Designs, Codes and Cryptography
https://doi.org/10.1007/s10623-023-01259-9

New method for combining Matsui's bounding conditions with sequential encoding method

Senpeng Wang[1,2] · Dengguo Feng[1] · Bin Hu[2] · Jie Guan[2] · Kai Zhang[2] · Tairong Shi[2]

Received: 29 January 2023 / Revised: 29 May 2023 / Accepted: 31 May 2023

Abstract
As the first generic method for finding the optimal differentialand linear characteristics, Matsui's branch and bound search algorithm has played an important role in evaluating the security of symmetric ciphers. By combining Matsui's bounding conditions with automatic search models, search efficiency can be improved. In this paper, by studying the properties of Matsui's bounding conditions, we give the general form of bounding conditions that can eliminate all the impossible solutions determined by Matsui's bounding conditions. Then, a new method of combining bounding conditions with sequential encoding method is proposed. With the help of some small size Mixed Integer Linear Programming (MILP) models, we can use fewer variables and clauses to build Satisfiability Problem (SAT) models. As applications, we use our new method to search for the optimal differential and linear characteristics of some SPN, Feistel, and ARX block ciphers. The number of variables and clauses and the solving time of the SAT models are decreased significantly. In addition, we find some new differential and linear characteristics covering more rounds.

Keywords Automatic search · SAT model · Matsui's bounding condition · Differential cryptanalysis · Linear cryptanalysis

Mathematics Subject Classification 94A60 · 65C10

1 Introduction

Differential cryptanalysis [5] and linear cryptanalysis [18] are two powerful methods which have been widely used in the security analysis of many symmetric ciphers. The core idea of

Communicated by D. Stebila.

✉ Senpeng Wang
wsp2110@126.com

1 State Key Laboratory of Cryptology, Beijing, China

2 PLA SSF Information Engineering University, Zhengzhou, China

these methods is to identify the differential or linear characteristics with high probabilities or correlations. However, searching for the optimal differences or linear masks is not an easy work. At EUROCRYPT 1994, Matsui [19] proposed a branch and bound search algorithm which could be used to identify the optimal differences and linear masks. Matsui's algorithm is one of the most powerful and efficient search tools. In a work concurrent to ours (after we submit this document to IACR Cryptol. ePrint Arch.), Kim et al. [12] accelerated Matsui's search algorithm to search for the optimal differences and linear masks of AES-like ciphers. Matsui's algorithm is powerful in searching distinguishers. However, the skills of controlling memory and selecting searching nodes are required when implementing Matsui's algorithm. By contrast, automatic search methods use solvers to deal with these problems which are easier to implement. In order to meet the demands of security analysis of ciphers, many automatic search methods have been proposed and widely used in the search for numerous distinguishers.

Mixed Integer Linear Programming (MILP) is a kind of optimization or feasibility program whose objective function and constraints are linear, and the variables are restricted to be integers or real numbers. MILP problem can be solved automatically with MILP solvers such as Gurobi [11]. In [21, 36], the first automatic search method based on MILP was proposed to evaluate the security of word-oriented block ciphers against differential and linear cryptanalysis. Later, Sun et al. [26, 27] proposed methods for generating inequalities to describe the bit-wise differential or linear characteristics for S-box. Therefore, their models can be used to obtain the minimum number of active S-boxes and search for the best differential and linear characteristics for bit-oriented block ciphers. However, the above methods only work on small size S-boxes (e.g. 4-bit). At FSE 2017, Abdelkhalek et al. [1] put forward the first MILP model for large S-boxes (e.g. 8-bit). Then, some efficient methods were proposed to generate inequalities of large S-boxes (e.g. [7, 34]). For ARX ciphers, Fu et al. [10] built the MILP models for the differential and linear characteristics of modular addition and applied them to search for the best differential and linear characteristics for SPECK. Moreover, as a powerful automatic search tool, MILP has been also widely used in other attacks, such as integral attacks [35, 38], cube attacks [33], impossible differential attacks [23], and zero-correlation linear attacks [8].

The Boolean Satisfiability Problem (SAT) is a problem which considers the satisfiability of a given boolean formula. And there are also many SAT solvers, such as CaDiCal [4]. The first automatic search method based on SAT is introduced by Mouha and Preneel [20]. Then, at CRYPTO 2015, K*ö*lbl et al. [13] used the SAT/SMT solver to find the optimal differential and linear characteristics for SIMON. And at ACNS 2016, Liu et al. [16] extended the SAT based automatic search algorithm to search for the linear characteristics for ARX ciphers. At FSE 2018, Sun et al. [30] built the SAT-based models for differential characteristics and got more accurate differential probability for LED64 and Midori64. Moreover, SAT can be used in the search for impossible differential trails [15] and integral distinguishers [29].

Automatic search tools bring great convenience to the security evaluation of ciphers. However, when the number of variables or constraints in the model is large, solvers may not return the result within a reasonable time. Therefore, it is of great importance to improve the efficiency of automatic search methods. And a lot of works have been done on this issue. We divide them into three main categories.

Reducing the Variables and Constraints in the Model. Although Sasaki and Todo [22] point out that the number of inequalities can not strictly determine the efficiency of solving model, it still has an important impact on the solving time. And a lot of methods have been proposed to reduce the variables and constraints modeling S-box or linear layers [1, 7, 14, 34].

Divide and Conquer Approach. In order to obtain the result of a large model in reasonable time, we can divide it into appropriate parts. In [27], Sun et al. split r-rounds cipher into two parts (the first r_0 and the last $(r - r_0)$ rounds). Then, they combine them after solving the models of the two parts respectively. At FSE 2019, Zhou et al. [41] proposed a divide-and-conquer approach which divided the whole search space according to the number of active S-boxes at a certain round. At FSE 2022, Erlacher et al. [9] proposed a new search strategy of dividing the search space into a large number of subproblems based on girdle patterns.

Combining Matsui's Bounding Conditions into the Model. Matsui's bounding conditions may reduce the feasible region of the original model. The first method of combining Matsui's branch and bound search algorithm with the MILP based search model is proposed by Zhang et al. [39]. Later, Sun et al. [31] put forward a new encoding method to convert Matsui's bounding conditions into boolean formulas of SAT model. Both methods are realized by adding the constraints derived from Matsui's bounding conditions into the original model.

From the perspective of application effect, the SAT model combining with Matsui's bounding conditions proposed by Sun et al. [31] is one of the best choices at present. This method can obtain the complete bounds (full rounds) on the number of active S-boxes, the differential probabilities and linear correlations for many block ciphers for the first time. The efficiency of automatic search has been greatly improved. Just like the MILP models of combining Matsui's bounding conditions, according to the experiment results in [31], adding more Matsui's bounding conditions may not necessarily improve the efficiency. This may be because that all the previous methods realize the bounding conditions by adding a set of constraints. And some added constraints increase the search complexity of models. Regrettably, there is no relevant theory for us to identify the constraints which have negative effects. By doing a considerable amount of experiments, Sun et al. [31] put forward a strategy on how to organise the sets of bounding conditions that potentially achieve better performance. Because this strategy is experimental and lacks sufficient theoretical guidance, we cannot really know its performance until completing its application. Therefore, it is meaningful to research a better way of combining Matsui's bounding conditions with the automatic search models and improve the search efficiency.

1.1 Our contributions

The efficiency of Matsui's bounding conditions comes from the fact that they can eliminate some impossible solutions and reduce the search space. When building SAT models, we have to convert Matsui's bounding conditions into other form of formulas. By studying the properties of Matsui's bounding conditions, we give the general form of inequality constraints that can eliminate all the impossible solutions determined by Matsui's bounding conditions. Then, we propose a new method of combining bounding conditions with sequential encoding method. With the help of some small size MILP models, we can use fewer variables and clauses to build SAT models. This will decrease the solving complexity of models. As applications, we use our new method to search for the optimal differential and linear characteristics for SPN, Feistel and ARX block ciphers. Compared with the previous method, the number of variables and clauses and the solving time of the SAT models are decreased significantly which can be seen in Table 2. For the block ciphers PRESENT, RECTANGLE, GIFT64, LBlock, TWINE, SPECK32, SPECK64, the optimal differential and linear characteristics of the full rounds are obtained which are consistent with the results in [31]. For SPECK48, SPECK96, SPECK128 and GIFT128, we find some new differential

Table 1 The comparison of the maximum length of optimal characteristics

Trail	GIFT128	SPECK48	SPECK96	SPECK128	Ref
Differential	⋆29	–	–	–	[12]
	⋆40	–	–	–	[12]
	–	12	8	8	[17]
	29	18	10	9	[31]
	40 (Full)	**20**	**11**	**10**	Sect. 4
Linear	⋆22	–	–	–	[12]
	⋆40	–	–	–	[12]
	–	13	9	9	[17]
	25	23 (Full)	14	10	[31]
	40 (Full)	23 (Full)	**16**	**11**	Sect. 4

⋆ The results were published after we submitted this work to IACR Cryptol. ePrint Arch. And their method is not based on automatic search solver and works only for AES-like ciphers

and linear characteristics covering more rounds. And a comparison of the maximum length of optimal differential and linear trails with previous results is provided in Table 1. For all the above ciphers, our results reach the maximum length of optimal differential and linear characteristics at present.

1.2 Outline

This paper is organized as follows: Sect. 2 provides the background of automatic search method based on SAT. In Sect. 3, by studying the properties of Matsui's bounding conditions and sequential encoding method, we propose a new SAT model of combining bounding conditions with sequential encoding method. In Sect. 4, we use the new method to search for the optimal differential and linear characteristics for block ciphers. In Sect. 5, we conclude the paper. And some auxiliary materials are supplied in Appendix.

2 Automatic search method based on SAT

2.1 Boolean satisfiability problem

For a formula, if it only consists of boolean variables, operators AND ($\wedge$), OR ($\vee$), NOT ($\bar{\cdot}$) and parentheses, we call it boolean formula. And SAT is the boolean satisfiability problem which considers whether there is a valid assignment to boolean variables such that the formula equals one. If such an assignment exists, the SAT problem is said satisfiable. This problem is NP-complete [25]. However, many problems with millions of variables can be solved by modern SAT solvers, such as CaDiCal [4].

For any boolean formula, we can convert it into Conjunctive Normal Form (CNF) denoted as $\bigwedge_{i=0}^{m}\left(\bigvee_{j=0}^{n_i} c_{i,j}\right)$, where $c_{i,j}$ is a boolean variable or constant or the NOT of a boolean variable. And each disjunction $\bigvee_{j=0}^{n_i} c_{i,j}$ is called a clause. CNF is a standard input format of SAT solvers. If we want to use SAT to solve a problem, we should translate it into a model consisted of boolean variables and clauses.

2.2 SAT models for some basic operations

When we use SAT to search for differential or linear characteristics, we should translate the search problem into a series of clauses. And the clauses should describe the propagation properties of differential or linear characteristics through the cipher. We call a pair of differences (linear masks) is valid when its differential probability (linear correlation) is nonzero. Here, we will briefly introduce the SAT models for some basic operations which will be used in this paper. For more information, please refer to [16, 31]. And in the following, we use x_0 to denote the most significant bit of the n-bit vector $x = (x_0, x_1, \dots, x_{n-1}) \in \mathbb{F}_2^n$.

Differential Model 1 (Branching) [31]. *Let $y = f(x)$ be a branching function, where $x \in \mathbb{F}_2$ is the input variable, and the output variables $y = (y_0, y_1, \dots, y_{n-1}) \in \mathbb{F}_2^n$ is calculated as $y_0 = y_1 = \cdots = y_{n-1} = x$. Then, $(\alpha, \beta_0, \beta_1, \dots, \beta_{n-1})$ is a valid differential of f if and only if it satisfies all the equations in the following:*

$$\left.\begin{array}{l} \alpha \vee \overline{\beta_i} = 1 \\ \overline{\alpha} \vee \beta_i = 1 \end{array}\right\}, 0 \leq i \leq n-1.$$

Differential Model 2 (Xor) [31]. *Let $y = f(x)$ be an Xor function, where $x = (x_0, x_1, \dots, x_{n-1}) \in \mathbb{F}_2^n$ are the input variables, and the output variable $y \in \mathbb{F}_2$ is calculated as $y = x_0 \oplus x_1 \oplus \cdots \oplus x_{n-1}$.*

When $n = 2$, $(\alpha_0, \alpha_1, \beta)$ is a valid differential of f if and only if it satisfies all the equations in the following:

$$\left.\begin{array}{l} \alpha_0 \vee \alpha_1 \vee \overline{\beta} = 1 \\ \alpha_0 \vee \overline{\alpha_1} \vee \beta = 1 \\ \overline{\alpha_0} \vee \alpha_1 \vee \beta = 1 \\ \overline{\alpha_0} \vee \overline{\alpha_1} \vee \overline{\beta} = 1 \end{array}\right\}.$$

When $n \geq 3$, we can decompose the n-input Xor operation into $(n-1)$ 2-input Xor operations by introducing auxiliary boolean variables. After applying 2-input Xor model to the $(n-1)$ 2-input Xor operations one by one, the model of n-input Xor operation can be expressed with $4 \times (n-1)$ clauses.

According to [28], the linear masks propagation model for branching (resp. Xor) operation is the same as the differences propagation model for Xor (resp. branching) operation. Thus, we do not introduce the SAT models for linear masks propagation through branching and Xor operations.

Differential Model 3 (Modular Addition) [16, 31]. *Let $z = f(x, y)$ be a n-bit modular addition operation. Then, $(\alpha, \beta, \gamma) \in \mathbb{F}_2^{3\times n}$ is a valid differential if and only if it satisfies all the following equations:*

$$\begin{array}{l} \alpha_{n-1} \oplus \beta_{n-1} \oplus \gamma_{n-1} = 0; \\ \left.\begin{array}{l} \alpha_i \vee \beta_i \vee \overline{\gamma_i} \vee \alpha_{i+1} \vee \beta_{i+1} \vee \gamma_{i+1} = 1 \\ \alpha_i \vee \overline{\beta_i} \vee \gamma_i \vee \alpha_{i+1} \vee \beta_{i+1} \vee \gamma_{i+1} = 1 \\ \overline{\alpha_i} \vee \beta_i \vee \gamma_i \vee \alpha_{i+1} \vee \beta_{i+1} \vee \gamma_{i+1} = 1 \\ \overline{\alpha_i} \vee \overline{\beta_i} \vee \overline{\gamma_i} \vee \alpha_{i+1} \vee \beta_{i+1} \vee \gamma_{i+1} = 1 \\ \alpha_i \vee \beta_i \vee \gamma_i \vee \overline{\alpha_{i+1}} \vee \overline{\beta_{i+1}} \vee \overline{\gamma_{i+1}} = 1 \\ \alpha_i \vee \overline{\beta_i} \vee \overline{\gamma_i} \vee \overline{\alpha_{i+1}} \vee \overline{\beta_{i+1}} \vee \overline{\gamma_{i+1}} = 1 \\ \overline{\alpha_i} \vee \beta_i \vee \overline{\gamma_i} \vee \overline{\alpha_{i+1}} \vee \overline{\beta_{i+1}} \vee \overline{\gamma_{i+1}} = 1 \\ \overline{\alpha_i} \vee \overline{\beta_i} \vee \gamma_i \vee \overline{\alpha_{i+1}} \vee \overline{\beta_{i+1}} \vee \overline{\gamma_{i+1}} = 1 \end{array}\right\} 0 \leq i \leq n-2, \end{array}$$

where the Xor operation denoted by $\oplus$ is symbolic representation which can be converted into CNF formulas with the method in Differential Model 2 (Xor). *In order to model the different probability, we will introduce* $(n-1)$ *binary variables denoted as* $w_0, w_1, \ldots, w_{n-2}$. *When they satisfy the following equations:*

$$\left.\begin{aligned} \alpha_{i+1} \vee \gamma_{i+1} \vee w_i &= 1 \\ \beta_{i+1} \vee \overline{\gamma_{i+1}} \vee w_i &= 1 \\ \alpha_{i+1} \vee \overline{\beta_{i+1}} \vee w_i &= 1 \\ \alpha_{i+1} \vee \beta_{i+1} \vee \gamma_{i+1} \vee \overline{w_i} &= 1 \\ \overline{\alpha_{i+1}} \vee \overline{\beta_{i+1}} \vee \overline{\gamma_{i+1}} \vee \overline{w_i} &= 1 \end{aligned}\right\} 0 \le i \le n-2,$$

the differential probability can be computed as $p(\alpha, \beta, \gamma) = 2^{-\sum_{i=0}^{n-2} w_i}$.

