a given S-box and its DDT, if the probability of entries in the DDT is pb, the corresponding entry of the pb-DDT is 1. Otherwise, entries of the pb-DDT are 0.

Assuming that there are two entries, 2^{-7} and 2^{-6}, in the DDT, we separate the DDT into two DDTs, i.e., 2^{-7}-DDT and 2^{-6}-DDT. Note that the deterministic transition of zero input difference to zero output difference is handled a bit differently. Instead of constructing 1-DDT, we assign a binary variable Q, where $Q = 1$ when the S-box is active and $Q = 0$ otherwise.

Then, we generate linear inequalities for every pb-DDT, where the algorithm explained in Sect. 3 is applied. Moreover, we assign a binary variable Q_{pb} for every pb-DDT. The corresponding linear inequalities for pb-DDT becomes effective only when $Q_{pb} = 1$, and otherwise these inequalities are ignored. Such modeling for MILP is well known as conditional (big-M) constraints (See [Bis17, Section 7.4]). Let $\langle \vec{u}_1, \vec{u}_2 \rangle$ be an inner-product of two vectors $\vec{u}_1, \vec{u}_2$. Let $\left\langle \vec{a}, (\vec{x}, \vec{y}) \right\rangle \geq b$ be linear inequalities, then the conditional constraints are represented as

$$\left\langle \vec{a}, (\vec{x}, \vec{y}) \right\rangle + M(1 - Q_{pb}) \geq b,$$

where M is a sufficiently big integer (in our case, $M = 2n$ is enough). Then $(\vec{x}, \vec{y})$ can take arbitrary value when $Q_{pb} = 0$, otherwise the original linear inequalities for pb-DDT are adopted.

Once all the conditional constraints for pb-DDT are generated, one additional linear inequality to constraint active pb-DDT is introduced as

$$Q = \sum_{pb} Q_{pb}.$$

In other words, if the S-box is active, the set of linear inequalities for only one pb-DDT is imposed. Then, the base-2 logarithm of the probability is evaluated as

$$\sum \log_2(pb) \times Q_{pb}.$$

Therefore, the objective function of just one S-box will take the form:

$$\text{Minimize: } \sum -\log_2(pb) \times Q_{pb}$$

Note that we can assign not only integers but also real numbers to represent $\log_2(pb)$ in the application of MILP. As a concrete example, the DDT of the `SKINNY-128` S-box has entries whose $\log_2(pb)$ is not integer, but rather, a real number.

Table 4: The DDT of the 3-bit S-box given in Table 1

Input Difference (Δx)	Output Difference (Δy)							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
0x0	1	0	0	0	0	0	0	0
0x1	0	2^{-2}	2^{-2}	0	0	2^{-2}	2^{-2}	0
0x2	0	2^{-2}	2^{-2}	0	0	2^{-2}	2^{-2}	0
0x3	0	0	0	2^{-1}	0	0	0	2^{-1}
0x4	0	0	0	0	2^{-1}	0	0	2^{-1}
0x5	0	2^{-2}	2^{-2}	0	0	2^{-2}	2^{-2}	0
0x6	0	2^{-2}	2^{-2}	0	0	2^{-2}	2^{-2}	0
0x7	0	0	0	2^{-1}	2^{-1}	0	0	0

Table 5: Left table : 2^{-1}-DDT. Right table : 2^{-2}-DDT.

Δx	Δy							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
0x0	0	0	0	0	0	0	0	0
0x1	0	0	0	0	0	0	0	0
0x2	0	0	0	0	0	0	0	0
0x3	0	0	0	1	0	0	0	1
0x4	0	0	0	0	1	0	0	1
0x5	0	0	0	0	0	0	0	0
0x6	0	0	0	0	0	0	0	0
0x7	0	0	0	1	1	0	0	0

Δx	Δy							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
0x0	0	0	0	0	0	0	0	0
0x1	0	1	1	0	0	1	1	0
0x2	0	1	1	0	0	1	1	0
0x3	0	0	0	0	0	0	0	0
0x4	0	0	0	0	0	0	0	0
0x5	0	1	1	0	0	1	1	0
0x6	0	1	1	0	0	1	1	0
0x7	0	0	0	0	0	0	0	0

We explain these procedures by using the same example provided in Sect. 3. Table 4 shows the DDT, and Table 5 shows the 2^{-2}-DDT and 2^{-1}-DDT. After applying the algorithm shown in Sect. 3, we obtain

$$\begin{aligned} f_1(\vec{x},\vec{y}) =&(\overline{y_2} \vee y_1) \wedge (\overline{x_2} \vee x_1) \wedge (x_2 \vee \overline{x_1}) \wedge (y_2 \vee \overline{y_1}) \wedge (y_3 \vee y_1) \\ &(x_3 \vee y_1) \wedge (x_1 \vee y_3) \wedge (x_3 \vee x_1) \wedge (\overline{x_3} \vee \overline{x_1} \vee \overline{y_3} \vee \overline{y_1}), \end{aligned}$$

for 2^{-1}-DDT and

$$f_2(\vec{x},\vec{y}) =(\overline{y_2} \vee \overline{y_1}) \wedge (\overline{x_2} \vee \overline{x_1}) \wedge (x_2 \vee x_1) \wedge (y_2 \vee y_1),$$

for 2^{-2}-DDT. Therefore, 9 and 4 linear inequalities are used to represent 2^{-1}-DDT and 2^{-2}-DDT, respectively. Let Q, $Q_{2^{-2}}$, and $Q_{2^{-1}}$ be binary variables, where $Q = 1$ only when the S-box is active, and Q_{pb} is used for the conditional constraint of pb-DDT. Then, the additional constraint:

$$Q_{2^{-2}} + Q_{2^{-1}} = Q$$

is introduced, and the objective function of just one S-box will take the form:

$$\text{Minimize: } Q_{2^{-1}} + 2 \times Q_{2^{-2}}$$

Table 6: Number of constraints to represent *pb*-DDTs of AES and SKINNY-128 S-boxes.

Structure		# non-zero entries	QM	Espresso
AES S-box	2^{-7}	33406	-	8241
	2^{-6}	65281	-	350
SKINNY-128 S-box	2^{-7}	62848	206	208
	2^{-6}	60530	275	283
	$2^{-5.4}$	65472	33	34
	2^{-5}	62708	234	239
	$2^{-4.4}$	65458	42	52
	2^{-4}	64884	147	159
	$2^{-3.7}$	65534	15	15
	$2^{-3.4}$	65518	24	28
	$2^{-3.2}$	65534	15	15
	2^{-3}	65435	62	67
	$2^{-2.7}$	65534	16	16
	$2^{-2.4}$	65532	17	17
	2^{-2}	65513	37	40

4.2 Results for AES and SKINNY-128 S-boxes

We have generated the linear inequalities for the DDT of AES and SKINNY-128 S-boxes. While, the DDT of the AES S-box has two distinct probability values, the DDT of the SKINNY-128 S-box has 13 different probability values.

Table 6 shows the number of linear constraints to represent *pb*-DDTs of AES and SKINNY-128 S-boxes. The number of terms in the naive product-of-sum representation corresponds to the number of non-zero entries in *pb*-DDT. We then applied both the QM and the Espresso algorithms and constructed minimum representation of the product-of-sum representation. Unfortunately, as mentioned above, the QM algorithm, which is an exact algorithm for the minimization could not yield results for the AES S-box because of its complexity. However, as described in Sect. 3, we expect that the optimized output of the heuristic Espresso algorithm is enough for the application to the modeling of the MILP problem. To apply our new modeling to SKINNY-128 and AES, we used the Espresso algorithm to generate the set of linear inequalities the differential propagation through their 8-bit S-boxes.

5 Application to SKINNY-128

5.1 Specification of SKINNY-128

SKINNY [BJK+16] is a family of lightweight tweakable block ciphers designed by Beierle et al. at CRYPTO 2016. Users can choose the block size $n \in \{64, 128\}$ and the tweakey size $t \in \{n, 2n, 3n\}$, where tweakey is a combination of tweak and key [JNP14]. The 64-bit block versions adopt a nibble-oriented SPN structure and is called SKINNY-64 while the 128-bit block versions adopt a byte-oriented SPN structure and is called SKINNY-128. As we are interested in modeling large S-boxes, our target is SKINNY-128 and the analysis is applicable to any tweakey size.

An internal 128-bit state of SKINNY-128 is represented by a 4×4-byte array. A 128-bit plaintext is first loaded in the state, and then the round function is applied N_r times, where N_r is 40, 48 and 56 for 128-, 256- and 384-bit tweakey, respectively. Unlike AES, there is no whitening key at the beginning, and the last round transformation is the same as the other rounds.

The round function of SKINNY-128 consists of five operations: SubCells, AddConstants, AddRoundTweakey, ShiftRows and MixColumns which are explained below.

