Mind the Propagation of States

New Automatic Search Tool for Impossible Differentials and Impossible Polytopic Transitions

Xichao Hu[1,2], Yongqiang Li[1,2(✉)], Lin Jiao[3], Shizhu Tian[1,2], and Mingsheng Wang[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing China
yongq.lee@gmail.com,{huxichao,tianshizhu,wangmingsheng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] State Key Laboratory of Cryptology, Beijing, China jiaolin_jl@126.com

Abstract. Impossible differentials cryptanalysis and impossible polytopic cryptanalysis are the most effective approaches to estimate the security of block ciphers. However, the previous automatic search methods of their distinguishers, impossible differentials and impossible polytopic transitions, neither consider the impact of key schedule in the single-key setting and the differential property of large S-boxes, nor apply to the block ciphers with variable rotations.

Thus, unlike previous methods which focus on the propagation of the difference or s-difference, we redefine the impossible differentials and impossible $(s+1)$-polytopic transitions according to the propagation of state, which allow us to break through those limitations of the previous methods. Theoretically, we prove that traditional impossible differentials and impossible $(s+1)$-polytopic transitions are equivalent to part of our redefinitions, which have advantages from broader view. Technically, we renew the automatic search model and design an SAT-based tool to evaluate our redefined impossible differentials and impossible $(s+1)$-polytopic transitions efficiently.

As a result, for GIFT64, we get the 6-round impossible differentials which cannot be detected by all previous tools. For PRINTcipher, we propose the first modeling method for the key-dependent permutation and key-dependent S-box. For MISTY1, we derive 902 4-round impossible differentials by exploiting the differential property of S-boxes. For RC5, we present the first modeling method for the variable rotation and get 2.5-round impossible differentials for each version of it. More remarkable, our tool can be used to evaluate the security of given cipher against the impossible differentials, and we prove that there exists no 5-round 1 input active word and 1 output active word impossible differentials for AES-128 even consider the relations of 3-round keys. Besides, we also get the impossible $(s+1)$-polytopic transitions for PRINTcipher, GIFT64, PRESENT, and RC5, all of which can cover more rounds than their corresponding impossible differentials as far as we know.

Keywords: Impossible Differentials · Impossible Polytopic Transitions · $(s+1)$-ploygon · SAT.

1 Introduction

Impossible differential cryptanalysis was proposed by Biham et al. and Knudsen respectively, where Biham et al. used it to analyze the security of Skipjack [4], and Knudsen utilized it to analyze the security of DEAL [14]. Up to now, impossible differential cryptanalysis has been applied to lots of block ciphers, such as AES [18], SIMON [8], XTEA [9], and so on. There is no doubt that it is one of the most effective cryptanalytic approaches to evaluate the security of block ciphers.

In the impossible differential cryptanalysis, attackers derive the right keys by discarding the wrong keys that lead to the impossible differentials inherent to the given cipher. Thus how to find an impossible differential as longer as possible is the most essential and critical problem in regard to this kind of attacks.

Impossible $(s+1)$-polytopic cryptanalysis was proposed by Tiessen [29], which is a generalization of impossible differential cryptanalysis. Unlike the impossible differentials are constructed by considering the interdependencies of the differences of two plaintexts and the accordingly two ciphertexts, the distinguishers of impossible $(s+1)$-polytopic cryptanalysis, named impossible $(s+1)$-polytopic transitions, are constructed by considering the interdependencies between the s-differences of $(s+1)$ plaintexts and $(s+1)$ ciphertexts [4].

In the last 20 years, using automatic tools to search the distinguishers becomes a new trend. The first automatic tool for the impossible differentials is presented by Kim et al. [13], named $\mathcal{U}$-method. Then, Luo et al. [17] extended it as UID-method. After that, Wu and Wang [31] introduced another method using the idea of solving equations, called $\mathcal{WW}$-method. However, those tools to search impossible differentials cannot describe the details of S-boxes, which waste plenty of differential property of the propagation.

This problem is settled with the application of the Mixed Integer Linear Programming (MILP) method to symmetric cryptography. The MILP problem is a mathematical optimization problem that finds the minimum or maximum value of some objective function under the conditions of linear equations and inequalities of integer variables. Mouha et al. [22] first introduced it to symmetric cryptography to find the lower bound on the number of active S-boxes for both differential and linear cryptanalysis. Later, Sun et al. [28] proposed the modelling method to depict the valid differential propagation of small S-boxes (typically 4 bits), and Fu et al. [12] presented the modelling method to depict all the valid differential/linear characteristics propagations of modular addition. Thus, the differential propagation of any round for the small S-boxes based block ciphers and ARX block ciphers can be modeled by a set of linear inequalities accurately.

[4] **Convention.** In our paper, the impossible $(s+1)$-polytopic transition is uniformly defined for $(s \geq 2)$, excluding the case of the impossible differential, since it has been studied in-depth separately.

On that basis, Cui et al. [10] proposed a MILP-based tool to search the impossible differentials for lightweight block ciphers, and an algorithm to verify the impossible differentials. Soon after, Sasaki and Todo [27] presented a MILP-based tool to search the impossible differential for SPN block ciphers. In particular, they proposed the best search method at present for large S-boxes based block ciphers, named the *arbitrary S-box* mode, which only treats the large S-boxes as permutations in order to make their tool valid to detect the contradiction in linear components.

However, the previous automatic search tools for impossible differentials have the following limitations in general.

- Previous tools cannot take into account the key schedule in the single-key setting.
- Previous tools cannot consider the differential property of large S-boxes.
- Previous tools cannot be applied to the block ciphers with variable rotation.

As to impossible polytopic transitions, there was only a search method proposed for DES and AES in the original paper [29]. However, due to the limitation that the searching spaces increase rapidly with the number of rounds, this method can only be confined to a small number of rounds. Besides, this tool cannot take into account the key schedule in the single-key setting and be applied to the block ciphers with variable rotations either.

Our Contributions. In this paper, we define a series of new notations, s-polygon to describe a tuple with s states, s-polygonal trail to depict the propagation of s-polygon, possible s-polygons and impossible s-polygons to depict the relations between two s-polygons.

Then, unlike the traditional impossible differentials and impossible $(s+1)$-polytopic transitions that are constituted according to the propagation of difference and s-difference, we redefine the impossible differentials and impossible $(s+1)$-polytopic transitions based on the propagation of the s-polygon[5]. Thus, the key schedule in the single-key setting can be considered in the construction of redefined impossible differentials and impossible $(s+1)$-polytopic transitions. We define the i-impossible differential (resp. i-impossible $(s+1)$-polytopic transition) to represent the redefined impossible differential (resp. impossible $(s+1)$-polytopic transition) which is constituted in the round key independent setting and d-impossible differential (resp. d-impossible $(s+1)$-polytopic transition) to

[5] This idea can be traced back to [21]. In [21], Mironov et al. used the idea of the transition of states to search two states that satisfy a fixed differential path, which is the critical step to find a collision of the hash function. Recently, two papers [16,26] that also used the idea of the transition of states appeared in the ePrint. As we understand, [16] applied the transition of two states to the non-linear layer. [26] utilized the idea to determine whether a given differential path of ARX based block ciphers is compatible or not. In our paper, we exploit the idea of the transition of multi-states to search the impossible differential and the impossible $(s+1)$-polytopic transition for block ciphers.

represent the redefined impossible differential (resp. impossible $(s+1)$-polytopic transition) which is constituted by considering the key schedule.

