IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2019, No. 4, pp. 438–469. DOI:10.13154/tosc.v2019.i4.438-469

Improving the MILP-based Security Evaluation Algorithm against Differential/Linear Cryptanalysis Using A Divide-and-Conquer Approach

Chunning Zhou[1,2], Wentao Zhang[1,2], Tianyou Ding[1,2] and Zejun Xiang[3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, {zhouchunning,zhangwentao,dingtianyou}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan, China, xiangzejun@hubu.edu.cn

Abstract. In recent years, Mixed Integer Linear Programming (MILP) has been widely used in cryptanalysis of symmetric-key primitives. For differential and linear cryptanalysis, MILP can be used to solve two kinds of problems: calculation of the minimum number of differentially/linearly active S-boxes, and search for the best differential/linear characteristics. There are already numerous papers published in this area. However, the efficiency is not satisfactory enough for many symmetric-key primitives. In this paper, we greatly improve the efficiency of the MILP-based search algorithm for both problems. Each of the two problems for an r-round cipher can be converted to an MILP model whose feasible region is the set of all possible r-round differential/linear characteristics. Generally, high-probability differential/linear characteristics are likely to have a low number of active S-boxes at a certain round. Inspired by the idea of a divide-and-conquer approach, we divide the set of all possible differential/linear characteristics into several smaller subsets, then separately search them. That is to say, the search of the whole set is split into easier searches of smaller subsets, and optimal solutions within the smaller subsets are combined to give the optimal solution within the whole set. In addition, we use several techniques to further improve the efficiency of the search algorithm. As applications, we apply our search algorithm to five lightweight block ciphers: PRESENT, GIFT-64, RECTANGLE, LBLOCK and TWINE. For each cipher, we obtain better results than the best-known ones obtained from the MILP method. For the minimum number of differentially/linearly active S-boxes, we reach 31/31, 16/15, 16/16, 20/20 and 20/20 rounds for the five ciphers respectively. For the best differential/linear characteristics, we reach 18/18, 15/13, 15/14, 16/15 and 15/16 rounds for the five ciphers respectively.

Keywords: Block cipher · Differential cryptanalysis · Linear cryptanalysis · MILP · Divide-and-conquer

1 Introduction

As a fundamental primitive of cryptography, block ciphers have received extensive attention from academia and industry. The most important criterion for designing a block cipher is to ensure that it can resist all known attacks, especially differential and linear cryptanalysis [BS91, Mat93].

Licensed under Creative Commons License CC-BY 4.0.
Received: 2019-03-01, Revised: 2019-09-01, Accepted: 2019-11-01, Published: 2020-01-31

To evaluate the security of a block cipher against differential/linear cryptanalysis, there are usually two approaches. One is to calculate the minimum number of differentially/linearly active S-boxes to obtain an upper bound of the maximum probability/absolute linear bias. The other approach is to search for the best differential/linear characteristics to calculate the maximum probability/absolute linear bias. For some block ciphers, e.g., for block ciphers with large block sizes, this needs a huge workload and is likely to be impossible to be accomplished in a reasonable time.

In [Mat94], Matsui proposed a branch and bound search algorithm to search for the best differential characteristic. Matsui's search algorithm is one of the most powerful and classic search tools, but difficult to implement in some cases. In recent years, a method based on Mixed Integer Linear Programming (MILP) is proposed to evaluate the security of recent symmetric-key primitive designs. Due to its easy-to-master and general-to-use features, the MILP-based method has been widely used.

In [WW11, MWGP11], for the first time, the authors applied MILP to evaluate the security of a block cipher against differential and linear cryptanalysis. Mouha *et al.* [MWGP11] introduced a model framework to calculate lower bounds of the minimum number of active S-boxes for word-oriented ciphers, then Sun *et al.* [SHS+13] extended their framework to SPN ciphers with bit-wise permutation diffusion layers. In [SHS+13], bit-wise operations are described as linear inequalities. However, the description of an S-box is rough, which results in a solution of the model is not guaranteed to be a valid differential characteristic. At ASIACRYPT 2014, Sun *et al.* [SHW+14b] improved the MILP-based method for automatically evaluating the security of a block cipher against (related-key) differential cryptanalysis, and proposed a heuristic algorithm for finding actual (related-key) differential characteristics. They introduced two systematic methods for generating inequalities to describe the bit-wise S-box operation more accurately: logical condition modeling and convex hull computation. By using the two methods, some impossible differential characteristics are removed from the feasible region of the model, and tighter security bounds are obtained. Later in [SHW+14a], Sun *et al.* encoded differential probabilities/linear approximations of S-boxes into the MILP model, and argued that the feasible region of the model built by using the convex hull computation method for S-boxes is exactly the set of all possible (related-key) differential/linear characteristics. Therefore, their model can be used to obtain the minimum number of active S-boxes and find the best differential/linear characteristic. In [FWG+16], Fu *et al.* extended the MILP-based automatic search algorithm to ARX ciphers and applied it to search for the best differential and linear characteristics for SPECK [BSS+13]. Abdelkhalek *et al.* [AST+17] introduced a new method for modelling large S-boxes, e.g., 8-bit S-boxes, and evaluated the maximum probability of differential characteristics for SKINNY-128 [BJK+16].

The problem of the calculation of the minimum number of differentially/linearly active S-boxes or the search for the best differential/linear characteristic for an r-round block cipher can be converted to an MILP model. As the size of the model increases significantly with the increasing of the number of rounds, the model can't be solved within a reasonable time when r is too large. Thus a lot of papers are published to address this issue. A simple split approach was introduced in [MWGP11, SHW+14b], which splits r rounds into the first r_1 and the last $(r - r_1)$ rounds then combines them, $1 \leq r_1 < r$. In [SHQ+15], the authors restricted difference patterns of S-boxes to search for improved differential characteristics. An interesting method that incorporates Matsui's branch and bound search algorithm and the MILP-based technique was introduced by Zhang *et al.* [ZSCH18]. They added the constraints derived from the bounding condition of Matsui's algorithm into the MILP model, which results in the model having a reduced feasible region and being solved with a shorter time. However, the efficiency of solving existing MILP models is still not satisfactory enough. New techniques are of necessity to push the limitation of our ability to evaluate the security of block ciphers based on MILP methods. Thus, the motivation of

this paper is to make the MILP-based method an efficient enough tool upon which we can entirely rely to evaluate the security of block ciphers against differential/linear attack.

Our Contributions. Due to the duality between differential and linear cryptanalysis [Mat94], we mainly focus on the security evaluation algorithm against differential cryptanalysis. Calculating the minimum number of active S-boxes or searching for the best differential characteristic is equivalent to an MILP-based search for the optimal differential characteristic with the minimum number of active S-boxes or with the minimum weight (defined as the negative of the binary logarithm of the probability). Our main contributions are:

1. We propose an improved MILP-based search algorithm to evaluate the security of block ciphers against differential cryptanalysis. Inspired by the idea of a divide-and-conquer approach, we divide the set of all possible differential characteristics into smaller subsets, then separately search them. We observe that high-probability differential characteristics are likely to have a low number of active S-boxes at a certain round, thus smaller subsets are partitioned based on the information of S-boxes. Take an SPN cipher as an example, we assume that the differential characteristics in a subset have exactly one or two active S-boxes at a certain round. By using the MILP technique, searching a subset can be transformed into an MILP model whose feasible region is exactly the subset. Then the optimal differential characteristics within the subsets are combined to give the optimal differential characteristic within the whole set. To further improve efficiency, we use three techniques:

   (a) At the beginning of the search, we generate a valid differential characteristic and use it as the currently optimal differential characteristic. Its number of active S-boxes (or weight) is served as an upper bound of the minimum number of active S-boxes (or weight). The currently optimal differential characteristic and the upper bound are dynamically updated during the search.

   (b) When searching a subset, we calculate a lower bound of the minimum number of active S-boxes (or weight) within it. If the lower bound is greater than or equal to the upper bound introduced in the first technique, there is no better differential characteristic and we terminate the search of this subset;

   (c) We choose a proper search order of subsets, such that better differential characteristics can be searched as early as possible.

   Finally, an MILP-based search algorithm is proposed. The algorithm can be used to both calculate the minimum number of active S-boxes and search for the best differential characteristic. Also, the algorithm can be extended to the security evaluation of block ciphers against linear cryptanalysis with a slight modification.

2. We apply our search algorithm to five lightweight block ciphers: PRESENT [BKL+07], GIFT-64 [BPP+17], RECTANGLE [ZBL+15], LBLOCK [WZ11] and TWINE [SMMK12]. For each cipher, we obtain better results than previous best-known results obtained from the MILP method. For the minimum number of differentially/linearly active S-boxes, we reach 31/31, 16/15, 16/16, 20/20 and 20/20 rounds for the five ciphers respectively. For the best differential/linear characteristics, we reach 18/18, 15/13, 15/14, 16/15 and 15/16 rounds for the five ciphers respectively. To compare with previous MILP-based work, we implement Sun *et al.*'s model [SHW+14b, SHW+14a] and Zhang *et al.*'s model [ZSCH18] to search for the best differential characteristic for the five ciphers. We implement the three methods (ours, Sun *et al.*'s and Zhang *et al.*'s) on a PC, and summarize the experimental results in Table 1. From the table, we see that our search algorithm has an advantage over the other two methods when the number of rounds is large, and for each cipher, our

Table 1: Comparison of the results on the best differential and linear characteristics

Ciphers	Differential characteristic				Linear characteristic			
	Rounds	t_1	t_2	t_3(Sect. 5)	Rounds	t_1	t_2	t_3(Sect. 5)
PRESENT	8	764s	444s	284s	8	5188s	258s	557s
	9	3426s	1143s	298s	9	14207s	663s	572s
	10	31000s	6286s	1596s	10	58048s	2361s	872s
	18	-	-	6.90h	18	-	-	10.8h
GIFT-64	6	494s	343s	1477s	8	7414s	1442s	829s
	7	9846s	1910s	1862s	9	48546s	6899s	9542s
	8	138603s	27352s	21796s	10	>4d	77353s	21157s
	15	-	-	28.29h	13	-	-	53.2h
RECTANGLE	8	3345s	824s	216s	8	1881s	1066s	101s
	9	14931s	3551s	257s	9	6218s	8204s	179s
	10	38461s	20041s	353s	10	64526s	38074s	439s
	15	-	-	15.98h	14	-	-	60.31h
LBLOCK	8	1330s	646s	7s	8	1064s	573s	6s
	9	8293s	1833s	9s	9	3451s	6848s	8s
	16	-	-	2.36h	15	-	-	2.17h
TWINE	9	7332s	4282s	8s	8	2631s	1078s	10s
	10	30026s	15404s	60s	9	2408s	4684s	14s
	15	-	-	15.02h	16	-	-	758s

t_1 and t_2 respectively denote the time of solving Sun *et al.*'s model [SHW+14b, SHW+14a]) and Zhang *et al.* 's model [ZSCH18], and t_3 denotes the runtime of our search algorithm. For the three methods, we use Sun's Greedy algorithm [SHW+14a]) to reduce the number of inequalities from the convex-hull modeling, and use Gurobi software [Gur] with MipFocus equal to 2 to solve models.

search algorithm covers more rounds with less time. The source code is available at https://github.com/Chunning-Zhou/MILPBasedSearchAlgorithmDiff.

Organization. In Section 2, we recall the existing automatic MILP-based tool for evaluating the security of block ciphers against differential cryptanalysis. An improved MILP-based search algorithm by incorporating the idea of a divide-and-conquer approach is proposed in Section 3. We use our search algorithm to evaluate the security of block ciphers against differential/linear cryptanalysis, and apply it to five lightweight block ciphers in Sections 4 and 5. In Section 6, we conclude the paper and provide some ideas for future work. More details and experimental results are given in Appendices.

2 Related Work

Calculating the minimum number of active S-boxes and searching for the best differential characteristic are two ways to evaluate the security of block ciphers against differential attacks. In this section, we introduce the MILP-based method for solving the two problems.

