New Automatic Search Tool for Impossible Differentials and Zero-Correlation Linear Approximations

Tingting Cui[1], Shiyao Chen[2], Keting Jia[3], Kai Fu[4], Meiqin Wang[2⋆]

[1] School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[2] Key Laboratory of Cryptologic Technology and Information Security,
Ministry of Education, Shandong University, Jinan 250100, China
[3] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[4] China Academy of Information and Communications Technology, Beijing 100191, China

Abstract. Impossible differential and zero-correlation linear cryptanalysis are two of the most powerful cryptanalysis methods in the field of symmetric key cryptography. There are several automatic tools to search such trails for ciphers with S-boxes. These tools focus on the properties of linear layers, and idealize the underlying S-boxes, i.e., assume any input and output difference pairs are possible. In reality, such S-box never exists, and the possible output differences with any fixed input difference can be at most half of the entire space. Hence, some of the possible differential trails under the ideal world become impossible in reality, possibly resulting in impossible differential trails for more rounds. In this paper, we firstly take the differential and linear properties of non-linear components such as S-box into consideration and propose a new automatic tool to search impossible differential trails for ciphers with S-box. We then generalize the tool to modulo addition, and apply it to ARX ciphers. To demonstrate the usefulness of the tool, we apply it to HIGHT, SHACAL-2, LEA, LBlock. As a result, it improves the best existing results of each cipher.
keywords Impossible differential cryptanalysis, zero-correlation linear cryptanalysis, MILP, automatic tool

1 Introduction

Impossible differential cryptanalysis (IDC) was introduced by Biham *et al.* and Knudsen to attack Skipjack in [2] and DEAL [18], respectively. Unlike the differential cryptanalysis [3] that aims to find a differential characteristic with high probability, IDC tries to find the best impossible differentials, i.e., to find the longest differentials with probability 0. It is a very powerful cryptanalysis method. Since it was proposed, it has been used to analyze security of lots of block ciphers such as AES [22], Camellia [5, 8]. As the counterpart of IDC, zero-correlation linear cryptanalysis (ZCLC), a variant of linear cryptanalysis [23], was proposed by Bogdanov *et al.* in [6]. Similar to the idea of IDC, its purpose is to find a linear approximation with probability exactly 1/2. In [30], Sun *et al.* proposed that in some cases, a zero-correlation linear approximation was equivalent to an impossible differential.

How to find the best impossible differential for a target cipher is a focus point in the field of symmetric ciphers. It is not always possible to find the best impossible differentials by hand as the number possibilities can be far beyond the reach of human. Hence, the automatic search of impossible differentials received lots of attention, ad several approaches have been proposed such as $\mathcal{U}$-method [17], UID-method [21] and the extended tool by Wu and Wang in [40] [5]. So far, all these methods above treat the underlying S-box (substitution-box) used in the target cipher as an ideal S-box, i.e., all input and output difference transitions are possible. Under such assumption, the length

[⋆] The corresponding author

[5] This method is renamed as WW-method through this paper.

of IDC depends only on the linear layers. However, S-box used in practical ciphers can never be ideal, i.e., some input/output difference transitions under an S-box in reality will never happen, or happen only under some constraints when the actual value falls in a small set. Due to this, some possible differentials in the ideal world will become impossible. In other words, it is possible to find IDCs of possibly more rounds which could not be found in previous tools.

The second limitation of the previous tools are their inapplicability to ARX ciphers due to the complication of modelling the modular addition. In this paper, we, for the first time, take the differential property of non-linear components such as S-box and modular addition into consideration. Under this model, it will be more accurate to evaluate the security of target block ciphers and more likely to find longer impossible differentials. In parallel to this work, Sasaki and Todo [28] proposed a similar tool, in which they only consider to search impossible differentials for the ciphers with S-box and Sun *et al.* presented a search tool for impossible differential by using constraint programming in [29].

Inspired by the automatic search of differentials and linear approximations with Mixed Integer Linear Programming (MILP) method introduced by [31, 10], we aim to search the impossible differentials and zero-correlation linear approximations[6] with MILP as well. MILP problem is a mathematical optimization problem in which only some variables are constrained to be integers and the goal is to find the minimum or maximum of the objective function, for instance, covering problem and packing problem. It was introduced into differential and linear cryptanalysis by Mouha *et al.* and Wu *et al.* in [24] and [39] respectively, later improved in [32, 31, 10, 33]. According to its applications on the search of differentials and linear approximations for block ciphers, every operation in a certain cipher can be exactly described with linear inequalities system including non-linear operations such as S-box and modular addition. By exploiting mathematical optimization software which can expedite the feasible and optimized solution, we can search the optimal characteristic for the target cipher with suitable executable time. In this paper, we propose an algorithm to automatically search impossible differentials and zero-correlation linear approximations based on MILP method.

1.1 Contributions

MILP method uses the idea of inequalities system to describe the propagation of difference. By taking input and output differences of each component as variables, a special set of inequalities are given to link these variables. If one input/output pair is a possible differential propagation pattern for this component, its corresponding variables must be a solution of this set of inequalities, otherwise, it is not a solution. Take 4-bit S-box $(x_0, x_1, x_2, x_3) \xrightarrow{S} (y_0, y_1, y_2, y_3)$ as an example, we can find out an inequalities system whose variables are $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3)$ and solutions exactly denote its all possible differential patterns. By combining all inequalities for all components in a target cipher together, we can use the whole system to describe the propagation of difference in this target cipher. If we fix the input and output differences of this target cipher, it is easy to solve this linear inequalities system to get one specific differential characteristic. But sometimes the system cannot be solved which means it is overcontraint. So detecting whether the system is infeasible or not can be a new way to find impossible differential characteristics. In this paper, we exactly utilize this idea to construct a

[6] For the sake of simplicity, zero-correlation linear approximation is renamed as ZC approximation through this paper.

new automatic search tool. So do ZC approximations. Our contributions are shown as follows.

Propose an automatic tool to search impossible differentials and ZC approximations for both ARX ciphers and ciphers with S-box. IDC and ZCLC are two efficient cryptanalysis methods in the field of symmetric ciphers. In this paper, we propose an automatic method to search impossible differentials and ZC linear approximations for ARX ciphers and ciphers with S-box by MILP method. It is the first published widely applicable search tool for ARX ciphers and a strong general tool for ciphers with S-box, which takes the differential and linear properties of non-linear components into consideration based on [10] and [31]. Comparing with previous methods which regard S-box used in ciphers as ideal S-box, we can not only find the previous results, but also be able to find longer impossible differentials and ZC approximations with this new tool. Further more, with this tool, the security of lots of ARX ciphers and lightweight block ciphers against IDC and ZCLC can be evaluated more accurately and easily.

Application on HIGHT Cipher. HIGHT cipher, introduced by Hong *et al.* at CHES 2006 [14], is an ISO standard lightweight block cipher. Its block size and key size are 64 bits and 128 bits respectively, and it totally has 32 rounds. Its longest previous impossible differential and ZC approximation are both 16 rounds, which are introduced in [20, 9, 26] and in [36] respectively. In our work, we use our automatic tool to search all cases of 17-round impossible differentials (ZC approximations) that both hamming weights of input and output differences (masks) are one. As a result, we totally find 4 impossible differentials and 4 ZC approximations for 17-round HIGHT, which are the longest ones until now. The results are summarized in Table 1.