Next, we study the relation between our redefined impossible differential (resp. impossible $(s+1)$-polytopic transition) and traditional impossible differential (resp. impossible $(s+1)$-polytopic transition). We show that the i-impossible differential (resp. i-impossible $(s+1)$-polytopic transition) is equivalent to traditional impossible differential (resp. impossible $(s+1)$-polytopic transition) which is constructed by taking into account the inside property of S-boxes for the block ciphers with SPN or Feistel structures and the block cipher MISTY1.

Finally, we model the propagations of states by the statements in the CVC format of STP[6] (a solver of the SAT problem) for each operation, and design an SAT-based unified automatic tool for searching the redefined impossible differential and impossible $(s+1)$-polytopic transition. Since traditional impossible differential is equivalent to the i-impossible differential and traditional impossible $(s+1)$-polytopic transition is equivalent to the i-impossible $(s+1)$-polytopic transition, our tool can be used to search the traditional impossible differential and traditional impossible $(s+1)$-polytopic transition. Furthermore, our tool has the following advantages.

- **Able to search the distinguishers by considering the impact of key schedule in the single-key setting.** Our automatic search tool focuses on the propagations of states, which are impacted by the value of key. By adding the constraints of key variables according to the key schedule, it can be used to search the impossible differentials and impossible $(s+1)$-polytopic transitions in the single-key setting confirming the key schedule. As far as we know, this is the first automatic search tool that considers the impact of key schedule in the single-key setting for impossible differentials and impossible $(s+1)$-polytopic transitions.
- **Able to search the distinguishers for the block ciphers with variable rotation.** In this paper, by exploiting the conditional term of the CVC format, we propose a novel method to model the propagations of states for variable rotation. This method allows us to search the impossible differentials and impossible $(s+1)$-polytopic transitions for block ciphers with variable rotation automatically. As far as we know, this is the first automatic search method for such type of block ciphers.
- **Able to search impossible differentials for block ciphers with large S-boxes by considering the differential property of large S-boxes.** We make use of the conditional terms to model the propagations of states for large S-boxes. This way allows us to search the impossible differentials for the block ciphers with large S-boxes by considering the differential property of large S-boxes. As far as we know, this is the first automatic tool to search the impossible differentials for such ciphers taking account in the differential property of large S-boxes.
- **New proving tool for resisting impossible differentials in aspect of cipher design.** Our tool not only can be used to evaluate the security of

[6] http://stp.github.io/

block ciphers against traditional impossible differentials for block ciphers with large S-box in the case of considering the differential property of large S-boxes, but also can be used to evaluate the security of block ciphers (includes block ciphers with key-dependent permutation) against the impossible differentials in the case of considering the key schedule in the single-key setting. It is very favorable in aspect of block ciphers design and assessment.

We apply our tool to various block ciphers, these results can be divided into three aspects[7].

Deriving new impossible differentials.

- For GIFT64 [2], we get the 6-round impossible differentials, which cannot be detected by Sun et al.'s method or Sasaki et al.'s method. This result shows that, our tool can detect more contradictions than the previous methods.
- For PRINTcipher48/96 [15], we can not only give the first modeling method for the key-dependent permutation, but also give the first direct modeling method for the key-dependent S-box, which is consisted of the key-dependent permutation and the fixed S-box. Take either of the two modeling methods, by considering all the details of the key schedule, we found 730 4-round impossible differentials for PRINTcipher48 and 234 5-round impossible differentials for PRINTcipher96.
- For MISTY1 [20], we found 902 4-round i-impossible differentials by exploiting the differential property of S-boxes, while only 28 4-round i-impossible differentials were got by implementing the *arbitrary S-box* mode of Sasaki et al.'s method.
- For RC5-32/64/128 [24], we propose the first modeling method for variable rotation, which allows us to get the 2.5-round impossible differentials for them in the key independent setting.

Evaluating the resistance against the impossible differentials. Besides applying our tool directly, we also propose three phases technique and inside value technique to speed up our proving process.

- For GIFT64, PRESENT [6], Midori64 [1], PRINTcipher48, and PRINTcipher96, we prove that, in the search space where the input difference only actives one S-box in the first substitution and the output difference only

[7] **Illustrantion.** Note that, when to search the r-round distinguishers by considering the key schedule in our model, different beginning round lead to different final models, since the round constants are different from each round. To a common format, we place the distinguishers of our model in the 1st round by default (except GIFT64, since the round key is not XORed with plaintext in the first round, we place the distinguishers in the 2nd). That is, when we say a distinguisher is an r-round distinguisher, it is an r-round distinguisher placed from 1st round to the r-th round. Similarly, when we say there exists no r-round impossible differentials in the search space, it means that for all the input differences and output differences where the input differences placed at the 1st round and the output differences placed at the r-th round, the differences cannot be connected. Actually, in other cases that the distinguishers do not begin with the 1st round, the distinguisher can be searched similarly.

actives one S-box in the last substitution, there exists no 7-round, 7-round, 6-round, 5-round, and 6-round impossible differentials for GIFT64, PRESENT, Midori64, PRINTcipher48, and PRINTcipher96 even taking account in the details of the key schedule.
- For AES [11], by adopting the new proposed three phases technique, we prove that even considering the relations of middle three-round keys, there still exists no 5-round 1 input active word and 1 output active word impossible differentials.
- For 5-round MISTY1 [20] with the FL layers placed at the even rounds, by adopting the three phases technique and inside value technique, we prove that there exists no 1 input active bit and 1 output active bit impossible differentials.

Resulting in new impossible $(s+1)$-polytopic transition $(s \geq 2)$. Besides applying our tool directly, we further propose the step by step strategy to speed up the search.

- For PRINTcipher, by considering all the details of the key schedule, we obtain the 6-round d-impossible 3-polytopic transition and 7-round d-impossible 4-polytopic transition for PRINTcipher48, and 7-round d-impossible 3-polytopic transition and 8-round d-impossible 4-polytopic transition for PRINTcipher96. Moreover, we investigate the impact of the restraints of the **xor** keys (i.e. the keys which are xored with the state) and **control** keys (i.e. the keys which are used to control the key-dependent permutation). The result shows that, both the restraints of the xor keys and control keys will lead to more contradictions.
- For GIFT64, we get a 7-round d-impossible 3-polytopic transition.
- For RC5-32, we get 108 3-round i-impossible 3-polytopic transitions. Similarly, we get a 3-round i-impossible 3-polytopic transition for RC5-64.
- For PRESENT, we get a 7-round i-impossible 4-polytopic transition.