2.1 Model Framework for Calculating the Minimum Number of Active S-boxes

Mouha *et al.* [MWGP11] introduced a model framework to calculate lower bounds of the minimum number of active S-boxes for word-oriented block ciphers. Then, Sun *et al.* extended their model framework to bit-oriented block ciphers [SHS+13, SHW+14b, SHW+14a].

2.1.1 Mouha *et al.*'s Model Framework for Word-Oriented Block Ciphers

Mouha *et al.* [MWGP11] considered truncated differences, and used a 0-1 variable to describe a word-level difference, such that the variable equals 1 if and only if the input

word is non-zero. Assume that a cipher is composed of three operations: XOR, linear transformation and S-box, the following constraints are introduced to describe word-level difference propagations through a cipher.

Equations Describing the XOR Operation. Let a, b and c denote word-level input and corresponding output differences of the XOR operation, the following equations are used to describe the XOR operation:

$$\begin{cases} a+b+c \geq 2d_{\oplus}, \\ d_{\oplus} \geq a, d_{\oplus} \geq b, d_{\oplus} \geq c, \end{cases} \tag{1}$$

where $d_{\oplus}$ is a dummy variable taking values in $\{0, 1\}$.

Equations Describing the Linear Transformation. Let $(x_0, ..., x_{m-1})$ and $(y_0, ..., y_{m-1})$ denote word-level input and output differences of the linear transformation L respectively. Given the differential branch number $\mathcal{B}_{\mathcal{D}}$ of L, the linear transformation is described by:

$$\begin{cases} \sum_{k=0}^{m-1} x_k + \sum_{k=0}^{m-1} y_k \geq \mathcal{B}_{\mathcal{D}} d_L, \\ d_L \geq x_k, d_L \geq y_k, k \in \{0, ..., m-1\}, \end{cases}$$

where d_L is a dummy variable taking values in $\{0, 1\}$. In [SHS+13], the authors pointed out that if L is not MDS, additional constraints are needed to ensure that non-zero input difference must result in non-zero output difference and vice versa. The constraints are similar to Equation (4) describing an S-box.

Additional Constraints. To avoid a trivial solution, it needs an additional constraint to ensure that at least one S-box is active. Besides, all dummy d-variables, and the variables representing the plaintext differences are restricted to be 0-1 variables.

Objective Function. The objective function is to minimize the number of active S-boxes, *i.e*, the sum of all variables representing word-level input differences of S-boxes of each round.

By using the model framework above, an MILP model is built to calculate a lower bound of the minimum number of active S-boxes for a word-oriented block cipher. However, this framework did not consider bit-wise operations, thus it is not applicable to bit-oriented block ciphers.

2.1.2 Sun *et al.*'s Bit-wise Model Framework

In [SHS+13, SHW+14b, SHW+14a], Sun *et al.* described operations of bit-oriented block ciphers. They used a 0-1 variable to denote the bit-level difference, such that the variable equals 1 if and only if the bit-level difference is non-zero. For each S-box, they used a 0-1 variable A to denote the word-level input difference, such that $A = 1$ if and only if the input word of the S-box is non-zero. Therefore, the objective function is to minimize the sum of all A variables.

For bit-wise XOR, an additional inequality $a + b + c \leq 2$ is added into Equation (1). For an $w \times v$ S-box marked by A, suppose $(x_0, ..., x_{w-1})$ and $(y_0, ..., y_{v-1})$ are input and output differences respectively, the following constrains are introduced to describe the S-box. Firstly, $A = 1$ holds if and only if $x_0, ..., x_{w-1}$ are not all zero:

$$\begin{cases} A - x_k \geq 0, k \in \{0, ..., w-1\}, \\ x_0 + x_1 + ... + x_{w-1} - A \geq 0. \end{cases} \tag{2}$$

Besides, the Hamming weight of $(x_0, ..., x_{w-1}, y_0, ..., y_{v-1})$ is greater than or equal to the branch number $\mathcal{B}_\mathcal{S}$ of the S-box for a non-zero input difference:

$$\begin{cases} \sum_{k=0}^{w-1} x_k + \sum_{k=0}^{v-1} y_k \geq \mathcal{B}_\mathcal{S} d_S, \\ d_S \geq x_k, k \in \{0, ..., w-1\}, \\ d_S \geq y_k, k \in \{0, ..., v-1\}, \end{cases} \tag{3}$$

where d_S is a dummy variable taking values in $\{0, 1\}$, and the branch number $\mathcal{B}_\mathcal{S}$ of an S-box is defined as

$$\mathcal{B}_\mathcal{S} = \min_{a \neq b} \{\mathrm{wt}((a \oplus b)||S(a) \oplus S(b)) : a, b \in \mathbb{F}_2^w\},$$

where $\mathrm{wt}(\cdot)$ is the standard Hamming weight. In addition, for bijective S-boxes, non-zero input difference must result in non-zero output difference and vice versa:

$$\begin{cases} wy_0 + wy_1 + ... + wy_{v-1} - (x_0 + x_1 + ...x_{w-1}) \geq 0, \\ vx_0 + vx_1 + ... + vx_{w-1} - (y_0 + y_1 + ...y_{v-1}) \geq 0. \end{cases} \tag{4}$$

By using the objective function and the constraints introduced above, an MILP model is built to calculate a lower bound of the minimum number of active S-boxes for a bit-oriented cipher. The constraints used to describe S-boxes are rough, thus a feasible solution of the model is not guaranteed to be a valid differential characteristic.

To describe an S-box more accurately, Sun *et al.* [SHW+14b] proposed two systematic methods for generating inequalities: the logical condition modeling and the convex hull computation. The inequalities are used to remove invalid differential characteristics from the feasible region of the model. Later in [SHW+14a], Sun *et al.* proved that the feasible region of the model built by using the convex hull computation method for S-boxes is exactly the set of all possible differential characteristics. In the convex hull computation method, a possible difference propagation $(x_0, ..., x_{w-1}) \to (y_0, ..., y_{v-1})$ of an $w \times v$ S-box is treated as a point in $\mathbb{F}_2^{w+v}$: $(x_0, ..., x_{w-1}, y_0, ..., y_{v-1}) \in \mathbb{F}_2^{w+v}$. All possible difference propagations of the S-box constitute a set of finitely many discrete points. By computing the H-Representation of the convex hull of the set with the help of SageMath software [sag], inequalities are generated and their feasible solutions are exactly the points in the set. The number of inequalities computed from SageMath is generally very large, thus a greedy algorithm [SHW+14a] and an MILP-based reduction algorithm [ST17] were proposed to select a small number of inequalities. By using these reduced inequalities to describe the S-box, all impossible difference propagations of S-boxes are removed from the feasible region of the model. By modeling S-boxes with the convex hull computation method and restricting all variables involved to be 0-1 variables, the MILP model can be used to obtain the minimum number of active S-boxes.

2.2 Model Framework for Searching for the Best Differential Characteristic

In [SHW+14a], Sun *et al.* encoded differential probabilities of an S-box into an MILP model and searched for the best differential characteristic for a block cipher.

Take PRESENT cipher as an example. PRESENT uses a 4×4 S-box, and there are 3 nontrivial probabilities in the difference distribution table of the S-box. A possible difference propagation of an S-box $(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)$ with the probability Pr is treated as a 10-dimensional point:

$$(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, p_0, p_1) \in \mathbb{F}_2^{10},$$

where

$$(p_0, p_1) = \begin{cases} (0,0), \text{if Pr} = 1; \\ (0,1), \text{if Pr} = 2^{-2}; \\ (1,1), \text{if Pr} = 2^{-3}, \end{cases} \tag{5}$$

that is to say, $\text{Pr} = 2^{-(p_0+2p_1)}$. By using the convex hull computation method, inequalities are generated to describe all the possible points. To find the best differential characteristic, *i.e.*, the differential characteristic with the maximal probability, the objective function is to minimize the sum of all $(p_0 + 2p_1)$. Note that in Equation (5), p_1 equals 1 if and only if the input difference is non-zero, thus p_1 also indicates the input word of the S-box.

3 Improving the MILP-based Search Algorithm by Incorporating with A Divide-and-Conquer Approach

The difficulty of the problem of the calculation of the minimum number of active S-boxes or the search for the best differential characteristic usually increases exponentially with the increasing of the number of rounds. Existing MILP models can hardly be used to solve the problem when the number of rounds is large. The main reason is that in existing MILP models built for an r-round cipher, the feasible region is the set of all possible r-round differential characteristics. Inspired by the idea of a divide-and-conquer approach, we improve the efficiency of the MILP-based search algorithm basing on our empirical knowledge about the valid differential characteristics. We first divide the set of all possible r-round differential characteristics into several subsets, we then search each subset separately. By using the MILP technique, searching a subset can be equivalent to solving an MILP model whose feasible region is exactly the subset. Then the optimal differential characteristic within the whole set is given by combining all the results returned from subsets. To further improve efficiency, we introduce several techniques. Finally, an improved MILP-based search algorithm is proposed. For the convenience of illustration, we mainly focus on searching for the best differential characteristic for SPN ciphers. The search for Feistel ciphers is similar and it is illustrated in Appendix A.

3.1 Dividing the Set of All Possible Differential Characteristics

Unlike Feistel ciphers, for SPN ciphers, valid differential characteristics have at least one active S-box in each round, and those with the highest probability are likely to have at most two active S-boxes at a certain round. Based on this observation, we propose a proper way to divide the set of all possible differential characteristics into smaller subsets then conquer them separately.

For an r-round SPN cipher, we first divide the set of all possible r-round differential characteristics into three kinds of subsets:

Subset-1 In this kind of subsets, differential characteristics have at least one active S-box in each round, and there is at least one round that contains exactly one active S-box;

Subset-2 In this kind of subsets, differential characteristics have at least two active S-boxes in each round, and there is at least one round that contains exactly two active S-boxes;

Subset-3 In this subset, differential characteristics have at least three active S-boxes in each round.

For Subset-1 and Subset-2, we further divide them by fixing the index i $(i = 1, 2, ..., r)$, such that round i has exactly 1 or 2 active S-boxes. By doing this, the number of all

possible input differences of round i decreases significantly. Then, we traverse the input difference of round i to further divide the subsets.

For the sake of brevity, we use $\mathcal{D}_{r,N_A,i,\Delta}$ to denote the subset of r-round differential characteristics that satisfy the following two constraints:

Constraint 1 there are at least N_A active S-boxes in each round except for round i;

Constraint 2 the input difference of round i equals Δ.

Specially, when $i = 0$, there is no constraint on the input difference, and we ignore the second constraint by (not rigorously) writing $\Delta = 0$. Differential characteristics in the subset are only required to satisfy the Constraint 1, *i.e.*, there are at least N_A active S-boxes in each round. Moreover, if $i = r + 1$, we regard the constraint of "input difference of round $(r + 1)$ equals Δ" as the constraint of "output difference of round r equals Δ".

In summary, the set of all possible r-round differential characteristics for an r-round SPN cipher is divided into the following subsets:

$$\left(\bigcup_{N_A,i,\Delta} \mathcal{D}_{r,N_A,i,\Delta}\right) \bigcup \mathcal{D}_{r,3,0,0}, \tag{6}$$

where $N_A \in \{1, 2\}, i \in \{1, 2, ..., r\}$, Δ belongs to the set of all possible input differences of round i that has exactly N_A active S-boxes. Take PRESENT as an example, its block size is 64 bits and it uses a 4×4 S-box. Thus it applied 16 S-boxes in parallel in each round. When one round has exactly one active S-box, there are $16 \times (2^4 - 1) = 240$ possibilities of input differences of this round; when one round has exactly two active S-boxes, there are $120 \times (2^4 - 1)^2 = 27000$ possibilities of the input difference of this round. Based on Equation (6), the set of all possible differential characteristics for r-round PRESENT is divided into $(r \times 240 + r \times 27000 + 1)$ subsets.

3.2 Building MILP Models for Searching Subsets

By using the MILP technique described in Section 2, searching for the best differential characteristic for an r-round cipher is transformed into an MILP model. The feasible region of this model is the set of all possible r-round differential characteristics. To search for the best differential characteristic within a subset divided in Equation (6), we introduce additional constraints and build an MILP model whose feasible region is exactly the subset.