The papers [16, 31] have showed the model for the linear correlations through modular addition. Because the most significant bit of modular addition is a constant value, we can omit this variable. So we give a new linear model for modular addition which is a little different from the previous.

Linear Model 1 (Modular Addition). *For an n-bit modular addition operation* $z = f(x, y)$, *we denote the two input linear masks as* α *and* β *and the output mask as* γ. *And in order to model the correlation,* $(n-1)$ *binary variables denoted as* $w = (w_0, w_1, \ldots, w_{n-2})$ *are introduced. Then, the correlation of the linear approximation* $(\alpha, \beta, \gamma) \in \mathbb{F}_2^{3\times n}$ *is nonzero if and only if* $(\alpha, \beta, \gamma, w)$ *satisfies all the following equations:*

$$\begin{aligned} &\alpha_0 \oplus \beta_0 \oplus \gamma_0 \oplus w_0 = 0; \\ &\alpha_{j+1} \oplus \beta_{j+1} \oplus \gamma_{j+1} \oplus w_j \oplus w_{j+1} = 0, 0 \le j \le n-3; \\ &\alpha_0 = \beta_0 = \gamma_0; \\ &\left.\begin{aligned} \alpha_i \vee \overline{\gamma_i} \vee w_{i-1} &= 1 \\ \overline{\alpha_i} \vee \gamma_i \vee w_{i-1} &= 1 \\ \beta_i \vee \overline{\gamma_i} \vee w_{i-1} &= 1 \\ \overline{\beta_i} \vee \gamma_i \vee w_{i-1} &= 1 \end{aligned}\right\} 1 \le i \le n-1. \end{aligned}$$

Then, the linear correlation is computed as $p(\alpha, \beta, \gamma) = 2^{-\sum_{i=0}^{n-2} w_i}$.

For S-box, the paper [30] showed an example of building the differential SAT model of 4-bit S-box. Then, the paper [31] proposed the SAT model of active n-bit S-box. Based on the above two methods, we will show a general method for building SAT model of S-box.

Differential Model 4 (S-box). *For an S-box* $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, *the differential probability is denoted as* $p(\alpha, \beta)$, *where* $\alpha \in \mathbb{F}_2^n$ *is the input difference and* $\beta \in \mathbb{F}_2^m$ *is the output difference. If the minimal non-zero differential probability of S-box is* 2^{-s}, *we introduce* s auxiliary variables $w = (w_0, w_1, \ldots, w_{s-1})$ *satisfying* $w_{i+1} \le w_i, 0 \le i \le s-2$ *to calculate the non-zero differential probability. In order to build the differential SAT model of S-box, we introduce a boolean function as follows:*

$$g(\alpha, \beta, w) = \begin{cases} 1, \text{ if } p(\alpha, \beta) = 2^{-\sum_{i=0}^{s-1} w_i}; \\ 0, \text{ otherwise.} \end{cases}$$

Let A *be a set which contains all vectors satisfying* $g(a, b, c) = 0$ *denoted as*

$$A = \left\{(a, b, c) \in \mathbb{F}_2^{n+m+s} | g(a, b, c) = 0\right\}.$$

Then, the following $|A|$ clauses form a primary SAT model of the given S-box

$$\bigvee_{i=0}^{n-1}\left(\alpha_i \oplus a_i^l\right) \vee \bigvee_{j=0}^{m-1}\left(\beta_j \oplus b_j^l\right) \vee \bigvee_{k=0}^{s-1}\left(w_k \oplus c_k^l\right) = 1, 0 \leq l \leq |A| - 1.$$

where $|A|$ is the number of vectors in the set A and $\left(a^l, b^l, c^l\right), 0 \leq l \leq |A| - 1$ is the l-th vector in the set A.

Note that the solution space of the above $|A|$ clauses about (α, β, γ) is the same as that of the following boolean function:

$$h(\alpha, \beta, \gamma) = \bigwedge_{l=0}^{|A|-1}\left(\bigvee_{i=0}^{n-1}\left(\alpha_i \oplus a_i^l\right) \vee \bigvee_{j=0}^{m-1}\left(\beta_j \oplus b_j^l\right) \vee \bigvee_{k=0}^{s-1}\left(w_k \oplus c_k^l\right)\right) = 1. \quad (1)$$

Equivalently, we have

$$h(\alpha, \beta, \gamma) = \bigwedge_{(a,b,c)\in\mathbb{F}_2^{n+m+s}}\left(h(a, b, c) \vee \bigvee_{i=0}^{n-1}(\alpha_i \oplus a_i) \vee \bigvee_{j=0}^{m-1}\left(\beta_j \oplus b_j\right) \vee \bigvee_{k=0}^{s-1}(w_k \oplus c_k)\right),$$

where $h(a, b, c)$ is the value of Eq. (1) by assigning $\alpha = a, \beta = b, \gamma = c$. This equation is called the product-of-sum representation of h. The issue of reducing the number of clauses is turned into the problem of simplifying the product-of-sum representation of the boolean function. According to [1], we know that this simplification problem can be solved by the Quine-McCluskey (QM) algorithm and Espresso algorithm, theoretically.

Using the same method of differential SAT model for S-box, the SAT model for linear correlations through S-box can be built easily. Here, we omit it.

2.3 Sequential encoding method

When building SAT models for ciphers, we always aim at getting some cryptographic properties such as the number of active S-boxes, the differential probability or the linear correlation. All kinds of these objections can be abstracted as the boolean cardinality constraint $\sum_{i=0}^{n-1} w_i \leq m$, where w_i is a boolean variable, and m is a non-negative integer. However, addition over integers is not a natural operation in SAT language, which is not easy to be described with only OR and AND operations. The sequential encoding method is one of the best methods for characterising boolean cardinality constraint. Many papers [16, 30, 31] use the sequential encoding method [24] to convert the constraint into CNF formulas.

When $m = 0$, the cardinality constraint $\sum_{i=0}^{n-1} w_i \leq m$ can be translated to n clauses as $\overline{w_i} = 1, 0 \leq i \leq n - 1$ which means all variables are zero.

When $m \geq 1$, in order to model constraint $\sum_{i=0}^{n-1} w_i \leq m$, auxiliary boolean variables $u_{i,j}$ $(0 \leq i \leq n - 2, 0 \leq j \leq m - 1)$ are introduced to return contradiction when the cardinality is larger than m. More specifically, for the partial sum $\sum_{i=0}^{k} w_i = m_k$, the values of the auxiliary boolean variables $u_{k,j}$ $(0 \leq j \leq m - 1)$ should satisfy the following equations:

$$u_{k,j} = \begin{cases} 0, \text{ if } m_k \leq j \leq m - 1; \\ 1, \text{ if } 0 \leq j \leq m_k - 1. \end{cases}$$

Then, $\sum_{i=0}^{k} w_i = \sum_{j=0}^{m-1} u_{k,j}$, and the sequence $\left\{\sum_{i=0}^{k} w_i | 0 \leq k \leq n - 2\right\}$ is non-decreasing. Therefore, the constraint $\sum_{i=0}^{n-1} w_i \leq m$ holds if the following implication predicates are satisfied.

$$
\begin{array}{l}
\text{if } w_0 = 1 \text{ then } u_{0,0} = 1 \\
u_{0,j} = 0, 1 \le j \le m-1 \\
\left.\begin{array}{l}
\text{if } w_i = 1 \text{ then } u_{i,0} = 1 \\
\text{if } u_{i-1,0} = 1 \text{ then } u_{i,0} = 1 \\
\left.\begin{array}{l}
\text{if } w_i = 1 \text{ and } u_{i-1,j-1} = 1 \text{ then } u_{i,j} = 1 \\
\text{if } u_{i-1,j} = 1 \text{ then } u_{i,j} = 1
\end{array}\right\} 1 \le j \le m-1 \\
\text{if } w_i = 1 \text{ then } u_{i-1,m-1} = 0
\end{array}\right\} 1 \le i \le n-2 \\
\text{if } w_{n-1} = 1 \text{ then } u_{n-2,m-1} = 0
\end{array}
$$

The above predicates can be interpreted as the following $2 \cdot m \cdot n - 3 \cdot m + n - 1$ clauses which are the SAT model for the cardinality constraint $\sum_{i=0}^{n-1} w_i \le m$.

$$
\begin{array}{l}
\overline{w_0} \vee u_{0,0} = 1 \\
\overline{u_{0,j}} = 1, 1 \le j \le m-1 \\
\left.\begin{array}{l}
\overline{w_i} \vee u_{i,0} = 1 \\
\overline{u_{i-1,0}} \vee u_{i,0} = 1 \\
\left.\begin{array}{l}
\overline{w_i} \vee \overline{u_{i-1,j-1}} \vee u_{i,j} = 1 \\
\overline{u_{i-1,j}} \vee u_{i,j} = 1
\end{array}\right\} 1 \le j \le m-1 \\
\overline{w_i} \vee \overline{u_{i-1,m-1}} = 1
\end{array}\right\} 1 \le i \le n-2 \\
\overline{w_{n-1}} \vee \overline{u_{n-2,m-1}} = 1
\end{array}
$$

2.4 Combining Matsui's bounding conditions with sequential encoding method

At EUROCRYPT 1994, Matsui [19] proposed a branch and bound search algorithm which can be used to identify the optimal difference probability. Let $P_{ini}(R)$ be the initial estimation for the probability bound achieved by R-round trails and $P_{opt}(i), 0 \le i \le R-1$ be the maximum probability achieved by i-round trails. Then, a partial trail $\left(\alpha^0 \to \alpha^1 \to \cdots \to \alpha^r\right)$ covering the first r rounds will never extend to be a better R-round trail if it does not satisfy the following condition:

$$\prod_{i=0}^{r-1} p\left(\alpha^i \to \alpha^{i+1}\right) \cdot P_{opt}(R-r) \ge P_{ini}(R), \tag{2}$$

where $p\left(\alpha^i \to \alpha^{i+1}\right)$ is the probability of the i-th round. Therefore, we can give up the partial trail. In this way, the efficiency of search algorithm can be improved greatly.

To facilitate the description of Matsui's bounding conditions, we introduce the probability weight as following.

$$\begin{cases} -\log_2\left(P_{ini}(R)\right) = W_{ini}(R), \\ -\log_2\left(P_{opt}(i)\right) = W_{opt}(i), \\ -\log_2\left(p\left(\alpha^i \to \alpha^{i+1}\right)\right) = \sum_{j=0}^{\varpi-1} w_j^i, \end{cases} \tag{3}$$

where $w_j^i, 0 \le j \le \varpi - 1$ are the boolean variables used to calculate the probability weight of the trail $\alpha^i \to \alpha^{i+1}$. By defining the symbol $w_{\varpi \times i+j} = w_j^i$, Eq. (2) can be rewritten as follows:

$$\sum_{i=0}^{r-1}\sum_{j=0}^{\varpi-1} w_j^i = \sum_{i=0}^{r\times\varpi-1} w_i \leq W_{ini}(R) - W_{opt}(R-r). \tag{4}$$

Note that the right-hand side of this equation is a constant, and the left-hand side of it matches the probability weight of the trail covering the first r rounds. All the above bounding conditions can be replaced with inequalities as the form:

$$\sum_{i=e_1}^{e_2} w_i \leq m_{e_1,e_2}, 0 \leq e_1 \leq e_2. \tag{5}$$

For the boolean cardinality constraint $\sum_{i=0}^{n-1} w_i \leq m$, based on the sequential encoding method, Sun et al. [31] realized bounding conditions without claiming any new variables as follows.

Case 1. *Bounding condition* $\sum_{i=e_1}^{e_2} w_i \leq m_{e_1,e_2}$ *with* $e_1 = 0$ *and* $e_2 < n-1$ *can be modeled by the following* e_2 *clauses:*

$$\overline{w_i} \vee \overline{u_{i-1,m_{e_1,e_2}-1}} = 1, 1 \leq i \leq e_2.$$

Case 2. *Bounding condition* $\sum_{i=e_1}^{e_2} w_i \leq m_{e_1,e_2}$ *with* $e_1 > 0$ and $e_2 < n-1$ *can be modeled by the following* $m - m_{e_1,e_2}$ *clauses:*

$$u_{e_1-1,j} \vee \overline{u_{e_2,j+m_{e_1,e_2}}} = 1, 0 \leq j \leq m - m_{e_1,e_2} - 1.$$

Case 3. *Bounding condition* $\sum_{i=e_1}^{e_2} w_i \leq m_{e_1,e_2}$ *with* $e_1 > 0$ *and* $e_2 = n-1$ *can be modeled by the following* $2\cdot\left(m - m_{e_1,e_2}\right)+1$ *clauses:*

$$\begin{cases} u_{e_1-1,j} \vee \overline{u_{n-2,j+m_{e_1,e_2}}} = 1,\ 0 \leq j \leq m - m_{e_1,e_2} - 1; \\ u_{e_1-1,j} \vee \overline{w_{n-1}} \vee \overline{u_{n-2,j+m_{e_1,e_2}-1}} = 1,\ 0 \leq j \leq m - m_{e_1,e_2}. \end{cases}$$

The above method can intermix multiple Matsui's bounding conditions into one SAT model with an increment on the number of clauses. At the same time, the number of variables remains the same as the original SAT model.

3 New SAT model of combining bounding conditions with sequential encoding method

Although numerous Matsui's bounding conditions can be obtained, it is not sure which bounding condition can accelerate the solve efficiency of SAT model accurately. According to the experiments, adding all Matsui's bounding conditions into the SAT model is not the best choice. With the observations and experiences in the tests, Sun et al. [31] put forward a strategy on how to create the sets of bounding conditions that probably achieve extraordinary advances. But this is an experimental strategy. It is worth studying how to combine bounding conditions with sequential encoding method in a better way.