Outline. We introduce the notations and related work in Section 2. Our redefined impossible differentials and impossible $(s+1)$-polytopic transitions and the relations between our redefinitions and traditional definitions are shown in Section 3. The SAT modeling methods and our search algorithm are detailed in Section 4. We apply our method to impossible differentials from the cryptanalysis aspect and design aspect in Section 5 and Section 6, respectively. In Section 7, we apply our method to impossible polytopic transitions. In Section 8, we conclude this paper.

2 Preliminaries

2.1 Notations

The following notations are used in this paper.

- $\boldsymbol{x^{m,s}}$: the tuple $(x_0, \ldots, x_{s-1})$, where $x_i \in \mathbb{F}_2^m$ $(0 \leq i \leq s-1)$.

- $\boldsymbol{x_i^{m,s}}$: the tuple $(x_{i,0}, \ldots, x_{i,s-1})$, where $x_{i,j} \in \mathbb{F}_2^m$ $(0 \leq j \leq s-1)$.
- $\boldsymbol{x^{m,s}}||\boldsymbol{y^{m,s}}$: the tuple $(x_0||y_0, \ldots, x_{s-1}||y_{s-1})$, where $x_i, y_i \in \mathbb{F}_2^m$ $(0 \leq i \leq s-1)$.
- $\boldsymbol{x^{m,s+1}} \rhd \boldsymbol{\alpha^{m,s}}$: the tuple $\boldsymbol{x^{m,s+1}}$ satisfy $x_0 \oplus x_{j+1} = \alpha_j$ $(0 \leq j \leq s-1)$.
- 0^p1^q: the concatenation of p successive 0s and q successive 1s.
- a^pb^q: the concatenation of p-bit constant a and q-bit constant b.
- $W(a)$: the hamming weight of a, i.e., the 1's number in the bit representation of a.
- e_I^n: an n bits value, whose i-th bit is 1 for $i \in I$, and 0 otherwise.
- $BC(n, m, l)$: the set of all iterated block ciphers whose block size is n-bit, master key size is m-bit, and round key size is l-bit.
- $E_k^r(x)$: the output of encryption $E \in BC(n, m, l)$ on the state $x \in \mathbb{F}_2^n$ after r-round under $k \in (\mathbb{F}_2^l)^r$.
- $E_k^r(\boldsymbol{x^{n,s}})$: the tuple $(E_k^r(x_0), \ldots, E_k^r(x_{s-1}))$.
- IKS_r^l: the set $\{(k_1, \ldots, k_r) | k_i \in \mathbb{F}_2^l, 1 \leq i \leq r\}$.
- $DKS_r^{m,l}$: the set $\{(k_1, \ldots, k_r) | k \in \mathbb{F}_2^m, k_i \in \mathbb{F}_2^l, k_i = G_i(k), 1 \leq i \leq r\}$, where G_i denotes the key schedule to generate the round key k_i from the master key k for a block cipher $E \in BC(n, m, l)$.

2.2 A Brief Introduction of Impossible Differentials and Impossible $(s+1)$-polytopic Transitions

Impossible differential is the distinguisher of impossible differential cryptanalysis, and impossible $(s+1)$-polytopic transition is the distinguisher of the impossible polytopic cryptanalysis. Here, we only recall the definitions of impossible $(s+1)$-polytopic transition, since impossible differential is the special case of $s = 1$. First, let us recall the definition of s-polytope and s-difference.

Definition 1 (s-polytope [29]). *An s-polytope in $\mathbb{F}_2^n$ is an s-tuple of values in $\mathbb{F}_2^n$.*

Definition 2 (s-difference [29]). *An s-difference over $\mathbb{F}_2^n$ is an s-tuple of values in $\mathbb{F}_2^n$. For an $(s+1)$-polytope $(m_0, m_1, \ldots, m_s)$, the corresponding s-difference is defined as $(m_0 \oplus m_1, m_0 \oplus m_2, \ldots, m_0 \oplus m_s)$.*

Next, we recall the propagation rule of s-difference and the valid $(s+1)$-polytopic trail.

Definition 3 (The Propagation Rule of The s-difference [29]). *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^q$ be a function. For the input s-difference $\boldsymbol{\alpha^{n,s}}$ and the output s-difference $\boldsymbol{\beta^{q,s}}$, if there exists x such that, $f(x \oplus \alpha_i) \oplus f(x) = \beta_i (0 \leq i \leq s-1)$, we call that $\boldsymbol{\alpha^{n,s}}$ can propagate to $\boldsymbol{\beta^{q,s}}$, denoted as $\boldsymbol{\alpha^{n,s}} \xrightarrow{f} \boldsymbol{\beta^{q,s}}$. Otherwise, we call that $\boldsymbol{\alpha^{n,s}}$ cannot propagate to $\boldsymbol{\beta^{q,s}}$, denoted as $\boldsymbol{\alpha^{n,s}} \overset{f}{\nrightarrow} \boldsymbol{\beta^{q,s}}$.*

Definition 4 (Valid $(s+1)$-polytopic Trail [29]). *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a function that is the iterated composition of round functions $f_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$:*

$$f := f_r \circ \cdots \circ f_2 \circ f_1.$$

Let $\boldsymbol{\alpha}_0^{n,s}$ *be the input* s*-difference and* $\boldsymbol{\alpha}_r^{n,s}$ *be the output* s*-difference. Then, a valid* $(s+1)$*-polytopic trail for* $(\boldsymbol{\alpha}_0^{n,s}, \boldsymbol{\alpha}_r^{n,s})$ *on* f *is an* $(r+1)$*-tuple* $(\boldsymbol{\alpha}_0^{n,s}, \boldsymbol{\alpha}_1^{n,s}, \ldots, \boldsymbol{\alpha}_r^{n,s})$*, where* $\boldsymbol{\alpha}_i^{n,s} \xrightarrow{f_{i+1}} \boldsymbol{\alpha}_{i+1}^{n,s} (0 \leq i \leq r-1)$.

By exploiting the definition of the valid $(s+1)$-polytopic trail, the definitions of possible $(s+1)$-polytopic transition and impossible $(s+1)$-polytopic transition can be re-expressed as follows.

Definition 5 (Possible $(s+1)$-polytopic Transition [29]). *A pair of input and output* s*-differences* $(\boldsymbol{\Delta_i}^{n,s}, \boldsymbol{\Delta_0}^{n,s})$ *is called an* r*-round possible* $(s+1)$*-polytopic transition if and only if there exists an* r*-round valid* $(s+1)$*-polytopic trail for* $(\boldsymbol{\Delta_i}^{n,s}, \boldsymbol{\Delta_0}^{n,s})$.