For the subset named $\mathcal{D}_{r,N_A,i,\Delta}$, differential characteristics in it are constrained by Constraints 1 and 2. Let $A_{j,k}$ denote the word-level input difference of the kth S-box at round j, the Constraint 1 is described by Equation (7):

$$\sum_{k=0}^{N_S-1} A_{j,k} \geq N_A, j \in \{1, 2, .., r\}, j \neq i, \tag{7}$$

where N_S is the total number of S-boxes in each round. Moreover, let $x_{i,j}$ denote the jth bit input difference of round i, the Constraint 2 is described by Equation (8):

$$x_{i,j} = \Delta_j, j \in \{0, 1, ..., n - 1\}, \tag{8}$$

where Δ_j is the jth bit of Δ, and n is the block size of the cipher. By adding Equations (7, 8) into the original MILP model that is used to search for the best r-round differential characteristic, we obtain a model whose feasible region is the subset $\mathcal{D}_{r,N_A,i,\Delta}$. By solving this model, we obtain the best differential characteristic within the subset.

By separately searching all the subsets divided, namely, solving all the corresponding MILP models, the best differential characteristics within the subsets are returned. They

are combined to give the best differential characteristic for an r-round cipher. Although the model with a smaller feasible region is easier to be solved, it will cost a huge amount of time if we solve all the models. This is mainly because the number of the subsets divided is generally large and some of the models are time-consuming (*e.g.*, the model whose feasible region is $\mathcal{D}_{r,3,0,0}$ takes a long time to be solved). Therefore, we introduce several techniques to further improve the efficiency of the algorithm in the next section.

3.3 Techniques to further Improve Efficiency

In this section, we introduce some techniques to improve the efficiency of our search algorithm. With the help of the techniques, the search of a subset can be early terminated, thus the runtime of the algorithm is greatly reduced.

The weight of a differential characteristic is defined as the negative of the binary logarithm of the probability. Searching for the best differential characteristic for an r-round cipher is equivalent to searching for the r-round differential characteristic with the minimum weight. For the search for r rounds, three techniques are used:

Technique 1. Setting an Upper Bound. Similar to the strategy used in [Mat94, AST+17], at the beginning of the search, we generate a valid r-round differential characteristic. This differential characteristic is used as the currently best r-round differential characteristic, and its weight is served as an upper bound of the minimum weight. We generate such an r-round differential characteristic by exploiting one of the best $(r-1)$-round differential characteristics we have found. The difference pattern of an S-box is defined as its truncated difference, *i.e.*, a 0-1 variable. In the model built for searching for the best r-round differential characteristic, we fix the S-boxes of the first $(r-1)$ rounds (or the last $(r-1)$ rounds) to have the same difference patterns as those of the best $(r-1)$-round differential characteristic. By solving this model, we obtain an r-round differential characteristic and its weight. To reduce unnecessary searches, we only focus on the subsets in which the differential characteristics have weights smaller than the obtained upper bound. During the search of subsets, if we find an r-round differential characteristic having a weight smaller than the current upper bound, the currently best r-round differential characteristic is updated by it and the current upper bound is updated by its weight.

Technique 2. Calculating Lower Bounds within Subsets. The upper bound introduced in Technique 1 uses the weight of the currently best differential characteristic as a threshold. When searching a subset, we calculate a lower bound of the minimum weight of the differential characteristics within it. If the lower bound is greater than or equal to the upper bound, there is no differential characteristic with a weight smaller than the weight of the currently best differential characteristic. At this point, we terminate the search of this subset and search the next one. With the help of this technique, the number of MILP models to be solved is reduced.

Technique 3. Choosing a Proper Search Order of Subsets. In order to make Technique 2 more efficient, we aim to find better differential characteristics as early as possible. Thus, we choose a proper search order of the subsets divided in Equation (6). There are two cases we can take into account: (1) we preferentially search the subsets which are more likely to provide better differential characteristics; (2) we preferentially search the subsets whose corresponding MILP models can be solved with less time. According to the two cases, we traverse the subsets $\mathcal{D}_{r,N_A,i,\Delta}$ in a proper order, where $N_A \in \{1,2\}, i \in \{1,2,...,r\}$, Δ belongs to the set of all possible input differences of round i that has exactly N_A active S-boxes. By default, N_A is traversed from 1 to 2 (*i.e.*, $N_A \in [1,2]$), namely, we first search

the subsets $\mathcal{D}_{r,1,i,\Delta}$ then $\mathcal{D}_{r,2,i,\Delta}$. If we can predict that the best differential characteristics have at least two active S-boxes in each round from some prior knowledge, we preferentially search the subsets $\mathcal{D}_{r,2,i,\Delta}$ (*i.e.*, $N_A \in [2,1]$). The search complexity usually increases exponentially with the increasing of the number of rounds. By our empirical observation, the closer the i is to the middle of r, the smaller the complexity of searching the subset $\mathcal{D}_{r,N_A,i,\Delta}$. This might due to the input difference of round i is fixed to equal Δ, which results in the complexity of searching the subset is determined by the sum of complexities of searching the first $(i-1)$ rounds and the last $(r-i+1)$ rounds. Therefore, we traverse the index i by:

$$i \in \texttt{SearchR} = \begin{cases} \left[\frac{r+2}{2}, \frac{r+2}{2}-1, \frac{r+2}{2}+1, ..., 1\right], \text{if } r \text{ is even;} \\ \left[\lfloor\frac{r+2}{2}\rfloor, \lfloor\frac{r+2}{2}\rfloor+1, \lfloor\frac{r+2}{2}\rfloor-1, ..., 1\right], \text{if } r \text{ is odd.} \end{cases}$$

For example, when $r = 10$, $\texttt{SearchR} = [6,5,7,4,8,3,9,2,10,1]$; and when $r = 11$, $i \in \texttt{SearchR} = [6,7,5,8,4,9,3,10,2,11,1]$.

3.4 Methods for Calculating Lower Bounds within Subsets

Based on Technique 2 introduced in Section 3.3, the search of a subset is terminated if a lower bound of the minimum weight of the differential characteristics within the subset is bigger than or equal to the weight of the currently best differential characteristic. In this section, we provide several methods to calculate the lower bound.

For an r-round SPN cipher, the set of all possible r-round differential characteristics is divided into several subsets according to Equation (6). For the subset named $\mathcal{D}_{r,N_A,i,\Delta}$, we use

$$\texttt{LB}[r, N_A, i, \Delta]$$

to store a lower bound of the minimum weight of the differential characteristics within it, where $N_A \in \{1,2,3\}$, $i \in \{0,1,...,r+1\}$, $\Delta \in \mathbb{F}_2^n$ (n is the block size of the cipher). The array LB is called the lower bound array in which the values are dynamically and repeatedly updated. In the following, we provide three methods for estimating lower bounds of the minimum weight of the differential characteristics within $\mathcal{D}_{r,N_A,i,\Delta}$. Methods 1-2 use other values that have been determined and stored in the lower bound array LB, and Method 3 uses the results returned from MILP models. The three methods can be used to assign a value to $\texttt{LB}[r, N_A, i, \Delta]$, and they are written as Functions `AssignByMethod1`(), `AssignByMethod2`() and `AssignByMethod3`() respectively.

Method 1. By using the simple split method introduced in [MWGP11, SHW+14b], we split r rounds into the first r_1 rounds and the last $(r-r_1)$ rounds then combine the two smaller parts, $1 \le r_1 < r$.

1. For the subset $\mathcal{D}_{r,N_A,0,0}$, the differential characteristics of the first r_1 rounds and the last $(r-r_1)$ rounds respectively belong to $\mathcal{D}_{r_1,N_A,0,0}$ and $\mathcal{D}_{r-r_1,N_A,0,0}$. Thus the lower bound within $\mathcal{D}_{r,N_A,0,0}$ can be estimated by:

$$\max_{1 \le r_1 \le r-1} (\texttt{LB}[r_1, N_A, 0, 0] + \texttt{LB}[r-r_1, N_A, 0, 0]). \tag{9}$$

2. For the subset $\mathcal{D}_{r,N_A,i,\Delta}$ with $i \in \{1,...,r\}, \Delta \neq 0$, the input difference of round i, namely, the output difference of round $(i-1)$ equals Δ. Thus the lower bound within $\mathcal{D}_{r,N_A,i,\Delta}$ can be estimated by:

$$\max \left\{ \begin{array}{l} \texttt{LB}[i-1, N_A, i, \Delta] + \texttt{LB}[r-i+1, N_A, 1, \Delta], (i \ge 2), \\ \max\limits_{1 \le r_1 \le i-1} (\texttt{LB}[r_1, N_A, 0, 0] + \texttt{LB}[r-r_1, N_A, i-r_1, \Delta]), (i \ge 2), \\ \max\limits_{i \le r_1 \le r-1} (\texttt{LB}[r_1, N_A, i, \Delta] + \texttt{LB}[r-r_1, N_A, 0, 0]) \end{array} \right\}. \tag{10}$$

3. For the subset $\mathcal{D}_{r,N_A,r+1,\Delta}$, the lower bound within it can be estimated by:

$$\max_{1\le r_1\le r-1}(\texttt{LB}[r_1, N_A, r_1+1, \Delta] + \texttt{LB}[r-r_1, N_A, 0, 0]) \tag{11}$$

Method 2. When searching the subset $\mathcal{D}_{r,N_A,i,\Delta}$, $N_A \in \{1,2\}, i \in \texttt{SearchR}, \Delta$ belongs to the set of all possible input differences of round i that has exactly N_A active S-boxes, we avoid searching the differential characteristics that have been searched. Take the search for $r = 10$ as an example, $\texttt{SearchR} = [6,5,7,4,8,3,9,2,10,1]$. Firstly, we search the subsets $\mathcal{D}_{r,N_A,6,\Delta}$. After searching $\mathcal{D}_{r,N_A,6,\Delta}, \forall\Delta$, we begin to deal with the subsets $\mathcal{D}_{r,N_A,5,\Delta}$. At this point, the differential characteristics satisfying round 6 with exactly N_A active S-boxes have already been searched. To avoid duplicate searches, we assume that the differential characteristics to be searched later have at least (N_A+1) active S-boxes at round 6. We add the additional constraint into the differential characteristics in $\mathcal{D}_{r,N_A,5,\Delta}$, and we calculate a lower bound of the minimum weight of these differential characteristics. If the lower bound is greater than or equal to the weight of the currently best r-round differential characteristic, there is no better differential characteristic than the currently best one within $\mathcal{D}_{r,N_A,5,\Delta}$. Suppose $i = \texttt{SearchR}[j], 1 \le j \le r-1$, let R_{front} denote the number in front of i, *i.e.*, $R_{front} = \texttt{SearchR}[j-1]$. In the following, we provide a method to estimate a lower bound of the minimum weight of the differential characteristics satisfying the additional constraint.

- For the case $i < \texttt{SearchR}[0]$, the differential characteristics in $\mathcal{D}_{r,N_A,i,\Delta}$ satisfy the additional constraint: "there are at least (N_A+1) active S-boxes in each of rounds from $(i+1)$ to R_{front}". Similar to Method 1, we split r rounds into smaller parts then combine them. Thus a lower bound of the minimum weight of the differential characteristics with the additional constraint is estimated by:

$$\max\left\{\begin{array}{l}\texttt{LB}[i, N_A, 0, 0] + \texttt{LB}[R_{front}-i, N_A+1, 0, 0] + \texttt{LB}[r-R_{front}, N_A, 0, 0]),\\ \texttt{LB}[i-1, N_A, i, \Delta] + \texttt{LB}[R_{front}-i+1, N_A+1, 1, \Delta] + \texttt{LB}[r-R_{front}, N_A, 0, 0])\end{array}\right\}, \tag{12}$$

  where in the first expression, the r rounds are split into three parts: (1) round 1 to i; (2) round $(i+1)$ to R_{front}; (3) round $(R_{front}+1)$ to r; and in the second expression, the r rounds are split into: (1) round 1 to $(i-1)$; (2) round i to R_{front}; (3) round $(R_{front}+1)$ to r.