3.1 Further insights into Matsui's bounding conditions

We all know that the efficiency of Matsui's algorithm comes from the fact that it can eliminate some impossible solutions and reduce the search space. When building SAT models, we have to convert Matsui's bounding conditions into other form of formulas. With the same mathematical symbols defined in Sect. 2, let $w_i \in \mathbb{F}_2, 0 \le i \le n-1$ be the variables which are used to calculate the differential probability or linear correlation of a cipher. According to Sect. 2.4, Sun et al. [31] summarize all Matsui's bounding conditions as the form of $\sum_{i=e_1}^{e_2} w_i \le m_{e_1,e_2}$. However, we find that constraints of the form $\sum_{i=e_1}^{e_2} w_i \le m_{e_1,e_2}$ can not always eliminate all the impossible solutions determined by Matsui's bounding conditions. We will give an example to show this phenomenon.

For a toy cipher E which has 3 rounds, let $\alpha^0 \to \alpha^1 \to \alpha^2 \to \alpha^3$ be the 3-round trail. By introducing 6 boolean variables $\left(w_0^0, w_1^0, w_0^1, w_1^1, w_0^2, w_1^2\right)$, the probability weight of round function is calculated as follows:

$$-\log_2\left(p\left(\alpha^i \to \alpha^{i+1}\right)\right) = w_0^i + w_1^i. \tag{6}$$

When Matsui's bounding conditions satisfy $W_{opt}(1) = 1$, $W_{opt}(2) = 2$ and $W_{ini}(3) = 3$, the boolean variables $\left(w_0^0, w_1^0, w_0^1, w_1^1, w_0^2, w_1^2\right)$ should satisfy the following conditions:

$$\begin{cases} w_0^0 + w_1^0 \ge W_{opt}(1), \\ w_0^1 + w_1^1 \ge W_{opt}(1), \\ w_0^2 + w_1^2 \ge W_{opt}(1), \\ w_0^0 + w_1^0 + w_0^1 + w_1^1 \ge W_{opt}(2), \\ w_0^1 + w_1^1 + w_0^2 + w_1^2 \ge W_{opt}(2), \\ w_0^0 + w_1^0 + w_0^1 + w_1^1 + w_0^2 + w_1^2 = W_{ini}(3). \end{cases} \tag{7}$$

Then, the solutions of $\left(w_0^0, w_1^0, w_0^1, w_1^1, w_0^2, w_1^2\right)$ satisfying Eq. (7) are as follows:

$$\begin{array}{l}\{0,1,0,1,0,1\}, \{0,1,0,1,1,0\}, \{0,1,1,0,0,1\}, \{0,1,1,0,1,0\}, \\ \{1,0,0,1,0,1\}, \{1,0,0,1,1,0\}, \{1,0,1,0,0,1\}, \{1,0,1,0,1,0\}.\end{array}$$

Thus, the number of impossible solutions eliminated by $W_{opt}(1) = 1$, $W_{opt}(2) = 2$ and $W_{ini}(3) = 3$ is $2^6 - 8 = 56$.

According to Sect. 2.4, all the form of $\sum_{i=e_1}^{e_2} w_i \le m_{e_1,e_2}$ conditions deduced from Matsui's bounding conditions are as follows:

$$\begin{cases} w_0^0 + w_1^0 \le W_{ini}(3) - W_{opt}(2), \\ w_0^0 + w_1^0 + w_0^1 + w_1^1 \le W_{ini}(3) - W_{opt}(1), \\ w_0^1 + w_1^1 \le W_{ini}(3) - W_{opt}(1) - W_{opt}(1), \\ w_0^1 + w_1^1 + w_0^2 + w_1^2 \le W_{ini}(3) - W_{opt}(1), \\ w_0^2 + w_1^2 \le W_{ini}(3) - W_{opt}(2), \\ w_0^0 + w_1^0 + w_0^1 + w_1^1 + w_0^2 + w_1^2 \le W_{ini}(3). \end{cases} \tag{8}$$

Then, the solutions of $\left(w_0^0, w_1^0, w_0^1, w_1^1, w_0^2, w_1^2\right)$ satisfying Eq. (8) are as follow:

$$\begin{aligned}
&\{0,0,0,0,0,0\},\{0,0,0,0,0,1\},\{0,0,0,0,1,0\},\{0,0,0,1,0,0\},\\
&\{0,0,0,1,0,1\},\{0,0,0,1,1,0\},\{0,0,1,0,0,0\},\{0,0,1,0,0,1\},\\
&\{0,0,1,0,1,0\},\{0,1,0,0,0,0\},\{0,1,0,0,0,1\},\{0,1,0,0,1,0\},\\
&\{0,1,0,1,0,0\},\{0,1,0,1,0,1\},\{0,1,0,1,1,0\},\{0,1,1,0,0,0\},\\
&\{0,1,1,0,0,1\},\{0,1,1,0,1,0\},\{1,0,0,0,0,0\},\{1,0,0,0,0,1\},\\
&\{1,0,0,0,1,0\},\{1,0,0,1,0,0\},\{1,0,0,1,0,1\},\{1,0,0,1,1,0\},\\
&\{1,0,1,0,0,0\},\{1,0,1,0,0,1\},\{1,0,1,0,1,0\}.
\end{aligned}$$

Thus, the number of impossible solutions eliminated by Eq. (8) is $2^6 - 27 = 37$. Therefore, the bounding conditions in Eq. (8) do not eliminate all the impossible solutions determined by $W_{opt}(1) = 1$, $W_{opt}(2) = 2$ and $W_{ini}(3) = 3$.

Here, we analyze the reasons for this phenomenon. When using Matsui's branch and bound algorithm to search for R-round optimal trails, we will firstly obtain a partial trail denoted as $\alpha^0 \to \alpha^1 \to \cdots \to \alpha^r$ covering the first r rounds. Then, we can use Eq. (2) to deduce the bound conditions of the form $\sum_{i=e_1}^{e2} w_i \le m_{e_1,e_2}$. But, it should be noted that all the obtained partial trails are valid. That is, the partial trails should satisfy

$$\sum_{i=0}^{r-1} \sum_{j=0}^{\varpi-1} w_j^i \ge W_{opt}(r).$$

Therefore, when combining Matsui's bounding conditions with automatic search algorithm, this kind of bounding conditions should also be considered.

Theorem 1 *For an R-round cipher, the same impossible solutions determined by Matsui's bounding conditions $W_{ini}(R)$ and $W_{opt}(i)$, $0 \le i \le R-1$ can be eliminated by the following bounding conditions*

$$W_{opt}(r_2 + 1 - r_1) \le \sum_{i=r_1}^{r_2} \sum_{j=0}^{\varpi-1} w_j^i \le W_{ini}(R) - W_{opt}(r_1) - W_{opt}(R - 1 - r_2), \quad (9)$$

where $0 \le r_1 \le r_2 \le R - 1$.

Proof Let $\alpha^{r_1} \to \alpha^{r_1+1} \to \cdots \to \alpha^{r_2+1}$ be a feasible partial trail covering $(r_2 + 1 - r_1)$ rounds, where $0 \le r_1 \le r_2 \le R-1$. Because of the constraint $W_{opt}(r_2 + 1 - r_1)$, the partial trail should satisfy the following bounding condition:

$$W_{opt}(r_2 + 1 - r_1) \le \sum_{i=r_1}^{r_2} \sum_{j=0}^{\varpi-1} w_j^i.$$

Then, due to the constraint of $W_{ini}(R)$, the partial trail will not be extended to a better R-round trail if the following bounding condition is violated

$$\sum_{i=r_1}^{r_2} \sum_{j=0}^{\varpi-1} w_j^i \le W_{ini}(R) - W_{opt}(r_1) - W_{opt}(R - 1 - r_2),$$

Therefore, the bounding conditions in Eq. (9) are converted from $W_{ini}(R)$ and $W_{opt}(r)$, $0 \le i \le R - 1$. That is, all the feasible trails will not be eliminated by the bounding conditions in Eq. (9).

Let $\alpha^0 \rightarrow \alpha^1 \rightarrow \cdots \rightarrow \alpha^R$ be a trail which does not satisfy all Matsui's bounding conditions $W_{ini}(R)$ and $W_{opt}(i), 0 \le i \le R-1$. Thus, there is at least a partial trail that does not satisfy $W_{ini}(R)$ or $W_{opt}(i)$. We denote this partial trail as $\alpha^{r_1} \rightarrow \alpha^{r_1+1} \rightarrow \cdots \rightarrow \alpha^{r_2-r_1+1}$. Then, this partial trail will violate the bounding condition as following

$$W_{opt}(r_2+1-r_1) \le \sum_{i=r_1}^{r_2} \sum_{j=0}^{\varpi-1} w_j^i \le W_{ini}(R) - W_{opt}(r_1) - W_{opt}(R-1-r_2). \quad (10)$$

Therefore, the trail $\alpha^0 \rightarrow \alpha^1 \rightarrow \cdots \rightarrow \alpha^R$ will not satisfy all the bounding conditions in Eq. (9). That is, all the infeasible trails determined by Matsui's bounding conditions will be eliminate by the bounding conditions in Eq. (9). □

Using the same mathematical symbols with Eq. (5), we have the following corollary.

Corollary 1 *All Matsui's bounding conditions can be replaced with inequality constraints of the form* $l_{e_1,e_2} \le \sum_{i=e_1}^{e_2} w_i \le m_{e_1,e_2}$.

3.2 A new method of combining bounding conditions with sequential encoding method

From Corollary 1, we know that the general form of bounding condition is $l_{e_1,e_2} \le \sum_{i=e_1}^{e_2} w_i \le m_{e_1,e_2}$. If we get the condition $l_{0,e_2} \le \sum_{i=0}^{e_2} w_i \le m_{0,e_2}$, according to the rules of sequential encoding method, we have

$$u_{e_2,j} = \begin{cases} 0, & \text{if } m_{0,e_2} \le j \le m-1, \\ 1, & \text{if } 0 \le j \le l_{0,e_2}-1, \\ uncertain, & \text{otherwise.} \end{cases}$$

Therefore, the value of some auxiliary variables are determined. We can reduce the variables and clauses which characterise these determined values. Because there are at least $m_{0,e_2} - l_{0,e_2}$ auxiliary variables whose values are uncertain. We have to introduce the boolean variables denoted as $\{u_{e_2,j} | l_{0,e_2} \le j \le m_{0,e_2}-1\}$ to represent these uncertain values. Then, we can use the following equation to compute the partial sum of $\sum_{i=0}^{e_2} w_i$.

$$\sum_{i=0}^{e_2} w_i = \sum_{j=l_{0,e_2}}^{m_{0,e_2}-1} u_{e_2,j} + l_{0,e_2}.$$

Base on this idea, we propose a new method of combining bounding conditions with sequential encoding method.

Lemma 1 *Let* $\sum_{i=0}^{n-1} w_i \le m, 1 \le n$ *be a cardinality constraint. Based on the sequential encoding method, the following clauses can eliminate the same impossible solutions determined by the condition* $l_{0,0} \le w_0 \le m_{0,0}$*:*

$$\begin{aligned} &\textbf{if } l_{0,0} = 0 \textbf{ and } m_{0,0} = 1: \\ &\quad \overline{w_0} \vee u_{0,0} = 1 \\ &\textbf{if } l_{0,0} = 0 \textbf{ and } m_{0,0} = 0: \\ &\quad \overline{w_0} = 1 \\ &\textbf{if } l_{0,0} = 1 \textbf{ and } m_{0,0} = 1: \\ &\quad w_0 = 1 \end{aligned}$$

Proof When using sequential encoding method to model the cardinality constraint $\sum_{i=0}^{n-1} w_i \leq m$, we have to introduce m auxiliary boolean variables $u_{0,0}, u_{0,1}, \ldots, u_{0,m-1}$ to represent the value of partial sum w_0. Different from the method in Sect. 2.4, we can realise the bounding condition $l_{0,0} \leq w_0 \leq m_{0,0}$ in the following way.

When $l_{0,0} = 0$ and $m_{0,0} = 1$, only the value of auxiliary variable $u_{0,0}$ is uncertain. Thus, the value of partial sum w_0 can be represented by the rules of sequential encoding method as $\overline{w_0} \vee u_{0,0} = 1$.

When $l_{0,0} = m_{0,0} = 0$, all the values of auxiliary variables are determined. Thus, no auxiliary variables need to be introduced. The value of partial sum w_0 can be represented as the clause $\overline{w_0} = 1$.

When $l_{0,0} = m_{0,0} = 1$, all the values of auxiliary variables are determined. Thus, no auxiliary variables need to be introduced. The value of partial sum w_0 can be represented as the clause $w_0 = 1$. □

Lemma 2 *Let $\sum_{i=0}^{n-1} w_i \leq m, 3 \leq n$ be a cardinality constraint. If the bounding condition $l_{0,e_2-1} \leq \sum_{i=0}^{e_2-1} w_i \leq m_{0,e_2-1}, 1 \leq e_2 \leq n-2$ is known, the following clauses can eliminate the same impossible solutions determined by bounding condition $l_{0,e_2} \leq \sum_{i=0}^{e_2} w_i \leq m_{0,e_2}$.*

$$
\begin{aligned}
&\textbf{if } m_{0,e_2} = 0:\\
&\quad \overline{w_{e_2}} = 1\\
&\textbf{if } m_{0,e_2} > 0:\\
&\quad \textbf{if } l_{0,e_2} = 0:\\
&\quad\quad \overline{w_{e_2}} \vee u_{e_2,0} = 1\\
&\quad\quad \textbf{if } l_{0,e_2-1} < m_{0,e_2-1}:\\
&\quad\quad\quad \overline{u_{e_2-1,0}} \vee u_{e_2,0} = 1\\
&\quad \left.\begin{aligned}
&\textbf{if } j = l_{0,e_2-1}:\\
&\quad \overline{w_{e_2}} \vee u_{e_2,j} = 1\\
&\textbf{if } j > l_{0,e_2-1} \textbf{ and } j \leq m_{0,e_2-1}:\\
&\quad \overline{w_{e_2}} \vee \overline{u_{e_2-1,j-1}} \vee u_{e_2,j} = 1\\
&\textbf{if } j \geq l_{0,e_2-1} \textbf{ and } j \leq m_{0,e_2-1} - 1:\\
&\quad \overline{u_{e_2-1,j}} \vee u_{e_2,j} = 1
\end{aligned}\right\} max(l_{0,e_2}, 1) \leq j \leq m_{0,e_2} - 1\\
&\quad \textbf{if } m_{0,e_2-1} = m_{0,e_2} \textbf{ and } l_{0,e_2-1} < m_{0,e_2}:\\
&\quad\quad \overline{w_{e_2}} \vee \overline{u_{e_2-1,m_{0,e_2}-1}} = 1\\
&\quad \textbf{if } l_{0,e_2-1} = m_{0,e_2}:\\
&\quad\quad \overline{w_{e_2}} = 1
\end{aligned}
\tag{11}
$$

Proof When using original sequential encoding method to model the cardinality constraint $\sum_{i=0}^{n-1} w_i \leq m$, we have to introduce m auxiliary boolean variables $u_{e_2,0}, u_{e_2,1}, \ldots, u_{e_2,m-1}$ to represent the value of partial sum $\sum_{i=0}^{e_2} w_i$. Different from the method in Sect. 2.4, we can realise the bounding condition $l_{0,e_2} \leq \sum_{i=0}^{e_2} w_i \leq m_{0,e_2}$ in the following way.