Definition 6 (Impossible $(s+1)$-polytopic Transition [29]). *A pair of input and output* s*-differences* $(\boldsymbol{\Delta_i}^{n,s}, \boldsymbol{\Delta_0}^{n,s})$ *is called an* r*-round impossible* $(s+1)$*-polytopic transition if and only if there exists no* r*-round valid* $s+1$*-polytopic trail for* $(\boldsymbol{\Delta_i}^{n,s}, \boldsymbol{\Delta_0}^{n,s})$.

2.3 SAT Problem & STP

The Boolean Satisfiability Problem (SAT) is a classic scientific computation problem aiming to determine whether a given boolean formula has a solution. STP is the openly available solver for the SAT problem, which supports the CVC format as the file-based input formats.

When to solve an SAT problem, we first model it by the statements in CVC format and save those statements as a file. Then, we invoke the STP for this file. If the target SAT problem has no solution, STP will return "Valid.". Otherwise, it will return a solution of the SAT problem and "Invalid.".

In particular, it is worth to mention that the CVC format supports the conditional term, i.e., the statement "IF a THEN b ELSE c ENDIF", where a is a boolean term, and b and c are bitvector terms. By exploiting the conditional term, we give our modeling methods for S-boxes and variable rotation in Sections 4.1.

3 New Definitions of Impossible Differentials and Impossible $(s+1)$-polytopic Transitions

In this section, we define the notations of s-polygon, possible s-polygons, and impossible s-polygons. Based on this, we redefine the impossible differentials and impossible $(s+1)$-polytopic transitions. Then, we study the relations between our redefinitions and traditional definitions of impossible differentials and impossible $(s+1)$-polytopic transitions.

3.1 New Definitions of Impossible Differentials and Impossible $(s+1)$-Polytopic Transitions

Let us think over the definitions of traditional impossible differentials and impossible $(s+1)$-polytopic transitions. For $E \in BC(n,m,l)$, suppose $(\boldsymbol{\Delta_i}^{n,s}, \boldsymbol{\Delta_o}^{n,s})$ is an r-round traditional impossible $(s+1)$-polytopic transition of it. Then, for $\forall k \in (F_2^l)^r$, $\forall \boldsymbol{x_i}^{n,s+1} \rhd \boldsymbol{\Delta_i}^{n,s}$ and $\forall \boldsymbol{y_i}^{n,s+1} \rhd \boldsymbol{\Delta_o}^{n,s}$, it holds $E_k^r(\boldsymbol{x_i}^{n,s+1}) \neq \boldsymbol{y_i}^{n,s+1}$. In particular, if (Δ_i, Δ_0) is an r-round impossible differential. Then, for $\forall k \in (F_2^l)^r$, $\forall x \in \mathbb{F}_2^n$ and $\forall y \in \mathbb{F}_2^n$, it holds $(E_k^r(x), E_k^r(x \oplus \Delta_i)) \neq (y, y \oplus \Delta_o)$. Thus, it is important to research the relations between two (resp. $s+1$) input states and two (resp.$s+1$) output states for forming the impossible differentials (resp.impossible $(s+1)$-polytopic transitions). To investigate such relations, we define the s-polygon firstly.

Definition 7 (s-polygon). *For $\forall E \in BC(n,m,l)$, its s-polygon is a tuple with s elements, where each element belongs to $\mathbb{F}_2^n$.*

For an iterated block cipher, the s-polygon propagates through round by round, which constitutes the s-polygonal trail.

Definition 8 (s-polygonal Trail). *Let $E \in BC(n,m,l)$ and $r \in \mathbb{Z}$. For any s-polygon $\boldsymbol{x}^{n,s}$ and $\forall k = (k_1, \ldots, k_r) \in (\mathbb{F}_2^l)^r$, we have the following chain of propagation:*

$$\boldsymbol{x}^{n,s} \to E_{(k_1)}^1(\boldsymbol{x}^{n,s}) \to E_{(k_1,k_2)}^2(\boldsymbol{x}^{n,s}) \to \cdots \to E_k^r(\boldsymbol{x}^{n,s}).$$

*We call $(\boldsymbol{x}^{n,s}, E_{(k_1)}^1(\boldsymbol{x}^{n,s}), \ldots, E_k^r(\boldsymbol{x}^{n,s}))$ an **r-round s-polygonal trail**. Moreover, if $k \in IKS_r^l$, the trail is called an **r-round i-s-polygonal trail**; if $k \in DKS_r^{m,l}$, the trail is called an **r-round d-s-polygonal trail**.*

Based on the definitions of s-polygon and s-polygonal trail, according to the compatibility of a pair of input and output s-polygons, the possible s-polygon and impossible s-polygon are defined as follows.

Definition 9 (Possible s-polygons). *Let $E \in BC(n,m,l)$, a pair of input and output s-polygons $(\boldsymbol{x}^{n,s}, \boldsymbol{y}^{n,s})$ is called **r-round possible s-polygons** of E, if there exists $k = (k_1, \ldots, k_r) \in (\mathbb{F}_2^l)^r$ and s-polygonal trail $(\boldsymbol{x}^{n,s}, E_{(k_1)}^1(\boldsymbol{x}^{n,s}), \ldots, E_k^r(\boldsymbol{x}^{n,s}))$ s.t. $y_i = E_k^r(x_i)(0 \leq i \leq s-1)$. Moreover, if $k \in IKS_r^l$, $(\boldsymbol{x}^{n,s}, \boldsymbol{y}^{n,s})$ is called **r-round i-possible s-polygons**; if $k \in DKS_r^{m,l}$, $(\boldsymbol{x}^{n,s}, \boldsymbol{y}^{n,s})$ is called **r-round d-possible s-polygons**.*

Definition 10 (Impossible s-polygons). *Let $E \in BC(n,m,l)$, a pair of input and output s-polygons $(\boldsymbol{x}^{n,s}, \boldsymbol{y}^{n,s})$ is called **r-round i-impossible s-polygons** (resp. **r-round d-impossible s-polygons**) of E, if $(\boldsymbol{x}^{n,s}, \boldsymbol{y}^{n,s})$ is not the r-round i-possible s-polygons (resp. r-round d-possible s-polygons).*

Now, based on the definition of impossible s-polygons, we propose two definitions of impossible $(s+1)$-polytopic transitions: i-impossible $(s+1)$-polytopic transition and d-impossible $(s+1)$-polytopic transition.

Definition 11 (The i-impossible (resp.d-impossible) $(s+1)$-polytopic Transition). *Let $E \in BC(n,m,l)$, a pair of input and output tuples $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is called an r-**round i-impossible (resp.d-impossible) $(s+1)$-polytopic transition**, if for $\forall \boldsymbol{x^{n,s+1}} \rhd \boldsymbol{\alpha}^{n,s}$ and $\forall \boldsymbol{y^{n,s+1}} \rhd \boldsymbol{\beta}^{n,s}$, $(\boldsymbol{x^{n,s+1}}, \boldsymbol{y^{n,s+1}})$ are r-round i-impossible (resp.d-impossible) $(s+1)$-polygons.*

Here, we give the definitions of i-impossible differential and d-impossible differential independently for clarity, while actually impossible differential is a particular case of impossible $(s+1)$-polytopic transition.