Application on SHACAL-2 Cipher. SHACAL-2 [11], proposed by Handschuch and Naccache, was selected as one of the four block ciphers by NESSIE. Its block size is 256 bits and key size is various from 128 bits to 512 bits. SHACAL-2 has totally 64 rounds and its round function is based on the compression function of the hash function SHA-2. The longest previous impossible differential of SHACAL-2 was presented by Hong *et al* in [12] and had 14 rounds. In this paper, by using our new automatic search tool, we search 2^{16} cases and find out 15-round impossible differentials, which are the longest ones so far. The results for SHACAL-2 are summarized in Table 1.

Application on LEA Cipher. LEA is a block cipher proposed by Hong *et al.* in [13], which can provide a high-speed software encryption on general-purpose processors. It has three versions (blocksize/keysize/rounds): 128/128/24, 128/192/28 and 128/256/32. This cipher adopts ARX construction and operates on 32-bit word. The previous best impossible differential and ZC approximation are proposed in [13], which had 10 rounds and 7 rounds respectively. In this paper, based on analyzing LEA cipher's construction, we search out 3 ZC approximations for 10-round LEA with our search tool. These trails are the best ones so far. What's more, the security bound should be extended more rounds than the given rounds claimed in [13] according to our results.

Application on LBlock Cipher. LBlock, designed by Wu and Zhang in [41], is an efficient lightweight block cipher. Its block size and key size are 64 bits and 80 bits. It applies a 32-round modified Feistel structure. Under the related-key setting, Minier and Naya-Plasencia found a 15-round related-key impossible differential in [38], then Wen *et al.* found two 16-round related-key impossible differentials in [37]. But Wen *et al.*'s two differentials are right only under part of master key pairs which satisfy one of the given two key differences. With our new search tool, we build a MILP model for LBlock and find out eighteen 16-round related-key impossible differentials. These impossible differentials break the limitation existed in Wen *et al.*'s work. As long as the master key pair satisfies one of the given differences, such related-key impossible differential is right in our work. The results for LBlock are summarized in Table 1.

Table 1. Summary of results for HIGHT, SHACAL-2, LEA, LBlock

Cipher	Type	Round	Resource
HIGHT	Imp. diff.	16	[20]
	Imp. diff.	16	[9]
	Imp. diff.	16	[26]
	Imp. diff.	**17**	**Sec. 4.1**
	ZC approx.	16	[36]
	ZC approx.	**17**	**Sec. 4.1**
SHACAL-2	ZC approx.	12	[35]
	Imp. diff.	14	[12]
	Imp. diff.	**15**	**Sec. 4.2**
LEA	Imp. diff.	10	[13]
	Imp. diff.	**10**	**Sec. 4.3**
	ZC approx.	7	[13]
	ZC approx.	**10**	**Sec. 4.3**
LBlock	RK Imp. diff.	15	[38]
	RK Imp. diff. *	16	[37]
	RK Imp. diff.	**16**	**Sec.4.4**

[1] (RK) Imp. diff.: (Related-key) impossible differential.
[2] ZC approx.: Zero correlation approximation.
[3] *: This related-key impossible differential is right only for part of master key pairs under the given difference of master key.

1.2 Outline

This paper is organized as follows. In section 2, we propose an automatic tool to search impossible differentials and ZC approximations for both ARX ciphers and ciphers with S-box. Then in section 3, a verification algorithm is presented. As applications, we use this tool to search impossible differentials and ZC approximations for several ciphers in section 4. At last, section 5 concludes this paper.

2 Automatic Tool for Search of Impossible Differentials and ZC Approximations

In this section, we propose an automatic tool to search impossible differentials for both ARX ciphers and ciphers with S-box. Similar to the idea of MILP models for differential cryptanalysis in previous work, we firstly utilize linear inequalities to exactly describe every component in the target cipher as well. But we are indifferent to the objective function, only interested in whether there is a solution for the whole inequalities system with fixed input and output differences or not. If not, the fixed input and output differences can lead to an impossible differential, which is expected. In section 2.1 and 2.2, we build the models to search impossible differentials for ARX ciphers and ciphers with S-box respectively. Since we do not care about the probability of each differential pattern for non-linear component, we redescribe the constraints for modular addition with 8 linear inequalities, reducing about 40% comparing with these proposed by Fu *et al.* for searching differentials in [10]. In section 2.3, we briefly introduce the model to search ZC approximations.

2.1 Impossible Differential Model for ARX Ciphers

ARX ciphers are designed by combining modular addition, bit rotation and XOR operations. For each operation, there is a set of linear inequalities to equivalently depict it.

Constraints for XOR and Bit Rotation. Both XOR and bit rotation are linear operations. For every XOR operation with bit-level input and output differences a, b and c, the constraint below can perfectly describe it:

$$a + b + c = 2d_{\oplus}, \tag{1}$$

where $d_{\oplus}$ is a dummy bit variable.

For the case of circular shift, since it only transforms the position of its input bits, so we can easily build linear equations between related bits.

Constraints for Modular Addition. In [19], Lipmaa and Moriai proposed a method to verify whether a given differential characteristic is possible or not. For sake of simplicity, Fu *et al.* summarized this method into a theorem in [10] as follows:

Theorem 1 (see [19, 10]). *The differential* $(\alpha, \beta \to \gamma)$ *satisfies* $\gamma = \alpha + \beta$ *iff* $(\alpha[0] \oplus \beta[0] \oplus \gamma[0]) = 0$ *and* $\alpha[i-1] = \beta[i-1] = \gamma[i-1] = \alpha[i] \oplus \beta[i] \oplus \gamma[i]$ *when* $\alpha[i-1] = \beta[i-1] = \gamma[i-1]$, $i \in [1, n-1]$.

In order to describe the first condition $\alpha[0] \oplus \beta[0] \oplus \gamma[0] = 0$ in Theorem 1, we utilize one equality to satisfy it as follows:

$$\alpha[0] + \beta[0] + \gamma[0] = 2d_{\oplus}, \tag{2}$$

where $d_{\oplus}$ is a dummy bit variable.