SubCells. An 8-bit S-box whose maximum differential probability is 2^{-2} is applied to all bytes.

AddConstants. A seven-bit constant updated by LFSR in every round is added to three bytes of the state. Adding constants have negligible impact in differential cryptanalysis, and we ignore this operation for the remaining explanation.

AddRoundTweakey. A 64-bit value is extracted from n-, $2n$- or $3n$-bit tweakey state, and is XORed to the upper half of the state. To measure the probability of differential characteristics in the single-key setting, adding subtweakeys has the same impact as adding constants, and we omit details of the tweakey scheduling algorithm.

ShiftRows. Each byte in Row j is rotated to the right (opposite to AES) by j bytes.

MixColumns. Four bytes in each column are multiplied by the binary matrix $\mathcal{M}$ that is defined and illustrated in Figure 1.

$$\mathcal{M} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

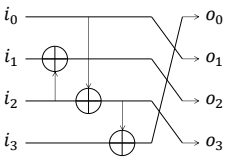

Figure 1: Matrix for MixColumns

The designers of SKINNY evaluated a lower bound on the number of active S-boxes, N_A, for each round by using MILP with word-wise modeling. The lower bounds in the single-key setting are listed in Table 7 in the row denoted by "LB (word)[BJK+16]". Because the maximum differential probability for each S-box is 2^{-2}, upper bounds on the probability of differential characteristics can be computed as $2^{-2 \cdot N_A}$. Considering that the block size is 128 bits, ensuring at least 64 active S-boxes is sufficient to show the resistance against the basic differential cryptanalysis. From Table 7, 15 rounds of SKINNY-128 resist differential cryptanalysis.

Table 7: Bounds on the number of active S-boxes in the single-key setting.

rounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LB (word) [BJK+16]	1	2	5	8	12	16	26	36	41	46	51	55	58	61	66
simple UB (bit)	1	2	5	8	12	16	26	36	43	48	52	56	62	68	-

"LB" denotes lower bound and "UB" denotes upper bound.

5.2 Simple Lower Bounds on Probability with Word-wise Search

The lower bounds in Table 7 were generated by the word-wise modeling. These are tight under the assumption that any non-zero input difference can generate any non-zero output difference through the S-box. However, if the properties of the DDT are considered, tightness is no longer ensured. In order to take into account the properties of the DDT, bit-wise modeling is required. It was not known how to do create such model for 8-bit S-boxes at the time when SKINNY-128 was designed. This motivates us to shift from word-wise modeling to bit-wise modeling and to derive tight bounds with bit-wise modeling.

Thanks to the new modeling in the previous sections, bit-wise modeling for SKINNY-128's 8-bit S-box is feasible, it still takes long time, though. To efficiently search for the tight bounds, our strategy first generates simple upper bounds, which can be searched with word-wise modeling but guaranteed to be valid even if the properties of the DDT are considered.

Such upper bounds can be searched by slightly modifying the word-wise modeling for lower bounds by the designers [BJK+16]. In details, the search for lower bounds allows XOR of two active bytes to be either active or inactive.[¶] We restrict this transition such that the XOR of two active bytes always cancel each other. In short, this strategy assumes that differences of all the bytes in a certain state are identical, and they propagate to another difference through the S-box with the highest probability. Then in a subsequent XOR, differences from the two active bytes cancel each other.

Such a characteristic always exists if an input difference Δ to the S-box remains unchanged with the highest probability after the S-box. Unfortunately, such a fixed point with respect to the difference with probability 2^{-2} does not exist in SKINNY-128. Table 8 lists the number of pairs of input and output differences for each probability of differential propagation. Among the 65536 entries in the DDT of SKINNY-128, there are only 23 pairs of input and output differences that achieve 2^{-2} probability and none of these 23 entries have an identical input and output differences.

Table 8: Distribution of non-trivial probabilities in DDT of SKINNY-128's S-box.

probability	2^{-7}	2^{-6}	$2^{-5.4}$	2^{-5}	$2^{-4.4}$	2^{-4}	$2^{-3.7}$	$2^{-3.4}$	$2^{-3.2}$	2^{-3}	$2^{-2.7}$	$2^{-2.4}$	2^{-2}
DDT value	2	4	6	8	12	16	20	24	28	32	40	48	64
# of entries	2688	5006	64	2828	78	652	2	18	2	101	2	4	23

Table 8 shows that most of the valid differential propagations have a relatively low probability. Only 0.2% of the pairs of input and output differences can be propagated with probability larger than 2^{-4}. This, at a short glance, implies that the security of SKINNY-128's S-box is not bad even though the maximum differential probability is 2^{-2}. Nevertheless, we can ensure that the above simple approach derives valid upper bounds.

We list the 23 pairs of input and output differences that can be propagated with probability 2^{-2}. (Δ_i, Δ_o) denotes that the input difference is Δ_i and the output difference is Δ_o and the values are written in hexadecimal numbers.

```
(01,20),(02,08),(02,09),(04,01),(05,01),(08,10),(09,10),(0A,10),
(10,40),(10,50),(20,80),(20,90),(21,20),(30,40),(30,50),(40,04),
(50,04),(80,02),(80,03),(90,02),(90,03),(C0,04),(D0,04)
```

Using these pairs, we can generate the following 8-round iterative differential propagations that only consist of propagations with probability 2^{-2}.

$$01 \xrightarrow{S} 20 \xrightarrow{S} 80 \text{ or } 90 \xrightarrow{S} 02 \xrightarrow{S} 08 \text{ or } 09 \xrightarrow{S} 10 \xrightarrow{S} 40 \text{ or } 50 \xrightarrow{S} 04 \xrightarrow{S} 01$$

Therefore, by setting the differences of all bytes in Round 1 to 0x01, Round 2 to 0x20, Round 3 to 0x80, etc, we can always pass all active S-boxes in each round with probability 2^{-2}.[‖]

The results of searching for simple upper bounds are listed in Table 7 in the row denoted "simple UB (bit)." Up to 8 rounds, the lower bound generated by the designers

[¶] Here we assume that the MixColumns operation is modeled as a sequence of three XORs.

[‖] Obviously, multiple characteristics can be exploited to increase the probability of the 8-round iteration. We can gain the advantage by a factor of 2^3 every 8 rounds. Because it is cryptographers' convention to evaluate the probability of the single best characteristic for measuring the resistance against differential cryptanalysis, we do not discuss the impact of multiple characteristics in this paper.

is tight, which is $2^{-2 \cdot N_A}$. A gap arises from 9 rounds. The maximum probability (the worst case for the designers) is $2^{-2 \cdot \mathrm{LB}}$ while the minimum probability (the best case for the designers) is $2^{-2 \cdot \mathrm{UB}}$, which are summarized in Table 9. In particular, the probability of the best differential characteristic for 14 rounds is of a great interest. In the remaining of this section, we evaluate the probability of the best differential characteristic for 9 to 14 rounds based on the model presented in Sect. 4 with various optimization techniques for `SKINNY-128`.

Table 9: Range of the probability of the best differential characteristic.

Rounds	1 - 8	9	10	11	12	13	14	15
Maximum	$2^{-2 \cdot N_A}$	2^{-82}	2^{-92}	2^{-102}	2^{-110}	2^{-116}	2^{-122}	2^{-132}
Minimum	$2^{-2 \cdot N_A}$	2^{-86}	2^{-96}	2^{-104}	2^{-112}	2^{-124}	2^{-132}	-

5.3 Searching for the Differential Characteristic with the Best Probability

The new model explained in Sect. 4 is for modeling a single S-box. We adopt two-stage search introduced by Sun et al. [SGL+17] for constraint programming, which runs as follows.**

Stage 1. The threshold of the number of active S-boxes is firstly defined, and all the truncated differential characteristics with active S-boxes less than a prespecified threshold are searched efficiently by using MILP with the byte-wise model. At this stage, it is unclear whether each truncated differential characteristic can be instantiated with actual differences.

Note that all truncated differential characteristics with a given number of active S-boxes can be generated by using the method in [SHW+14a]. In short, each time we detect a truncated differential characteristic, we add a constraint to the system so that the detected truncated differential characteristic can be excluded from the solution space.

Stage 2. With information about the active byte positions, the maximum probability of each truncated differential characteristic is searched by using MILP with the bit-wise model. As we know the upper bound, we can add another constraint such that the probability of the characteristic must be higher than the upper bound.

5.3.1 Optimization techniques for `SKINNY-128`

As we explain later, a simple application of the two-stage search in [SGL+17] to `SKINNY-128` cannot finish searching for bounds because of the too expensive computational cost. Here, we introduce two techniques to reduce the computational time, which are particularly useful for `SKINNY-128`.