Definition 12 (The i-impossible (resp. d-impossible) Differential). *Let $E \in BC(n,m,l)$, $\alpha \in \mathbb{F}_2^n$, and $\beta \in \mathbb{F}_2^n$, (α, β) is called an r-**round i-impossible (resp. d-impossible) differential**, if for $\forall (x_0, x_1) \in \{(\alpha_0, \alpha_1) \in \mathbb{F}_2^n \times \mathbb{F}_2^n | \alpha_0 \oplus \alpha_1 = \alpha\}$ and $\forall (y_0, y_1) \in \{(\beta_0, \beta_1) \in \mathbb{F}_2^n \times \mathbb{F}_2^n | \beta_0 \oplus \beta_1 = \beta\}$, (x_0, x_1) and (y_0, y_1) are r-round i-impossible (resp. d-impossible) 2-polygons.*

According to the definitions of d-possible $(s+1)$-polygons and i-possible $(s+1)$-polygons, the relation between i-impossible $(s+1)$-polytopic transition and d-impossible $(s+1)$-polytopic transition is obviously as follows.

Theorem 1. *Let $E \in BC(n,m,l)$. Then an i-impossible $(s+1)$-polytopic transition of E must be a d-impossible $(s+1)$-polytopic transition of E. In particular, an i-impossible differential of E must be a d-impossible differential of E.*

3.2 The Equivalence of i-impossible $(s+1)$-polytopic Transitions and Traditional Impossible $(s+1)$-polytopic Transitions

SPN structure and Feistel structure are widely used in the design of block ciphers. In this subsection, we show that the i-impossible $(s+1)$-polytopic transitions are equivalent to traditional impossible $(s+1)$-polytopic transitions for the block ciphers with SPN structure or Feistel structure. Moreover, with the same approach, the equivalence also holds for the block cipher MISTY1. Note that, since impossible differentials are the particular case of impossible $(s+1)$-polytopic transitions, we are not going to state the equivalency for impossible differentials solely here.

First, for narrative purposes, we define a class of round function, which is widely used in block ciphers.

Definition 13 (Common Round Function). *A function F_r is called common round function(CRF), if it can be represented as $F_r = (P_r^{'} \circ S_r \circ P_r \circ K_r) \circ \cdots \circ (P_1^{'} \circ S_1 \circ P_1 \circ K_1) \circ (P_0^{'} \circ S_0 \circ P_0)$, where $S_i (0 \le i \le r)$ denotes the substitution layer which is composed of a set of S-boxes in parallel, $P_i (0 \le i \le r)$ and $P_i^{'} (0 \le i \le r)$ denote the linear permutation layers, and $K_i (1 \le i \le r)$ denotes the key mixing layer, where the key is fully Xored with the state. In particular, in the case of $r = 0$, denote $F_0 = (P_0^{'} \circ S_0 \circ P_0)$.*

The above definition of CRF includes a lot of round functions, which are broadly used in block ciphers. For example, the round function of AES [11] is of the "SP" structure, in which the substitution layer precedes the linear layer. It is the CRF in the case of $r = 0$ and P_0 is the identical permutation. The round function of Prince [7] in the last half rounds is of the "PS" structure, in which the linear layer precedes the substitution layer. It is the CRF in the case of $r = 0$ and $P_0^{'}$ is the identical permutation. The round function of RoadRunneR [3] is of the "SPKSPKSPKS" structure. It is the CRF in the case of $r = 3$ and $P_3^{'}$ is the identical permutation.

Since the common round function is widely used in block ciphers, we study the relationship between the valid $(s+1)$-polytopic transitions and i-possible $(s+1)$-polygons of it.

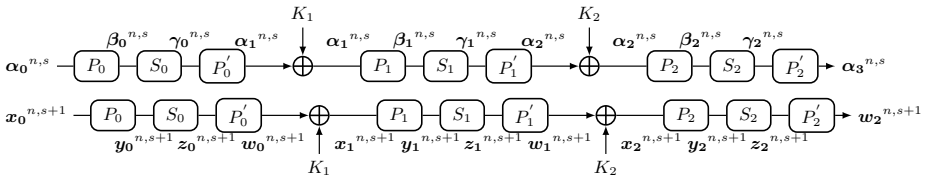

Fig. 1. The Valid $(s+1)$-polytopic Trail and $(s+1)$-polygonal Trail for CRF

Theorem 2 (The Equivalence of CRF). *Let F_r be a CRF. Then, $(\boldsymbol{\alpha_0}^{n,s}, \boldsymbol{\alpha_{r+1}}^{n,s})$ is a valid polytopic transition of F_r if and only if there exist i-possible $(s+1)$-polygons $(\boldsymbol{x_0}^{n,s+1}, \boldsymbol{w_r}^{n,s+1})$ of F_r , where $\boldsymbol{x_0}^{n,s+1} \rhd \boldsymbol{\alpha_0}^{n,s}$ and $\boldsymbol{w_r}^{n,s+1} \rhd \boldsymbol{\alpha_{r+1}}^{n,s}$.*

Proof. We only prove this theorem in the case of $r = 2$. The other cases can be proved analogously.

Suppose $(\boldsymbol{\alpha_0}^{n,s}, \boldsymbol{\alpha_3}^{n,s})$ is a valid polytopic transition of F_2. Then there exists a valid $(s+1)$-polytopic trail $(\boldsymbol{\alpha_0}^{n,s}, \boldsymbol{\alpha_1}^{n,s}, \boldsymbol{\alpha_2}^{n,s}, \boldsymbol{\alpha_3}^{n,s})$, as shown in the upper half of Figure 1. For $0 \le i \le 2$, since $(\boldsymbol{\beta_i}^{n,s}, \boldsymbol{\gamma_i}^{n,s})$ is a possible $(s+1)$-polytopic transition of S_i, there exists a_i such that $S_i(a_i) \oplus S_i(a_i \oplus \beta_{i,j}) = \gamma_{i,j} (0 \le j \le s-1)$. Let $\boldsymbol{y_i}^{n,s+1} = (y_{i,0}, \ldots, y_{i,s})$ and $\boldsymbol{z_i}^{n,s+1} = (z_{i,0}, \ldots, z_{i,s})$, where $y_{i,0} = a_i$, $y_{i,j+1} = a_i \oplus \beta_{i,j}$, $z_{i,0} = S_i(a_i)$ and $z_{i,j+1} = S(a_i) \oplus \gamma_{i,j}$, then we have $S(y_{i,j}) = z_{i,j} (0 \le j \le s)$. Denote $\boldsymbol{x_i}^{n,s+1} = (x_{i,0}, \ldots, x_{i,s})$ and $\boldsymbol{w_i}^{n,s+1} = (w_{i,0}, \ldots, w_{i,s})$, where $x_{i,j} = P_i^{-1}(y_{i,j})$ and $w_{i,j} = P_i^{'}(z_{i,j}) (0 \le j \le s)$. Since $\alpha_{i,j} = P_i^{-1}(\beta_{i,j})$, we have $x_{i,0} \oplus x_{i,j+1} = \alpha_{i,j} (0 \le j \le s-1)$. Similar, we have $w_{i,0} \oplus w_{i,j+1} = \alpha_{i+1,j} (0 \le j \le s-1)$. Thus, for $1 \le i \le 2$, we have $w_{i-1,0} \oplus w_{i-1,j+1} = \alpha_{i,j} = x_{i,0} \oplus x_{i,j+1} (0 \le j \le s-1)$. Let $K_i = w_{i-1,0} \oplus x_{i,0}$, then we have $x_{i,j} = w^{i-1,j} \oplus K_i (0 \le j \le s)$. Therefore, we have constructed i-possible $(s+1)$-polygons of F_2, which is $(\boldsymbol{x_0}^{n,s+1}, \boldsymbol{w_2}^{n,s+1})$ with $\boldsymbol{w_2}^{n,s+1} \rhd \boldsymbol{\alpha_3}^{n,s}$ and $\boldsymbol{x_0}^{n,s+1} \rhd \boldsymbol{\alpha_0}^{n,s}$, as shown in the lower half of Figure 1.