When $m_{0,e_2} = 0$, all the values of auxiliary variables are determined. Thus, all the auxiliary variables and related clauses can be reduced. And the value of w_{e_2} can be represented as the clauses $\overline{w_{e_2}} = 1$.

When $m_{0,e_2} > 0$, in order to characterise the value of $\sum_{i=0}^{e_2} w_i$, the $m_{0,e_2} - l_{0,e_2}$ auxiliary variables whose values are uncertain must be introduced, denoted as $\{u_{e_2,j} | l_{0,e_2} \leq j \leq$

$m_{0,e_2} - 1\}$. And all the other auxiliary variables whose values are determined are not needed. Then, we use the rules of sequential encoding method to model these uncertain variables one by one.

If $l_{0,e_2} = 0$, the value of $u_{e_2,0}$ should satisfy the following rules of sequential encoding method.

$$\begin{cases} \text{if } w_{e_2} = 1 \text{ then } u_{e_2,0} = 1; \\ \text{if } u_{e_2-1,0} \text{ is uncertain , when } u_{e_2-1,0} = 1 \text{ then } u_{e_2,0} = 1. \end{cases}$$

For $max(l_{0,e_2}, 1) \leq j \leq m_{0,e_2} - 1$, the value of $u_{e_2,j}$ should satisfy the following rules of sequential encoding method.

$$\begin{cases} \text{if } u_{e_2-1,j-1} \text{ is determined as 1 and } w_{e_2} = 1 \text{ then } u_{e_2,j} = 1; \\ \text{if } u_{e_2-1,j-1} \text{ is uncertain, when } u_{e_2-1,j-1} = 1 \text{ and } w_{e_2} = 1 \text{ then } u_{e_2,j} = 1; \\ \text{if } u_{e_2-1,j} \text{ is uncertain, when } u_{e_2-1,j} = 1 \text{ then } u_{e_2,j} = 1. \end{cases}$$

Because of the bounding condition $l_{0,e_2} \leq \sum_{i=0}^{e_2} w_i \leq m_{0,e_2}$ and the rules of sequential encoding method, auxiliary boolean variables $u_{e_2,j}$ will return contradiction when $\sum_{i=0}^{e_2} w_i > m_{0,e_2}$. Thus, the following clauses should be satisfied.

$$\begin{cases} \text{if } m_{0,e_2-1} = m_{0,e_2}, u_{e_2-1,m_{0,e_2}-1} \text{ is uncertain, } w_{e_2} = 1 \text{ then } u_{e_2-1,m_{0,e_2}-1} = 0; \\ \text{if } l_{0,e_2-1} = m_{0,e_2} \text{ then } w_{e_2} = 0. \end{cases}$$

The above predicates can be interpreted as the clauses as Eq. (11). □

Lemma 3 *Let* $\sum_{i=0}^{n-1} w_i \leq m, 2 \leq n$ *be a constraint. If the bounding condition* $l_{0,n-2} \leq \sum_{i=0}^{n-2} w_i \leq m_{0,n-2}$ *is known, the following clauses can eliminate the same impossible solutions determined by* $l_{0,n-1} \leq \sum_{i=0}^{n-1} w_i \leq m_{0,n-1}$.

$$\begin{cases} \textbf{if } m_{0,n-1} = 0 : \\ \quad \overline{w_{n-1}} = 1 \\ \textbf{if } m_{0,n-1} > 0 : \\ \quad \textbf{if } m_{0,n-2} = m_{0,n-1} \textbf{ and } l_{0,n-2} < m_{0,n-1} : \\ \quad\quad \overline{w_{n-1}} \vee \overline{u_{n-2,m_{0,n-1}-1}} = 1 \\ \quad \textbf{if } l_{0,n-2} = m_{0,n-1} : \\ \quad\quad \overline{w_{n-1}} = 1 \end{cases} \tag{12}$$

Proof According to Lemma 1 and 2, the auxiliary variables $u_{n-2,j}, l_{0,n-2} \leq j \leq m_{0,n-2} - 1$ are introduced to describe the value of $\sum_{i=0}^{n-2} w_i$. For the bounding condition $l_{0,n-1} \leq \sum_{i=0}^{n-1} w_i \leq m_{0,n-1}$, we only need to know whether the condition is valid or not. Therefore, no auxiliary variables need to be introduced. Then, the value of w_{n-1} should satisfy the following rules of sequential encoding method.

$$\begin{cases} \text{if } m_{0,n-1} = 0 \text{ then } w_{n-1} = 0; \\ \text{if } l_{0,n-2} < m_{0,n-1} = m_{0,n-2}, w_{n-1} = 1 \text{ then } u_{n-2,m_{0,n-1}-1} = 0; \\ \text{if } m_{0,n-1} > 0, l_{0,n-2} = m_{0,n-1} \text{ then } w_{n-1} = 0. \end{cases}$$

The above predicates can be interpreted as the clauses as Eq. (12). □

Theorem 2 *Based on the sequential encoding method, the following clauses are the SAT model which can eliminate the same impossible solutions determined by the bounding conditions* $l_{0,e_2} \leq \sum_{i=0}^{e_2} w_i \leq m_{0,e_2}$, $0 \leq e_2 \leq n-1$*:*

$$
\begin{array}{l}
\textbf{if } l_{0,0}=0 \textbf{ and } m_{0,0}=1:\\
\quad \overline{w_0} \vee u_{0,0}=1\\
\textbf{else if } l_{0,0}=m_{0,0}=0:\\
\quad \overline{w_0}=1\\
\textbf{else if } l_{0,0}=1 \textit{ and } m_{0,0}=1:\\
\quad w_0=1\\
\left.\begin{array}{l}
\textbf{if } m_{0,e_2}=0:\\
\quad \overline{w_{e_2}}=1\\
\textbf{if } m_{0,e_2}>0:\\
\quad \textbf{if } l_{0,e_2}=0:\\
\quad\quad \overline{w_{e_2}} \vee u_{e_2,0}=1\\
\quad\quad \textbf{if } l_{0,e_2-1}<m_{0,e_2-1}:\\
\quad\quad\quad \overline{u_{e_2-1,0}} \vee u_{e_2,0}=1\\
\quad\quad \left.\begin{array}{l}
\textbf{if } j=l_{0,e_2-1}\\
\quad \overline{w_{e_2}} \vee u_{e_2,j}=1\\
\textbf{if } j>l_{0,e_2-1} \textbf{ and } j \leq m_{0,e_2-1}\\
\quad \overline{w_{e_2}} \vee \overline{u_{e_2-1,j-1}} \vee u_{e_2,j}=1\\
\textbf{if } j \geq l_{0,e_2-1} \textbf{ and } j \leq m_{0,e_2-1}-1\\
\quad \overline{u_{e_2-1,j}} \vee u_{e_2,j}=1
\end{array}\right\} \begin{array}{l} max(l_{0,e_2},1) \leq j\\ \leq m_{0,e_2}-1\end{array}\\
\quad \textbf{if } m_{0,e_2-1}=m_{0,e_2} \textbf{ and } l_{0,e_2-1}<m_{0,e_2}\\
\quad\quad \overline{w_{e_2}} \vee \overline{u_{e_2-1,m_{0,e_2}-1}}=1\\
\quad \textbf{if } l_{0,e_2-1}=m_{0,e_2}\\
\quad\quad \overline{w_{e_2}}=1
\end{array}\right\} \begin{array}{l} 1 \leq e_2\\ \leq n-2\end{array}\\
\textbf{if } m_{0,n-1}=0:\\
\quad \overline{w_{n-1}}=0\\
\textbf{if } m_{0,n-1}>0:\\
\quad \textbf{if } m_{0,n-2}=m_{0,n-1} \textbf{ and } l_{0,n-2}<m_{0,n-1}:\\
\quad\quad \overline{w_{n-1}} \vee \overline{u_{n-2,m_{0,n-1}-1}}=1\\
\quad \textbf{if } l_{0,n-2}=m_{0,n-1}:\\
\quad\quad \overline{w_{n-1}}=1
\end{array}
\tag{13}
$$

Proof Any bounding condition $l_{0,e_2} \leq \sum_{i=0}^{e_2} w_i \leq m_{0,e_2}$ belongs to only one case of Lemma 1-3. Therefore, we can integrate them into Eq. (13) which is the SAT model based on sequential encoding method. □

According to Theorem 2, if we want to build the SAT model in Eq. (13), we need the lower and upper bounds of partial sum $\sum_{i=0}^{e_2} w_{e_2}$, $0 \leq e_2 \leq n-1$. From Theorem 1, we know that all the Matsui's bounding conditions can be converted into a series of linear inequalities.

Thus, we propose a method based on MILP to obtain these bounds. The whole procedure is demonstrated in Algorithm 1.

Algorithm 1 Determining the lower and upper bounds of conditions

Require: Matsui's bounding conditions $W_{ini}(R)$ and $W_{opt}(i), 0 \le i \le R-1$
Ensure: The lower bound l_{0,e_2} and upper bound m_{0,e_2} of $\sum_{i=0}^{e2} w_i$
1: Let $\mathcal{M}$ be an empty MILP model
2: **for** $0 \le r_1 \le r_2 \le R-1$ **do** ▷ Add the linear conditions in Eq. (9) into models
3: $\quad\mathcal{M}$.addConstr $\left(W_{opt}(r_2+1-r_1) \le \sum_{i=r_1}^{r_2} \sum_{j=0}^{\varpi-1} w_j^i\right)$
4: $\quad\mathcal{M}$.addConstr $\left(\sum_{i=r_1}^{r_2} \sum_{j=0}^{\varpi-1} w_j^i \le W_{ini}(R) - W_{opt}(r_1) - W_{opt}(R-1-r_2)\right)$
5: **end for**
—— **Lower bound** ——
6: Let $\mathcal{M}_l = \mathcal{M}$
7: $\mathcal{M}_l$.setObjective($\sum_{i=0}^{e_2} w_i$,Minimize) ▷ Set the objective function
8: $l_{0,e_2} = \mathcal{M}_l.optimize()$ ▷ (Solve the MILP model and obtain the lower bound)
—— **Upper bound** ——
9: Let $\mathcal{M}_m = \mathcal{M}$
10: $\mathcal{M}_m$.setObjective($\sum_{i=0}^{e_2} w_i$,Maximize) ▷ Set the objective function
11: $m_{0,e_2} = \mathcal{M}_m.optimize()$ ▷ (Solve the MILP model and obtain the upper bound)
12: **return** $\left(l_{0,e_2}, m_{0,e_2}\right)$

For all partial sums $\sum_{i=0}^{e_2} w_i, 0 \le e_2 \le n-1$, we can use Algorithm 1 to get their lower and upper bounds easily. Then, according to Theorem 2, the SAT model of combining Matsui's bounding conditions with sequential encoding method can be obtained. And we can use it to search for the optimal characteristics of ciphers.

4 Applications to block ciphers

We apply our new method to several block ciphers and compare it with the traditional method of combining Matsui's bounding conditions with sequential encoding method proposed by Sun et al. [31]. In order to make the comparison as fair as possible, we implement the two methods on the same platform (AMD Ryzen 9 5950X 16-Core 3.4G GHz) and the same SAT solver (CaDiCal [4]). All the source codes can be found in https://github.com/RNG2022/simplest-Sat-model

4.1 Description of some block ciphers

SPN Ciphers. PRESENT [6] has an SPN structure and uses 80- and 128-bit keys with 64-bit blocks through 31 rounds. In order to improve the hardware efficiency, it uses a fully wired diffusion layer. RECTANGLE [40] is very similar to PRESENT. It is a 25-round SPN cipher with the 64-bit block size. As an improved version of PRESENT, GIFT [2] is composed of two versions. GIFT-64 is a 28-round SPN cipher with the 64-bit block size, and GIFT-128 is a 40-round SPN cipher with the 128-bit block size.

Feistel Ciphers. LBlock [37] is a lightweight block cipher proposed by Wu and Zhang. The block size is 64 bits and the key size is 80 bits. It employs a variant Feistel structure and consists of 32 rounds. And TWINE [32] is a 64-bit lightweight block cipher supporting 80- and 128-bit keys. It has the same structure as LBlock and consists of 36 rounds.

Table 2 The comparison results of the two methods

Cipher	Total round	Property	K_{var}	K_{cnf}	K_{sol}
PRESENT	31 (Full)	differential	7.1%	11.1%	36.6%
		linear	2.0%	4.7%	46.6%
RECTANGLE	25 (Full)	differential	16.2%	20.0%	35.0%
		linear	14.1%	27.4%	94.0%
GIFT64	28 (Full)	differential	8.7%	12.3%	44.8%
		linear	19.0%	24.1%	94.7%
GIFT128	29	differential	19.0%	22.9%	30.7%
	25	linear	24.2%	28.5%	61.2%
LBlock	32 (Full)	differential	18.8%	52.5%	52.0%
		linear	18.0%	31.8%	58.7%
TWINE	36 (Full)	differential	14.4%	19.6%	45.5%
		linear	18.0%	30.8%	60.0%
SPECK32	22 (Full)	differential	23.0%	28.5%	69.0%
		linear	32.8%	43.0%	89.5%
SPECK48	18	differential	22.1%	33.5%	84.0%
	23 (Full)	linear	29.9%	39.5%	67.0%
SPECK64	27 (Full)	differential	18.3%	22.7%	76.5%
		linear	24.9%	34.2%	69.3%
SPECK96	10	differential	49.3%	54.5%	82.7%
	14	linear	47.2%	56.7%	67.8%
SPECK128	9	differential	51.8%	57.8%	90.3%
	10	linear	59.7%	68.3%	71.8%

ARX Ciphers. SPECK [3] is a family of lightweight block ciphers published by National Security Agency (NSA). It adopts ARX structure which takes the modular addition as its nonlinear operation. According to block size, SPECK family of ciphers are composed of SPECK2n, where $n \in \{16, 24, 32, 48, 64\}$.

4.2 The results of applications

In order to better illustrate our results, the following notations are introduced.

- M_{new} and M_{sun}: the methods proposed in Sect. 3 and [31], respectively.
- Var, Cnf, and T^{sol}: the number of variables, clauses and solving time of models, respectively.
- $K_{var} = \frac{Var_{new}}{Var_{sun}}$, $K_{cnf} = \frac{Cnf_{new}}{Cnf_{sun}}$ and $K_{sol} = \frac{T^{sol}_{new}}{T^{sol}_{sun}}$: The ratio of the total number of variables, total number of clauses and total solving time of models, respectively.
- P_{opt} and Cor_{opt}: the optimal probability and correlation of differential trails and linear trails, respectively.