When $i \in [1, n-1]$, there are 56 possible patterns for $(\alpha[i], \beta[i], \gamma[i], \alpha[i+1], \beta[i+1], \gamma[i+1])$ to meet the second condition in Theorem 1. We propose 8 linear inequalities whose solution set is exactly these 56 possible patterns for each $i \in [1, n-1]$ as follows.

$$\begin{aligned}
-\alpha[i] - \beta[i] - \gamma[i] + \alpha[i+1] + \beta[i+1] + \gamma[i+1] &\geq -2,\\
\alpha[i] + \beta[i] + \gamma[i] - \alpha[i+1] - \beta[i+1] - \gamma[i+1] &\geq -2,\\
\alpha[i] + \beta[i] + \gamma[i] + \alpha[i+1] + \beta[i+1] - \gamma[i+1] &\geq 0,\\
\alpha[i] + \beta[i] + \gamma[i] + \alpha[i+1] - \beta[i+1] + \gamma[i+1] &\geq 0,\\
\alpha[i] + \beta[i] + \gamma[i] - \alpha[i+1] + \beta[i+1] + \gamma[i+1] &\geq 0,\\
-\alpha[i] - \beta[i] - \gamma[i] + \alpha[i+1] - \beta[i+1] - \gamma[i+1] &\geq -4,\\
-\alpha[i] - \beta[i] - \gamma[i] - \alpha[i+1] + \beta[i+1] - \gamma[i+1] &\geq -4,\\
-\alpha[i] - \beta[i] - \gamma[i] - \alpha[i+1] - \beta[i+1] + \gamma[i+1] &\geq -4.
\end{aligned} \tag{3}$$

Note that in [10], Fu *et al.* used 13 linear inequalities to exactly describe the differential patterns of modular addition because of taking the probability of each pattern into consideration. In our model, we don't care about the probability, but just focus on whether the probability of each pattern is nonzero or not. Thus, we only need 8 new linear inequalities to describe the differential propagation on modular addition, which can accelerate the search process.

Up to now, by setting the input and output differences of each operation in the target ARX cipher as corresponding binary variables and constructing linear inequalities system among these variables following the rules in this section, the differential propagation on each operation can be exactly described. By combining all inequalities together, the whole inequalities system can perfectly describe the target cipher, and its every solution is a differential characteristic. When we fixed the input and output differences, if the inequalities system is infeasible, it means this is an impossible differential. By traversing a special set of input/output differences in the MILP model, we can confirm whether there exists an impossible differential or not for a certain reduced-round ARX cipher within this set. Note that due to the time complexity, it is hard to search all input/output differences, so this special set needs to be decided carefully and it always depends on the feature of the given cipher. Without loss of generality, we denote such set as $(\Delta \to \Gamma)$, where Δ and Γ are chosen sets of input and output differences respectively. In Algorithm 1, we explain how to implement the search process of impossible differentials.

Algorithm 1: General search process for impossible differentials

```
// Step 1: Construct the MILP model.
Set the input and output differences of each operation as binary variables;
Add linear inequalities for each operation of the target cipher so as to link all variables in the
  model;
// Step 2: Search out all impossible differentials within a certain set of input
   and output differences.
Decide the set of input differences $\Delta$ and the set of output differences $\Gamma$;
for input differences $\Delta x_i \in \Delta$ do
    for output differences $\Delta y_j \in \Gamma$ do
        Add all constraints about the fixed input and output differences into MILP model;
        Start to solve this model;
        if solver finds a solution then
            // The current input and output differences is a possible
               differential.
            Break;
        else
            // The current input and output differences lead an impossible
               differential.
            Store current input and output differences;
```

2.2 Impossible Differential Model for Ciphers with S-box

Unlike ARX ciphers, lots of block ciphers use S-box layer as the non-linear operations rather than modular addition and their linear operations may be more complicated by including many XOR, rotation operations and simple permutations. For the sake of simplicity, we don't depict the linear operations in detail as they have been exactly described in section 2.1.

Constraints for S-box operation. Assume S is an $m \times l$-bit S-box that $(y_0, y_1, \dots, y_{l-1}) = S(x_0, x_1, \dots, x_{m-1})$ and $(\Delta x, \Delta y)$ are input and output differences. The set of all its differential patterns is $DT = \{(\Delta x, \Delta y) | Pr[\Delta x \xrightarrow{S} \Delta y] > 0\}$. According to Sun *et al.*'s work in [31], we can build linear inequalities system to exactly depict DT with the help of software SAGE [7] and the greed algorithm in [33]. For more details, please refer to [33].

Remark: In this section, we describe the differential propagations of S-box, modular addition and linear operations with MILP model. Actually, some ciphers are designed by other nonlinear components such as bit-level AND and OR and so on. These operations can be described similarly as that for S-box. As far as we know, most operations used in current ciphers can be exactly described. Especially, the large S-box such as 8-bit size can already be exactly described [27], althrough it needs lots of linear inequalities.

2.3 Zero-Correlation Linear Model for ARX Ciphers and Ciphers with S-box

In order to search ZC approximations for ARX ciphers and ciphers with S-box, it is necessary to consider about the linear approximations of basic operations such as XOR, branching, bit rotation and modular addition operations. Before studying the construction of MILP model for search of ZC approximations, we introduce the linear approximations over XOR and branching operations proposed by Biham in [4] as follows, where "$\cdot$" means the scalar product of binary vectors.

Lemma 1 (XOR operation [4]). *Let $h(x_1, x_2) = x_1 \oplus x_2$, α_1, α_2 are the input masks of x_1 and x_2 respectively, β is the output mask, then the correlation $C(\beta \cdot h(x_1, x_2), \alpha_1 \cdot x_1 \oplus \alpha_2 \cdot x_2) \neq 0$ if and only if $\beta = \alpha_1 = \alpha_2$.*

Lemma 2 (Branching operation [4]). *Let $h(x) = (x, x)$, α is the input mask, β_1, β_2 are the output masks of $h(x)$, then the correlation $C((\beta_1, \beta_2) \cdot h(x), \alpha \cdot x) \neq 0$ if and only if $\alpha = \beta_1 \oplus \beta_2$.*

Following Lemma 1 and 2, we start to construct MILP model to search ZC approximations for ARX ciphers.

Constraints for Branching, XOR and Bit Rotation. Assumed that the input mask of braching operation is α, the output masks are β_1 and β_2. According to Lemma 2, $\alpha = \beta_1 \oplus \beta_2$, so similar to (1) in section 2.1, we have the following equality to exactly describe its each bit operation.

$$\alpha[i] + \beta_1[i] + \beta_2[i] = 2d_{\oplus}, \tag{4}$$

where $d_{\oplus}$ is a dummy bit variable.

In the light of Lemma 1, some linear equations between input masks and output mask can perfectly describe the linear approximation of XOR operation. Besides, the bit rotation operation is a simple permutation that we can list some equations for the related bits.

Constraints for Modular Addition. In [34, 25], a method to calculate the correlation of modular addition is given as follows.

[7] Inequality_generator() function in the sage.geometry.polyhedron class of SAGE. The website of SAGE is: http://www.sagemath.org/.

Theorem 2 ([34, 25]). *For the linear approximation of addition modulo 2^n, let the input masks and output mask be $\alpha_1 = (\alpha_1[n-1], \dots, \alpha_1[0])$, $\alpha_2 = (\alpha_2[n-1], \dots, \alpha_2[0])$ and $\beta = (\beta[n-1], \dots, \beta[0])$ respectively, where $\alpha_1, \alpha_2, \beta \in \mathcal{F}_2^n$, and let the vector $u = (u[n-1], \dots, u[0])$ satisfy $u[i] = 4\beta[i] + 2\alpha_1[i] + \alpha_2[i], 0 \le u[i] < 8, 0 \le i < n$. Then the correlation can be computed as follows:*

$$cor_{\boxplus}(\beta, \alpha_1, \alpha_2) = LA_{u[n-1]}A_{u[n-2]} \cdots A_{u[0]}C, \tag{5}$$

where $A_r, 0 \le r < 7$, is 2×2 matrice,

$$A_0 = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, A_1 = A_2 = -A_4 = \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

$$A_7 = \frac{1}{2}\begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}, -A_3 = A_5 = -A_6 = \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

L is a row vector $L = (1, 0)$, and C is a column vector $C = (1, 1)^T$.