Since all the procedures above are invertible, it is easy to show that if there exist $\boldsymbol{x_0}^{n,s+1} \rhd \boldsymbol{\alpha_0}^{n,s}$ and $\boldsymbol{w_2}^{n,s+1} \rhd \boldsymbol{\alpha_3}^{n,s}$, such that $(\boldsymbol{x_0}^{n,s+1}, \boldsymbol{w_2}^{n,s+1})$ is the

i-possible $(s+1)$-polygons of F_2, then $(\boldsymbol{\alpha_0}^{n,s}, \boldsymbol{\alpha_3}^{n,s})$ is the valid polytopic transition of F_2. □

With the same technique, we also can show the equivalence between traditional impossible $(s+1)$-polytopic transition and the i-impossible $(s+1)$-polytopic transition for the block ciphers with SPN structure and Feistel structure as follows. The specific process of proofs are shown in the Full Version of our paper in the ePrint because of space cause.

Theorem 3 (The Equivalence of SPN Structure Block Ciphers). *Let $E \in BC(n, m, l)$ be an SPN structure block cipher whose round function is a CRF, and the round keys are fully Xored with the state. Then, $(\boldsymbol{\alpha_0}^{n,s}, \boldsymbol{\alpha_r}^{n,s})$ is an r-round traditional impossible $(s+1)$-polytopic transition if and only if it is an r-round i-impossible $(s+1)$-polytopic transition.*

Theorem 4 (The Equivalence of Feistel Structure Block Ciphers). *Let $E \in BC(2n, m, l)$ be a Feistel structure block cipher whose round function is a CRF and the round keys are fully Xored with the branch. Then, $(\boldsymbol{\alpha_0}^{n,s}||\boldsymbol{\beta_0}^{n,s}$, $\boldsymbol{\alpha_r}^{n,s}||\boldsymbol{\beta_r}^{n,s})$ is an r-round traditional impossible $(s+1)$-polytopic transition if and only if it is an r-round i-impossible $(s+1)$-polytopic transition.*

The block cipher MISTY1 [20] is designed by adopting the theory of provable security [23]. We can also show that traditional impossible $(s+1)$-polytopic transition is equivalent to the i-impossible $(s+1)$-polytopic transition for the block cipher MISTY1 as the following theorem. The specific process of proof is also shown in the Full Version of our paper.

Theorem 5 (The Equivalence of The Block Cipher MISTY1). *Let E denote the block cipher MISTY1. Then, $(\boldsymbol{\alpha_0}^{32,s}||\boldsymbol{\beta_0}^{32,s}, \boldsymbol{\alpha_r}^{32,s}||\boldsymbol{\beta_r}^{32,s})$ is an r-round traditional impossible $(s+1)$-polytopic transition if and only if it is an r-round i-impossible $(s+1)$-polytopic transition.*

The advantages of i-impossible differentials and i-impossible $(s+1)$-polytopic transitions. Since i-impossible differentials (resp. i-impossible $(s+1)$-polytopic transitions) are equivalent to traditional impossible differentials (resp. traditional impossible $(s+1)$-polytopic transitions), our method gives new view of traditional impossible differentials and impossible $(s+1)$-polytopic transitions, which allows us to get the distinguishers for the block cipher with large S-boxes or variable rotation in the key independent setting using full knowledge of their differential or s-differential property. In particular, by exploiting this new view, we can evaluate the security of block ciphers against traditional impossible differentials for block ciphers with large S-box in the case of considering the differential property of large S-boxes.

4 Automatic Search Method

In this section, we propose an unified automatic search algorithm for our redefined impossible differentials and impossible $(s+1)$-polytopic transitions. Firstly,

we give the statements in CVC format to model the propagation of the state under each operation.

4.1 Model the Propagation of the State by Statements in CVC Format

Here, we model the propagation of the state under the operations (Generalized-) Copy, (Generalized-) Xor, (Generalized-) Modular Addition, Linear Transformations, S-box and Variable Rotation by statements in CVC format.

Model 1 ((Generalized-)Copy) *Let F be a (Generalized-)Copy function, where the input x takes value from $\mathbb{F}_2^q$, and the output is calculated as $(y_0, y_1, \ldots, y_{t-1}) = (x, x, \ldots, x)$. Then, the following statements can describe the propagation of the state under the (Generalized-)Copy operation.*

$$\begin{cases} ASSERT(y_0 = x); \\ ASSERT(y_1 = x); \\ \quad \vdots \\ ASSERT(y_{t-1} = x); \end{cases}$$

Model 2 ((Generalized-)Xor) *Let F be a (Generalized-)Xor function, where the input $(x_0, x_1, \ldots, x_{t-1})$ take values from $(\mathbb{F}_2^q)^t$, and the output is calculated as $y = \oplus_{i=0}^{i=t-1} x_i$. Then, the following statement can describe the propagation of the state under the (Generalized-)Xor operation.*

$$ASSERT(y = BVXOR(\cdots(BVXOR(BVXOR(x_0, x_1), x_2), \ldots, x_{t-1}));$$ [8]

Model 3 ((Generalized-)Modular Addition) *Let F be a (Generalized-) Modular Addition function, where the input $(x_0, x_1, \ldots, x_{t-1})$ take values from $(\mathbb{F}_2^q)^t$, and the output is calculated as $y = \boxplus_{i=0}^{i=t-1} x_i$. Then, the following statement can describe the propagation of the state under the (Generalized-)Modular Addition operation.*

$$ASSERT(y = BVPLUS(q, x_0, \ldots, x_{t-1}));$$ [9]

The linear transformations of block ciphers have various representations, such as the permutation layer of PRESENT [6], and the MDS matrix in AES [11]. Since all the representations of linear transformations can be converted to the binary matrix multiplication, we only show the modeling method for the binary matrix multiplication here.