- For the case $i > \texttt{SearchR}[0]$, the differential characteristics in $\mathcal{D}_{r,N_A,i,\Delta}$ satisfy the additional constraint: "there are at least (N_A+1) active S-boxes in each of rounds from R_{front} to $(i-1)$ ". Similar to the first case, we estimate a lower bound of the minimum weight of the differential characteristics with the additional constraint by:

$$\max\left\{\begin{array}{l}\texttt{LB}[R_{front}-1, N_A, 0, 0] + \texttt{LB}[i-R_{front}, N_A+1, 0, 0] + \texttt{LB}[r-i+1, N_A, 0, 0]),\\ \texttt{LB}[R_{front}-1, N_A, 0, 0] + \texttt{LB}[i-R_{front}, N_A+1, i-R_{front}+1, \Delta] + \texttt{LB}[r-i+1, N_A, 1, \Delta])\end{array}\right\}. \tag{13}$$

Moreover, tighter lower bounds can be estimated by splitting the r rounds more carefully, and the details and implementation can be seen in the public source code.

Method 3. We estimate lower bounds of the minimum weight of the differential characteristics within a subset by solving MILP models. The constraints for describing a subset have been introduced in Section 3.2, and there are two kinds of MILP models for us to choose from:

1. The first kind of model is called a "Rough" model that provides a rough lower bound. We notice that the models built by using Mouha *et al.*'s [MWGP11] and Sun *et*

al.'s [SHS+13] model frameworks are solved quickly, and they are used to calculate a lower bound of the minimum number of active S-boxes for word-oriented and bit-oriented ciphers respectively. By solving this model, a rough lower bound of the minimum weight is obtained by multiplying the number of active S-boxes with the minimum weight of a single S-box. Note that if the cipher is bit-oriented and its branch number of an S-box equal to 2, the constraints for modelling an S-box is insufficient, and this model can't provide a useful lower bound.

2. The second kind of model is called a "Tightest" model that provides the tightest lower bound (*i.e.*, the lower bound is exactly the minimum weight). The model built by using Sun *et al.*'s model framework [SHW+14b, SHW+14a] is used to search for the best differential characteristic, and it returns the minimum weight.

3.5 Improved Search Algorithm

Based on the approaches and techniques presented in Section 3.1 to Section 3.4, we are ready to present an MILP-based search algorithm for searching for the best differential characteristics for a block cipher, as illustrated in Algorithm 1. In the following, we give a detailed description of the algorithm.

Given a block cipher and a number of rounds R, Algorithm 1 is used to search for the best r-round differential characteristic, $r \in \{1, 2, ..., R\}$. At the beginning of the search for r rounds, we generate the currently best r-round differential characteristic and obtain an upper bound of the minimum weight (denoted as `UpperBound`) according to Technique 1. Nextly, we initialize the lower bound array `LB` and search the subsets divided for the r-round cipher. During the search, we dynamically update the currently best r-round differential characteristic and `UpperBound`. After searching all the subsets, the currently best r-round differential characteristic is exactly the best r-round differential characteristic, and `UpperBound` equals the minimum weight for the r-round cipher.

Function `InitLBArray()`. This function is used to initialize the lower bound array `LB`, specifically, assign values to $\texttt{LB}[r, N_A, 0, 0]$, $N_A = 1, 2, 3$. These values will be used later when searching the subsets divided in Equation (6). To assign values to $\texttt{LB}[r, N_A, 0, 0]$, we estimate lower bounds of the minimum weight of the differential characteristics within $\mathcal{D}_{r,N_A,0,0}$ by using Methods 1 and 3 (introduced in Section 3.4). In Method 3, a "Rough" model or a "Tightest" model is solved. The "Rough" model is usually easier to be solved than the "Tightest" model for the same subset, thus we preferentially solve the "Rough" model to obtain a lower bound. If the lower bound returned from the "Rough" model is a good one, we don't need to solve the "Tightest" model. Usually, either of the two models will be time-consuming when the number of rounds r exceeds a certain value. Thus Method 3 is adopted only when r is small which makes it possible to solve the model in a short time.

Function `SearchSubset12()`. This function is used to search the subsets $\mathcal{D}_{r,N_A,i,\Delta}$, where $N_A \in \{1, 2\}, i \in \texttt{SearchR}, \Delta$ belongs to the set of all possible input differences of round i that has exactly N_A active S-boxes. We traverse the parameters N_A and i according to Technique 3, and search a specific subset $\mathcal{D}_{r,N_A,i,\Delta}$ in Line 30-36. According to Technique 2, we calculate a lower bound of the minimum weight of the differential characteristics within the subset and use it to determine whether there is a differential characteristic better than the currently best one. We first estimate the lower bound within $\mathcal{D}_{r,N_A,i,\Delta}$ by using the value stored in $\texttt{LB}[r, N_A, 0, 0]$ based on the fact that all differential characteristics in $\mathcal{D}_{r,N_A,i,\Delta}$ belong to $\mathcal{D}_{r,N_A,0,0}$, and we store it in $\texttt{LB}[r, N_A, i, \Delta]$. Then, the value stored in $\texttt{LB}[r, N_A, i, \Delta]$ is updated by using Method 1 and Method 2. There is no model solved until

Function Three functions used to assign values to the lower bound array

```
1  AssignByMethod1(r, N_A, i, Δ)
2  begin
3      if i = 0, Δ = 0 then
4          LB[r, N_A, i, Δ] ← max(LB[r, N_A, i, Δ], the result of Equation (9));
5      end
6      if i ∈ {1, 2, ..., r}, Δ ≠ 0 then
7          LB[r, N_A, i, Δ] ← max(LB[r, N_A, i, Δ], the result of Equation (10));
8      end
9      if i = r + 1, Δ ≠ 0 then
10         LB[r, N_A, i, Δ] ← max(LB[r, N_A, i, Δ], the result of Equation (11));
11     end
12 end
13
14 AssignByMethod2(r, N_A, i, Δ)
15 begin
16     if i < SearchR[0] and the result of Equation (12) is greater than or equal to
         the weight of the currently best differential characteristic (UpperBound) then
17         LB[r, N_A, i, Δ] ← max(LB[r, N_A, i, Δ], UpperBound);
18     end
19     if i > SearchR[0] and the result of Equation (13) is greater than or equal to
         the weight of the currently best differential characteristic (UpperBound) then
20         LB[r, N_A, i, Δ] ← max(LB[r, N_A, i, Δ], UpperBound);
21     end
22 end
23
24 AssignByMethod3(r, N_A, i, Δ, ModelType)
25 begin
26     if ModelType = "Rough" then
27         if cipher is bit-oriented and its branch number of an S-box equal to 2 then
28             Return;
29         end
30         Build an MILP model for calculating a lower bound of the minimum
             number of active S-boxes for an r-round cipher by using Mouha et al.'s
             framework [MWGP11] for word-oriented ciphers or Sun et al.'s
             framework [SHS+13] for bit-oriented ciphers;
31         Add Equations (7-8) into the model;
32         Solve the model;
33         LB[r, N_A, i, Δ] ← max(LB[r, N_A, i, Δ], the number of active S-boxes returned
             from the model multiplies the minimum weight of a single S-box);
34     end
35     if ModelType = "Tightest" then
36         Build an MILP model for searching for the best differential characteristic for
             an r-round cipher by using Sun et al.'s framework [SHW+14b, SHW+14a];
37         Add Equations (7-8) into the model;
38         Solve the model;
39         LB[r, N_A, i, Δ] ← the minimum weight returned from the model;
40     end
41 end
42
```

now. If the current value stored in $\texttt{LB}[r, N_A, i, \Delta]$ is greater than or equal to the weight of the currently best r-round differential characteristic (`UpperBound`), we terminate the search of this subset and search the next one. Otherwise, we estimate a tighter lower bound by calling Function `UpdateLBSubset12`(), and update the value stored in $\texttt{LB}[r, N_A, i, \Delta]$.

Function `UpdateLBSubset12`(). In this function, we estimate lower bounds of the minimum weight of the differential characteristics within the subset $\mathcal{D}_{r,N_A,i,\Delta}$ by solving MILP models, where $N_A \in \{1, 2\}, i \in \texttt{SearchR}, \Delta$ belongs to the set of all possible input differences of round i that has exactly N_A active S-boxes. We observe that the running time of solving an MILP model usually increases dramatically as the number of rounds increases. Therefore, we first solve a model built for the subset of r_1-round differential characteristics, then use Method 1 to estimate lower bounds within the subsets of r_2-round differential characteristics, r_1 increases from 1, $r_2 = r_1 + 1, ..., R$. For each r_1, we obtain a new lower bound within the subset $\mathcal{D}_{r,N_A,i,\Delta}$. The lower bound is constantly updated by increasing the number r_1, until it is greater than or equal to `UpperBound` or it equals the minimum weight of the differential characteristics within the subset. After calling this function, if the value stored in $\texttt{LB}[r, N_A, i, \Delta]$ is still smaller than `UpperBound`, we find a differential characteristic that has a weight smaller than the weight of the currently best r-round differential characteristic.

Function `SearchSubset3`(). This function is used to search the subset $\mathcal{D}_{r,3,0,0}$. A lower bound of the minimum weight of the differential characteristics within $\mathcal{D}_{r,3,0,0}$ had been calculated and it was stored in $\texttt{LB}[r, 3, 0, 0]$. For most lightweight SPN ciphers, the best differential characteristics are likely to have exactly 1 or 2 active S-boxes at a certain round. The value stored in $\texttt{LB}[r, 3, 0, 0]$ is generally greater than or equal to `UpperBound`. In this case, the search of the subset is terminated and no other model needs to be solved.

4 Automatic Security Evaluation against Differential/Linear Cryptanalysis

In this section, we use our improved search algorithm to evaluate the security of block ciphers against differential and linear Cryptanalysis.

4.1 Security Evaluation Against Differential Cryptanalysis

Algorithm 1 is used to search for the best differential characteristics for a block cipher. It can be used to calculate the minimum number of active S-boxes by modifying the aim of searching for the differential characteristic with the minimum weight to that with the minimum number of active S-boxes. At this point, the lower bound array `LB` is used to store a lower bound of the minimum number of active S-boxes. Besides, the "Tightest" model in Method 3 is built to calculate the minimum number of active S-boxes by using Sun *et al.*'s framework [SHW+14b, SHW+14a]. Applying our algorithm to an r-round block cipher, we obtain the minimum weight or its lower bound calculated from the minimum number of active S-boxes. If the minimum weight or its lower bound is greater than or equal to cipher's block size, it can be concluded that the r-round cipher is secure against differential cryptanalysis.

4.2 Security Evaluation Against Linear Cryptanalysis

According to the duality between differential and linear cryptanalysis [Mat94], Algorithm 1 is easily extended to the security evaluation against linear cryptanalysis. The linear

Algorithm 1: Process of searching for the best differential characteristics for SPN ciphers by using a divide-and-conquer approach.

Data: An R-round block cipher;
Result: Best differential characteristics covered from 1 to R rounds.