We apply the two methods M_{sun} and M_{new} to the above SPN, Feistel and ARX ciphers to search for their optimal differential probabilities and linear correlations. The detailed results are shown in Table 4-15. The comparison of the two methods on the total number of variables, clauses and solving time of models are presented in Table 2. Take PRESENT as an example,

Table 3 New optimal differential probabilities and linear correlations

(a) Differential property

Cipher	Round	$\log_2^{P_{opt}}$	Var	Cnf	T^{sol}
GIFT128	30	-193	838 882	2 119 484	430.20h
	31	−198.415	473 100	1 176 426	38.28h
	32	−204.415	527 361	1 331 711	53.29h
	33	−210.415	523 013	1 331 731	55.56h
	34	−217.415	607 170	1 550 500	67.38h
	35	−224.83	627 866	1 601 828	58.78h
	36	−234.415	947 853	2 384 355	330.88h
	37	−240.415	642 079	1 604 643	71.70h
	38	−246.415	633 699	1 596 599	86.96h
	39	−253.415	729 939	1 845 704	31.96h
	40	−260.415	644 931	1 633 919	131.86h
SPECK48	19	-89	68 632	177 696	482.23h
	20	-96	77 548	197 656	673.51h
SPECK96	11	-58	125 910	311 320	674.98h
SPECK128	10	-49	150 920	381 667	358.21h

(b) Linear property

Cipher	Round	$\log_2^{Cor_{opt}}$	Var	Cnf	T^{sol}
GIFT128	26	-91	147 345	379 885	994.45h
	27	-94	91 807	236 723	631.82h
	28	-98	123 898	321 268	347.7h
	29	-101	93 844	244 787	156.13h
	30	-105	126 614	332 020	319.5h
	31	-108	95 881	252 851	125.83h
	32	-112	129 330	342 772	272.14h
	33	-117	173 725	455 905	306.97h
	34	-121	148 366	386 148	314.38h
	35	-126	197 520	510 125	764.42h
	36	-130	167 402	429 524	524.84h
	37	-133	125 704	324 443	145.39h
	38	-137	168 070	436 180	196.20h
	39	-140	126 205	329 435	155.27h
	40	-143	122 722	324 467	147.33h
SPECK96	15	-43	50 325	165 960	74.47h
	16	-48	69 323	222 298	289.07h
SPECK128	11	-31	55 745	175 540	261.10h

when searching for the optimal differential probabilities of every round from 1 to 31, the total number of variables, clauses and the time of solving SAT models needed by our method is only 7.1%, 11.1% and 36.6% of the method M_{sun}, respectively.

For full-round PRESENT, RECTANGLE, GIFT64, LBlock, TWINE, SPECK32 and SPECK64, the optimal differential probabilities and linear correlations of ciphers have been obtained. For GIFT128, SPECK48, SPECK96 and SPECK128, our method M_{new} finds some new differential probabilities and linear correlations covering more rounds which are listed in Table 3.

5 Conclusion

In this paper, we aim at finding a better way of combining Matsui's bounding conditions with sequential encoding method. By studying the properties of Matsui's bounding conditions, the general form of inequality constraint which can eliminate all the impossible solutions determined by Matsui's bounding conditions is proposed. Because the values of some auxiliary boolean variables in sequential encoding method can be determined, we propose a new method of integrating bounding conditions into SAT model. When applying our new method to search for the optimal differential probability and linear correlation of block ciphers, the total number of variables, clauses and solving time of SAT models are decreased. In addition, we find some new differential and linear characteristics covering more rounds. As a result, we obtain a more efficient search tool.

Because our method of combining bounding condition with sequential encoding method is general, it can be used to search for other kinds of distinguishers for ciphers. The wide applications will be done in the future. And for SPECK48, SPECK96 and SPECK128, some optimal differential probabilities or linear correlations of the full-round ciphers can not be obtained by the existing methods. How to speed up the search of these ciphers is a problem worth studying.

Acknowledgements The authors would like to thank the anonymous reviewers for their detailed comments and suggestions. This work is supported by the National Natural Science Foundation of China [Grant No.62102448,62202493].

Appendix

Table 4 Experimental results of PRESENT

(**a**) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	−2	669	3 112	0.1s	667	3 059	0.1s
2	−4	668	2 659	0.1s	472	2 217	0.1s
3	−8	4 203	14 763	0.2s	2 443	10 799	0.2s
4	−12	7 839	24 564	0.3s	3 739	15 479	0.3s
5	−20	32 809	92 575	3.7s	14 973	53 459	2.4s
6	−24	22 011	58 386	2.2s	8 491	29 135	1.1s
7	−28	29 679	76 683	2.4s	9 211	32 663	1.7s
8	−32	38 499	97 428	2.8s	9 931	36 191	1.5s
9	−36	48 471	120 621	3.0s	10 651	39 719	1.0s
10	−41	80 418	196 930	3.9s	8 999	31 662	1.6s
11	−46	98 990	238 786	8.1s	14 923	52 427	2.4s
12	−52	150 790	358 715	32.4s	28 420	97 945	9.7s

Table 4 continued

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
13	−56	107 355	252 813	5.4s	18 889	64 523	3.3s
14	−62	209 460	489 035	28.9s	35 040	118 125	16.7s
15	−66	145 437	337 053	10.0s	22 861	76 631	3.1s
16	−70	164 337	379 110	18.8s	22 717	78 431	2.1s
17	−74	184 389	423 615	8.3s	22 573	80 231	2.3s
18	−78	205 593	470 568	6.4s	22 429	82 031	2.5s
19	−82	227 949	519 969	5.1s	8 334	29 753	1.3s
20	−86	251 457	571 818	7.1s	8 334	30 449	1.3s
21	−90	276 117	626 115	7.6s	8 334	31 145	1.3s
22	−96	508 490	1 148 645	15.6s	28 141	101 795	4.0s
23	−100	335 511	755 283	11.8s	27 697	102 995	4.6s
24	−106	612 280	1 374 005	33.3s	34 129	117 935	16.6s
25	−110	400 665	896 547	17.2s	33 397	118 559	4.9s
26	−116	725 670	1 619 525	60.0s	40 117	134 075	36.3s
27	−120	471 579	1 049 907	31.8s	39 097	134 123	12.5s
28	−124	505 167	1 123 068	20.8s	14 034	47 405	1.4s
29	−128	539 907	1 198 677	18.2s	13 746	47 525	2.3s
30	−132	575 799	1 276 734	19.1s	13 458	47 645	4.9s
31	−136	612 843	1 357 239	18.3s	13 170	47 765	3.5s
Total		7 575 051	17 154 948	403.0s	539 417	1 895 896	147.3s

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	−1	351	1 790	0.6s	351	1 758	0.1s
2	−2	382	1 977	0.4s	318	1 817	0.1s
3	−4	1 369	6 599	0.7s	983	5 634	0.1s
4	−6	2 293	9 945	0.7s	1 391	7 754	0.1s
5	−8	3 473	13 867	0.7s	1 799	9 874	0.2s
6	−10	4 909	18 365	1.0s	2 207	11 994	0.3s
7	−12	6 601	23 439	1.2s	2 615	14 114	0.4s
8	−14	8 549	29 089	1.0s	3 023	16 234	0.4s
9	−16	10 753	35 315	1.1s	3 431	18 354	0.7s
10	−18	13 213	42 117	1.3s	3 839	20 474	0.8s
11	−20	15 929	49 495	1.7s	4 247	22 594	0.6s
12	−22	18 901	57 449	2.1s	4 655	24 714	1.1s
13	−24	22 129	65 979	2.2s	5 063	26 834	0.8s
14	−26	25 613	75 085	2.5s	5 471	28 954	0.9s
15	−28	29 353	84 767	2.8s	5 879	31 074	1.1s

Table 4 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
16	−30	33 349	95 025	2.7s	6 287	33 194	1.6s
17	−32	37 601	105 859	5.0s	6 695	35 314	1.9s
18	−34	42 109	117 269	3.5s	7 103	37 434	2.1s
19	−36	46 873	129 255	5.3s	7 511	39 554	1.6s
20	−38	51 893	141 817	5.5s	7 919	41 674	1.7s
21	−40	57 169	154 955	3.4s	8 327	43 794	2.2s
22	−42	62 701	168 669	6.0s	8 735	45 914	2.2s
23	−44	68 489	182 959	6.3s	9 143	48 034	3.0s
24	−45	74 533	197 825	7.7s	9 551	50 154	3.3s
25	−48	80 833	213 267	8.0s	9 959	52 274	3.6s
26	−50	87 389	229 285	8.8s	10 367	54 394	3.7s
27	−52	94 201	245 879	8.9s	10 775	56 514	4.6s
28	−54	101 269	263 049	8.5s	11 183	58 634	5.1s
29	−56	108 593	280 795	9.3s	11 591	60 754	3.7s
30	−58	116 173	299 117	10.0s	11 999	62 874	4.9s
31	−60	124 009	318 015	14.1s	12 407	64 994	9.5s
Total		9 731 820	22 048 710	133.3s	194 824	1 027 681	62.1s

Table 5 Experimental results of RECTANGLE

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	−2	669	2 392	2.9s	667	2 339	1.1s
2	−4	668	2 179	0.4s	472	1 737	0.3s
3	−7	2 659	8 117	0.8s	1 491	5 486	0.7s
4	−10	4 653	13 313	1.2s	2 129	7 678	0.7s
5	−14	11 193	30 351	1.3s	4 501	15 503	1.1s
6	−18	16 845	43 752	1.7s	6 085	20 039	1.1s
7	−25	50 313	125 223	7.6s	18 281	55 018	5.0s
8	−31	60 335	145 130	15.8s	21 455	60 545	9.9s
9	−36	63 766	150 466	18.8s	20 654	57 228	14.1s
10	−41	80 418	187 330	23.0s	23 402	64 540	16.6s
11	−46	98 990	228 226	70.5s	26 150	71 852	42.8s
12	−51	119 482	273 154	103.0s	28 898	79 164	27.1s
13	−56	141 894	322 114	227.8s	31 646	86 476	52.7s
14	−61	166 226	375 106	140.7s	34 394	93 788	57.1s
15	−66	192 478	432 130	256.9s	37 142	101 100	58.8s

Table 5 continued

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
16	−71	220 650	493 186	203.8s	39 890	108 412	75.2s
17	−76	250 742	558 274	354.1s	42 638	115 724	76.6s
18	−81	282 754	627 394	242.8s	45 386	123 036	98.5s
19	−86	316 686	700 546	287.3s	48 134	130 348	132.7s
20	−91	352 538	777 730	406.6s	50 882	137 660	137.9s
21	−96	390 310	858 946	479.1s	53 630	144 972	106.8s
22	−101	430 002	944 194	497.5s	56 378	152 284	111.5s
23	−106	471 614	1 033 474	335.0s	59 126	159 596	175.3s
24	−111	515 146	1 126 786	560.1s	61 874	166 908	170.5s
25	−116	560 598	1 224 130	621.7s	64 622	174 220	324.8s
Total		4 801 629	10 683 643	4 860.6s	779 927	2 135 653	1 698.9s

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	−1	367	1 246	1.6s	351	1 214	0.9s
2	−2	446	1 433	0.7s	318	1 273	0.4s
3	−4	1 705	4 967	1.4s	983	4 002	0.7s
4	−6	2 997	7 769	1.2s	1 391	5 578	0.8s
5	−8	4 673	11 147	1.3s	1 799	7 154	0.7s
6	−10	6 733	15 101	1.3s	2 207	8 730	1.0s
7	−13	14 268	30 114	3.6s	4 252	16 115	2.5s
8	−16	19 731	39 396	6.6s	5 473	19 691	4.5s
9	−19	26 058	49 926	9.8s	6 694	23 267	10.8s
10	−22	33 249	61 704	20.9s	7 915	26 843	21.6s
11	−25	41 304	74 730	48.2s	9 136	30 419	44.1s
12	−28	50 223	89 004	104.5s	10 357	33 995	74.6s
13	−31	60 006	104 526	234.6s	11 578	37 571	220.5s
14	−34	70 653	121 296	292.6s	12 799	41 147	271.6s
15	−37	82 164	139 314	380.6s	14 020	44 723	429.5s
16	−40	94 539	158 580	0.30h	15 241	48 299	778.5s
17	−42	71 037	118 311	368.5s	10 435	33 506	205.9s
18	−45	119 292	197 409	507.8s	16 162	52 415	875.7s
19	−48	134 115	220 227	0.36h	17 479	56 183	0.32h
20	−51	149 802	244 293	0.36h	18 796	59 951	0.30h
21	−54	166 353	269 607	0.34h	20 113	63 719	0.35h

Table 5 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
22	−57	183 768	296 169	0.41h	21 430	67 487	0.38h
23	−60	202 047	323 979	0.49h	22 747	71 255	0.48h
24	−63	221 190	353 037	0.52h	24 064	75 023	0.52h
25	−66	241 197	383 343	1.52h	25 381	78 791	1.39h
Total		1 997 917	3 316 628	4.86h	281 121	908 351	4.57h

Table 6 Experimental results of GIFT64

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	−1.415	590	2 747	0.3s	590	2 699	0.2s
2	−3.415	1 560	6 677	0.3s	1 268	5 947	0.2s
3	−7	4 554	16 630	0.5s	2 990	12 916	0.3s
4	−11.415	11 663	36 670	3.2s	6 281	24 437	0.5s
5	−17	28 744	81 820	15.5s	13 678	48 259	2.4s
6	−22.415	38 950	103 956	33.8s	16 090	53 830	19.4s
7	−28.415	65 899	168 535	110.9s	24 275	78 099	66.7s
8	−38	136 625	334 925	433.1s	49 795	147 570	343.9s
9	−42	73 534	175 738	74.6s	23 962	69 556	25.8s
10	−48	136 911	323 127	191.0s	38 249	112 630	62.1s
11	−52	110 934	259 130	33.0s	26 634	79 812	43.5s
12	−58	198 771	460 311	189.2s	42 257	128 014	54.8s
13	−62	156 014	358 650	56.6s	29 306	90 068	20.7s
14	−68	272 151	621 687	70.7s	46 265	143 398	60.1s
15	−72	208 774	474 298	46.8s	31 978	100 324	5.1s
16	−78	357 051	807 255	107.8s	28 561	86 231	38.6s
17	−82	269 214	606 074	51.2s	27 205	85 367	13.7s
18	−88	453 471	1 017 015	119.7s	30 997	94 787	56.1s
19	−92	337 334	753 978	59.5s	29 353	93 347	34.6s
20	−98	561 411	1 250 967	133.5s	33 433	103 343	59.6s
21	−102	413 134	918 010	82.6s	31 501	101 327	16.2s
22	−108	680 871	1 509 111	125.7s	35 869	111 899	75.3s
23	−112	496 614	1 098 170	87.5s	33 649	109 307	35.5s
24	−118	811 851	1 791 447	239.1s	38 305	120 455	142.2s
25	−122	587 774	1 294 458	120.8s	35 797	117 287	40.4s
26	−128	954 351	2 097 975	251.9s	40 741	129 011	137.8s

Table 6 continued

(**a**) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun}			M_{new}		
		Var	Cnf	T^{sol}	Var	Cnf	T^{sol}
27	−132	686 614	1 506 874	155.6s	37 945	125 267	11.8s
28	−138	1 108 371	2 428 695	365.3s	43 177	137 567	100.2s
Total		9 163 735	20 504 930	3 160.9s	800 151	2 512 754	1 416.4s