In order to quickly calculate the correlation shown in Theorem 2, Nyberg and Wellén utilized the automaton to calculate (5) by multiplication from left to right [25]. They let $e_0 = L = (1, 0)$ and $e_1 = (0, 1)$, then the state transitions for addition modulo 2^n is as follows:

$$\varepsilon_n = e_0 \xrightarrow{u[n-1]} \varepsilon_{n-1} \xrightarrow{u[n-2]} \varepsilon_{n-2} \rightarrow \dots \rightarrow \varepsilon_1 \xrightarrow{u[0]} \varepsilon_0.$$

Where $\varepsilon_j \in \{e_0, e_1\}, 0 \le j < n$. For more details, please refer to [25].

Based on the work above, Fu *et al.* in [10] set a $0-1$ variable s_i that $s_i = 0$ if $\varepsilon_i = e_0$, otherwise, $s_i = 1$, then utilized $(s_{i+1}, \beta[i], \alpha_1[i], \alpha_2[i], s_i)$ to describe the state transition from ε_{i+1} to ε_i, namely $e_{s_{i+1}}A_{u[i]} = e_{s_i}$. They found that there were 10 possible transitions for the vector $(s_{i+1}, \beta[i], \alpha_1[i], \alpha_2[i], s_i)$, and listed eight linear inequalities exactly satisfying these 10 possible transitions with the help of SAGE and the greedy algorithm in [33], which are shown as follows:

$$\begin{aligned}
s_{i+1} - \beta[i] - \alpha_1[i] + \alpha_2[i] + s_i \ge 0, &\quad s_{i+1} + \beta[i] + \alpha_1[i] - \alpha_2[i] - s_i \ge 0, \\
s_{i+1} + \beta[i] - \alpha_1[i] - \alpha_2[i] + s_i \ge 0, &\quad s_{i+1} - \beta[i] + \alpha_1[i] - \alpha_2[i] + s_i \ge 0, \\
s_{i+1} + \beta[i] - \alpha_1[i] + \alpha_2[i] - s_i \ge 0, &\quad s_{i+1} - \beta[i] + \alpha_1[i] + \alpha_2[i] - s_i \ge 0, \\
-s_{i+1} + \beta[i] + \alpha_1[i] + \alpha_2[i] + s_i \ge 0, &\quad s_{i+1} + \beta[i] + \alpha_1[i] + \alpha_2[i] + s_i \le 4.
\end{aligned}$$

Note that there is an additional constraint $\varepsilon_n = e_0$, hence, the constraints include $8 \times n + 1$ linear inequalities for linear approximation of addition modulo 2^n.

Constraints for S-box operation. Assume S is an arbitrary $m \times l$ S-box that $(y_0, y_1, \dots, y_{l-1}) = S(x_0, x_1, \dots, x_{m-1})$, and α, β are input and output masks respectively, then the set of all its meaningful linear approximations is $LT = \{(\alpha, \beta) | Pr[\alpha \xrightarrow{S} \beta] \ne \frac{1}{2}\}$. Similar to the construction of constraints for S-box in impossible differential cryptanalysis in section 2.2, we can build linear inequalities system to exactly depict LT, with the help of SAGE and Greedy algorithm in [33].

Until now, every operation in a certain reduced-round ARX cipher or cipher with S-box can be exactly described with inequalities in MILP method. The corresponding MILP model for search of ZC approximations is built by combining the whole inequalities system of all operations, and it is similar as the building process for searching of impossible differentials.

3 Algorithm to Verify the Impossible Differentials and ZC Approximations

In section 2, we propose a method to search impossible differentials and ZC approximations by judging whether the model is infeasible or not. However, with this method, we can not directly find out the contradictory place, even we can not judge whether infeasible status is caused by writing wrong code or not. What's more, the contradictions found by this method are often not traditional contradiction between 0 and 1 on a certain bit, but two sets of differences on some bits calculated from input and output differences respectively which have no intersection. So in this section, we propose an algorithm to verify the corretness of impossible differentials and ZC approximations searched by our new tools.

The idea of our new search method for impossible differentials and ZC approximations based on MILP is that the differential propagation on cipher can be exactly described by inequalities system. Specifically, we first set variables on both sides of each operation in the cipher to represent the possible differences, then link them with suitable inequalities system so that the solutions' set of this system is exactly the set of all possible differential patterns. Without loss of generality, assume that there is a R-round impossible differential for the target cipher. Obviously, if we remove some inequalities from its MILP model such that the infeasible model becomes feasible, the contradiction must be happened on the variables existed in those removed inequalities. In our method, we find the contradiction between the $\lceil \frac{R}{2} \rceil$-th and $\lceil \frac{R}{2} \rceil + 1$-th rounds[8], i.e. that consider the contradiction on the input difference of $\lceil \frac{R}{2} \rceil + 1$-th round. Based on such observation, we propose a method to verify the impossible differentials and ZC approximations. Take the verification process of an impossible differential as an example, see algorithm 2. So does the verification process for the ZC approximation.

To search the related-key impossible differential of a target cipher, the process is similar to that under single key setting, except that the key schedule and conditions on master key should be described into the MILP model. However, note that in the phase of finding out sets A and B, the set of linear inequalities for whole key schedule and constraints on master key must be put into two small models for rounds $1 \sim \lceil \frac{R}{2} \rceil$ and rounds $\lceil \frac{R}{2} \rceil + 1 \sim R$ simultaneously.

4 Applications

4.1 Application on HIGHT

HIGHT, introduced by Hong *et al.* at CHES 2006 [14], is a lightweight block cipher approved by Korea Information Security Agency (KISA) and is adopted as an International Standard by ISO/IEC 18033-3 [15]. Its block size and key size are 64 bits and 128 bits respectively. HIGHT employs the Type-II generalized Feistel network consisting of 32 rounds with four parallel Feistel functions in each round. Whitening keys are applied before the first round and after the last round. The round function is shown in Figure 1, where $(X_7^i|X_6^i,\ldots,|X_0^i)$ and $(SK_{4i+3}|SK_{4i+2}|SK_{4i+1}|SK_{4i})$ indicate the 64 bits input and 32 bits subkey of the i-th round respectively.

[8] Actually, we can find out a contradiction between any two adjacent round functions. However, we believe that the contradictions happen between the $\lceil \frac{R}{2} \rceil$-th and $\lceil \frac{R}{2} \rceil + 1$-th rounds are more intuitive and cover less related bits.