Model 4 (Binary Matrix Multiplication) *Let $M = (m_{i,j})_{0 \le i \le s-1, 0 \le j \le t-1}$ be a binary matrix, where the input $x = (x_0, x_1, \ldots, x_{t-1})$ take values from $\mathbb{F}_2^t$, and the output of multiplication $y = (y_0, y_1, \ldots, y_{s-1})$ is calculated as*

$$y_i = \begin{cases} x_k, \text{ if } m_{i,k} = 1 \text{ and } |\{j | m_{i,j} \neq 0\}| = 1, \\ \oplus_{\{j | m_{i,j} \neq 0\}} x_j, \text{ otherwise.} \end{cases}$$

[8] BVXOR: Bitwise XOR function which is supported by the CVC format of STP
[9] BVPLUS: Bitvector Add function which is supported by the CVC format of STP

Then, the statements to describe the propagation of the state under binary matrix multiplication operation can be combined by the modeling methods for Copy and (Generalized-) Xor.

S-box is often used to provide confusion for block ciphers. By exploiting the conditional term, we can describe the propagation of the state under it specifically.

Algorithm 1 *Function for Modeling S-box*

1: **Input**: S, x, y
2: **Output**: *The statement to describe the propagation of the state under S-box*
3: $statement_1 = S[0]$
4: **for** $j = 1$ to $2^t - 1$ **do**
5: $\quad statement_1$ = "*IF* $x = j$ *THEN* $S[j]$ *ELSE* $statement_1$"
6: **endfor**
7: $statement$ = "$ASSERT(y = statement_1)$;"
8: **return** *statement*

Model 5 (S-box) *Let S be an S-box which substitutes t-bit to s-bit, where the input x takes values from $\mathbb{F}_2^t$, and the output $y \in \mathbb{F}_2^s$ is calculated as $y = S(x)$. Then the statement generated by Algorithm 1 can describe the propagation of the state under S-box operation.*

Variable rotation is a novel operation used in some typical block ciphers, such as RC5 [24] and RC6 [25]. Due to the output of variable rotation operation is closely related to the input values, it is hard to model the propagation of difference and s-difference under it. In our new model, we exploit the conditional term to describe the propagation of the state under the variable rotation.

Algorithm 2 *Function for Modeling Variable Rotation*

1: **Input**: q, x, y, z
2: **Output**: The statement to describe the propagation of the state under variable rotation
3: $statement_1 = x$
4: **for** $j = 1$ to $q - 1$ **do**
5: $\quad statement_1$ = "*IF* $(y \bmod q) = j$ *THEN* $x \lll_j$ *ELSE* $statement_1$"
6: **endfor**
7: $statement$ = "$ASSERT(z = statement_1)$;"
8: **return** *statement*

Model 6 (Variable Rotation) *Let F be a variable rotation function, the input (x, y) take values from $\mathbb{F}_2^q \times \mathbb{F}_2^q$, and the output is calculated as $z = x \lll_y \in \mathbb{F}_2^q$. Then, the statement generated by the Algorithm 2 can describe the propagation of the state under variable rotation operation.*

4.2 The Automatic Search Method for Redefined Impossible Differentials and Impossible $(s+1)$-polytopic Transitions

In this subsection, we show our automatic search algorithm for the i-impossible (resp. d-impossible) $(s+1)$-polytopic transitions. Since an i-impossible (resp. d-impossible) differential is an i-impossible (resp. d-impossible) 2-polytopic transition, the automatic search algorithm for i-impossible (resp. d-impossible) differentials can be derived from the algorithm for i-impossible (resp. d-impossible) $(s+1)$-polytopic transitions with $s=1$. First, we propose our method for determining whether a pair of input and output s-differences is an i-impossible (resp. d-impossible) $(s+1)$-polytopic transition. Then, we discuss the selection of parameter s and the search space of our method.

The i-impossible (resp. d-impossible) $(s+1)$-polytopic Transition Determining Method.
Our method for determining whether a pair of input and output s-differences $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an i-impossible (resp. d-impossible) $(s+1)$-polytopic transition can be divided into two phases: statements generated phase and STP invoked phase. In the statements generated phase, we generate a system of statements as a file to describe the $(s+1)$-polygons $\boldsymbol{x}^{n,s+1}$ propagate to $\boldsymbol{y}^{n,s+1}$ with $\boldsymbol{x}^{n,s+1} \rhd \boldsymbol{\alpha}^{n,s}$ and $\boldsymbol{y}^{n,s+1} \rhd \boldsymbol{\beta}^{n,s}$. In the STP invoked phase, we invoke the STP for the file to determine whether $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an i-impossible (resp. d-impossible) $(s+1)$-polytopic transition.

Specification of the statements generated phase.
The algorithm shown in Algorithm 3 generates the statements for judging whether a pair of input and output s-differences $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an r-round impossible $(s+1)$-polytopic transition.

Algorithm 3 Generating statements in CVC format

1: **Input**: the number of rounds r, the input s-difference $\boldsymbol{\alpha}^{n,s}$, the output s-difference $\boldsymbol{\beta}^{n,s}$ and keyflag$\in$ {True, False}
2: **Output**: System of statements in CVC format
3: Declare the input and output $(s+1)$-polygons of $\boldsymbol{x}^{n,s+1}$ and $\boldsymbol{y}^{n,s+1}$.
4: Declare the intermediate variables and key variables.
5: **for** $i=0$ to s **do**
6: Model the r-round propagation of (x_i, y_i).
7: **endfor**
8: Generate the constraint of $\boldsymbol{x}^{n,s+1}$ such that $\boldsymbol{x}^{n,s+1} \rhd \boldsymbol{\alpha}^{n,s}$.
9: Generate the constraint of $\boldsymbol{y}^{n,s+1}$ such that $\boldsymbol{y}^{n,s+1} \rhd \boldsymbol{\beta}^{n,s}$.
10: **if** keyflag **then**
11: Generate the constraint of key variables according to key shedule.
12: **endif**
13: Add the statements "QUERY(FALSE);" and "COUNTEREXAMPLE;".

We present certain illustrations for Algorithm 3 as follows.