Global R, `UpperBound`, `LB`;
begin
  for $r \leftarrow 1$ **to** R **do**
    Generate the currently best r-round differential characteristic and an upper bound of the minimum weight `UpperBound` by using Technique 1;
    Call `InitLBArray`(r); `// initialize the lower bound array LB`
    Call `SearchSubset12`(r); `// search Subset-1 and Subset-2`
    Call `SearchSubset3`(r); `// search Subset-3`
    `Weight`[r] $\leftarrow$ `UpperBound`;
  end
  return `Weight`
end

Function `InitLBArray`(r)
begin
  foreach N_A **in** $[1, 2, 3]$ **do**
    Call `AssignByMethod1`($r, N_A, 0, 0$) to initialize `LB`[$r, N_A, 0, 0$];
    foreach $ModelType$ **in** ["Rough","Tightest"] **do**
      if r is a number smaller than a certain value **then**
        Call `AssignByMethod3`($r, N_A, 0, 0, ModelType$) to update `LB`[$r, N_A, 0, 0$];
      end
    end
  end
end

Function `SearchSubset12`(r)
begin
  `// Traverse` N_A `and` i `by a proper order based on Technique 3`
  foreach N_A **in** $[1, 2]$ or $[2, 1]$ **do**
    foreach i **in** `SearchR` **do**
      foreach $\Delta \leftarrow$input difference of round i that has N_A active S-boxes **do**
        `LB`[r, N_A, i, Δ] $\leftarrow \max($`LB`[r, N_A, i, Δ], `LB`[$r, N_A, 0, 0$]);
        Call `AssignByMethod1`(r, N_A, i, Δ) to update `LB`[r, N_A, i, Δ];
        Call `AssignByMethod2`(r, N_A, i, Δ) to update `LB`[r, N_A, i, Δ];
        if `LB`[r, N_A, i, Δ] $<$ `UpperBound` **then**
          Call `UpdateLBSubset12`(r, N_A, i, Δ) to update `LB`[r, N_A, i, Δ];
          `UpperBound` $\leftarrow \min($`UpperBound`, `LB`[r, N_A, i, Δ]);
        end
      end
    end
  end
end

Function `SearchSubset3`(r)
begin
  if `LB`[$r, 3, 0, 0$] $<$ `UpperBound` **then**
    Call `AssignByMethod3`($r_1, 3, 0, 0$,"Tightest") to update `LB`[$r_1, 3, 0, 0$];
    `UpperBound` $\leftarrow \min($`UpperBound`, `LB`[$r, 3, 0, 0$]);
  end
end

```
UpdateLBSubset12(r, N_A, i, Δ)
  // Estimate tighter lower bounds within D_{r,N_A,i,Δ} in two ways:
  // (1) Due to the input difference of round i is determinate to
     equal Δ, the minimum weight within D_{r,N_A,i,Δ} equals the sum of
     minimum weights within D_{i-1,N_A,i,Δ} and D_{r-i+1,N_A,1,Δ}. Based on this
     fact, we first update LB[i-1, N_A, i, Δ] and LB[r-i+1, N_A, 1, Δ] by
     combining Methods 3 and 1, then update LB[r, N_A, i, Δ] by using
     Equation (10) in Method 1;
  // (2) We first update the values in Equations (12,13) by combining
     Methods 3 and 1, then update LB[r, N_A, i, Δ] by using Method 2.
  // When both values stored in LB[i-1, N_A, i, Δ] and LB[r-i+1, N_A, 1, Δ]
     are updated by using Method 3 with ModelType = "Tightest", the
     value stored in LB[r, N_A, i, Δ] after updating by the function is
     exactly the minimum weight within D_{r,N_A,i,Δ}.
begin
    foreach ModelType in ["Rough","Tightest"] do
        r_1 = 1;
        while r_1 ≤ max(i-1, r-i+1) and LB[r, N_A, i, Δ] < UpperBound do
            // Estimate lower bounds according to the first way
            if r_1 ≤ i-1 and LB[r, N_A, i, Δ] < UpperBound then
                Call AssignByMethod3(r_1, N_A, r_1+1, Δ,ModelType);
                Call AssignByMethod1(r_2, N_A, r_2+1, Δ), r_2 = r_1+1, ..., R;
                Call AssignByMethod1(r, N_A, i, Δ) to update LB[r, N_A, i, Δ];
            end
            if r_1 ≤ r-i+1 and LB[r, N_A, i, Δ] < UpperBound then
                Call AssignByMethod3(r_1, N_A, 1, Δ,ModelType);
                Call AssignByMethod1(r_2, N_A, 1, Δ), r_2 = r_1+1, ..., R;
                Call AssignByMethod1(r, N_A, i, Δ) to update LB[r, N_A, i, Δ];
            end
            // Estimate lower bounds according to the second way
            if i < SearchR[0] and LB[r, N_A, i, Δ] < UpperBound then
                R_front ← SearchR[j-1], j is the index of i in SearchR;
                if r_1 ≤ R_front - i + 1 then
                    Call AssignByMethod3(r_1, N_A+1, 1, Δ,ModelType);
                    Call AssignByMethod1(r_2, N_A+1, 1, Δ), r_2 = r_1+1, ..., R;
                    Call AssignByMethod2(r, N_A, i, Δ) to update LB[r, N_A, i, Δ];
                end
            end
            if i > SearchR[0] and LB[r, N_A, i, Δ] < UpperBound then
                R_front ← SearchR[j-1], j is the index of i in SearchR;
                if r_1 ≤ i - R_front then
                    Call AssignByMethod3(r_1, N_A+1, r_1+1, Δ,ModelType);
                    Call AssignByMethod1(r_2, N_A+1, r_2+1, Δ), r_2 = r_1+1, ..., R;
                    Call AssignByMethod2(r, N_A, i, Δ) to update LB[r, N_A, i, Δ];
                end
            end
            r_1 = r_1 + 1;
        end
    end
end
```

characteristics with the maximum absolute linear bias are likely to have a low number of active S-boxes at a certain round, thus we divide the set of all possible linear characteristics into smaller subsets by using the partition method introduced for differential characteristics. In [MWGP11, SHS+13, SHW+14b, SHW+14a], the authors described linear mask propagations through a block cipher, and built MILP models to search for the best linear characteristic and calculate the minimum number of linearly active S-boxes. By using their work, we build an MILP model to search each of the subsets divided. For linear characteristics, the weight is defined as the negative of the binary logarithm of the correlation contribution. Searching for the best linear characteristic or calculating the minimum number of active S-boxes for an r-round cipher is transformed into the search for the r-round linear characteristic with the minimum weight or with the minimum number of active S-boxes. If the minimum weight or its lower bound calculated from the minimum number of active S-boxes is greater than or equal to the half of cipher's block size, it can be concluded that the r-round cipher is secure against linear cryptanalysis.

5 Applications to PRESENT, GIFT-64, RECTANGLE, LBLOCK and TWINE

In this section, we apply our search algorithm to five lightweight block ciphers: three SPN ciphers PRESENT, GIFT-64, RECTANGLE and two Feistel ciphers LBLOCK, TWINE. For each of the five ciphers, we obtain the minimum number of differentially and linearly active S-boxes and find the best differential and linear characteristics. The experimental results are summarized in Tables 2-6, where $\#\{AS_D\}$ and $\#\{AS_L\}$ respectively denote the minimum number of differentially and linearly active S-boxes, Pr_D denotes the probability of the best differential characteristic, and Cor_L denotes the correlation contribution of the best linear characteristic. We study the number of MILP models to be solved in Appendix C, and give examples of best differential and linear characteristics in Appendix D.

Our experiment is performed on a PC (Intel(R) Core(TM) i7-4790 CPU, 3.60 GHz, 10.00GB RAM, 4 cores, Linux), and we use the openly available software Gurobi [Gur] to solve MILP models. We observe that in the experiment, the running time is usually improved when setting MipFocus (a parameter in Gurobi) to 2. Therefore, we implement our algorithm with setting MipFocus equal to 2, and we obtain the results in Table 1 by using the same parameter for a fair comparison. We recommend readers to refer [Gur] for more information and select appropriate parameters for other problems.

5.1 PRESENT

PRESENT is an SPN cipher designed by Bogdanov *et al.* [BKL+07]. The differential and linear branch numbers of PRESENT S-box are equal to 3 and 2 respectively. For PRESENT, we obtain the minimum number of differentially/linearly active S-boxes for up to 31/31 rounds (full rounds) and find the best differential/linear characteristics for up to 18/18 rounds.

Although our results on the minimum number of differentially active S-boxes are the same as those in [SHS+13], they didn't prove the results are exact values because their description for modelling an S-box is rough, while ours is exact. In [Wan08], the authors provided the best differential characteristics for PRESENT for 5 to 10 rounds, and good ones for 11 to 15 rounds, while we cover more rounds. Although the weight of the best differential characteristic for 15-round PRESENT is larger than the block size 64, it can be used to analyze the differential clustering [DR02] of PRESENT, whose clustering effect is very strong as shown in [WSTP12].

Table 2: Experimental results of PRESENT

Rounds	Differential cryptanalysis				Linear cryptanalysis			
	$\#\{AS_D\}$	Time	Pr_D	Time	$\#\{AS_L\}$	Time	Cor_L	Time
1	1	0s	2^{-2}	1s	1	0s	2^{-1}	0s
2	2	1s	2^{-4}	2s	2	2s	2^{-2}	2s
3	4	2s	2^{-8}	3s	3	3s	2^{-4}	71s
4	6	4s	2^{-12}	4s	4	6s	2^{-6}	88s
5	10	5s	2^{-20}	5s	5	9s	2^{-8}	152s
6	12	8s	2^{-24}	249s	6	8s	2^{-10}	128s
7	14	10s	2^{-28}	9s	7	7s	2^{-12}	18s
8	16	11s	2^{-32}	11s	8	8s	2^{-14}	98s
9	18	15s	2^{-36}	14s	9	10s	2^{-16}	15s
10	20	16s	2^{-41}	1298s	10	11s	2^{-18}	300s
11	22	18s	2^{-46}	438s	11	12s	2^{-20}	11s
12	24	22s	2^{-52}	311s	12	14s	2^{-22}	978s
13	26	24s	2^{-56}	22s	13	15s	2^{-24}	14s
14	28	31s	2^{-62}	18859s	14	17s	2^{-26}	3507s
15	30	32s	2^{-66}	2594s	15	19s	2^{-28}	16s
16	32	19s	2^{-70}	370s	16	21s	2^{-30}	3080s
17	34	20s	2^{-74}	20s	17	23s	2^{-32}	16302s
18	36	22s	2^{-78}	629s	18	24s	2^{-34}	14105s
19	38	34s			19	26s		
20	40	29s			20	28s		
21	42	28s			21	30s		
22	44	29s			22	34s		
23	46	37s			23	35s		
24	48	34s			24	37s		
25	50	36s			25	40s		
26	52	38s			26	42s		
27	54	40s			27	44s		
28	56	42s			28	46s		
29	58	42s			29	49s		
30	60	44s			30	49s		
31	62	47s			31	51s		
Total time	740s		6.90h		720s		10.8h	

Table 3: Experimental results of GIFT-64

Rounds	Differential cryptanalysis				Linear cryptanalysis			
	$\#\{AS_D\}$	Time	Pr_D	Time	$\#\{AS_L\}$	Time	Cor_L	Time
1	1	1s	$2^{-1.415}$	1s	1	0s	2^{-1}	0s
2	2	2s	$2^{-3.415}$	47s	2	1s	2^{-2}	2s
3	3	3s	2^{-7}	108s	3	3s	2^{-3}	3s
4	5	69s	$2^{-11.415}$	291s	5	61s	2^{-5}	77s
5	7	61s	2^{-17}	849s	7	60s	2^{-7}	99s
6	10	144s	$2^{-22.415}$	181s	9	65s	2^{-10}	160s
7	13	115s	$2^{-28.415}$	385s	12	177s	2^{-13}	225s
8	16	271s	2^{-38}	19934s	15	243s	2^{-16}	263s
9	18	28s	2^{-42}	32s	18	493s	2^{-20}	8713s
10	20	124s	2^{-48}	7569s	20	681s	2^{-25}	11615s
11	22	77s	2^{-52}	121s	22	392s	2^{-29}	34019s
12	24	19s	2^{-58}	61001s	24	3206s	2^{-31}	14644s
13	26	75s	2^{-62}	604s	26	11229s	2^{-34}	121716s
14	28	15s	2^{-68}	9121s	28	7982s		
15	30	17s	2^{-72}	1595s	30	18410s		
16	32	18s						
Total time	1039s		28.29h		11.95h		53.2h	

Table 4: Experimental results of RECTANGLE

Rounds	Differential cryptanalysis				Linear cryptanalysis			
	$\#\{AS_D\}$	Time	Pr_D	Time	$\#\{AS_L\}$	Time	Cor_L	Time
1	1	1s	2^{-2}	1s	1	1s	2^{-1}	0s
2	2	1s	2^{-4}	1s	2	1s	2^{-2}	1s
3	3	1s	2^{-7}	8s	3	1s	2^{-4}	5s
4	4	2s	2^{-10}	27s	4	2s	2^{-6}	9s
5	6	11s	2^{-14}	128s	6	6s	2^{-8}	41s
6	8	13s	2^{-18}	6s	8	8s	2^{-10}	6s
7	11	11s	2^{-25}	17s	10	5s	2^{-13}	15s
8	13	11s	2^{-31}	28s	12	9s	2^{-16}	24s
9	15	11s	2^{-36}	41s	14	11s	2^{-19}	78s
10	17	25s	2^{-41}	96s	16	25s	2^{-22}	260s
11	19	47s	2^{-46}	297s	18	38s	2^{-25}	1772s
12	21	120s	2^{-51}	669s	20	131s	2^{-28}	5927s
13	23	597s	2^{-56}	2798s	22	428s	2^{-31}	31491s
14	25	2218s	2^{-61}	12410s	24	1615s	2^{-34}	177473s
15	27	12753s	2^{-66}	40989s	26	5588s		
16	29	36891s			28	21352s		
Total time	14.64h		15.98h		8.12h		60.31h	

5.2 GIFT-64

GIFT [BPP+17] is an SPN cipher which is similar to PRESENT. It has two versions: GIFT-64 and GIFT-128, whose block sizes are 64 bits and 128 bits respectively. In this paper, we focus on version GIFT-64. Both differential and linear branch numbers of GIFT-64 S-box are equal to 2. For GIFT-64, we obtain the minimum number of differentially/linearly active S-boxes for up to 16/15 rounds and find the best differential/linear characteristics for up to 15/13 rounds.