Algorithm 2: Verification process of an impossible differential

```
Input: The MILP model of this impossible differential. Assume that the input and output
       differences are $\Delta x_{in}$ and $\Delta x_{out}$ respectively;
Output: The detailed contradiction.
Collect all inequalities linking the $\lceil \frac{R}{2} \rceil$-th and $\lceil \frac{R}{2} \rceil + 1$-th rounds, then put those into a set $\mathbb{I}_{mid}$;
for Each inequality in $\mathbb{I}_{mid}$ do
    Remove the same inequality from the MILP model;
    Solve the new model;
    if The model is infeasible then
        Delete this inequality from set $\mathbb{I}_{mid}$;
        Continue;
    else
        Put this inequality back to MILP model;
Extract all variables corresponding to the input difference of $\lceil \frac{R}{2} \rceil + 1$-th round from remained
    set $\mathbb{I}_{mid}$, and put them into a set $Var_{contradiction}$;
// The contradiction happens on $Var_{contradiction}$.
Output $Var_{contradiction}$;
Set two empty set $A$ and $B$;
for Each possible difference value $\Delta x_{contradiction}$ on $Var_{contradiction}$ do
    if $\Delta x_{in} \rightarrow \Delta x_{contradiction}$ is possible then
        put $\Delta x_{contradiction}$ into set $A$;
    if $\Delta x_{contradiction} \rightarrow \Delta x_{out}$ is possible then
        put $\Delta x_{contradiction}$ into set $B$;
Output sets $A$ and $B$.
```

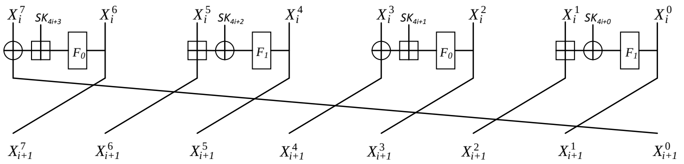

Fig. 1. Round function of HIGHT cipher

Denote exclusive-or, addition modulo 2^{32} and left rotation operations as $\oplus$, $\boxplus$ and $\lll$ respectively through this paper. F_0 and F_1, used in the round function, are defined as follows:

$$F_0(x) = (x \lll 1) \oplus (x \lll 2) \oplus (x \lll 7),$$
$$F_1(x) = (x \lll 3) \oplus (x \lll 4) \oplus (x \lll 6).$$

Since the key schedule is not related to the search of impossible differentials and ZC approximations, we omit it in this paper. For further details, please refer to [14].

17-Round Impossible Differentials of HIGHT. For HIGHT block cipher, the longest impossible differential, firstly proposed by Lu in [20], is 16 rounds. Based on the property that the modular addition $\boxplus$ operation definitely preserves the least significant difference in the original positions, he exploited the miss-in-the-middle manner to find two impossible differentials for 16 rounds HIGHT cipher. In this part, we use the method in section 2 to build a MILP model for 17-round HIGHT cipher. In this model, the differences of all subkeys are set to be zero. Since traversing all input and output differences is impossible due to the time complexity, we only try the cases that the hamming weights of both input and output differences are exactly one, we find four

17-round impossible differentials as follows, where e_i means only the i-th bit is 1 in this byte or word through this paper.

$$(e_7,0,0,0,0,0,0,0) \nrightarrow (0,0,0,0,0,0,0,e_7), \quad (0,0,e_7,0,0,0,0,0) \nrightarrow (0,e_7,0,0,0,0,0,0),$$
$$(0,0,0,0,e_7,0,0,0) \nrightarrow (0,0,0,e_7,0,0,0,0), \quad (0,0,0,0,0,0,e_7,0) \nrightarrow (0,0,0,0,0,e_7,0,0).$$

Taking the first impossible differential above as an example, we verify it by using the algorithm in Section 3. One contradiction is found on the last byte of output of round 9 (input of round 10). The set A of values on this 8-bit contradiction place calculated from fixed input difference includes 255 possible values except $0x80$, and the set B of values on the same 8-bit state calculated from fixed output difference only has the value $0x80$. This means sets A and B have no intersection. In other words, this is actually a 17-round impossible differential.

17-Round ZC Approximations of HIGHT. Until now, for HIGHT block cipher, the longest ZC approximation is 16 rounds presented by Wen *et al.* in [36], which utilized the mask property of addition that the correlation is not zero if and only if two input masks and output mask have the same high non-zero bit position in [7]. They tried to set the non-zero bits of mask on the highest position of each branch of input and output, and found 128 ZC approximations, see Theorem 1 in [36]. In this part, we utilize the MILP model proposed for ZC approximations to search longer ZC approximations for HIGHT cipher. Note that the masks of all subkeys are set as free variables in this model. Because of the time complexity as well, we only try the cases that the hamming weights of both input and output masks are exactly one. As a result, we found four 17-round ZC approximations as follows, where e_0 means 8-bit value 00000001.

$$(0,e_0,0,0,0,0,0,0) \nrightarrow (e_0,0,0,0,0,0,0,0), \quad (0,0,0,e_0,0,0,0,0) \nrightarrow (0,0,e_0,0,0,0,0,0),$$
$$(0,0,0,0,0,e_0,0,0) \nrightarrow (0,0,0,0,e_0,0,0,0), \quad (0,0,0,0,0,0,0,e_0) \nrightarrow (0,0,0,0,0,0,e_0,0).$$

Taking the second ZC approximation as an example, one contradiction is found on the first byte of output of round 9 (input of round 10). Set A of masks on this 8-bit contradiction place calculated from the fixed input mask involves 255 values except $0x01$, and set B of masks on this 8-bit state calculated from the fixed output mask only has one value $0x01$. This means such approximation is a valid ZC approximation.

4.2 Application on SHACAL-2

SHACAL-2 [11], introduced by Handschuch and Naccache, was selected as one of the four block ciphers by NESSIE. Its block size is 256 bits and key size is various from 128 bits to 512 bits. SHACAL-2 has totally 64 rounds and its round function is based on the compression function of the hash function SHA-2, which is shown in Figure 2. The input of round $i = 0,1,\ldots,63$ is divided into 8 words $A^i \| B^i \| C^i \| D^i \| E^i \| F^i \| G^i \| H^i$. W^i and K^i are 32-bit subkey and constant respectively.

The functions used in round function is defined as follows, where $S_i(X)$ denotes the right rotation of 32-bit word X by i-bit position, $\neg X$ means the complement of 32-bit word X.

$$\begin{aligned} Ch(X,Y,Z) &= (X\&Y) \oplus (\neg X\&Z) \\ Maj(X,Y,Z) &= (X\&Y) \oplus (X\&Z) \oplus (Y\&Z) \\ \textstyle\sum_0 &= S_2(X) \oplus S_{13}(X) \oplus S_{22}(X) \\ \textstyle\sum_1 &= S_6(X) \oplus S_{11}(X) \oplus S_{25}(X) \end{aligned}$$

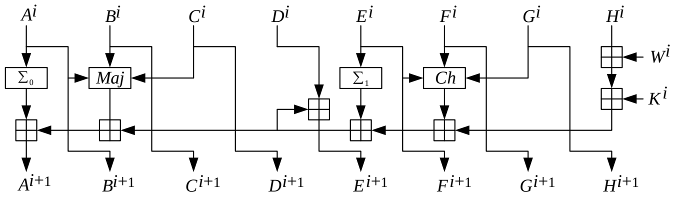

Fig. 2. Round function of SHACAL-2

Since we do not use the key schedule in this paper, we omit it here.