Cutting-off low probability transitions. We observe that the gap between the lower bound and the upper bound is not very big for some rounds. According to Table 7, the gap is only 1 active S-box, probability of 2^{-2}, for 11 rounds and 12 rounds. In such a case, we can regard the DDT entries with low probabilities as impossible, which makes the system of constraint inequalities smaller.

** Without the two-stage search, obtaining the best probability for 10 rounds is already infeasible.

For example, for 11 rounds, the simple upper bound has probability 2^{-104}, while the lower bound on the number of active S-boxes is 51 in which the best possible probability is 2^{-102}. Hence, we are interested in searching for differential characteristics with 51 active S-boxes whose probability is higher than 2^{-104}. Recall that the DDT of the SKINNY-128's S-box has 13 different non-trivial probabilities which range from 2^{-2} to 2^{-7} as shown in Table 8. If 1 active S-box requires differential transition with a probability 2^{-4} or smaller, the entire characteristic cannot be better than the simple upper bound. Thus, we can regard the DDT entries with probability 2^{-4} to 2^{-7} as impossible. As mentioned before, the DDT entries with relatively high probabilities is sparse. Thus, this optimization saves a lot of inequalities in the system. Table 6 shows the exact number of constraints to represent *pb*-DDT. By using the Espresso algorithm, the total number of inequalities for all the 2^{-7}-DDT to 2^{-2}-DDT is 1173 per S-box, and 83% of the linear inequalities is for the 2^{-7}-DDT to 2^{-4}-DDT. This means that this technique reduces the number of inequalities to 17%. Note that the inequalities generated by the QM algorithm can also be used. However as explained in Sect 4.2, we have no clue whether minimizing the number of inequalities for each S-box would surely reduce the running time of the entire model.

Equivalence class of truncated differential characteristics. Suppose that there exists a truncated differential characteristic with a certain number of active S-boxes. Then, we can obtain 4 rotation-variants of this truncated differential characteristic due to the symmetric structure of the SKINNY's round function. Obviously, the best differential characteristics obtained from rotation-variants of truncated differential characteristics are rotation-variants. This reduces the computational cost to 1/4. Note that equivalence classes are available only for the single-key, because the tweakey schedule is not symmetric.

5.3.2 Evaluation of SKINNY-128

To avoid redundancy, we focus our attention on the results of 10, 13 and 14 rounds.

Tight bound for 10 Rounds. From Table 7 and Table 9, our goal is to search for truncated differential characteristics with 46 or 47 active S-boxes that can be instantiated with actual differences with probability higher than 2^{-96}. With the method from [SHW+14a], we found 8 truncated differential characteristics (2 equivalence classes) with 46 active S-boxes and 32 truncated differential characteristics (8 equivalence classes) with 47 active S-boxes.

8 classes with 47 active S-boxes can be tested by regarding the DDT entries with probability of 2^{-4} or smaller as impossible, and in this case the MILP solver finishes relatively fast. In our experiments, we used Gurobi Optimizer [Inc15] with Xeon Processor E5-2699 (18 cores) in 128 GB RAM. In this environment, we needed about 90 seconds to test each class and none of them can be instantiated with actual differences with probability higher than 2^{-96}.

To test the 2 classes with 46 active S-boxes, we can only regard the DDT entries with 2^{-7} as impossible. Hence it requires more computational time than the case with 47 active S-boxes. For example, we needed about 5,000 seconds to test each truncated differential characteristic. As a result, none of them can be instantiated with probability higher than 2^{-96}.

Finally, it is proven that the probability of the best differential characteristic for 10-round SKINNY-128 is 2^{-96}, which is achieved by the simple upper bound.

Tight bound for 13 Rounds. In this case, the gap between the lower bound and the upper bound is large. From Table 7 and Table 9, our goal is searching for truncated differential characteristics with 58, 59, 60 or 61 active S-boxes that can be instantiated

with actual differences with probability higher than 2^{-124}. After the first stage of the search, we found 6 classes of truncated differential characteristics with 58 active S-boxes, no truncated differential characteristics with 59 active S-boxes, 8 classes with 60 active S-boxes, and 4 classes with 61 active S-boxes.

Thanks to the technique of cutting-off low probability transitions, truncated differential characteristics with 60 and 61 active S-boxes could be tested in a reasonable time. We confirmed that none of them can be instantiated with a probability higher than 2^{-124}.

Regarding the 6 classes with 58 active S-boxes, cutting-off low probability transitions can no longer be applied due to the big gap. Indeed, the best possible probability is 2^{-116}. Even if one active S-box takes the worst transition with probability 2^{-7}, the probability is 2^{-121} which is still higher than 2^{-124}. We then simply applied the model in Section 4 and accepted the relatively long computational time. In our environment, the test of 6 classes finished in 16 days. We found that 5 classes cannot be instantiated with probability higher than 2^{-124}, while 1 class has a differential characteristic with probability 2^{-123} that consists of 51 active S-boxes propagating with probability 2^{-2} and 7 active S-boxes propagating with probability 2^{-3}. The discovered characteristic is shown in Table 11 in Appendix D. Finally, it is proven that the tight bound on the probability of differential characteristic for 13 rounds is 2^{-123}.

Lower bound for 14 Rounds. The gap between lower bound and upper bound is even larger than in the 13-round case. Given that the search on 13 rounds took 16 days, we decided not to compute the tight bound for 14 rounds, but to concentrate on whether or not the upper bound on the probability of the characteristic is higher than 2^{-128}. Without using MILP and by combining the results of the 13-round search and the property of the extended round, we prove that there does not exist any 14-round differential characteristic with probability higher than 2^{-128}.

The minimum number of active S-boxes for 14-round is 61. We are only interested in truncated differential characteristics with less than 64 active S-boxes. Hence, the number of active S-boxes can be 61, 62 or 63. We searched for all the 14-round truncated differential characteristics up to 63 active S-boxes, and found 2 classes with 61 active S-boxes, 4 classes with 62 active S-boxes and no solution with 63 active S-boxes.

We then manually verified that all of these 6 classes are 1-round extension of the truncated differential characteristics for 13 rounds with 58 active S-boxes (58 active S-boxes for 13 rounds and 3 or 4 active S-boxes for the extended round). Recall that the probability of the best differential characteristic for 13 rounds is proven above to be 2^{-123}. If the number of active S-boxes increases by 3, the upper bound on the probability becomes $2^{-123-2\times 3} = 2^{-129}$. Hence, there does not exist any 14-round differential characteristic achieving probability higher than 2^{-128}, which shows that `SKINNY-128` is secure against differential cryptanalysis with 1-round less than the designers' original expectation.

Remarks on Approximated Probabilities. One of the anonymous reviewers pointed out that our model rounds the probability described as $\log_2 pb$ at one decimal place, and this may overlook some good differential characteristics. For example, our model approximates the probability $40/256 \approx 2^{-2.678}$ as $2^{-2.7}$, so clearly our model underestimates the power of this differential transition. The same applies to $28/256 \approx 2^{-3.193}$ and $20/256 \approx 2^{-3.678}$ that are evaluated in our model as $2^{-3.2}$ and $2^{-3.7}$, respectively. This is a valid concern and we have the following two remarks about it:

- First, the $\log_2 pb$ values appear only as coefficients in the objective function and we can increase the precision as much as we want.
- Second, this issue does not affect our conclusion about the non-existence of any 14-round differential characteristic whose probability is higher than 2^{-128}. The proof

relies on the non-existence of any 13-round differential characteristic with probability higher than 2^{-122}. The gap in our 13-round search caused by this rounding issue is maximized when the number of differential transition with $2^{-2.678}$ ($2^{-2.7}$ in our model) is maximized. To be higher than 2^{-122} with 58 S-boxes, this transition can occur at most 8 times, if it occurs for 9 times then the probability will be $2^{-2.678\times 9-2\times 49} < 2^{-122}$. This means that the maximum gap is $2^{0.022\times 8} = 2^{0.176}$. Therefore, our result, 2^{-123} by the approximated probabilities, ensures that the maximum probability for the 13 rounds is less than $2^{-122.82}$.

6 Application to AES-Based Constructions

In their FSE 2016 paper, Jean and Nikolić have investigated efficient AES-based constructions without impacting their security [JN16]. Their construction inherits the core design of AEGIS [WP16] and Tiaoxin-346 [Nik16], in which the internal state consists of parallel s 128-bit states, and during 1 step, AES-round function is applied to a 128-bit states then xored to neighboring states, m 128-bit message blocks are xored to m states, and x additional xors between 128-bit states are introduced. To make sure that their schemes have high efficiency, they have carefully studied and subsequently chosen the internal state size and the number of AES-rounds to be computed per step and which state blocks should the AES round function be applied to.