The designers of GIFT gave a 9-round differential characteristic with the probability $2^{-44.415}$. Then in [ZDY19], Zhu *et al.* provided 9/12/13-round differential characteristics with probabilities $2^{-42}/2^{-60}/2^{-64}$ based on MILP technique. Recently, Li *et al.* [LWZZ19] found better 12/13-round differential characteristics with probabilities $2^{-58}/2^{-62}$ by using MILP technique. However, the differential characteristics they found are not proven the best ones, while we find the best differential characteristics covered from 1 to 15 rounds.

5.3 RECTANGLE

RECTANGLE is an SPN cipher proposed in 2015 [ZBL+15]. Both differential and linear branch numbers of RECTANGLE S-box are equal to 2. For RECTANGLE, we obtain the minimum number of differentially/linearly active S-boxes for up to 16/16 rounds and find the best differential/linear characteristics for up to 15/14 rounds.

The 64-bits plaintext of RECTANGLE is pictured as a rectangular array with 4 rows and 16 columns. The permutation of RECTANGLE is a left rotation to each row. If there is exactly 1 or 2 active S-boxes at a certain round, the index of the 1st active S-box has no influence on the minimum number of active S-boxes (or minimum weight) for r-round RECTANGLE. Therefore, different from the partition of the set of all possible r-round differential/linear characteristics for PRESENT and GIFT, we divide the set for RECTANGLE into $(1 \times (2^4-1) \times r + 15 \times (2^4-1)^2 \times r + 1)$ smaller subsets. The number of resulting subsets for RECTANGLE is much less than those for PRESENT and GIFT.

5.4 LBLOCK and TWINE

LBLOCK [WZ11] and TWINE [SMMK12] are two similar Feistel ciphers, and both them are word-oriented ciphers. For LBLOCK, we obtain the minimum number of differen-

Table 5: Experimental results of LBLOCK

Rounds	Differential cryptanalysis				Linear cryptanalysis			
	$\#\{AS_D\}$	Time	Pr_D	Time	$\#\{AS_L\}$	Time	Cor_L	Time
1	0	0s	2^0	0s	0	0s	2^{-0}	0s
2	1	0s	2^{-2}	1s	1	0s	2^{-1}	0s
3	2	0s	2^{-4}	0s	2	0s	2^{-2}	0s
4	3	1s	2^{-6}	1s	3	0s	2^{-3}	0s
5	4	1s	2^{-8}	1s	4	1s	2^{-4}	1s
6	6	1s	2^{-12}	1s	6	1s	2^{-6}	1s
7	8	1s	2^{-16}	1s	8	1s	2^{-8}	2s
8	11	2s	2^{-22}	2s	11	2s	2^{-11}	2s
9	14	2s	2^{-28}	2s	14	2s	2^{-14}	2s
10	18	6s	2^{-36}	6s	18	6s	2^{-18}	8s
11	22	4s	2^{-44}	4s	22	4s	2^{-22}	4s
12	24	5s	2^{-48}	5s	24	6s	2^{-24}	5s
13	27	25s	2^{-56}	812s	27	38s	2^{-27}	2103s
14	30	8s	2^{-62}	848s	30	10s	2^{-30}	15s
15	32	19s	2^{-66}	820s	32	28s	2^{-33}	5669s
16	35	30s	2^{-72}	6002s	35	55s		
17	36	29s			36	31s		
18	39	10s			39	11s		
19	41	190s			41	6s		
20	44	18 28s			44	40s		
Total time	352s		2.36h		242s		2.17h	

Table 6: Experimental results of TWINE

Rounds	Differential cryptanalysis				Linear cryptanalysis			
	$\#\{AS_D\}$	Time	Pr_D	Time	$\#\{AS_L\}$	Time	Cor_L	Time
1	0	0s	2^0	0s	0	0s	2^0	0s
2	1	0s	2^{-2}	0s	1	0s	2^{-1}	0s
3	2	0s	2^{-4}	0s	2	0s	2^{-2}	1s
4	3	0s	2^{-6}	1s	3	1s	2^{-3}	1s
5	4	1s	2^{-8}	1s	4	1s	2^{-4}	1s
6	6	1s	2^{-12}	1s	6	1s	2^{-6}	2s
7	8	1s	2^{-16}	1s	8	2s	2^{-8}	2s
8	11	1s	2^{-22}	2s	11	3s	2^{-11}	3s
9	14	2s	2^{-28}	2s	14	3s	2^{-14}	4s
10	18	6s	2^{-38}	52s	18	10s	2^{-18}	17s
11	22	4s	2^{-46}	49s	22	4s	2^{-22}	12s
12	24	4s	2^{-51}	63s	24	6s	2^{-24}	8s
13	27	24s	2^{-58}	7905s	27	53s	2^{-27}	364s
14	30	14s	2^{-64}	17153s	30	12s	2^{-30}	16s
15	32	14s	2^{-68}	28840s	32	37s	2^{-32}	261s
16	35	19s			35	49s	2^{-35}	66s
17	36	17s			36	57s		
18	39	9s			39	11s		
19	41	5s			41	7s		
20	44	17s			44	42s		
Total time	139s		15.02h		299s		758s	

tially/linearly active S-boxes for up to 20/20 rounds and find the best differential/linear characteristics for up to 16/15 rounds. For TWINE, we obtain the minimum number of differentially/linearly active S-boxes for up to 20/20 rounds and find the best differential/linear characteristics for up to 15/16 rounds.

6 Conclusion

In this paper, we propose a new MILP-based search algorithm for the security evaluation against differential/linear cryptanalysis by incorporating the idea of a divide-and-conquer approach. For the search for an r-round block cipher, we first divide the set of all possible r-round differential/linear characteristics into several subsets, then separately search each subset; we also use several techniques to early terminate the search of a subset, which improves efficiency remarkably; finally the optimal solutions within smaller subsets are combined to give the optimal solution within the whole set. As a result, we obtain a more efficient search algorithm.

We only apply our new algorithm to five lightweight block ciphers in this paper. We point out that the permutation layers of these five ciphers are all bit permutations. For each of five ciphers, the best differential and linear characteristics we found have a low number (0, 1 or 2) of active S-boxes at a certain round. In future work, we will consider applying our algorithm to the ciphers with stronger permutation layer, such as AES [DR02], NOEKEON [DPAR00], SERPENT [BAK98], etc. In Tables 2-6, although the weights of the best differential/linear characteristics for some reduced rounds are larger than cipher's block size n/half of n, we argue that it is possibly useful when the differential/linear clustering is taken into consideration [DR02, Nyb94]. We explain how to search for related-key differential characteristics [Bih93] in Appendix B, but it seems to be more difficult and requires more work. We leave these researches as our future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China (No.61379138).

References

[AST+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics. *IACR Trans. Symmetric Cryptol.*, 2017(4):99–129, 2017.

[BAK98] Eli Biham, Ross J. Anderson, and Lars R. Knudsen. Serpent: A New Block Cipher Proposal. In *Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998, Proceedings*, pages 222–238, 1998.

[Bih93] Eli Biham. New Types of Cryptanalytic Attacks Using related Keys. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 398–409, 1993.

[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

Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, pages 123–153, 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 *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, pages 450–466, 2007.

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 321–345, 2017.

[BS91] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.

[BSS+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. https://eprint.iacr.org/2013/404.

[DPAR00] Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie Proposal: NOEKEON. 2000. http://gro.noekeon.org.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[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 *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 268–288, 2016.

[Gur] http://www.gurobi.com.

[LWZZ19] Lingchen Li, Wenling Wu, Yafei Zheng, and Lei Zhang. The Relationship between the Construction and Solution of the MILP Models and Applications. Cryptology ePrint Archive, Report 2019/049, 2019. https://eprint.iacr.org/2019/049.

[Mat93] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 386–397, 1993.

[Mat94] Mitsuru Matsui. On Correlation Between the Order of S-boxes and the Strength of DES. In *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, pages 366–375, 1994.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, pages 57–76, 2011.

[Nyb94] Kaisa Nyberg. Linear Approximation of Block Ciphers. In *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, pages 439–444, 1994.

[sag] http://www.sagemath.org.

[SHQ+15] Siwei Sun, Lei Hu, Kexin Qiao, Xiaoshuang Ma, Jinyong Shan, and Ling Song. Improvement on the Method for Automatic Differential Analysis and Its Application to Two Lightweight Block Ciphers DESL and LBlock-s. In *Advances in Information and Computer Security - 10th International Workshop on Security, IWSEC 2015, Nara, Japan, August 26-28, 2015, Proceedings*, pages 97–111, 2015.

[SHS+13] Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang. Automatic Security Evaluation of Block Ciphers with S-bP Structures Against Related-Key Differential Attacks. In *Information Security and Cryptology - 9th International Conference, Inscrypt 2013, Guangzhou, China, November 27-30, 2013, Revised Selected Papers*, pages 39–51, 2013.

[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. https://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 *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*, pages 158–178, 2014.

[SMMK12] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A Lightweight Block Cipher for Multiple Platforms. In *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, pages 339–354, 2012.

[ST17] Yu Sasaki and Yosuke Todo. 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*, pages 150–165, 2017.

[Wan08] Meiqin Wang. Differential Cryptanalysis of Reduced-Round PRESENT. In *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*, pages 40–49, 2008.

[WSTP12] Meiqin Wang, Yue Sun, Elmar Tischhauser, and Bart Preneel. A Model for Structure Attacks, with Applications to PRESENT and Serpent. In *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, pages 49–68, 2012.

[WW11] Shengbao Wu and Mingsheng Wang. Security evaluation against differential cryptanalysis for block cipher structures. Cryptology ePrint Archive, Report 2011/551, 2011. https://eprint.iacr.org/2011/551.

[WZ11] Wenling Wu and Lei Zhang. LBlock: A Lightweight Block Cipher. In *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, pages 327–344, 2011.

[ZBL+15] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. *SCIENCE CHINA Information Sciences*, 58(12):1–15, 2015.

[ZDY19] Baoyu Zhu, Xiaoyang Dong, and Hongbo Yu. MILP-Based Differential Attack on Round-Reduced GIFT. In *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, pages 372–390, 2019.

[ZSCH18] Yingjie Zhang, Siwei Sun, Jiahao Cai, and Lei Hu. Speeding up MILP aided differential characteristic search with matsui's strategy. In *Information Security - 21st International Conference, ISC 2018, Guildford, UK, September 9-12, 2018, Proceedings*, pages 101–115, 2018.