15-Round Impossible Differentials of SHACAL-2. For SHACAL-2 block cipher, the longest impossible differential so far was proposed by Hong *et al.* in [12] and totally had 14 rounds. In [12], Hong *et al.* firstly proposed a 9-round impossible differential from round 2 to 10 as follows, where Δh_0^{11}, i.e. the lsb of ΔH^{11}, is 1.

$$(0, 0, 0, e_{31}, 0, 0, 0, e_{31}) \nrightarrow (?, ?, ?, ?, ?, ?, ?, \Delta H^{11}).$$

By adding two rounds before this 9-round impossible differential, Hong *et al.* constructed two types of 11-round impossible differential which were determined by the msb of the 0^{th} round key W^0. That is, if the value of the key bit is 0, one of the two 11-round impossible differentials holds with probability 1, otherwise, the other one holds with probability 1. Consequently, by representing h_0^{11} as a nonlinear equation of some bits in $A^{14}, B^{14}, \ldots, H^{14}, K^{11}, K^{12}, K^{13}, W^{11}, W^{12}$ and W^{13}, Hong *et al.* continuely combined a nonlinear equation of 3 rounds to those 11-round impossible differentials. In the end, they had two types of 14-round impossible differentials for rounds $0 \sim 13$ with respect to the msb of W^0. By utilizing these two 14-round trails, they attacked 30-round SHACAL-2 with 512-bit key.

In this part, we use our automatic search tool to find out longer impossible differential for SHACAL-2. The functions Ch, Maj, $\sum_0$ and $\sum_1$ used in round function are all not mentioned in section 2. But they are easy to be described, since the first two functions can be regarded as S-box operation and the last two ones are similar to single XOR operation. We describe them one by one as follows.

Constraints for Ch function. It is easy to find that there are 14 out of 16 possible differential patterns for each bit of $Ch(X, Y, Z)$ function except the patterns $(0, 0, 0) \xrightarrow{Ch} 1$ and $(0, 1, 1) \xrightarrow{Ch} 0$. So we use the following constraints to exactly describe all differential patterns on each bit of this function, where $i = 0, 1 \ldots, 31$.

$$\begin{aligned} X[i] + Y[i] + Z[i] - Ch[i] &\geq 0, \\ X[i] - Y[i] - Z[i] + Ch[i] &\geq -1. \end{aligned}$$

Constraints for Maj function. By analyzing the differential property of $Maj(X, Y, Z)$ function, we find that only 14 out of 16 differential patterns are possible for each bit of this function. Patterns $(0, 0, 0) \xrightarrow{Maj} 1$ and $(1, 1, 1) \xrightarrow{Maj} 0$ never happen. With the help of software SAGE, we use the following constraints to just describe all possible differential patterns on each bit.

$$\begin{aligned} X[i] + Y[i] + Z[i] - Maj[i] &\geq 0, \\ Maj[i] - X[i] - Y[i] - Z[i] &\geq -2. \end{aligned}$$

Constraints for $\sum_0$ and $\sum_1$ functions. Take $\sum_0 = S_2(X) \oplus S_{13}(X) \oplus S_{22}(X)$ as an example, so does that for $\sum_1$. Firstly we denote $S_2(X)$, $S_{13}(X)$ and $S_{22}(X)$ by Y, Z, W, as a result, $\sum_0 = Y \oplus Z \oplus W$. Then for each bit of $\sum_0(Y, Z, W)$, we can use the equation as follows to limit it, where d_1 and d_2 are free 32-bit variables.

$$Y[i] + Z[i] + W[i] + \textstyle\sum_0[i] - 2d_1[i] - 2d_2[i] = 0.$$

Until now, we can build the MILP model for SHACAL-2. In practice, we only search the cases that the nonzero bits only happen on the msb of each word of input and output differences. Totally there are $2^8 \times 2^8 = 2^{16}$ cases. In the end, we find out eight 13-round impossible differentials of SHACAL-2 as follows.

$$\begin{aligned}
(0,0,0,e_{31},0,0,0,e_{31}) &\nrightarrow (0,0,0,0,e_{31},0,0,0),\\
(0,0,0,e_{31},0,0,0,e_{31}) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0),\\
(0,0,e_{31},0,0,0,0,0) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0),\\
(0,0,e_{31},e_{31},0,0,0,e_{31}) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0),\\
(0,e_{31},0,0,0,0,0,0) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0),\\
(0,e_{31},0,e_{31},0,0,0,e_{31}) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0),\\
(0,e_{31},e_{31},0,0,0,0,0) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0),\\
(0,e_{31},e_{31},e_{31},0,0,0,e_{31}) &\nrightarrow (e_{31},0,0,0,e_{31},0,0,0).
\end{aligned}$$

Take the first 13-round impossibe differential as an example, we check its correctness. The difference on the least significant seven bits of F^7 calculated from the input difference is 1000000, but the difference on the same place calculated from output difference is 0000000. This is a contradiction.

Since the first two 13-round impossible differentials have the same input difference with the 9-round impossible differential proposed by Hong *et al.*, we can add two rounds before them and obtain two types of 15-round impossible differentials for each one. These are the longest impossible differentials of SHACAL-2 so far.

4.3 Application on LEA

LEA is a block cipher proposed by Hong *et al.* in [13], which can provide a high-speed software encryption on general-purpose processors. It has 128-bit block size and 128, 192, or 256-bit key size. The number of rounds is 24, 28 and 32 respectively according to key size. The round function is shown in Figure 3, where $X_i[0]\|X_i[1]\|X_i[2]\|X_i[3]$ and $RK_i[0]\|RK_i[1]\|RK_i[2]\|RK_i[3]\|RK_i[4]\|RK_i[4]$, $i = 0, 1, \ldots, 31$ denote 4-word input and 6-word subkey of the i-th round respectively, while ROL_i and ROR_j means the left rotation of 32-bit value by i-bit and right rotation of 32-bit value by j-bit respectively. In [13], the designers claimed that LEA cipher's impossible differntial and ZC approximation were 10 rounds and 7 rounds respectively.

In this part, we combine our basic analysis on the construction of LEA with the automatic search tool in section 2 to search its impossible differentials and ZC approximations again. As a result, our longest impossible differential is 10 rounds as well. But we find out 3 ZC approximations for 10-round LEA, which are three more rounds than previous ones, They are shown as follows, where $e_{i,j}$ means only the i-th bit and the j-th bit are 1 in this word. With these trails, the security bound (11 rounds) for ZCLC claimed in [13] is really not enough.