On the other hand, the security of the construction is measured by counting the number of active S-boxes to generate an internal collision: a differential characteristic starting and ending with zero internal-state difference after introducing some difference from the message blocks. Considering that the key size is 128 bits and the maximum differential probability of the AES S-box is 2^{-6}, a lower bound on the number of active S-boxes is 22.

To search for secure constructions whose best differential has a minimum of 22 active S-boxes, Jean and Nikolić have used MILP. They have shown seven secure constructions that provide good trade-offs between state size and efficiency (see Figures 5-11 in [JN16]). The state size of these constructions ranges from 6-9 and 12 128-bit blocks. Two constructions process three message blocks per step, i.e., an iteration of their designs, while the others process two message blocks per step. In all these constructions, the AES round function is applied, if any, once per step. This was found and proven to be more efficient than cascading r AES round function calls.

However, Jean and Nikolić have noted the limitation of MILP-based search as follows:

> ***Limitations.*** *"MILP only yields upper bounds on the actual probabilities of the differential characteristics as, theoretically, they can be impossible. ⋯ the fact that partially undetermined behavior of the XOR operation (mentioned before) may result in inconsistent systems that produce truncated differential characteristics which are impossible to instantiate with actual differences."*

This has motivated us to attempt instantiate two of their constructions using our new model.

6.1 Results on the Fifth Construction (C5)

C5 processes two message blocks per step with an internal state of seven 128-bit blocks and the AES round function call is applied to five out of the seven blocks. The designers reported that the minimum number of active S-boxes for this construction is 22 but without stating the number of steps needed to reach it. We have found that there are only 4 truncated differential characteristics that achieve the minimum number of active S-boxes after 4 steps. We checked up to 8 steps and the number of truncated differential characteristics remained constant. All these differentials were found to be infeasible. One

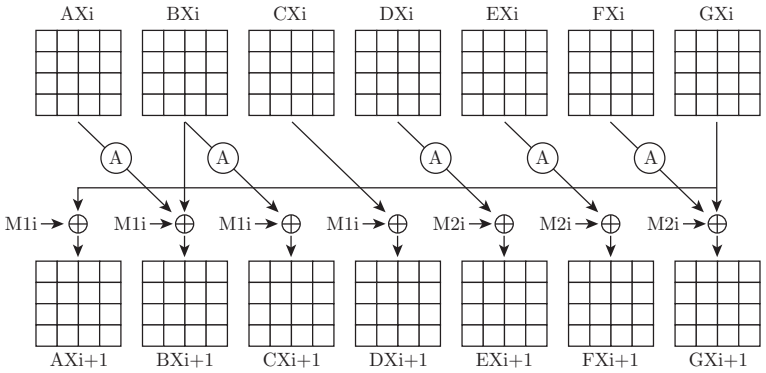

Figure 2: C5 construction. 'A' represents AES round function.

of which is depicted in Figure 3. Verifying it manually, for $BX4$ and $DX4$ to be of zero difference, bytes 12-15 of $BX3$, $CX3$, and $M1\_3$ must be equal. As bytes 12-15 in $BZ2$ are zero, this means that these bytes also equal bytes 12-15 in $M1\_2$, i.e., bytes 12-15 in $BX3$, $CX3$, $M1\_2$, and $M1\_3$ are equal. As $AZ2$ is all zeros, this means that bytes $BX2$ must be inactive which contradicts the truncated differential characteristic. Similar reasoning applies to the other 3 truncated differential characteristics with 22 active S-boxes.

Then, we increased the number of active S-boxes and found 580 truncated differential characteristics that have 23 active S-boxes. Again, we verified that with higher number of steps, the number of truncated differential characteristics with 23 active S-boxes does not change. All these truncated differential characteristics were found to be infeasible by the solver. One of these differentials is depicted in Figure 7. We found that it is invalid because of the reasoning mentioned above, i.e., for $BX4$ and $DX4$ to be of zero difference, bytes 13-15 of $BX2$ should be inactive contradicting the differential characteristic. Moreover, for $GX4$ to be of no difference, it requires that $MC[0, a, 0, 0]$ from $FZ2$ to be equal to $MC[b, 0, 0, 0]$ from $FZ3$ and this is not possible because there are no non-zero differences at different positions that would yield the same output after the MC operation.

This concludes that the lower bound on the number of active S-boxes for C5 is 24, which improves 22 as early estimated by the designers.

6.2 Results on the First Construction (C1)

C1 processes two message blocks per step with an internal state of six 128-bit blocks and the AES round function call is applied to all these 6 blocks. The minimum number of active S-boxes reported by the designers is 22 and the number of steps was not stated. Therefore, our first step was to find the truncated differential characteristics that achieve that minimum number of active S-boxes and in how many steps. We have found that there are 256 truncated differential characteristics that achieve that minimum number of active S-boxes after 3 steps. On the chance that there might be other truncated differential characteristics that have 22 active S-boxes but for larger number of steps. We ran the MILP model for 4, 5, 6 and 7 steps and the number of differential characteristics did not change.

These 256 differential characteristics are formed by having one active byte in the first block of both messages. We have applied our method on all of these differential characteristics and 252 were found by the solver to be infeasible. Figure 8 depicts one of these invalid differential characteristics. We have verified it manually and found out that this characteristic is indeed invalid. As shown in Figure 8, for $AX3$ to be of zero difference, this means that $FZ2$ must equal $AX2$ and as $AX1$ is all zeros, $AX2$ equals $FZ1$. So, for $AX3$ to be of zero difference $FZ1 = MC[0, a, 0, 0]$ must equal $FZ2 = MC[b, 0, 0, 0]$ and

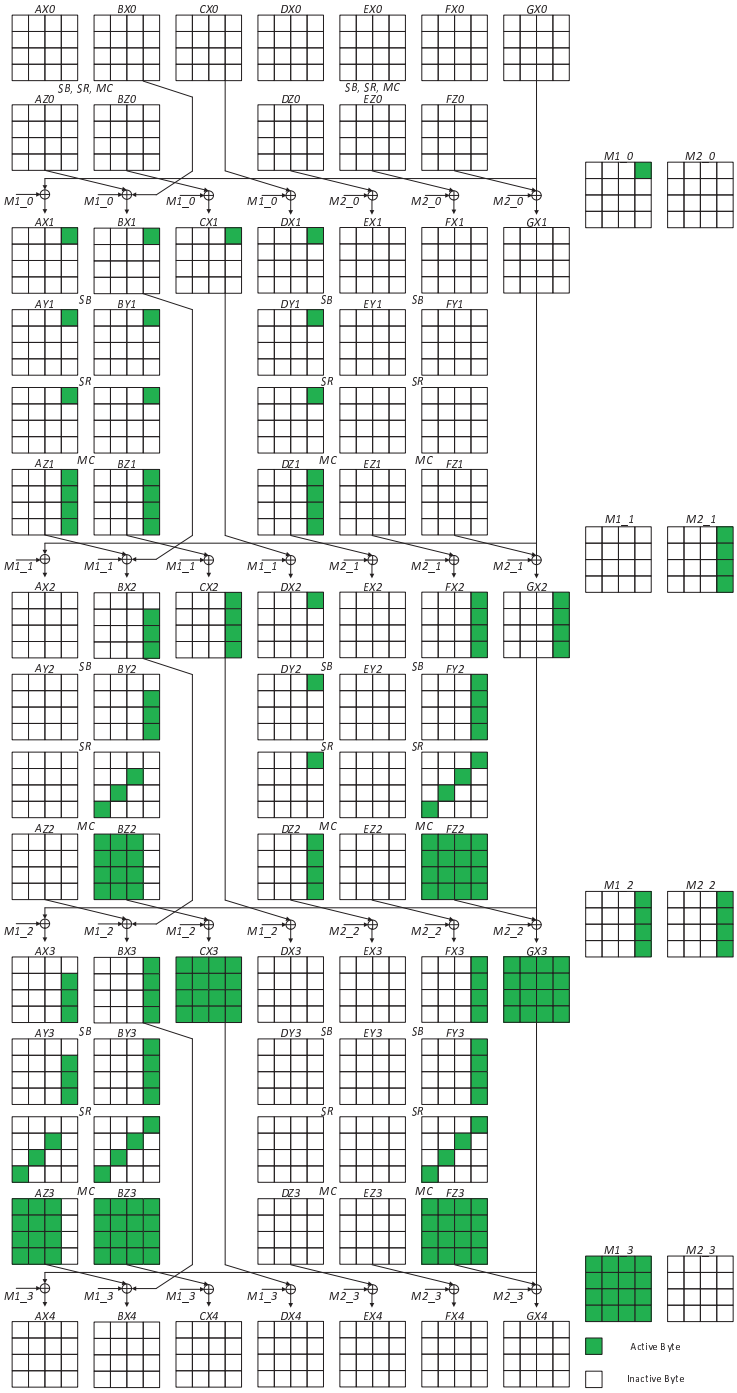

Figure 3: Invalid differential characteristic for C5 with 22 active S-boxes

this is not possible as explained above. Following the same logic, the truncated differential characteristic can be valid if and only if the active bytes are at the same position and that position is not impacted by the shift row operation along the different steps, i.e., the active bytes of the first block of the two messages are at positions (0,0), (4,4), (8,8) and (12,12) and these are the 4 truncated differential characteristics that the solver found to

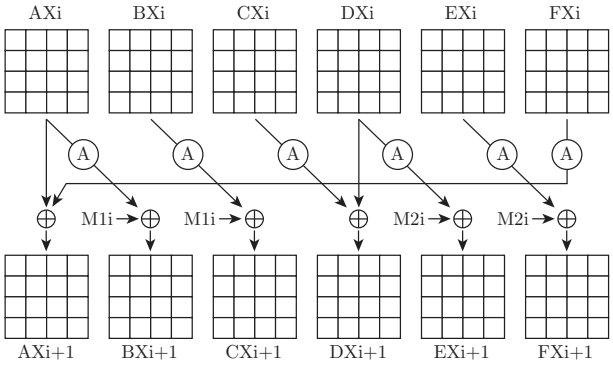

Figure 4: C1 construction

be feasible.