A Search Algorithm for Feistel Ciphers

In this section, we evaluate the security of Feistel ciphers against differential/linear cryptanalysis by using the idea of a divide-and-conquer approach.

Dividing the Set of All Possible Differential/Linear Characteristics. For Feistel ciphers, the differential/linear characteristics with the highest probability/absolute linear bias are likely to have no active S-box at a certain round. Therefore, we first divide the set of all possible r-round differential/linear characteristics for an r-round Feistel cipher into:

Subset-0 In this kind of subset, differential/linear characteristics have no active S-box at a certain round;

Subset-1 In this subset, differential/linear characteristics have at least one active S-box in each round.

For Subset-0, we further divide it by fixing the index i ($i = 1, 2, ..., r$), such that round i contains no active S-box. The subsets divided now are not small enough, thus we further divide the set by traversing the difference/linear mask patterns of the S-boxes at the $(i+1)$th round. The reason why we don't divide the set by traversing the input difference of round i is that in this partition, the number of resulting subsets will be too large.

Similar to the case for SPN ciphers, we use $\mathcal{D}_{r,N_A,i,P}$ to denote the subset of r-round differential/linear characteristics that satisfy the following two constraints:

Constraint 1 there are at least N_A active S-boxes in each round except for round i;

Constraint 2 the difference/linear mask patterns of the S-boxes at round i and $(i+1)$ are equal to zeros and P respectively.

When $i = 0$, there is no constraint on the difference/linear mask patterns, and we ignore the second constraint by (not rigorously) writing $P = 0$. Differential/linear characteristics in the subset are only required to satisfy the Constraint 1, *i.e.*, there are at least N_A active S-boxes in each round. And if $i = r$, based on the Feistel structure, we regard the constraint of "difference/linear mask patterns of the S-boxes at round $(i+1)$ equal P" as the constraint of "difference/linear mask patterns of the S-boxes at round $(i-1)$ are determined by P".

In summary, the set of all possible r-round differential/linear characteristics for an r-round Feistel cipher is divided into the following subsets:

$$\left(\bigcup_{i,P} \mathcal{D}_{r,0,i,P}\right) \bigcup \mathcal{D}_{r,1,0,0}, \tag{14}$$

where $i \in \{1, 2, ..., r\}$, P belongs to the set of all possible difference/linear mask patterns of the S-boxes at round $(i+1)$. Take LBLOCK as an example, its block size is 64 bits and it uses a 4×4 S-box. Thus it applied 8 S-boxes in each round. If round i has no active S-box, there is at least one active S-box at round $(i+1)$. Based on Equation (14), the set of all possible differential/linear characteristic for r-round LBLOCK is divided into $(r \times (2^8 - 1) + 1)$ subsets.

Building MILP Models for Searching Subsets. For a subset divided in Equation (14), we build an MILP model to search it. Let $A_{j,k}$ denote the word-level input difference of the kth S-box at round j, Constraint 1 is described by Equation (7), and Constraint 2 is described by:

$$\begin{cases} A_{i,k} = 0, k \in \{0, 1, ..., N_S - 1\}; \\ A_{i+1,k} = p_k, k \in \{0, 1, ..., N_S - 1\}, \end{cases} \tag{15}$$

where N_S is the total number of S-boxes in each round, and P is expressed as $P = (p_0, p_1, ..., p_{N_S-1})$. By using Sun *et al.*'s model framework [SHW+14b, SHW+14a] and Equations (7,15), we build an MILP model whose feasible region is the subset.

Search Algorithm for Feistel Ciphers. For a subset named $\mathcal{D}_{r,N_A,i,P}$, we use $\texttt{LB}[r, N_A, i, P]$ to store a lower bound of the minimum weight (or number of active S-boxes) of the differential/linear characteristics within it. Techniques for the improvement and methods for calculating lower bounds introduced for SPN ciphers are also used for Feistel ciphers. It should be noted that for the subset $\mathcal{D}_{r,0,i,P}$ with $i \neq 0, P \neq 0$, difference patterns of the S-boxes at round i equal zeros, but the input difference of round i is indeterminate. Therefore, the sum of the minimum weights (or numbers of active S-boxes) of the differential/linear characteristics within $\mathcal{D}_{i,0,i,P}$ and $\mathcal{D}_{r-i+1,0,1,P}$ is smaller than or equal to the minimum weight (or number of active S-boxes) of the differential/linear characteristics within $\mathcal{D}_{r,0,i,P}$. To obtain the minimum weight (or number of active S-boxes) of the differential/linear characteristics within $\mathcal{D}_{r,0,i,P}$, we need to solve an MILP model whose feasible region is $\mathcal{D}_{r,0,i,P}$. The process of searching for the best differential characteristics for Feistel ciphers is illustrated in Algorithm 2.

B Security Evaluation against Related-Key Differential Cryptanalysis

In this section, we extend our search algorithm to search for the optimal related-key differential characteristic. We observe that the related-key differential characteristics with the minimum number of active S-boxes or with the maximum differential probability are likely to have no active S-box at a certain round. Therefore, we use the partition method introduced for Feistel ciphers (see Appendix A) to divide the set of all possible related-key differential characteristics for an r-round cipher. In [SHS+13, SHW+14b, SHW+14a], the authors described the differential behaviour of the key schedule and introduced the related-key model. The model can be used to both calculate the minimum number of related-key differentially active S-boxes and search for the best related-key differential characteristic. By using previous work, we build a related-key model to search each of the subsets divided. We point out that the search of a related-key subset is more difficult than the search of a single-key subset. This is mainly because in a related-key subset, there are plenty of possibilities of round keys. Besides, a related-key model is more difficult to be solved than a single-key model because the size of the related-key model is larger. Applying our algorithm to related-key differential cryptanalysis seems to require a large calculation, thus more techniques are needed to improve efficiency. A method to shorten the running time is to interrupt the solving process of a model at some point when the model is proven to have no better solution than the obtained currently best one (it can be done with Gurobi software). Moreover, if good solutions are found by searching a reduced number of subsets, the remaining subsets can be discarded. By doing this, good solutions rather than the best ones are obtained in a shorter time.

C Number of MILP Models Solved in Our Algorithm

In previous MILP-based work, a problem is transformed into an MILP model. While in our search algorithm, a problem is solved by solving several MILP models that are easier to be solved. In this section, we study the number of models to be solved in our search algorithm.

Take searching for the best differential characteristics for the first 9-round PRESENT as an example, we list the number of models that have been solved in Table 7. PRESENT

Algorithm 2: Process of searching for the best differential characteristic for Feistel ciphers by using a divide-and-conquer approach.

Data: An R-round cipher;
Result: Best differential characteristics covered from 1 to R rounds.

Global R, `UpperBound`, `LB`;
begin
  for $r \leftarrow 1$ **to** R **do**
    Generate the currently best r-round differential characteristic and an upper bound of the minimum weight `UpperBound` by using Technique 1;
    Call `InitLBArray`(r);
    Call `SearchSubset0`(r);
    Call `SearchSubset1`(r);
    `Weight`[r] $\leftarrow$ `UpperBound`;
  end
  return `Weight`
end

Function `InitLBArray`(r)
begin
  foreach N_A **in** $[0, 1]$ **do**
    Initialize `LB`$[r, N_A, 0, 0]$ by using Method 1;
    if r is a number smaller than a certain value **then**
      Update `LB`$[r, N_A, 0, 0]$ by using Method 3;
    end
  end
end

Function `SearchSubset0`(r)
begin
  foreach i **in** `SearchR` **do**
    foreach $P \leftarrow$ difference patterns of the S-boxes at round $(i+1)$ **do**
      // search the subset $\mathcal{D}_{r,0,i,P}$.
      `LB`$[r, 0, i, P] \leftarrow \max($`LB`$[r, 0, i, P],$ `LB`$[r, 0, 0, 0])$;
      Update `LB`$[r, 0, i, P]$ by using Method 1 and Method 2;
      if `LB`$[r, 0, i, P] <$ `UpperBound` **then**
        Update `LB`$[r, 0, i, P]$ similar to the case for SPN ciphers;
        `UpperBound` $\leftarrow \min($`UpperBound`$,$ `LB`$[r, 0, i, P])$;
      end
    end
  end
end

Function `SearchSubset1`(r)
begin
  // search the subset $\mathcal{D}_{r,1,0,0}$
  if `LB`$[r, 1, 0, 0] <$ `UpperBound` **then**
    Update `LB`$[r, 1, 0, 0]$ by using Method 3;
    `UpperBound` $\leftarrow \min($`UpperBound`$,$ `LB`$[r, 1, 0, 0])$;
  end
end

Table 7: Number of MILP models we solved for searching for the best differential characteristics for r-round PRESENT by using our algorithm

r	r_1 (number of rounds of the model built for)								
	1	2	3	4	5	6	7	8	9
1	3+1								
2	0+0	3+0							
3	0+0	0+0	3+0						
4	0+0	0+0	0+0	3+0					
5	0+0	0+0	0+0	0+0	3+0				
6	11958+108	888+63	108+8	0+0	0+0	3+0			
7	0+0	0+0	0+0	0+0	0+0	0+0	3+0		
8	0+0	0+0	0+0	0+0	0+0	0+0	0+0	3+0	
9	0+0	0+0	0+0	0+0	0+0	0+0	0+0	0+0	3+0

The item "$*+*$" denotes the number of "Rough" models adding the number of "Tightest" models (introduced in Method 3 in Section3.4).

is a bit-oriented block cipher, and its differential branch number of an S-box is equal to 3. In our algorithm, "Rough" models and "Tightest" models (introduced in Method 3 in Section3.4) are used to estimate lower bounds of the minimum weights of the differential characteristics within the subsets divided. From Table 7 we see that, to search for the best differential characteristics for r-round PRESENT, several MILP models are solved. These models are built for r_1 rounds, $1 \leq r_1 \leq r$, and most of them are easy to be solved.

For the search for $r = 1$ rounds, we obtain the best 1-round differential characteristic by solving one "Tightest" model built for 1 round. Moreover, we solve three "Rough" models whose feasible regions are $\mathcal{D}_{1,N_A,0,0}$ to assign values to $\texttt{LB}[r, N_A, 0, 0]$, $N_A = 1, 2, 3$.

For the search for $r = 2, 3, 4, 5, 7, 8, 9$ rounds, the currently best r-round differential characteristic generated by using Technique 1 is exactly the best one. Three "Rough" models built for r rounds are solved to assign values to $\texttt{LB}[r, N_A, 0, 0]$, $N_A = 1, 2, 3$. When searching an arbitrary subset, the lower bound of the minimum weight of the differential characteristics within the subset, namely, the value stored in the corresponding lower bound array is greater than or equal to the weight of the currently best r-round differential characteristic. Therefore, no other model is solved.

For the search for $r = 6$ rounds, the currently best r-round differential characteristic generated by using Technique 1 is not the best one. Except for the three "Rough" models built for r rounds are solved to assign values to $\texttt{LB}[r, N_A, 0, 0]$, $N_A = 1, 2, 3$, several "Rough" and "Tightest" models built for r_1 rounds are solved to search the subsets divided, $1 \leq r_1 \leq r$.

D Examples of Best Differential and Linear Characteristics

We provide the best differential and linear characteristics searched by implementing our search algorithm.

Table 8: The best differential characteristic with probability 2^{-78} for 18-round PRESENT.