(**b**) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun}			M_{new}		
		Var	Cnf	T^{sol}	Var	Cnf	T^{sol}
1	−1	351	1 150	1.1s	351	1 118	0.8s
2	−2	382	1 337	0.3s	318	1 177	0.4s
3	−3	637	2 245	0.4s	445	1 765	0.4s
4	−5	2 039	6 879	0.8s	1 269	4 954	0.7s
5	−7	3 155	10 033	0.8s	1 741	6 562	0.8s
6	−10	7 077	21 216	1.5s	3 601	12 815	1.5s
7	−13	10 236	29 106	2.3s	4 822	16 247	2.2s
8	−16	13 971	38 244	4.5s	6 043	19 679	3.7s
9	−20	24 950	65 986	27.2s	10 250	31 940	18.8s
10	−25	41 805	106 810	218.3s	16 955	49 845	182.2s
11	−29	43 090	107 342	592.1s	16 742	47 540	460.1s
12	−31	25 795	63 539	175.1s	8 893	25 474	166.5s
13	−34	45 021	110 115	218.2s	13 705	39 935	215.0s
14	−37	52 500	127 317	250.5s	14 638	42 791	208.2s
15	−40	60 555	145 767	500.8s	15 571	45 647	345.1s
16	−43	69 186	165 465	462.0s	16 504	48 503	344.2s
17	−46	78 393	186 411	351.7s	17 437	51 359	357.0s
18	−49	88 176	208 605	256.1s	18 370	54 215	221.0s
19	−52	98 535	232 047	241.0s	19 303	57 071	330.8s
20	−55	109 470	256 737	227.0s	20 236	59 927	214.9s
21	−58	120 981	282 675	266.9s	21 169	62 783	338.5s
22	−61	133 068	309 861	253.0s	22 102	65 639	307.0s
23	−64	145 731	338 295	309.1s	23 035	68 495	310.4s
24	−67	158 970	367 977	271.8s	23 968	71 351	225.8s
25	−70	172 785	398 907	264.5s	24 901	74 207	456.5s
26	−73	187 176	431 085	283.2s	25 834	77 063	260.3s
27	−76	202 143	464 511	285.6s	26 767	79 919	262.8s
28	−79	217 686	499 185	311.7s	27 700	82 775	237.5s
Total		2 113 864	4 978 847	5 777.5s	402 670	1 200 796	5 473.2s

Table 7 Differential property of GIFT128

Round	$\log_2^{P_{opt}}$	M_{sun}			M_{new}		
		Var	Cnf	T^{sol}	Var	yCnf	T^{sol}
1	−1.415	1 182	5 499	0.2s	1 182	5 403	0.2s
2	−3.415	3 128	13 381	0.2s	2 548	11 931	0.2s
3	−7	11 939	42 911	0.7s	8 057	33 693	0.5s
4	−11.415	23 375	73 502	1.5s	12 713	49 269	1.4s
5	−17	48 201	137 955	7.9s	22 631	80 998	6.9s
6	−22.415	78 022	208 308	19.7s	32 698	108 934	17.8s
7	−28.415	131 979	337 655	98.1s	49 363	158 179	83.3s
8	−39	305 162	746 449	1.06h	115 588	337 447	0.71h
9	−45.415	272 180	645 604	0.74h	98 536	273 887	0.52h
10	−49.415	239 761	562 598	542.7s	72 419	206 125	201.9s
11	−54.415	345 062	802 966	726.5s	87 710	256 334	115.0s
12	−60.415	483 563	1 114 804	0.60h	110 573	324 151	229.8s
13	−67.83	664 028	1 515 923	2.00h	145 314	418 180	0.28h
14	−79	1 218 318	2 747 022	42.98h	316 984	856 761	8.06h
15	−85.415	856 156	1 912 402	22.88h	204 874	538 803	4.63h
16	−90.415	833 262	1 854 320	6.58h	176 946	472 134	0.63h
17	−96.415	1 095 855	2 430 141	7.86h	209 023	564 547	1.74h
18	−103.415	1 416 604	3 128 587	27.29h	255 346	687 908	2.79h
19	−110.83	1 597 380	3 513 947	42.54h	277 578	742 308	3.00h
20	−121.415	2 729 099	5 973 181	744.30h	495 133	1 285 212	151.29h
21	−126.415	1 528 822	3 334 794	35.71h	272 002	699 574	8.2h
22	−132.415	1 950 067	4 246 118	24.23h	314 263	818 523	5.52h
23	−139.415	2 444 925	5 311 943	44.26h	272 403	971 688	13.35h
24	−146.83	2 680 964	5 811 667	61.77h	394 602	1 026 020	26.42h
25	−157.415	4 447 707	9 611 825	744.50h	680 957	1 731 196	283.76h
26	−162.415	2 431 742	5 244 388	38.59h	367 058	927 014	20.19h
27	−168.415	3 046 199	6 562 735	79.10h	419 503	1 072 499	35.63h
28	−174.415	3 271 885	7 041 002	84.05h	419 187	1 080 583	39.76h
29	−181.83	4 018 764	8 637 027	126.33h	490 994	1 268 484	56.14h
Total		38 175 331	83 568 654	2137.78h	7 265 067	19 127 269	657.28h
30	−193	-	–	–	838 882	2 119 484	430.20h
31	−198.415	–	–	–	464 358	1 158 942	38.28h
32	−204.415	–	–	–	527 361	1 331 711	53.29h
33	−210.415	–	–	–	523 013	1 331 731	55.56h
34	−217.415	–	–	–	607 170	1 550 500	67.38h
35	−224.83	–	–	–	627 866	1 601 828	58.78h
36	−234.415	–	–	–	947 853	2 384 355	330.88h

Table 7 continued

Round	$\log_2^{P_{opt}}$	M_{sun} Var	Cnf	T^{sol}	M_{new} Var	yCnf	T^{sol}
37	−240.415	–	–	–	642 079	1 604 643	71.70h
38	−246.415	–	–	–	633 699	1 596 599	86.96h
39	−253.415	–	–	–	729 939	1 845 704	31.96h
40	−260.415	–	–	–	644 931	1 633 919	131.86h

Table 8 Linear property of GIFT128

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	Cnf	T^{sol}	M_{new} Var	Cnf	T^{sol}
1	−1	703	2 302	1.0s	703	2 238	0.8s
2	−2	766	2 681	0.4s	638	2 361	0.5s
3	−3	1 277	4 501	0.4s	893	3 541	0.4s
4	−5	4 087	13 791	0.9s	2 549	9 946	1.0s
5	−7	6 323	20 113	1.0s	3 501	13 186	1.3s
6	−10	14 181	42 528	2.1s	7 249	25 775	1.9s
7	−13	20 508	58 338	4.8s	9 718	32 711	4.6s
8	−17	38 338	104 234	24.0s	17 262	54 884	25.9s
9	−22	66 780	173 900	234.0s	29 480	87 725	224.1s
10	−26	70 814	178 870	640.3s	29 642	84 948	721.0s
11	−31	113 135	279 355	1.33h	44 955	125 305	1.55h
12	−36	142 550	345 035	7.85h	54 430	147 565	6.96h
13	−38	67 573	161 991	1.40h	23 083	62 978	0.37h
14	−41	115 848	276 465	2.83h	24 510	96 239	2.13h
15	−45	178 898	423 742	4.27h	49 422	137 796	4.23h
16	−48	153 843	342 028	2.99h	39 427	110 063	1.06h
17	−51	173 226	405 870	1.28h	40 360	113 927	1.17h
18	−56	328 690	765 185	5.46h	74 550	207 765	5.79h
19	−59	222 738	515 616	2.63h	48 706	134 603	3.74h
20	−64	416 330	958 975	22.39h	88 460	242 225	17.66h
21	−68	373 878	856 594	41.29h	78 746	212 388	23.98h
22	−74	629 715	1 434 747	536.54h	134 681	355 678	355.26h
23	−79	589 055	1 334 575	335.82h	129 035	333 305	192.01h
24	−82	387 213	874 722	57.26h	80 821	208,775	24.93h
25	−86	560 174	1 262 890	162.42h	109 634	284,772	84.86h
Total		4 676 643	10 859 048	1186.8h	1 132 455	3 090 699	725.96h

Table 8 continued

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	Cnf	T^{sol}	M_{new} Var	Cnf	T^{sol}
26	−91	–	–	–	147 345	379 885	994.45h
27	−94	–	–	–	91 807	236 723	631.82h
28	−98	–	-	-	123 898	321 268	347.7h
29	−101	–	–	–	93 844	244 787	156.13h
30	−105	–	–	–	126 614	332 020	319.5h
31	−108	–	–	–	95 881	252 851	125.83h
32	−112	–	–	–	129 330	342 772	272.14h
33	−117	–	–	–	173 725	455 905	306.97h
34	−121	–	–	–	148 366	386 148	314.38h
35	−126	–	–	–	197 520	510 125	764.42h
36	−130	–	–	–	167 402	429 524	524.84h
37	−133	–	–	–	125 704	324 443	145.39h
38	−137	–	–	–	168 070	436 180	196.20h
39	−140	–	–	–	126 205	329 435	155.27h
40	−143	–	–	–	122 722	324 467	147.33h

Table 9 Experimental results of LBlock

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	Cnf	T^{sol}	M_{new} Var	Cnf	T^{sol}
1	0	184	546	0.1s	184	522	0.1s
2	−2	1 053	3 524	0.2s	1 051	3 401	0.2s
3	−4	1 911	6 169	0.2s	1 615	5 360	0.2s
4	−6	3 057	9 511	0.2s	2 179	7 319	0.2s
5	−8	4 491	13 501	0.3s	2 743	9 278	0.2s
6	−12	11 070	31 656	0.5s	6 210	20 115	0.5s
7	−16	16 210	44 036	0.7s	8 410	25 880	0.5s
8	−22	32 571	84 505	1.8s	16 149	46 879	1.2s
9	−28	45 633	113 891	2.8s	21 609	59 682	1.8s
10	−36	80 208	193 906	5.0s	36 876	97 323	3.4s
11	−44	107 136	252 370	8.5s	47 748	121 452	5.8s
12	−48	73 530	170 916	4.0s	29 770	75 305	2.3s
13	−56	160 164	368 326	13.3s	60 420	151 638	9.2s
14	−62	150 563	342 553	14.1s	53 837	133 497	10.0s
15	−66	124 200	281 046	9.2s	40 110	100 020	6.2s
16	−72	198 877	447 903	13.4s	58 849	147 315	11.4s

Table 9 continued

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
17	−76	161 110	361 336	11.7s	43 690	109 890	7.7s
18	−82	253 911	567 365	19.0s	63 861	161 133	12.8s
19	−86	202 820	451 706	20.8s	47 270	119 760	13.6s
20	−92	315 665	700 939	20.7s	68 873	174 951	14.7s
21	−96	249 330	552 156	11.7s	50 850	129 630	6.5s
22	−102	384 139	848 625	18.2s	73 885	188 769	11.6s
23	−106	300 640	662 686	20.5s	54 430	139 500	9.7s
24	−112	459 333	1 010 423	21.8s	78 897	202 587	9.7s
25	−115	284 202	624 243	10.4s	45 218	117 120	5.7s
26	−121	536 886	1 177 618	22.3s	79 926	208 453	12.1s
27	−126	499 251	1 092 904	36.3s	72 563	188 404	16.5s
28	−131	537 885	1 175 710	26.5s	74 789	194 482	10.8s
29	−135	479 895	1 047 811	17.3s	62 455	163 690	8.4s
30	−141	720 202	1 570 430	34.3s	90 300	236 789	9.6s
31	−146	662 427	1 442 272	51.5s	81 743	213 268	18.5s
32	−151	706 821	1 537 174	39.2s	83 969	219 346	16.3s
Total		7 765 375	7 187 757	456.3s	1 460 479	3 772 758	237.3s

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	176	481	0.1s	176	465	0.1s
2	−1	623	1 981	0.1s	607	1 918	0.1s
3	−2	1 013	3 156	0.1s	877	2 934	0.1s
4	−3	1 499	4 524	0.1s	1 147	3 950	0.1s
5	−4	2 081	6 052	0.1s	1 417	4 966	0.1s
6	−6	4 353	11 893	0.2s	2 671	9 251	0.2s
7	−8	6 051	15 376	0.3s	3 331	11 279	0.3s
8	−11	11 098	26 227	0.5s	5 570	18 236	0.5s
9	−14	15 038	33 227	0.8s	6 910	21 852	0.8s
10	−18	25 040	52 116	1.4s	10 700	32 605	1.3s
11	−22	32 780	64 771	2.6s	13 100	38 565	2.2s

Table 9 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
12	−24	24 027	46 129	1.3s	8 737	25 595	1.2s
13	−27	37 802	71 199	3.2s	12 554	36 876	2.3s
14	−30	44 718	82 487	2.8s	13 830	40 364	1.9s
15	−33	52 210	94 607	3.7s	15 106	43 852	3.5s
16	−36	60 278	107 559	7.8s	16 382	47 340	3.7s
17	−37	33 647	59 590	2.5s	8 291	24 342	1.7s
18	−40	74 694	131 375	4.2s	16 918	50 300	2.4s
19	−42	62 541	109 018	3.4s	13 291	39 635	2.4s
20	−45	92 562	160 043	4.3s	18 594	55 532	3.1s
21	−47	76 662	131 575	4.1s	14 548	43 559	2.3s
22	−50	112 350	191 527	5.1s	20 270	60 764	3.1s
23	−52	92 223	156 244	4.5s	15 805	47 483	2.4s
24	−55	134 058	225 827	5.5s	21 946	65 996	3.6s
25	−56	72 217	121 220	2.8s	10 977	33 478	1.8s
26	−59	155 194	259 627	6.7s	22 098	68 188	2.1s
27	−62	168 926	280 835	9.3s	23 822	72 572	6.9s
28	−65	183 234	302 875	16.1s	25 546	76 956	5.2s
29	−66	97 669	161 024	4.3s	12 713	38 830	3.4s
30	−69	207 826	341 795	6.3s	25 442	78 636	5.7s
31	−72	223 670	366 075	16.2s	27 294	83 276	5.7s
32	−74	178 917	291 859	10.2s	21 097	64 415	6.2s
Total		2 285 177	3 912 294	130.4s	411 767	1 244 010	76.5s