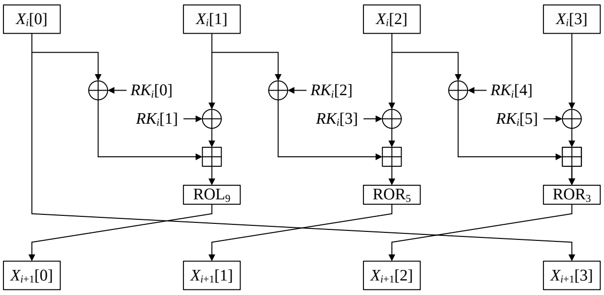

Fig. 3. Round function of LEA

$$\begin{aligned}(0, e_0, 0, 0) &\nrightarrow (e_9, e_{27}, e_{29}, e_{22,0}),\\ (e_0, 0, 0, 0) &\nrightarrow (e_9, e_{27}, e_{29}, e_{22,0}),\\ (e_0, e_0, 0, 0) &\nrightarrow (e_9, e_{27}, e_{29}, e_{22,0}).\end{aligned}$$

Taking the first ZC approximation as an example, the contradiction happens on the sixth most significant bit of $X_4[2]$.

4.4 Application on LBlock

LBlock, proposed by Wu and Zhang at ACNS in [41], is a lightweight block cipher. On account of its excellent hardware performance, software performance and security, it is widely focused on by cryptanalysts in the field of symmetric ciphers. Its block size and key size are 64 bits and 80 bits respectively. LBlock cipher adopts a 32-round modified Feistel network which adds an extra left rotation operation on one branch of general Feistel network. The round function is shown in Figure 4, where (X_1^i, X_0^i) and sk_i denote 64 bits input and 32 bits subkey of the i-th round respectively.

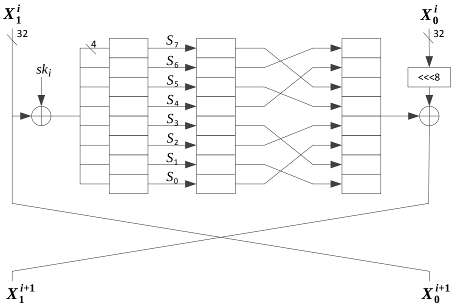

Fig. 4. Round function of LBlock cipher

In the round function, there are an XOR operation with subkey, a nonlinear layer and a simple permutation that the second component involves 8 parallel different S-boxes $S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7$ and the last component only changes the byte order of its input. It is worth noting that an 8-bit left rotation operation happens on the right branch in Figure 4.

The master key of LBlock cipher is 80 bits, denoted by $K = k_{79}, k_{78}, \ldots, k_0$. All subkeys $sk_i, i = 0, 1, \ldots, 31$ are produced by utilizing an 80-bit register. The process is illustrated in Algorithm 3.

Algorithm 3: Key schedule of LBlock cipher

```
$sk_0 = K_{79\sim48}$;
for $1 \leq r \leq 31$ do
    $k_{79\sim0} \leftarrow k_{79\sim0} \lll 29$;
    $k_{79\sim76} \leftarrow S_9(k_{79\sim76})$; $k_{75\sim72} \leftarrow S_8(k_{75\sim72})$; $k_{50\sim47} \leftarrow k_{50\sim47} \oplus [i]_2$;
    $sk_r \leftarrow k_{79\sim48}$.
```

In Algorithm 3, $k_{a\sim b}$ denotes all key bits from k_a to k_b, S_8 and S_9 are two different 4×4 S-boxes. For more details about LBlock, please refer to [41].

16-Round Related-Key Impossible Differentials of LBlock. Differential cryptanalysis and impossible differential cryptanalysis are both implemented under the single-key setting, i.e., all plaintexts are encrypted by one master key. In [1] and [16], related-key differential and related-key impossible differential cryptanalysis are proposed respectively, which exploited the relation of two master keys to recover the secret keys.

For LBlock cipher, The previous best related-key impossible differentials is found by Wen *et al.* in [37]. They designed a specialized algorithm to search such trails with some observations on key schedule and structure of the cipher. Finally they totally found two 16-round related-key impossible differentials. However, Wen *et al.*'s two impossible differentials are right only under part of master key pairs under each of the given two key differences.

In our work, we use the tool in Section 2 to build a MILP model for LBlock cipher including the key schedule and search the related-key impossible differentials. According to the key schedule, LBlock is a bit-level cipher. We only search the cases that the difference of two master keys has only one nonzero bit (80 cases) and the input and output differences both have no more than one nonzero bit ($65 \times 65 = 4225$ cases), so in total we search 338000 cases. In the end, we search out eighteen 16-round related-key impossible differentials shown in Table 2.

Table 2. 16-round related-key impossible differentials for LBlock

Δin	Δout	Key difference (ΔK)
0	0 or nonzero bit $\in \{20, 21, 22, 23\}$	only $k_{11} = 1$
0	0 or nonzero bit $\in \{8, 9, 10, 11\}$	only $k_{10} = 1$
0	0	only $k_6 = 1$
0	0 or nonzero bit $\in \{16, 17, 18, 19\}$	only $k_2 = 1$
0	0	only $k_1 = 1$
0	0	only $k_0 = 1$

[1] Δin and Δout are input and output differences respectively.

[2] 0 denotes zero difference;

[3] Nonzero bit $\in \{n_1, n_2, n_3, n_4\}$ means the difference can be one of four cases: $0x8000000000000000 \ggg n_i, i = 1, 2, 3, 4$;

[4] "only $k_i = 1$" denotes only the i-th bit of master key is 1, other bits are all 0.

Taking the related-key impossible differential with $k_0 = 1$ as example, one contradiction is found on the sixth most significant nibble of output of round 8 (input of round 9). Set A of differences on this 4-bit contradiction place calculated from the fixed input difference involves only one value 0000, and set B on the same 4-bit state calculated from the fixed output difference only has 11 values: 1000, 0100, 1100, 1010, 0110, 1110, 0001, 0101, 0011, 1011, 1111. Sets A and B have no intersection, which means such differential is a valid related-key impossible one.

5 Conclusion

In this paper, we propose a new automatic search tool for impossible differentials and ZC approximations based on MILP method. In this tool, the differential and linear properties of non-linear components are firstly taken into consideration, so we can not only find the previous impossible differentials and ZC approximations, but also may find longer ones for a target cipher. As applications, we apply this tool on search of impossible differential or ZC approximations for HIGHT, SHACAL-2, LEA, LBlock. As a result, we find the longest such types of trails for each cipher so far. Actually, this tool is useful in IDC and ZCLC for most ARX ciphers and lightweight block ciphers. Additionally, it can be used in evaluating the security of stream cipher and hash functions as well. However, there are still two problems in this tool to be solved in the future. Firstly, the search for cipher with 8-bit S-box is slow because of lots of linear inequalities to describe such S-box. Secondly, searching all case of target rounds of a cipher is difficult due to the time complexity, how to shrink searching scope to find the longest trail in suitable time is another meanful problem, especially under related-key setting. In the future, we will focus on these problems and improve this tool.