Rounds	input difference of S-boxes	output difference of S-boxes	probability
1^{th}	0x0000000000001001	0x0000000000009009	2^{-4}
2^{th}	0x0009000000000009	0x0004000000000004	2^{-4}
3^{th}	0x0000100100000000	0x0000300300000000	2^{-4}
4^{th}	0x0000000009000900	0x0000000004000400	2^{-4}
5^{th}	0x0000004400000000	0x0000005500000000	2^{-4}
6^{th}	0x0000030000000300	0x0000010000000100	2^{-6}
7^{th}	0x0000000000000404	0x0000000000000505	2^{-4}
8^{th}	0x0000000500000005	0x0000000100000001	2^{-6}
9^{th}	0x0000000000000101	0x0000000000000909	2^{-4}
10^{th}	0x0005000000000005	0x0001000000000001	2^{-6}
11^{th}	0x0000000000001001	0x0000000000009009	2^{-4}
12^{th}	0x0009000000000009	0x0004000000000004	2^{-4}
13^{th}	0x0000100100000000	0x0000900900000000	2^{-4}
14^{th}	0x0900000000000900	0x0400000000000400	2^{-4}
15^{th}	0x0000400400000000	0x0000500500000000	2^{-4}
16^{th}	0x0000090000000900	0x0000040000000400	2^{-4}
17^{th}	0x0000040400000000	0x0000050500000000	2^{-4}
18^{th}	0x0000050000000500	0x00000c0000000c00	2^{-4}

Table 9: The best linear characteristic with correlation 2^{-34} for 18-round PRESENT.

Rounds	input mask of S-boxes	output mask of S-boxes	correlation
1^{th}	0x0d00000000000000	0x0200000000000000	2^{-1}
2^{th}	0x0000000040000000	0x0000000040000000	2^{-2}
3^{th}	0x0000008000000000	0x0000002000000000	2^{-2}
4^{th}	0x0000000002000000	0x0000000004000000	2^{-2}
5^{th}	0x0000004000000000	0x0000008000000000	2^{-2}
6^{th}	0x0200000000000000	0x0400000000000000	2^{-2}
7^{th}	0x0000400000000000	0x0000200000000000	2^{-2}
8^{th}	0x0000000008000000	0x0000000002000000	2^{-2}
9^{th}	0x0000000000400000	0x0000000000400000	2^{-2}
10^{th}	0x0000002000000000	0x0000004000000000	2^{-2}
11^{th}	0x0000020000000000	0x0000020000000000	2^{-2}
12^{th}	0x0000000004000000	0x0000000002000000	2^{-2}
13^{th}	0x0000000000400000	0x0000000000200000	2^{-2}
14^{th}	0x0000000000200000	0x0000000000200000	2^{-2}
15^{th}	0x0000000000200000	0x0000000000800000	2^{-2}
16^{th}	0x0020000000000000	0x0020000000000000	2^{-2}
17^{th}	0x0000000020000000	0x0000000080000000	2^{-2}
18^{th}	0x0080000000000000	0x00f0000000000000	2^{-1}

Table 10: The best differential characteristic with probability 2^{-72} for 15-round GIFT-64

Rounds	input difference of S-boxes	output difference of S-boxes	probability
1^{th}	0x000f0000000c0000	0x000400000004000	2^{-4}
2^{th}	0x0000404000000000	0x000050500000000	2^{-4}
3^{th}	0x0500000005000000	0x020000000200000	2^{-6}
4^{th}	0x2020000000000000	0x505000000000000	2^{-4}
5^{th}	0x5000000050000000	0x200000002000000	2^{-6}
6^{th}	0x0000202000000000	0x000050500000000	2^{-4}
7^{th}	0x0500000005000000	0x020000000200000	2^{-6}
8^{th}	0x2020000000000000	0x505000000000000	2^{-4}
9^{th}	0x5000000050000000	0x200000002000000	2^{-6}
10^{th}	0x0000202000000000	0x000050500000000	2^{-4}
11^{th}	0x0500000005000000	0x020000000200000	2^{-6}
12^{th}	0x2020000000000000	0x505000000000000	2^{-4}
13^{th}	0x5000000050000000	0x200000002000000	2^{-6}
14^{th}	0x0000202000000000	0x000050500000000	2^{-4}
15^{th}	0x0500000005000000	0x0f0000000f000000	2^{-4}

Table 11: The best linear characteristic with correlation 2^{-34} for 13-round GIFT-64

Rounds	input mask of S-boxes	output mask of S-boxes	correlation
1^{th}	0x0c0c000000000000	0x0101000000000000	2^{-2}
2^{th}	0x0000100000001000	0x0000800000008000	2^{-2}
3^{th}	0x0000000000000808	0x0000000000000505	2^{-2}
4^{th}	0x0000000500000005	0x0000000a0000000a	2^{-2}
5^{th}	0x0808000002020000	0x0505000005050000	2^{-4}
6^{th}	0x0000505000005050	0x0000a0a00000a0a0	2^{-4}
7^{th}	0x00000a0a00000a0a	0x0000020800000208	2^{-6}
8^{th}	0x0a0a000000000000	0x0208000000000000	2^{-3}
9^{th}	0xa000000000000000	0x2000000000000000	2^{-1}
10^{th}	0x0000200000000000	0x0000800000000000	2^{-2}
11^{th}	0x0000000000000800	0x0000000000000500	2^{-1}
12^{th}	0x0000000100000004	0x0000000800000006	2^{-2}
13^{th}	0x0800000400020000	0x0500000600050000	2^{-3}

Table 12: The best differential characteristic with probability 2^{-66} for 15-round RECTANGLE.

Rounds	input difference of S-boxes	output difference of S-boxes	probability
1^{th}	0x0000f00009000000	0x0000200006000000	2^{-4}
2^{th}	0x0000000060000200	0x0000000020000600	2^{-5}
3^{th}	0x0200000000006000	0x0600000000002000	2^{-5}
4^{th}	0x6000020000000000	0x2000060000000000	2^{-5}
5^{th}	0x0000600002000000	0x0000200006000000	2^{-5}
6^{th}	0x0000000060000200	0x0000000020000600	2^{-5}
7^{th}	0x0200000000006000	0x0600000000002000	2^{-5}
8^{th}	0x6000020000000000	0x2000060000000000	2^{-5}
9^{th}	0xa000060000200000	0x0000200006000000	2^{-5}
10^{th}	0x0000000060000200	0x0000000020000600	2^{-5}
11^{th}	0x0000000000006800	0x0000000000002100	2^{-5}
12^{th}	0x3000000000000000	0x8000000000000000	2^{-3}
13^{th}	0x8000000000000000	0x1000000000000000	2^{-3}
14^{th}	0x0001000000000000	0x0006000000000000	2^{-2}
15^{th}	0x0040000200000000	0x00f0000d00000000	2^{-4}

Table 13: The best linear characteristic with correlation 2^{-34} for 14-round RECTANGLE.

Rounds	input mask of S-boxes	out mask of S-boxes	correlation
1^{th}	0x000f0000dc000000	0x0002000021000000	2^{-3}
2^{th}	0x0000000200003000	0x0000000a00004000	2^{-2}
3^{th}	0x0000000800060000	0x00000003000c0000	2^{-2}
4^{th}	0x00000000005a0000	0x0000000000850000	2^{-4}
5^{th}	0x0000000000c00010	0x0000000000100050	2^{-2}
6^{th}	0x0100000000000500	0x0800000000000a00	2^{-4}
7^{th}	0x0a00000000000800	0x0800000000000300	2^{-2}
8^{th}	0x1a00000000000000	0x8400000000000000	2^{-4}
9^{th}	0xc000000000000000	0x1000000000000000	2^{-1}
10^{th}	0x0001000000000000	0x0008000000000000	2^{-2}
11^{th}	0x0008000000000000	0x0004000000000000	2^{-2}
12^{th}	0x0040000000000000	0x0060000000000000	2^{-1}
13^{th}	0x0400002000000000	0x0600006000000000	2^{-2}
14^{th}	0x4000060000200000	0x6000080000600000	2^{-3}

Table 14: The best differential characteristic with probability 2^{-72} for 16-round LBLOCK.

Rounds	input difference of S-boxes	output difference of S-boxes	probability
1^{th}	0x00424000	0x00218000	2^{-6}
2^{th}	0x00040000	0x00020000	2^{-2}
3^{th}	0x40400000	0x40200000	2^{-4}
4^{th}	0x04420000	0x04650000	2^{-6}
5^{th}	0x05060040	0x04020040	2^{-6}
6^{th}	0x00000000	0x00000000	2^{0}
7^{th}	0x06004005	0x0100c00a	2^{-6}
8^{th}	0x10000ac0	0xf0000260	2^{-8}
9^{th}	0x00b02500	0x00a01c00	2^{-6}
10^{th}	0x00000000	0x00000000	2^{0}
11^{th}	0xb0250000	0x20120000	2^{-6}
12^{th}	0x02210000	0x02150000	2^{-6}
13^{th}	0x000100b0	0x00010020	2^{-4}
14^{th}	0x20000000	0xa0000000	2^{-2}
15^{th}	0x01a0b000	0x0a102000	2^{-6}
16^{th}	0xa0010000	0xe0050000	2^{-4}

Table 15: The best linear characteristic with correlation 2^{-33} for 15-round LBLOCK.

Rounds	input mask of S-boxes	output mask of S-boxes	correlation
1^{th}	0x00000000	0x00000000	2^{0}
2^{th}	0xd09d0000	0x10590000	2^{-3}
3^{th}	0x08053000	0x090dd000	2^{-3}
4^{th}	0x9d0000d0	0x31000010	2^{-3}
5^{th}	0x00000000	0x00000000	2^{0}
6^{th}	0x02007007	0x03001001	2^{-3}
7^{th}	0x30000033	0x70000027	2^{-3}
8^{th}	0x00700702	0x00100203	2^{-3}
9^{th}	0x00000000	0x00000000	2^{0}
10^{th}	0xf07b0000	0x20310000	2^{-3}
11^{th}	0x03032000	0x070bf000	2^{-3}
12^{th}	0x7b0000f0	0x22000020	2^{-3}
13^{th}	0x00000000	0x00000000	2^{0}
14^{th}	0x0f001009	0x02002002	2^{-3}
15^{th}	0xe00000c1	0x100000f9	2^{-3}

Table 16: The best differential characteristic with probability 2^{-68} for 15-round TWINE.

Rounds	input difference of S-boxes	output difference of S-boxes	probability
1^{th}	0x00000000	0x00000000	2^{0}
2^{th}	0xca0c0000	0xe7020000	2^{-7}
3^{th}	0xe0702000	0xa090a000	2^{-7}
4^{th}	0x00ca0050	0x00e70020	2^{-6}
5^{th}	0x00000000	0x00000000	2^{0}
6^{th}	0x00a0c005	0x00702002	2^{-7}
7^{th}	0x00020072	0x000a009a	2^{-6}
8^{th}	0x0500050a	0x02000207	2^{-6}
9^{th}	0x00000000	0x00000000	2^{0}
10^{th}	0x5a050000	0x27020000	2^{-6}
11^{th}	0x20702000	0xa050a000	2^{-7}
13^{th}	0x005a0000	0x00270000	2^{-4}
14^{th}	0x00000200	0x00000a00	2^{-2}
15^{th}	0x0aa05000	0x07702000	2^{-6}
16^{th}	0x00700070	0x00900090	2^{-4}

Table 17: The best linear characteristic with correlation 2^{-35} for 16-round TWINE.

Rounds	input mask of S-boxes	output mask of S-boxes	correlation
1^{th}	0x00000000	0x00000000	2^{0}
2^{th}	0xc60c0000	0x12010000	2^{-3}
3^{th}	0xb2200000	0x6cc00000	2^{-3}
4^{th}	0x60c000c0	0x20a00010	2^{-3}
5^{th}	0x00000000	0x00000000	2^{0}
6^{th}	0x0060c00c	0x0020100a	2^{-3}
7^{th}	0x000b0220	0x00060cc0	2^{-3}
8^{th}	0x0c006c00	0x0a002a00	2^{-3}
9^{th}	0x00000000	0x00000000	2^{0}
10^{th}	0xce010000	0xa20a0000	2^{-3}
11^{th}	0xad200000	0xe1c00000	2^{-3}
12^{th}	0xe00000c0	0xd00000a0	2^{-2}
13^{th}	0x000c0000	0x00010000	2^{-1}
14^{th}	0x02e01000	0x0cd0a000	2^{-3}
15^{th}	0xe00a0000	0x200e0000	2^{-2}
16^{th}	0x00a2000e	0x00ec000d	2^{-3}