Table 10 Experimental results of TWINE

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	184	761	0.6s	184	737	0.4s
2	−2	1 053	4 814	1.0s	1 051	4 691	1.1s
3	−4	1 911	8 104	1.1s	1 615	7 295	1.2s
4	−6	3 057	12 091	1.1s	2 179	9 899	1.3s
5	−8	4 491	16 726	1.1s	2 743	12 503	1.1s
6	−12	11 070	38 106	2.0s	6 210	26 565	1.9s
7	−16	16 210	51 561	2.1s	8 410	33 405	2.5s
8	−22	32 571	96 545	3.6s	16 149	58 919	3.3s
9	−28	45 633	127 436	4.1s	21 609	73 227	4.0s
10	−38	100 661	265 893	10.9s	47 575	147 587	8.6s
11	−46	111 870	283 105	15.2s	51 312	149 829	11.3s
12	−51	92 541	229 174	11.0s	38 657	111 682	7.9s
13	−58	148 588	362 181	22.8s	56 940	163 576	20.9s
14	−64	155 253	372 989	30.1s	55 307	157 479	15.8s
15	−68	127 790	304 341	14.3s	40 920	117 745	9.4s
16	−74	204 239	482 693	39.5s	59 647	172 963	28.9s
17	−77	131 330	308 567	15.0s	34 410	101 436	7.6s
18	−83	256 928	600 482	32.3s	61 348	183 183	17.8s
19	−88	247 479	574 738	35.2s	55 775	166 306	27.4s
20	−94	322 371	744 437	60.4s	68 985	205 247	21.8s
21	−97	202 482	465 815	14.0s	39 554	119 500	7.8s
22	−103	387 828	889 106	26.3s	70 014	214 123	12.6s
23	−107	303 395	692 916	10.5s	51 545	158 445	5.6s
24	−113	463 358	1 054 586	24.9s	74 690	230 279	13.5s
25	−116	286 598	650 531	11.1s	42 718	133 612	4.6s
26	−122	541 247	1 225 463	17.2s	75 383	238 483	7.9s
27	−126	417 660	943 011	18.5s	55 500	176 085	5.8s
28	−132	629 881	1 418 495	28.5s	60 760	189 025	6.6s
29	−136	483 370	1 085 931	21.8s	59 080	188 105	9.2s
30	−142	725 235	1 625 639	54.7s	64 580	201 525	12.6s
31	−146	553 880	1 238 931	28.3s	62 660	200 125	12.0s
32	−152	827 309	1 846 895	41.3s	68 400	214 025	15.1s
33	−155	501 770	1 118 447	22.8s	51 418	166 572	7.6s
34	−161	930 398	2 070 860	39.1s	56 350	178 372	6.8s
35	−166	848 643	1 885 174	68.0s	70 310	225 145	23.7s
36	−172	1 051 617	2 331 743	74.8s	76 510	239 965	21.4s
Total		11 169 901	25 428 287	805.3s	1 610 498	4 977 660	366.8s

Table 10 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	Cnf	T^{sol}	M_{new} Var	Cnf	T^{sol}
1	0	176	777	0.6s	176	761	0.3s
2	−1	607	3 165	0.7s	607	3 102	0.7s
3	−2	941	4 932	0.7s	877	4 710	0.7s
4	−3	1 339	6 892	0.8s	1 147	6 318	0.8s
5	−4	1 801	9 012	0.7s	1 417	7 926	0.7s
6	−6	3 633	17 221	1.2s	2 671	14 579	1.1s
7	−8	4 875	21 592	1.4s	3 331	17 495	1.4s
8	−11	8 666	35 699	2.3s	5 570	27 708	1.9s
9	−14	11 438	43 883	2.4s	6 910	32 508	2.3s
10	−18	18 640	66 916	4.0s	10 700	47 405	3.0s
11	−22	23 980	81 051	4.7s	13 100	54 845	3.6s
12	−24	17 403	56 785	2.7s	8 737	36 251	2.3s
13	−27	27 194	86 591	4.7s	12 554	52 268	3.6s
14	−30	31 950	99 063	5.0s	13 830	56 940	4.3s
15	−32	27 459	83 560	4.0s	10 975	45 503	2.8s
16	−35	41 594	124 467	6.2s	15 506	64 540	4.3s
17	−36	23 177	68 572	2.9s	7 885	33 598	1.6s
18	−39	51 370	150 395	5.3s	16 170	70 124	3.1s
19	−41	42 936	124 075	4.5s	12 778	55 487	3.0s
20	−44	63 446	181 175	6.4s	17 974	77 980	4.1s
21	−45	34 647	98 142	3.1s	9 087	40 254	1.2s
22	−48	75 398	211 967	5.2s	18 510	83 308	3.1s
23	−50	61 869	172 270	4.1s	14 581	65 471	2.2s
24	−53	89 906	248 123	5.8s	20 442	91 420	3.9s
25	−54	48 421	132 832	3.4s	10 289	46 910	2.0s
26	−57	104 034	283 779	5.6s	20 850	96 492	2.6s
27	−59	84 258	228 145	5.0s	16 384	75 455	3.2s
28	−62	120 974	325 311	8.0s	17 851	79 979	3.5s
29	−63	64 499	172 642	3.7s	11 491	53 566	2.2s
30	−66	137 278	365 831	7.8s	12 549	56 742	3.4s
31	−68	110 103	291 700	5.4s	12 619	57 946	2.5s
32	−71	156 650	412 739	7.0s	13 707	61 182	3.7s
33	−72	82 881	217 572	4.4s	12 693	60 222	2.3s
34	−75	175 130	458 123	7.4s	13 847	63 590	3.7s
35	−77	139 404	362 935	5.8s	13 885	64 730	2.9s
36	−80	196 934	510 407	9.4s	15 069	68 158	3.2s
Total		2 085 011	5 758 341	152.1s	396 769	1 775 473	91.2s

Table 11 Experimental results of SPECK32

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	79	294	0.5s	79	279	0.1s
2	−1	281	1 229	1.9s	281	1 170	0.1s
3	−3	783	3 154	2.1s	691	2 837	0.2s
4	−5	1 368	5 002	1.7s	1 000	3 995	0.2s
5	−9	3 925	12 826	2.6s	2 535	9 285	0.6s
6	−13	6 465	19 176	3.4s	3 665	12 425	1.8s
7	−18	11 838	32 782	9.3s	6 050	19 264	6.7s
8	−24	20 349	53 299	55.2s	9 653	28 875	41.9s
9	−30	28 511	71 702	417.5s	12 565	35 903	299.9s
10	−34	26 350	64 751	484.3s	10 340	29 245	248.0s
11	−38	32 265	78 226	805.1s	11 095	31 635	764.8s
12	−42	38 780	92 976	0.34h	11 850	34 025	852.1s
13	−45	36 328	86 427	680.1s	9 704	28 376	292.8s
14	−49	52 565	124 216	0.30h	12 495	37 085	698.4s
15	−54	73 638	172 510	0.61h	16 646	48 856	878.3s
16	−58	70 840	164 726	0.38h	15 160	44 165	690.1s
17	−63	97 188	224 542	1.34h	19 844	57 352	0.96h
18	−69	130 424	299 069	8.96h	26 796	75 411	5.81h
19	−74	127 386	290 218	28.08h	25 982	71 704	16.33h
20	−77	94 186	213 859	5.64h	17 642	49 148	4.25h
21	−81	129 125	292 506	9.80h	21 855	61 925	10.08h
22	−85	141 865	320 456	8.62h	22 385	63 865	5.92h
Total		1 124 539	2 623 946	64.74h	258 313	746 825	44.69h

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	111	455	0.1s	111	440	0.1s
2	0	190	924	0.1s	190	879	0.1s
3	−1	582	2 855	0.1s	582	2 722	0.1s
4	−3	1 398	6 232	0.2s	1 306	5 783	0.2s
5	−5	2 169	8 788	0.2s	1 801	7 604	0.3s
6	−7	3 120	11 749	0.5s	2 296	9 425	0.5s
7	−9	4 251	15 115	1.1s	2 791	11 246	0.8s
8	−12	7 654	25 655	3.8s	4 614	17 884	3.9s
9	−14	7 455	23 863	10.8s	4 081	15 482	6.1s
10	−17	12 526	38 639	46.1s	6 334	23 532	28.8s
11	−19	11 559	34 591	48.4s	5 371	19 718	37.6s

Table 11 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
12	−20	8 941	26 418	17.0s	3 673	13 886	30.0s
13	−22	15 399	44 977	41.7s	5 695	22 034	25.9s
14	−24	17 835	51 268	12.8s	6 145	23 765	26.4s
15	−26	20 451	57 964	15.8s	6 595	25 496	23.2s
16	−28	23 247	65 065	38.9s	7 045	27 227	35.7s
17	−30	26 223	72 571	62.2s	7 495	28 958	31.7s
18	−34	50 310	136 821	0.37h	14 570	53 795	622.0s
19	−36	34 419	92 200	0.37h	9 889	35 396	0.44h
20	−38	38 025	101 101	0.59h	10 249	36 947	0.43h
21	−40	41 811	110 407	0.34h	10 609	38 498	0.42h
22	−42	45 777	120 118	0.33h	10 969	40 049	0.33h
Total		373 453	1 047 776	2.09h	122 411	460 766	1.87h

Table 12 Experimental results of SPECK48

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	119	446	0.1s	119	423	0.1s
2	−1	425	1 869	0.3s	425	1 778	0.1s
3	−3	1 191	4 810	0.5s	1 051	4 325	0.2s
4	−6	2 966	10 551	0.8s	2 214	8 492	0.3s
5	−10	6 575	20 761	1.9s	4 215	14 875	1.3s
6	−14	10 590	30 741	6.4s	5 870	19 545	2.9s
7	−19	19 110	52 168	23.2s	9 494	29 980	18.4s
8	−26	37 868	97 805	174.1s	17 836	52 472	155.2s
9	−33	54 112	133 941	0.49h	24 176	67 280	0.60h
10	−40	72 932	175 413	4.18h	30 516	82 088	4.30h
11	−45	69 234	163 648	5.19h	26 174	69 748	5.29h
12	−49	69 125	161 871	3.08h	22 805	61 465	2.59h
13	−54	97 908	227 464	5.64h	28 712	78 076	4.66h
14	−58	95 090	219 421	1.33h	24 920	68 405	1.10h
15	−63	131 550	301 768	6.21h	31 250	86 404	4.06h
16	−68	151 335	345 052	8.63h	33 527	92 578	4.91h
17	−75	233 120	527 877	59.46h	50 800	137 776	55.15h
18	−82	269 972	606 885	192.27h	59 716	157 736	157.98h
Total		1 323 222	3 082 491	286.55h	373 820	1 033 446	240.69h
19	−89	-	-	-	68 632	177 696	482.23h
20	−96	-	-	-	77 548	197 656	673.51h

Table 12 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	167	695	0.1s	167	672	0.1s
2	0	286	1412	0.2s	286	1343	0.1s
3	−1	878	4367	0.4s	878	4162	0.2s
4	−3	2118	9544	0.4s	1978	8855	0.3s
5	−6	4624	18411	0.6s	3872	15988	0.5s
6	−8	5163	18832	1.4s	3757	14981	1.0s
7	−12	12405	41821	21.8s	8195	30970	10.4s
8	−15	13882	43731	79.5s	8266	29900	63.8s
9	−19	23105	69231	0.33h	12595	44030	0.36h
10	−22	23730	68419	0.95h	11786	40348	0.84h
11	−25	29116	81827	3.70h	13100	44684	3.44h
12	−28	35054	96431	6.67h	14414	49020	6.06h
13	−30	30711	83281	3.12h	11353	39134	2.50h
14	−33	47302	126663	10.28h	15958	55532	7.97h
15	−37	69365	182556	40.11h	22555	76760	36.48h
16	−39	47694	124006	29.34h	14476	49223	25.03h
17	−43	90305	232291	124.91h	25945	87810	106.15h
18	−45	61086	155641	86.19h	16510	55853	42.87h
19	−48	90332	228663	57.01h	22604	77364	38.81h
20	−51	100594	252651	51.20h	24010	81884	17.42h
21	−54	111408	277835	217.37h	25416	86404	151.05h
22	−57	122774	304215	335.77h	26822	90924	223.90h
23	−59	100227	247261	52.73h	20383	70010	20.59h
Total		1022326	2669784	1019.7h	305326	1055851	683.48h

Table 13 Experimental results of SPECK64

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	159	598	0.1s	159	567	0.1s
2	−1	569	2509	0.3s	569	2386	0.1s
3	−3	1599	6466	0.4s	1411	5813	0.2s
4	−6	3990	14199	1.0s	2982	11436	0.5s
5	−10	8855	27961	2.7s	5695	20075	2.4s
6	−15	17679	50812	15.8s	10079	32782	11.0s

Table 13 continued

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
7	−21	32 319	86 556	78.0s	16 779	50 841	78.6s
8	−29	62 991	159 427	0.39h	30 945	87 369	0.38h
9	−34	58 056	142 108	0.54h	25 640	70 444	0.46h
10	−38	60 690	146 291	0.35h	23 050	63 905	0.55h
11	−42	73 545	175 406	518.8s	24 065	67 775	551.8s
12	−46	87 640	207 156	524.9s	25 080	71 645	333.0s
13	−50	102 975	241 541	685.1s	26 095	75 515	508.7s
14	−56	170 401	396 040	0.50h	40 943	117 103	0.41h
15	−62	202 055	464 969	2.03h	48 083	133 931	2.21h
16	−70	308 286	702 316	47.58h	75 378	202 569	34.51h
17	−73	157 152	355 875	1.06h	36 120	96 728	1.01h
18	−76	173 082	391 331	0.73h	33 922	93 812	0.33h
19	−81	288 162	649 648	0.77h	51 086	143 332	0.53h
20	−85	266 705	599 311	0.35h	43 945	124 025	0.45h
21	−89	293 045	656 946	0.35h	43 875	125 725	0.30h
22	−94	386 793	864 742	0.52h	54 593	156 958	0.50h
23	−99	425 742	948 952	1.16h	58 454	166 876	0.85h
24	−107	709 857	1 575 649	14.96h	103 395	285 009	12.20h
25	−112	523 152	1 156 936	11.15h	78 776	211 876	10.08h
26	−116	471 520	1 040 961	3.88h	66 400	179 905	2.74h
27	−121	610 170	1 344 904	17.29h	80 786	220 300	11.97h
Total		5 497 189	12 409 610	104.43h	1 008 305	2 818 702	79.89h

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	223	935	0.1s	223	904	0.1s
2	0	382	1 900	0.2s	382	1 807	0.2s
3	−1	1 174	5 879	0.3s	1 174	5 602	0.2s
4	−3	2 838	12 856	0.4s	2 650	11 927	0.3s
5	−6	6 208	24 811	1.4s	5 200	21 556	1.0s
6	−9	9 622	34 583	3.9s	7 102	27 676	3.2s
7	−13	17 765	58 536	55.2s	11 785	43 300	40.1s
8	−17	25 205	77 401	452.1s	15 135	52 885	440.0s
9	−19	19 497	57 676	787.2s	10 267	35 840	417.7s
10	−21	23 502	68 269	161.9s	10 732	38 513	231.5s
11	−24	37 852	107 623	570.3s	15 604	56 260	377.1s
12	−27	45 730	127 067	742.3s	17 506	62 380	577.2s

Table 13 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun}			M_{new}		
		Var	Cnf	T^{sol}	Var	Cnf	T^{sol}
13	−30	54 352	148 123	0.57h	19 408	68 500	0.53h
14	−33	63 718	170 791	0.70h	21 310	74 620	0.65h
15	−37	93 445	246 156	16.18h	30 165	103 220	6.47h
16	−41	109 565	283 621	68.49h	34 755	115 285	46.92h
17	−43	74 577	191 080	4.35h	22 039	73 280	3.48h
18	−45	82 302	209 857	0.68h	21 760	74 465	0.55h
19	−47	90 399	229 471	0.61h	21 481	75 650	0.58h
20	−49	98 868	249 922	549.0s	21 202	76 835	643.7s
21	−52	144 912	364 211	108.0s	29 192	106 612	96.8s
22	−54	118 965	297 418	51.0s	22 489	82 889	32.2s
23	−59	263 694	653 938	0.76h	50 606	180 502	0.68h
24	−63	246 015	603 951	37.94h	50 065	169 085	29.54h
25	−66	215 848	526 530	41.97h	43 424	144 324	31.36h
26	−68	174 399	423 994	15.31h	32 791	110 429	8.52h
27	−70	186 123	451 606	0.51h	31 861	110 312	0.46h
Total		2 207 180	5 628 206	188.30h	550 308	1 924 658	130.53h