References

1. E Biham. New types of cryptanalytic attacks using related keys. Journal of Cryptology, 1991, 7(4): 229-246
2. E Biham, A Biryukov, A Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Workshop on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 1999. LNCS, vol. 1592: 12-23
3. E Biham, A Shamir. Differential cryptanalysis of the data encryption standard. Springer 1993, ISBN 978-1-4613-9316-0
4. E Biham. On Matsui's linear cryptanalysis. In: Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, 1994. LNCS, vol. 950: 341-355
5. C Blondeau. Impossible differential attack on 13-round Camellia-192. Information Processing Letters, 2015, 115(9): 660-666
6. A Bogdanov, V Rijmen. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Designs, codes and cryptography, 2014, 70(3): 369-383
7. A Bogdanov, M Wang. Zero correlation linear cryptanalysis with reduced data complexity. In: International Workshop on Fast Software Encryption, Washington, USA, 2012. LNCS, vol. 7549: 29-48
8. C Boura, M Naya-Plasencia, V Suder. Scrutinizing and improving impossible differential attacks: applications to CLEFIA, Camellia, LBlock and Simon. In: International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, 2014. LNCS, vol. 8873: 179-199
9. J Chen, M Wang, B Preneel. Impossible Differential Cryptanalysis of Lightweight Block Ciphers TEA, XTEA and HIGHT. In: International Conference on Cryptology in Africa, Ifrance, Morocco, 2012. LNCS, vol. 7374: 117-137
10. K Fu, M Wang, Y Guo, et al. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In: International Workshop on Fast Software Encryption, Bochum, Germany, 2016. LNCS, vol. 9783: 268-288

11. H Handschuh, D Naccache. SHACAL: A Family of Block Ciphers. Submission to the NESSIE project, 2002
12. S Hong, J Kim, G Kim, et al. Impossible Differential Attack on 30-Round SHACAL-2. In: International Conference on Cryptology in India, New Delhi, India, 2003. LNCS, vol. 2904: 97-106
13. D Hong, J -K Lee, D -C Kim, et al. LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors. In: International Workshop on Information Security Applications, Jeju Island, Korea, 2013. LNCS, vol. 8267:3-27
14. D Hong, J Sung, S Hong, et al. HIGHT: A new block cipher suitable for low-resource device. In: International Workshop on Cryptographic Hardware and Embedded Systems, Yokohama, Japan, 2006. LNCS, vol. 4249: 46-59
15. ISO/IEC 18033-3, Information technology-Security techniques-Encryption algorithms-Part 3: Block ciphers, 2010
16. G Jakimoski, Y Desmedt. Related-key differential cryptanalysis of 192-bit key AES variants. In: International Workshop on Selected Areas in Cryptography, Ottawa, Canada, 2003. LNCS, vol. 3006: 208-221
17. J Kim, S Hong, J Lim. Impossible differential cryptanalysis using matrix method. Discrete Mathematics, 2010, 310(5): 988-1002
18. L Knudsen. DEAL-a 128-bit block cipher. NIST AES Proposal, 1998
19. H Lipmaa, S Moriai. Efficient algorithms for computing differential properties of addition. In: International Workshop on Fast Software Encryption, Yokohama, Japan, 2001. LNCS, vol. 2355: 336-350
20. J Lu. Cryptanalysis of reduced versions of the HIGHT block cipher from CHES 2006. In: International Conference on Information Security and Cryptology, Seoul, Korea, 2007. LNCS, vol. 4817: 11-26
21. Y Luo, X Lai, Z Wu et al. A unified method for finding impossible differentials of block cipher structures. Information Sciences, 2014, vol. 263: 211-220
22. H Mala, M Dakhilalian, V Rijmen et al. Improved impossible differential cryptanalysis of 7-round AES-128. In: International Conference on Cryptology in India, Hyderabad, India, 2010. LNCS, vol. 6498: 282-291
23. M Matsui. Linear cryptanalysis method for DES cipher. In: Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, 1993. LNCS, vol. 765: 386-397
24. N Mouha, Q Wang, D Gu et al. Differential and linear cryptanalysis using mixed-integer linear programming. In: International Conference on Information Security and Cryptology, Beijing, China, 2011. LNCS, vol. 7537: 57-76
25. K Nyberg, J Wallén. Improved linear distinguishers for SNOW 2.0. In: International Workshop on Fast Software Encryption, Graz, Austria, 2006. LNCS, vol. 4047: 144-162
26. O Özen, K Varici, C Tezcan, et al. Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In: Australasian Conference on Information Security and Privacy, Brisbane, Australia, 2009. LNCS, vol. 5594: 90-107
27. A. Abdelkhalek, Y. Sasaki, Y. Todo, et al. MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics. IACR Transactions on Symmetric Cryptology, 2017(4): 99-129
28. Y Sasaki, Y Todo. New Impossible Differential Search Tool from Design and Cryptanalysis Aspects. In: International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 2017. LNCS, vol. 10212: 185-215
29. S Sun, D Gerault, P Lafourcade, et al. Analysis of AES, SKINNY, and Others with Constraint Programming. IACR Transactions on Symmetric Cryptology, 2017(1): 281-306
30. B Sun, Z Liu, V Rijmen et al. Links among impossible differential, integral and zero correlation linear cryptanalysis. In: Annual Cryptology Conference, Santa Barbara, USA, 2015. LNCS, vol. 9215: 95-115
31. S Sun, L Hu, P Wang, et al. Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers. In: International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, 2014. LNCS, vol. 8873: 158-178
32. S Sun, L Hu, L Song, et al. Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks. In: International Conference on Information Security and Cryptology, Guangzhou, China, 2013. LNCS, vol. 8567: 39-51
33. S Sun, L Hu, M Wang, et al. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Report 2014/747
34. J Wallén. Linear approximations of addition modulo 2^n. In: International Workshop on Fast Software Encryption, Lund, Sweden, 2003. LNCS, vol. 2887: 261-273

35. L Wen, M Wang. Integral Zero-Correlation Distinguisher for ARX Block Cipher, with Application to SHACAL-2. In: Australasian Conference on Information Security and Privacy, Wollongong, Australia, 2014. LNCS, vol. 8544: 454-461
36. L Wen, M Wang, A Bogdanov, et al. Multidimensional zero-correlation attacks on lightweight block cipher HIGHT: Improved cryptanalysis of an ISO standard. Information Processing Letters, 2014, 114(6): 322-330
37. L Wen, M Wang, J Zhao. Related-Key Impossible Differential Attack on Reduced-Round LBlock. J. Comput. Sci. Technol., 2014, 29(1): 165-176
38. M Minier, M Naya-Plasencia. A related key impossible differential attack against 22 rounds of the lightweight block cipher LBlock. Information Processing Letters, 2012, 112(16): 624-629
39. S Wu, M Wang. Security Evaluation against Differential Cryptanalysis for Block Cipher Structures. IACR Cryptology ePrint Archive, Report 2011/551
40. S Wu, M Wang. Automatic search of truncated impossible differentials for word-oriented block ciphers. In: International Conference on Cryptology in India, Kolkata, India, 2012. LNCS, vol. 7668: 283-302
41. W Wu, L Zhang. LBlock: A lightweight block cipher. In: International Conference on Applied Cryptography and Network Security, Nerja, Spain, 2011. LNCS, vol. 6715: 327-344