As the S-box is the same for every byte of the state and as the MC input will always have the same format, the probability of these 4 differentials will be the same and the actual differences will just be a permutation of each other. We have tried to instantiate one of these 4 differential characteristics and it was found that the best probability is 2^{-134} instead of 2^{-132} which means that there are two S-boxes that cannot be bypassed by the maximum probability of 2^{-6}. These are the S-boxes at byte 8 of $AX2$ and $EX2$ (highlighted in yellow in Figure 5 and the actual difference values are listed in Table 12 in Appendix E.). Looking for the reason, we have found that for the path to be valid, the input of the S-boxes at byte 8 of $AX2$ and $BX2$ (resp. $EX2$ and $FX2$) must be different and the output must be the same, i.e., $S(a) = S(b)$ where are a, b are two distinct non-zero differences. We have searched through the DDT of the AES S-box and found that the maximum probability that would fulfill this condition is 2^{-13} , one S-box is activated with 2^{-6} while the other is activated with 2^{-7}. This means that the probability found by the MILP solver is the highest probability. Therefore, we conclude that the best differential characteristic of this construction has a probability of 2^{-134}.

7 Concluding Remarks

In this paper, we presented a new S-box modeling that can represent a truncated version of the DDT of large S-boxes and can handle the probability of differential characteristics. As underlying ideas, we focused on the relationship between logical condition model and product-of-sum representation of Boolean functions, and introduced the Quine-McCluskey and the Espresso algorithms as a tool to generate constraint inequalities. We then separated the DDT for each probability, and used conditional constraints to deal with multiple tables.

With the proposed modeling, we first evaluated the upper bound on differential characteristics of `SKINNY-128`, and improved the number of rounds to resist simple differential distinguishers by 1 round. We then evaluated the upper bound on differential characteristics of two AES-round based constructions. We improved the lower number of active S-boxes for C5 construction. Lastly, the proposed techniques are quite general, thus they can be used to evaluate various designs. For example, the proposed techniques are expected to be extended to study the related-tweakey security of `SKINNY-128` in a straightforward manner. They can also be applied in other cryptanalysis techniques such as linear cryptanalysis.

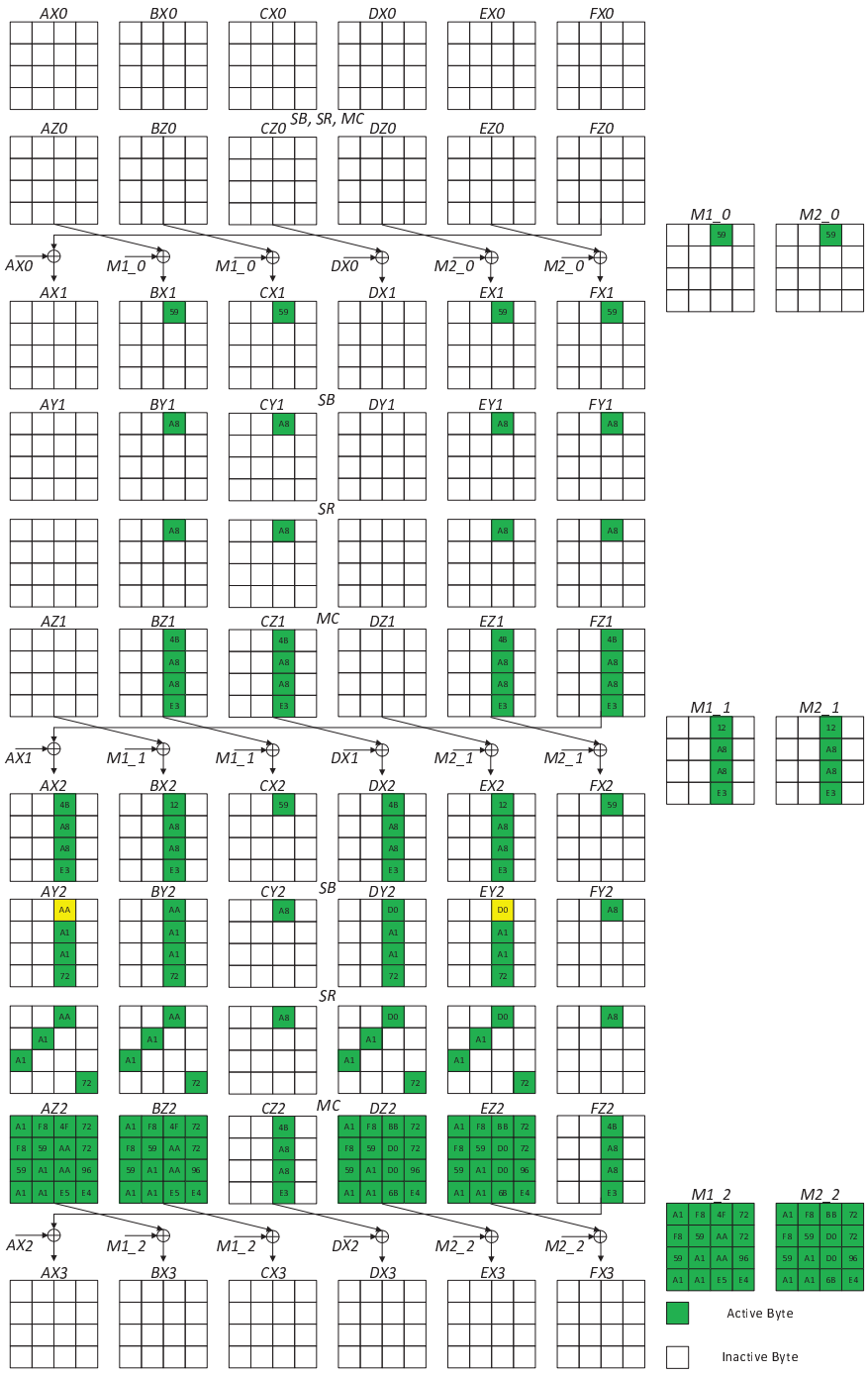

Figure 5: Valid differential characteristic for C1

References

[BAK98] Eli Biham, Ross J. Anderson, and Lars R. Knudsen. Serpent: A New Block Cipher Proposal. In Serge Vaudenay, editor, *Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998,*

Proceedings, volume 1372 of *Lecture Notes in Computer Science*, pages 222–238. Springer, 1998.

[Bis17] Johannes Bisschop. AIMMS - Optimization Modeling. https://download.aimms.com/aimms/download/manuals/AIMMS3_OM.pdf, 2017.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, 2007.

[BSVMH84] Robert King Brayton, Alberto L. Sangiovanni-Vincentelli, Curtis T. McMullen, and Gary D. Hachtel. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, 1984.

[CCZ98] Claude Carlet, Pascale Charpin, and Victor A. Zinoviev. Codes, bent functions and permutations suitable for des-like cryptosystems. *Des. Codes Cryptography*, 15(2):125–156, 1998.

[CHP+17] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. A Security Analysis of Deoxys and its Internal Tweakable Block Ciphers. *IACR Trans. Symmetric Cryptol.*, 2017(3):73–107, 2017.

[CJF+16] Tingting Cui, Keting Jia, Kai Fu, Shiyao Chen, and Meiqin Wang. New Automatic Search Tool for Impossible Differentials and Zero-Correlation Linear Approximations. *IACR Cryptology ePrint Archive*, 2016:689, 2016.

[FWG+16] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In Thomas Peyrin, editor, *FSE*, volume 9783 of *Lecture Notes in Computer Science*, pages 268–288. Springer, 2016.