Table 14 Experimental results of SPECK96

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun}			M_{new}		
		Var	Cnf	T^{sol}	Var	Cnf	T^{sol}
1	0	239	902	0.8s	239	855	0.1s
2	−1	857	3 789	1.9s	857	3 602	0.1s
3	−3	2 415	9 778	2.6s	2 131	8 789	0.2s
4	−6	6 038	21 495	4.2s	4 518	17 324	0.7s
5	−10	13 415	42 361	6.4s	8 655	30 475	3.4s
6	−15	26 799	77 020	24.4s	15 359	49 870	22.7s
7	−21	49 007	131 244	163.8s	25 627	77 497	230.4s
8	−30	108 025	272 406	1.53h	54 445	151 910	1.49h
9	−39	159 420	384 536	41.24h	76 920	202 360	40.56h
10	−49	243 782	570 615	452.48h	111 848	283 107	367.75h
Total		609 997	1 514 146	495.50h	300 599	825 789	409.86h
11	−58	–	–	–	125 910	311 320	674.98h

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun}			M_{new}		
		Var	Cnf	T^{sol}	Var	Cnf	T^{sol}
1	0	335	1 415	0.1s	335	1 368	0.1s
2	0	574	2 876	0.1s	574	2 735	0.1s

Table 14 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
3	−1	1 766	8 903	0.2s	1 766	8 482	0.2s
4	−3	4 278	19 480	0.2s	3 994	18 071	0.2s
5	−6	9 376	37 611	1.3s	7 856	32 692	1.1s
6	−9	14 550	52 439	12.6s	10 750	42 012	10.5s
7	−13	26 885	88 776	200.6s	17 865	65 780	180.4s
8	−18	46 923	143 128	1.25h	28 679	98 698	1.12h
9	−22	53 435	154 236	10.24h	29 685	98 220	7.03h
10	−27	83 859	232 396	127.10h	42 863	137 626	107.60h
11	−31	88 445	237 556	260.22h	41 505	130 660	173.60h
12	−33	62 940	166 486	36.05h	26 008	83 255	14.02h
13	−36	96 992	253 923	44.06h	35 328	115 844	24.34h
14	−39	112 318	290 559	44.82h	37 094	122 908	27.07h
Total		602 676	1 689 784	523.80h	284 302	958 351	354.82h
15	−43	–	–	–	50 325	165 960	74.47h
16	−48	–	–	–	69 323	222 298	289.07h

Table 15 Experimental results of SPECK128

(a) Differential property

Round	$\log_2^{P_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	319	1 206	0.1s	319	1 143	0.1s
2	−1	1 145	5 069	0.1s	1 145	4 818	0.1s
3	−3	3 231	13 090	0.3s	2 851	11 765	0.3
4	−6	8 086	28 791	1.0s	6 054	23 212	0.7s
5	−10	17 975	56 761	3.5s	11 615	40 875	4.2
6	−15	35 919	103 228	36.7s	20 639	66 958	30.3
7	−21	65 695	175 932	343.8s	34 475	104 153	286.3
8	−30	144 825	36 520	2.74h	73 325	204 390	2.71h
9	−39	213 740	365 206	76.38h	103 720	272 600	68.75h
Total		490 935	1 264 859	79.23h	254 143	729 914	71.56h
10	−49	–	–	–	150 920	381 667	358.21h

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
1	0	447	1 895	0.1s	447	1 832	0.1s
2	0	766	3 852	0.2s	766	3 663	0.1s

Table 15 continued

(b) Linear property

Round	$\log_2^{Cor_{opt}}$	M_{sun} Var	M_{sun} Cnf	M_{sun} T^{sol}	M_{new} Var	M_{new} Cnf	M_{new} T^{sol}
3	−1	2358	11927	0.2s	2358	11362	0.2s
4	−3	5718	26104	0.4s	5338	24215	0.3s
5	−6	12544	50411	3.6s	10512	43828	2.9s
6	−9	19478	70295	23.2s	14398	56348	18.1s
7	−13	36005	119016	463.5s	23945	88260	308.5s
8	−18	62859	191896	2.85h	38471	132490	2.34h
9	−22	71595	206796	3.08h	39845	131900	2.35h
10	−27	112371	311596	98.78h	57551	184858	70.51h
Total		324141	993788	104.85h	193631	678756	75.29h
11	−31	–	–	–	55745	175540	261.10h

References

1. Ahmed Abdelkhalek Yu., Sasaki Y.T., Tolba M., Youssef A.M.: MILP modeling for (large) s-boxes to optimize probability of differential characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017).
2. Banik S., Pandey S.K., Peyrin T., Sasaki Y., Sim S.M., Todo Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, volume 10529 of Lecture Notes in Computer Science, pages 321–345. Springer, 2017.
3. Beaulieu R., Shors D., Smith J., Treatman-Clark S., Weeks B., Wingers L.: The SIMON and SPECK families of lightweight block ciphers. IACR Cryptol. ePrint Arch., p. 404 (2013).
4. Biere A.: Cadical at the sat race 2019. In Heule M., Järvisalo M., Suda M. (Ed.) SAT Race 2019—Solver and Benchmark Descriptions,Theory and Applications of Satisfiability Testing—SAT 2009, volume B-2019-1, pp. 8–9. University of Helsinki (2019).
5. Biham E., Shamir A.: Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, Advances in Cryptology—CRYPTO '90, Santa Barbara, California, USA, August 11–15, 1990, Proceedings, volume 537 of Lecture Notes in Computer Science, pp. 2–21. Springer (1990).
6. Bogdanov A., Knudsen L.R., Leander G., Paar C., Poschmann A., Robshaw M.J.B., Seurin Y., Vikkelsoe C.: PRESENT: an ultra-lightweight block cipher. In: Paillier P., Verbauwhede I. (Eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10–13, 2007, Proceedings, volume 4727 of Lecture Notes in Computer Science, pp. 450–466. Springer (2007).
7. Boura C., Coggia D.: Efficient MILP modelings for sboxes and linear layers of SPN ciphers. IACR Trans. Symmetric Cryptol. **2020**(3), 327–361 (2020).
8. Cui T., Jia K., Fu K., Chen S., Wang M.: New automatic search tool for impossible differentials and zero-correlation linear approximations. IACR Cryptol. ePrint Arch., p. 689 (2016).
9. Erlacher J., Mendel F., Eichlseder M.: Bounds for the security of ascon against differential and linear cryptanalysis. IACR Trans. Symmetric Cryptol. **2022**(1), 64–87 (2022).
10. Fu K., Wang M., Guo Y., Sun S., Hu L.: Milp-based automatic search algorithms for differential and linear trails for speck. In: Peyrin T. (Ed.) Fast Software Encryption—23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers, volume 9783 of Lecture Notes in Computer Science, pp. 268–288. Springer (2016).
11. Gu Z., Rothberg E., Bixby R.: Gurobi optimizer. http://www.gurobi.com/.
12. Kim S., Hong D., Sung J., Hong S.: Accelerating the best trail search on aes-like ciphers. IACR Trans. Symmetric Cryptol. **2022**(2), 201–252 (2022).
13. Kölbl S., Leander G., Tiessen T.: Observations on the SIMON block cipher family. In Gennaro R., Robshaw M. (Eds.) Advances in Cryptology—CRYPTO 2015—35th Annual Cryptology Conference,

Santa Barbara, CA, USA, August 16–20, 2015, Proceedings, Part I, volume 9215 of Lecture Notes in Computer Science, pages 161–185. Springer (2015).
14. Li T., Sun Y.: Superball: a new approach for MILP modelings of boolean functions. IACR Trans. Symmetric Cryptol. **2022**(3), 341–367 (2022).
15. Liu Y., Liang H., Li M., Huang L., Hu K., Yang C. Wang M.: STP models of optimal differential and linear trail for s-box based ciphers. Sci. China Inf. Sci. **64**(5) (2021).
16. Liu Y., Wang Q., Rijmen V.: Automatic search of linear trails in ARX with applications to SPECK and chaskey. In Manulis M., Sadeghi A-.R., Schneider S.A. (Eds.) Applied Cryptography and Network Security—14th International Conference, ACNS 2016, Guildford, UK, June 19–22, 2016. Proceedings, volume 9696 of Lecture Notes in Computer Science, pp. 485–499. Springer (2016).
17. Liu Z., Li Y., Jiao L., Wang M.: A new method for searching optimal differential and linear trails in ARX ciphers. IEEE Trans. Inf. Theory **67**(2), 1054–1068 (2021).
18. Matsui M.: Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23–27, 1993, Proceedings, volume 765 of Lecture Notes in Computer Science, pp. 386–397. Springer (1993).
19. Matsui M.: On correlation between the order of s-boxes and the strength of DES. In Alfredo De Santis, editor, Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings, volume 950 of Lecture Notes in Computer Science, pp. 366–375. Springer (1994.)
20. Mouha M., Preneel B.: Towards finding optimal differential characteristics for arx: Application to salsa20. Cryptology ePrint Archive, Report 2013/328 (2013). https://ia.cr/2013/328.
21. Mouha N., Wang Q., Gu D., Preneel B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu C., Yung, M, Lin D., (Eds.), Information Security and Cryptology—7th International Conference, Inscrypt 2011, Beijing, China, November 30–December 3, 2011. Revised Selected Papers, volume 7537 of Lecture Notes in Computer Science, pp. 57–76. Springer (2011).
22. Sasaki Y., Todo, Y.: New algorithm for modeling s-box in MILP based differential and division trail search. In: Farshim P., Simion E. (Eds.) Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8–9, 2017, Revised Selected Papers, volume 10543 of Lecture Notes in Computer Science, pp. 150–165. Springer (2017).
23. Sasaki Y., Todo Y.: New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In: Coron J.S., Nielsen J.B. (Eds.) Advances in Cryptology—EUROCRYPT 2017—36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30–May 4, 2017, Proceedings, Part III, volume 10212 of Lecture Notes in Computer Science, pp. 185–215 (2017).
24. Sinz C.: Towards an optimal CNF encoding of boolean cardinality constraints. In van Beek P. (Ed.) Principles and Practice of Constraint Programming—CP 2005, Sitges, Spain, October 1–5, 2005, Proceedings, volume 3709 of Lecture Notes in Computer Science, pp. 827–831. Springer (2005).
25. Stephen A.C.: The complexity of theorem-proving procedures. In Harrison M.A., Banerji R.B., Ullman J.D. (Eds.) Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3–5, 1971, Shaker Heights, Ohio, USA, pp. 151–158. ACM (1971).
26. Sun S., Hu L., S L.ong, Xie Y., Wang P.: Automatic security evaluation of block ciphers with s-bp structures against related-key differential attacks. In Lin D., Xu S., Yung M. (Eds.), Information Security and Cryptology - 9th International Conference, Inscrypt 2013, Guangzhou, China, November 27–30, 2013, Revised Selected Papers, volume 8567 of Lecture Notes in Computer Science, pp. 39–51. Springer, (2013).
27. Sun S., Hu L., Wang P., Qiao K., Ma X., Song L.: Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, Advances in Cryptology—ASIACRYPT 2014—20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Proceedings, Part I, volume 8873 of Lecture Notes in Computer Science, pp. 158–178. Springer (2014.)
28. Sun B., Liu Z., Rijmen V., Li R., Cheng L., Wang Q., AlKhzaimi H., Li C.: Links among impossible differential, integral and zero correlation linear cryptanalysis. In Gennaro R., Robshaw M. (Eds.) Advances in Cryptology—CRYPTO 2015—35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16–20, 2015, Proceedings, Part I, volume 9215 of Lecture Notes in Computer Science, pp. 95–115. Springer (2015).
29. Sun L., Wang W., Wang M.: Automatic search of bit-based division property for ARX ciphers and word-based division property. In: Takagi T., Peyrin T. (Eds.) Advances in Cryptology—ASIACRYPT

2017—23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3–7, 2017, Proceedings, Part I, volume 10624 of Lecture Notes in Computer Science, pp. 128–157. Springer (2017).
30. Sun L., Wang W., Wang M.: More accurate differential properties of LED64 and midori64. IACR Trans. Symmetric Cryptol. **2018**(3), 93–123 (2018).
31. Sun L., Wang W., Wang M.: Accelerating the search of differential and linear characteristics with the SAT method. IACR Trans. Symmetric Cryptol. **2021**(1), 269–315 (2021).
32. Suzaki T., Minematsu K., Morioka S., Kobayashi E.: Twine: A lightweight block cipher for multiple platforms. In Lars R. Knudsen and Huapeng Wu, editors, Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers, volume 7707 of Lecture Notes in Computer Science, pages 339–354. Springer, 2012.
33. Todo Y., Isobe T., Hao Y., Meier W.: Cube attacks on non-blackbox polynomials based on division property. In: Katz J., Shacham H. (Ed.) Advances in Cryptology—CRYPTO 2017—37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part III, volume 10403 of Lecture Notes in Computer Science, pp. 250–279. Springer (2017).
34. Udovenko A.: MILP modeling of boolean functions by minimum number of inequalities. IACR Cryptol. ePrint Arch., page 1099 (2021).
35. Wang S., Hu B., Guan J., Zhang K., Shi T.: Milp-aided method of searching division property using three subsets and applications. In: Galbraith S.D., Moriai S. (Ed.) Advances in Cryptology—ASIACRYPT 2019—25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part III, volume 11923 of Lecture Notes in Computer Science, pp. 398–427. Springer (2019).
36. Wu S., Wang M.: Security evaluation against differential cryptanalysis for block cipher structures. IACR Cryptol. ePrint Arch. p. 551 (2011).
37. Wu W., Zhang L.: Lblock: A lightweight block cipher. In Javier López and Gene Tsudik, editors, Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings, volume 6715 of Lecture Notes in Computer Science, pages 327–344, 2011.
38. Xiang Z., Zhang W., Bao Z., Lin D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Cheon J.H., Takagi T. (Ed)., Advances in Cryptology—ASIACRYPT 2016, Hanoi, Vietnam, December 4–8, 2016, Proceedings, Part I, volume 10031 of Lecture Notes in Computer Science, pp. 648–678 (2016).
39. Zhang Y., Sun S., Cai J., Hu L.: Speeding up MILP aided differential characteristic search with matsui's strategy. In Chen L., Manulis M., Schneider S.A. (Eds.) ISC 2018, Guildford, UK, September 9–12, 2018, Proceedings, volume 11060 of Lecture Notes in Computer Science, pp. 101–115. Springer (2018)
40. Zhang W., Bao Z., Lin D., Rijmen V., Yang B., Verbauwhede I.: RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci. China Inf. Sci. **58**(12), 1–15 (2015).
41. Zhou C., Zhang W., Ding T., Xiang Z.: Improving the milp-based security evaluation algorithm against differential/linear cryptanalysis using A divide-and-conquer approach. IACR Trans. Symmetric Cryptol. **2019**(4), 438–469 (2019).