[GFG+16] Gerald Gamrath, Tobias Fischer, Tristan Gally, Ambros M. Gleixner, Gregor Hendel, Thorsten Koch, Stephen J. Maher, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Sebastian Schenker, Robert Schwarz, Felipe Serrano, Yuji Shinano, Stefan Vigerske, Dieter Weninger, Michael Winkler, Jonas T. Witt, and Jakob Witzig. The SCIP Optimization Suite 3.2. Technical Report 15-60, ZIB, Takustr.7, 14195 Berlin, 2016.

[ILO16] IBM ILOG. IBM ILOG CPLEX Optimization Studio V12.7.0 documentation. Official webpage, https://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition/, 2016.

[Inc15] Gurobi Optimization Inc. Gurobi Optimizer 6.5. Official webpage, http://www.gurobi.com/, 2015.

[JN16] Jérémy Jean and Ivica Nikolic. Efficient Design Strategies Based on the AES Round Function. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016,*

Revised Selected Papers, volume 9783 of *Lecture Notes in Computer Science*, pages 334–353. Springer, 2016.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. 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 II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.

[Log] Logic friday. http://sontrak.com/.

[McC56] Edward J. McCluskey. Minimization of Boolean functions. *The Bell System Technical Journal*, 35(6):1417–1444, 1956.

[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, *Inscrypt 2011*, volume 7537 of *LNCS*, pages 57–76. Springer, 2011.

[Nat01] National Institute of Standards and Technology. *Federal Information Processing Standards Publication 197: Advanced Encryption Standard (AES)*. NIST, November 2001.

[Nik16] Ivica Nikolić. Tiaoxin-346 VERSION 2.1. Submitted to CAESAR., 2016.

[Qui52] Willard Van Orman Quine. The Problem of Simplifying Truth Functions. *The American Mathematical Monthly*, 59(8):521–531, 1952.

[Qui55] Willard Van Orman Quine. A Way to Simplify Truth Functions. *The American Mathematical Monthly*, 62(9):627–631, 1955.

[SGL+17] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of AES, SKINNY, and Others with Constraint Programming. *IACR Trans. Symmetric Cryptol.*, 2017(1):281–306, 2017.

[SHW+14a] Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu. 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, Report 2014/747, 2014. http://eprint.iacr.org/2014/747.

[SHW+14b] 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, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 158–178. Springer, 2014.

[ST16] Yu Sasaki and Yosuke Todo. New Differential Bounds and Division Property of Lilliput: Block Cipher with Extended Generalized Feistel Network. In Roberto Avanzi and Howard Heys, editors, *SAC 2016*, volume 10532 of *LNCS*, pages 264–283. Springer, 2016.

[ST17a] Yu Sasaki and Yosuke Todo. New Algorithm for Modeling S-box in MILP Based Differential and Division Trail Search. In Emil Simion and Pooya Farshim, editors, *SecITC 2017*, volume 10543 of *LNCS*, pages 150–165. Springer, 2017.

[ST17b] Yu Sasaki and Yosuke Todo. New Impossible Differential Search Tool from Design and Cryptanalysis Aspects - Revealing Structural Properties of Several Ciphers. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 185–215. Springer, 2017.

[SWLW16] Ling Sun, Wei Wang, Ru Liu, and Meiqin Wang. MILP-Aided Bit-Based Division Property for ARX-Based Block Cipher. *IACR Cryptology ePrint Archive*, 2016:1101, 2016.

[SWW16] Ling Sun, Wei Wang, and Meiqin Wang. MILP-Aided Bit-Based Division Property for Primitives with Non-Bit-Permutation Linear Layers. *IACR Cryptology ePrint Archive*, 2016:811, 2016.

[Tod15] Yosuke Todo. Structural Evaluation by Generalized Integral Property. 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 287–314. Springer, 2015.

[WP16] Hongjun Wu and Bart Preneel. AEGIS: A Fast Authenticated Encryption Algorithm (v1.1). Submitted to CAESAR., 2016.

[WW11] Shengbao Wu and Mingsheng Wang. Security Evaluation against Differential Cryptanalysis for Block Cipher Structures. *IACR Cryptology ePrint Archive*, 2011:551, 2011.

[XZBL16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 648–678. Springer, 2016.

[ZR17] Wenying Zhang and Vincent Rijmen. Division Cryptanalysis of Block Ciphers with a Binary Diffusion Layer. *IACR Cryptology ePrint Archive*, 2017:188, 2017.

A Sketch of Quine-McCluskey Algorithm

The Quine-McCluskey (QM) algorithm [Qui52, Qui55, McC56] derives the minimum product-of-sum representation of a given Boolean function from its truth table. Let f be a Boolean function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$, and $x = (x_0, x_1, x_2, \ldots, x_{n-1})$ denotes its input. When we want to evaluate the minimum product-of-sum representation of f, we focus on x such that $f(x) = 0$. Let $S^1 \subseteq \mathbb{F}_2^n$ be the set of x such that $f(x) = 0$. Then, the size of S^1 is the number of terms in the trivial representation described in Sect. 3.1. The QM algorithm generates the minimum representation from the set S^1 by performing the following two steps:

1. Find all prime implicants.
2. Find the essential prime implicants, as well as the minimum necessary prime implicants to represent the Boolean function.

The first step evaluates combinable terms exhaustively. For instance, the two terms 0001 and 0011 can be combined as 00-1, where - represents an arbitrary bit. As another example, the combined terms 00-1 and 10-1 can be further combined as -0-1. Note that the Hamming distance between combinable terms is always 1. Therefore, all terms belonging to S^1 are first rearranged according to their Hamming weights. Then, we exhaustively evaluate all pairs whose Hamming distance is 1, combine them if possible, and insert these combined terms into a new set S^2. If a term $x \in S^1$ is combined with another term, these terms are removed from S^1. This procedure is iterated until there is no more combinable terms, and $S = S^1 \cup S^2 \cup \cdots \cup S^n$ becomes the set of terms that cannot be combined any further. Such terms are called "prime implicants."

The list of all prime implicants is redundant to represent the given Boolean function. Hence, the second step first searches S for the essential prime implicants. Here an essential prime implicant is a prime implicant that other prime implicants cannot cover, i.e., we cannot represent the given Boolean function accurately without the essential prime implicants. However, essential prime implicants are not enough to represent Boolean functions accurately. Therefore, we have to choose additional prime implicants to represent the given Boolean function with minimum number of terms accurately. This procedure is basically done heuristically.

We refer the reader to [Qui52, Qui55, McC56] for further details. Moreover, there are off-the-shelf software to implement the QM algorithm or more efficient heuristic algorithms, e.g., Logic Friday [Log].

B How to Generate Linear Inequalities by Logic Friday

To generate the linear inequalities for the DDT of an S-box, we used the Espresso algorithm implemented by Logic Friday. For the help of verification, we present how to generate linear inequalities by Logic Friday.

Logic Friday, a freeware tool for Boolean logic analysis, can be downloaded from http://sontrak.com/. For an n-bit to n-bit S-box, we first input the truth table whose input $2n$ bits and output is 1 bit. Here, the input corresponds to the difference $(\Delta x \| \Delta y)$, the output is 1 if and only if the corresponding entry of the pair of input-output differences is non-zero (possible transition) in the DDT. Logic Friday can import the truth table with the csv format from "Import Truth Table" in the "File" menu. Then the minimized *sum-of-product* representation can be obtained by choosing "Minimize" in the "fast" mode for the minimization from the "Operation" menu. Our MILP modeling requires the *product-of-sum* representation instead of sum-of-product. The product-of-sum representation can be obtained by selecting "product of sums" in the "Equation" menu.

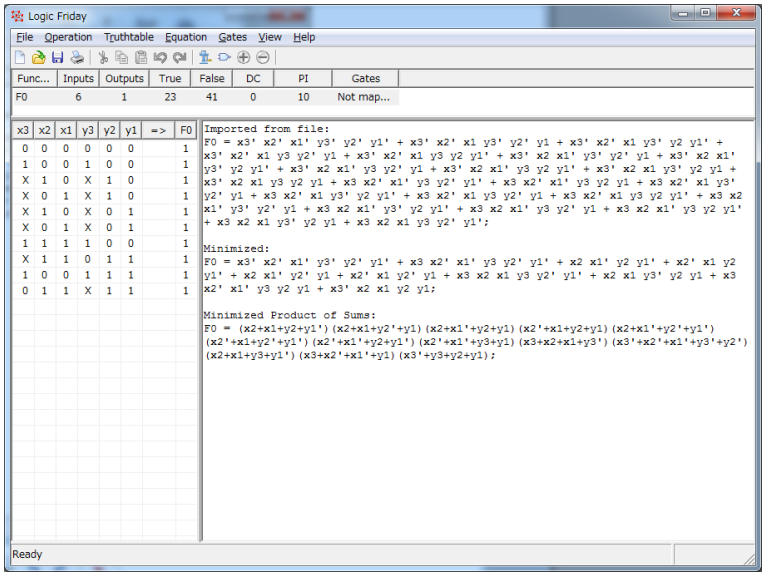

Figure 6: Logic Friday

Figure 6 shows the screenshot, where the information of the $*$-DDT of the 3-bit S-box shown in Table 1 is imported. As Figure 6 shows, Logic Friday returns the minimized product-of-sum representation, where the prime symbol "'" denotes the negation.

C Number of Inequalities in Various Models

Comparing the $*$-DDT Modeling. In Table 10, we compare the number of inequalities to model the $*$-DDT by using different models. Because computing the H-representation for 8-bit S-boxes is infeasible, we compare the number of inequalities for 4-bit S-boxes. In Table 10, the second and third columns show the number of inequalities after reducing the convex hull of the H-representation by using the greedy algorithm by Sun et al. [SHW+14a] and the sub-MILP problem by Sasaki and Todo [ST17b], respectively. These numbers are directly borrowed from those papers. The fourth and fifth columns show the case of logical condition model by using the Espresso heuristic algorithm in the Logic Friday software and the Quine-McCluskey algorithm, respectively. Note that the greedy and the Espresso algorithms are heuristic algorithms while the sub-MILP method and the QM algorithm ensure optimality.

Here, we give two important remarks. First, the coefficients of inequalities in the H-representation can take any integer whereas the coefficients of inequalities in the logical condition model only takes $\{-1, 0, 1\}$. It is unclear which model is faster even though the number of inequalities in the H-representation is generally smaller than that in the logical condition model. Second, Sasaki and Todo showed that minimizing the number of inequalities does not necessarily minimizes the runtime to solve the entire system [ST17b]. Given those facts, the issue of identifying the best model still remains open.

Comparing the Probability Evaluation. Another model proposed in this paper is to optimize probabilities of the differential characteristic. In this context, only one previous work is available [SHW+14a] that has two drawbacks. First, it cannot be applied to S-boxes larger than 5 bits. Second, it can evaluate the probability only when probabilities of all transitions are represented by 2^{-n}, where n is an integer. Note that the second drawback

Table 10: Number of Inequalities to Describe the $*$-DDT of Various 4-bit S-boxes.

Sbox	#inequalities			
	H-representation		Logical model	
	Greedy (heuristic)	sub-MILP (exact)	Espresso (heuristic)	QM (exact)
Kline	22	21	45	43
Piccolo	23	21	31	31
TWINE	23	23	47	45
PRINCE	26	22	52	51
MIBS	27	23	52	47
PRESENT/LED	22	21	39	36
LBlock S0	28	24	30	30
LBlock S1	27	24	30	30
LBlock S2	27	24	30	30
LBlock S3	27	24	31	30
LBlock S4	28	24	30	30
LBlock S5	27	24	30	30
LBlock S6	27	24	30	30
LBlock S7	27	24	30	30
LBlock S8	28	24	31	30
LBlock S9	27	24	31	30
Serpent S0	23	21	39	36
Serpent S1	24	21	38	36
Serpent S2	25	21	46	40
Serpent S3	31	27	48	47
Serpent S4	26	23	43	41
Serpent S5	25	23	43	41
Serpent S6	22	21	38	36
Serpent S7	30	27	49	47
Lilliput	—	23	47	45
Minalpher	—	22	51	50
RECTANGLE	—	21	31	30
`SKINNY-64`	—	21	31	31
Midori S0	—	21	47	47
Midori S1	—	22	57	56

is not a big issue as long as 4-bit S-boxes are evaluated. This is because probabilities of differential transitions in most of the 4-bit S-boxes are usually 2^{-2} or 2^{-3}. Whereas, the issue is a big matter when we evaluate 8-bit S-boxes like `SKINNY-128`.

Our model solved both issues. It can evaluate 8-bit S-box by splitting the DDT with respect to probability and by introducing conditional logic to control the entire behavior. Our model can increase the precision of the probability as much as we want.

D 13-Round Differential Characteristic for SKINNY-128

Table 11: 13-round differential characteristic for SKINNY-128 with probability 2^{-123}. Differences are represented by hexadecimal numbers. "Probability" shows logarithm of the probability of the S-box differential transition in the corresponding byte. The characteristic consists of 51 S-boxes with probability 2^{-2} and 7 S-boxes with probability 2^{-3}.

Round	Before SB	After SB	Probability	After SR
1	00 00 80 00	00 00 02 00	[0 0 -2 0]	00 00 02 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 00 80	00 00 00 02	[0 0 0 -2]	00 00 02 00
2	00 00 00 00	00 00 00 00	[0 0 -2 0]	00 00 00 00
	00 00 02 00	00 00 08 00	[0 0 0 0]	00 00 00 08
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 02 00	00 00 08 00	[0 0 -2 0]	00 09 00 00
3	00 09 00 00	00 10 00 00	[0 -2 0 0]	00 10 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 00 08	00 00 00 10	[0 0 0 -2]	00 10 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
4	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 10 00 00	00 50 00 00	[0 -2 0 0]	00 00 50 00
	00 10 00 00	00 40 00 00	[0 -2 0 0]	00 00 00 40
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
5	00 00 00 40	00 00 00 04	[0 0 0 -2]	00 00 00 04
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 50 40	00 00 04 04	[0 0 -2 -2]	04 04 00 00
	00 00 00 40	00 00 00 04	[0 0 0 -2]	00 00 04 00
6	04 04 04 04	05 05 05 01	[-3 -3 -3 -2]	05 05 05 01
	00 00 00 04	00 00 00 01	[0 0 0 -2]	01 00 00 00
	04 04 00 00	05 05 00 00	[-3 -3 0 0]	00 00 05 05
	04 04 00 04	01 01 00 05	[-2 -2 0 -3]	01 00 05 01
7	04 05 05 05	01 01 21 01	[-2 -2 -3 -2]	01 01 21 01
	05 05 05 01	01 01 01 20	[-2 -2 -2 -2]	20 01 01 01
	01 00 05 05	20 00 01 01	[-2 0 -2 -2]	01 01 20 00
	05 05 00 04	01 01 00 01	[-2 -2 0 -2]	01 00 01 01
8	01 00 00 00	20 00 00 00	[-2 0 0 0]	20 00 00 00
	01 01 21 01	20 20 20 20	[-2 -2 -2 -2]	20 20 20 20
	21 00 21 01	20 00 20 20	[-2 0 -2 -2]	20 20 20 00
	00 00 01 01	00 00 20 20	[0 0 -2 -2]	00 20 20 00
9	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	20 00 00 00	90 00 00 00	[-2 0 0 0]	00 90 00 00
	00 00 00 20	00 00 00 90	[0 0 0 -2]	00 90 00 00
	00 20 20 00	00 80 90 00	[0 -2 -2 0]	80 90 00 00
10	80 00 00 00	02 00 00 00	[-2 0 0 0]	02 00 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 90 00 00	00 02 00 00	[0 -2 0 0]	02 00 00 00
11	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	02 00 00 00	08 00 00 00	[-2 0 0 0]	00 08 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	02 00 00 00	08 00 00 00	[-2 0 0 0]	00 00 00 08
12	00 00 00 08	00 00 00 10	[0 0 0 -2]	00 00 00 10
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 08 00 00	00 10 00 00	[0 -2 0 0]	00 00 00 10
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
13	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
	00 00 00 10	00 00 00 40	[0 0 0 -2]	40 00 00 00
	00 00 00 10	00 00 00 40	[0 0 0 -2]	00 40 00 00
	00 00 00 00	00 00 00 00	[0 0 0 0]	00 00 00 00
CT	00 40 00 00			
	00 00 00 00			
	40 40 00 00			
	00 40 00 00			

E Figures for AES-Based Constructions

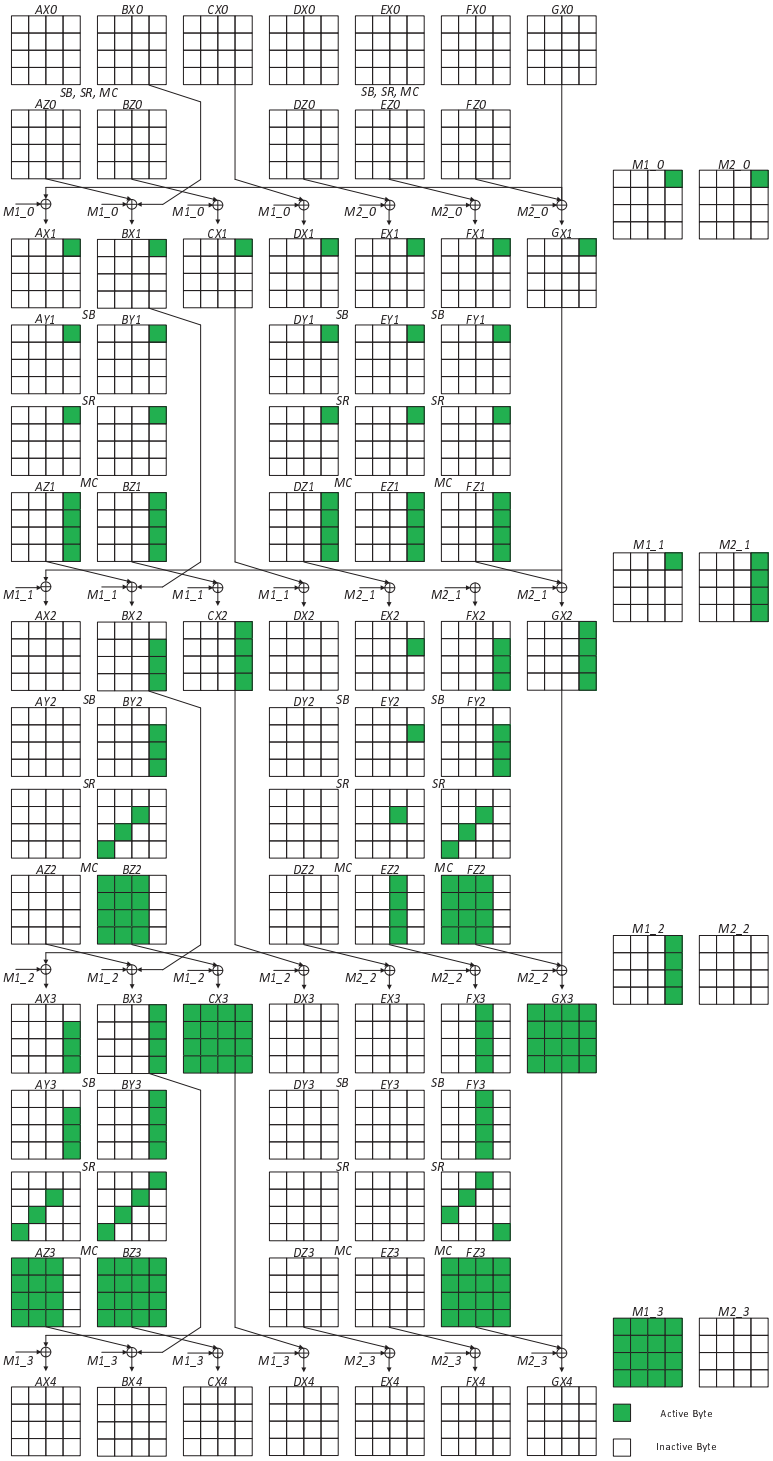

Figure 7: Invalid differential characteristic for C5 with 23 active S-boxes

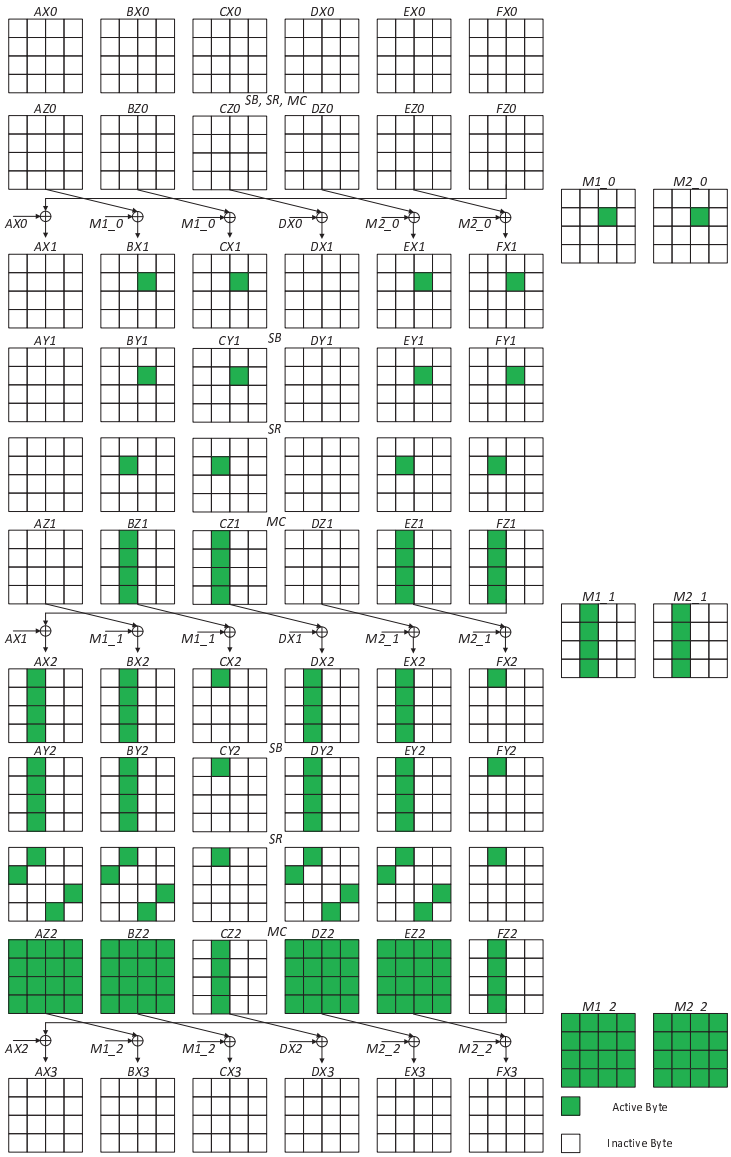

Figure 8: Invalid differential characteristic for C1

Table 12: 3-step differential characteristic for C1 with probability 2^{-134}. Differences are represented by hexadecimal numbers. "Probability" shows logarithm of the probability of the S-box differential transition in the corresponding byte of a particular state.

State	Values	Probability	State	Values	Probability
$M1\_0$ $M2\_0$	00 00 59 00 00 00 00 00 00 00 00 00 00 00 00 00	N/A	BX_1 CX_1 EX_1 FX_1	00 00 59 00 00 00 00 00 00 00 00 00 00 00 00 00	N/A
BY_1 CY_1 EY_1 FY_1	00 00 A8 00 00 00 00 00 00 00 00 00 00 00 00 00	[0 0 -6 0] [0 0 0 0] [0 0 0 0] [0 0 0 0]	BZ_1 CZ_1 EZ_1 FZ_1	00 00 4B 00 00 00 A8 00 00 00 A8 00 00 00 E3 00	N/A
$M1\_1$ $M2\_1$	00 00 12 00 00 00 A8 00 00 00 A8 00 00 00 E3 00	N/A	AX_2 DX_2	00 00 4B 00 00 00 A8 00 00 00 A8 00 00 00 E3 00	N/A
BX_2 EX_2	00 00 12 00 00 00 A8 00 00 00 A8 00 00 00 E3 00	N/A	CX_2 FX_2	00 00 59 00 00 00 00 00 00 00 00 00 00 00 00 00	N/A
AY_2	00 00 AA 00 00 00 A1 00 00 00 A1 00 00 00 72 00	[0 0 -7 0] [0 0 -6 0] [0 0 -6 0] [0 0 -6 0]	BY_2	00 00 AA 00 00 00 A1 00 00 00 A1 00 00 00 72 00	[0 0 -6 0] [0 0 -6 0] [0 0 -6 0] [0 0 -6 0]
CY_2 FY_2	00 00 A8 00 00 00 00 00 00 00 00 00 00 00 00 00	[0 0 -6 0] [0 0 0 0] [0 0 0 0] [0 0 0 0]	DY_2	00 00 D0 00 00 00 A1 00 00 00 A1 00 00 00 72 00	[0 0 -6 0] [0 0 -6 0] [0 0 -6 0] [0 0 -6 0]
EY_2	00 00 D0 00 00 00 A1 00 00 00 A1 00 00 00 72 00	[0 0 -7 0] [0 0 -6 0] [0 0 -6 0] [0 0 -6 0]	AZ_2 BZ_2	A1 F8 4F 72 F8 59 AA 72 59 A1 AA 96 A1 A1 E5 E4	N/A
CZ_2 FZ_2	00 00 4B 00 00 00 A8 00 00 00 A8 00 00 00 E3 00	N/A	DZ_2 EZ_2	A1 F8 BB 72 F8 59 D0 72 59 A1 D0 96 A1 A1 6B E4	N/A
$M1\_2$	A1 F8 4F 72 F8 59 AA 72 59 A1 AA 96 A1 A1 E5 E4	N/A	$M2\_2$	A1 F8 BB 72 F8 59 D0 72 59 A1 D0 96 A1 A1 6B E4	N/A