- Line 3-4. Declare the variables which are used in the system of statements, including the variables which are used to represent the input $(s+1)$-polygon and output $(s+1)$-polygon, the intermediate variables and key variables used to describe the propagation from the input $(s+1)$-polygon to the output $(s+1)$-polygon.
- Line 5-7. According to the propagation rules for each operation which are given in Section 4.1, model the propagation from the input $(s+1)$-polygon $\boldsymbol{x}^{n,s+1}$ to the output $(s+1)$-polygon $\boldsymbol{y}^{n,s+1}$ with the aid of the intermediate variables and key variables.
- Line 8-9. Generate the statements in CVC format such that the input $(s+1)$-polygon $\boldsymbol{x}^{n,s+1}$ satisfies the input s-difference $\boldsymbol{\alpha}^{n,s}$ and the output $(s+1)$-polygon $\boldsymbol{y}^{n,s+1}$ satisfies the output s-difference $\boldsymbol{\beta}^{n,s}$.
- Line 10-12. If "keyflag=True", then the algorithm generates the statements to constraint the key variables according to the key schedule. In this case, the algorithm generates the statements to judge whether a pair of input and output s-differences $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an r-round d-impossible $(s+1)$-polytopic transition; Otherwise, it generates the statements to judge whether a pair of input and output s-differences $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an r-round i-impossible $(s+1)$-polytopic transition.
- Line 13. The statements "QUERY(FALSE);" and "COUNTEREXAMPLE;" are added to the system of statements. This is a common method in STP to determine whether an SAT problem has a solution. By adding those two statements, if the SAT problem has solutions, the STP will return one of the solutions and the statement "Invalid."; Otherwise, it returns "Valid.".

Specification of the invoke STP phase.

We invoke the STP for the file which is consisted of the system of statements. If the statements generated in the case of *keyflag=True*, then the s-differences $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an r-round d-impossible $(s+1)$-polytopic transition when the STP returns "Valid.", and $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is not an r-round d-impossible $(s+1)$-polytopic transition when the STP returns an r-round d-$(s+1)$-polygonal trail and "Invalid.". Similarly, if the statements generated in the case of *keyflag=False*, then the s-differences $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is an r-round i-impossible $(s+1)$-polytopic transition when the STP returns "Valid.", and $(\boldsymbol{\alpha}^{n,s}, \boldsymbol{\beta}^{n,s})$ is not an r-round i-impossible $(s+1)$-polytopic transition when the STP returns an r-round i-$(s+1)$-polygonal trail and "Invalid.".

Work as a proof tool. Once the search space fixed, we can run our tool for all the input and output s-differences in such space. If none of the input and output s-differences is an r-round i-impossible (resp. d-impossible) $(s+1)$-polytopic transition, we can declare that there exists no r-round i-impossible (resp. d-impossible) $(s+1)$-polytopic transition in this space.

The Select of parameter s and Search Space.

In our automatic search method for impossible $(s+1)$-polytopic transition, the total time cost mainly depends on the size of the search space and the time

cost for determining whether an element in the search space is an impossible $(s+1)$-polytopic transition.

The time cost for determining whether an element in the search space is an impossible $(s+1)$-polytopic transition is closely related to operations contained in the block cipher and the value of parameter s we selected. In our experiment, we choose s at most 4, since the search time will cost quite a lot if s increases beyond this range.

For the search space, traditional automatic tools focus on search the μ input active bits (resp. nibbles) and ν output active bits (resp. nibbles) impossible differentials. Since the impossible $(s+1)$-polytopic transition is the generation of impossible differential, we define the $(\mu_0, \ldots, \mu_{s-1})$ active bits and $(\mu_0, \ldots, \mu_{s-1})$ active nibbles to generate the search space.

Definition 14 ($(\mu_0, \ldots, \mu_{s-1})$ Active Bits). *For a block cipher $E \in BC(n, m, l)$, we call the s-difference $\boldsymbol{\alpha}^{n,s}$ satisfied the $(\mu_0, \ldots, \mu_{s-1})$ active bits, if there are μ_i bits of the binary representation of $\alpha_i (0 \leq i \leq s-1)$ are non-zero.*

Definition 15 ($(\mu_0, \ldots, \mu_{s-1})$ Active Nibbles). *For a block cipher $E \in BC(n, m, l)$ whose S-box size is q, for any s-difference $\boldsymbol{\alpha}^{n,s}$, the binary representation of α_i $(0 \leq i \leq s-1)$ can be divided into $\frac{n}{q}$ pieces, where $\alpha_{i,j} = \{\alpha_{i,q\cdot j}, \ldots, \alpha_{i,q\cdot j+q-1}\}$ $(0 \leq j \leq \frac{n}{q} - 1)$. We call the s-difference $\boldsymbol{\alpha}^{n,s}$ satisfied the $(\mu_0, \ldots, \mu_{s-1})$ active nibbles, if there are μ_i pieces of $\alpha_i (0 \leq i \leq s-1)$ have non-zero items.*

Our method focuses on searching the $(\mu_0, \ldots, \mu_{s-1})$ input active bits and $(\nu_0, \ldots, \nu_{s-1})$ output active bits or $(\mu_0, \ldots, \mu_{s-1})$ input active nibbles and $(\nu_0, \ldots, \nu_{s-1})$ output active nibbles, or the subset of those two spaces according to the experimental result. Due to the limitation of the size of the executable search space, we mainly search some small values of active bits and active nibbles. Assume the value μ'_i $(0 \leq i \leq g)$ appears φ_i times in the tuple $(\mu_0, \ldots, \mu_{s-1})$ and value ν'_i $(0 \leq i \leq h)$ appears ϕ_i times in the tuple $(\nu_0, \ldots, \nu_{s-1})$. Then, for a block cipher $E \in BC(n, m, l)$, the number of pairs of input and output s-differences with $(\mu_0, \ldots, \mu_{s-1})$ input active bits and $(\nu_0, \ldots, \nu_{s-1})$ output active bits is

$$\binom{\binom{n}{\mu'_0}}{\varphi_0} \times \cdots \times \binom{\binom{n}{\mu'_g}}{\varphi_g} \times \binom{\binom{n}{\nu'_0}}{\phi_0} \times \cdots \times \binom{\binom{n}{\nu'_h}}{\phi_h} \sim O(n^{\mu'_0\varphi_0 + \cdots + \mu'_g\varphi_g + \nu'_0\phi_0 + \cdots + \nu'_h\phi_h}).$$

For a block cipher $E \in BC(n, m, l)$ whose S-box size is q, let $p = \frac{n}{q}$, the number of pairs of input and output s-differences with $(\mu_0, \ldots, \mu_{s-1})$ input active nibbles and $(\nu_0, \ldots, \nu_{s-1})$ output active nibbles is

$$\binom{\binom{p}{\mu'_0} \cdot (2^q - 1)}{\varphi_0} \times \cdots \times \binom{\binom{p}{\mu'_g} \cdot (2^q - 1)}{\varphi_g} \times \binom{\binom{p}{\nu'_0} \cdot (2^q - 1)}{\phi_0} \times \cdots \times \binom{\binom{p}{\nu'_h} \cdot (2^q - 1)}{\phi_h},$$

which is $O(p^{\mu'_0\varphi_0 + \cdots + \mu'_g\varphi_g + \nu'_0\phi_0 + \cdots + \nu'_h\phi_h} \cdot 2^{q\cdot(\mu'_0 + \cdots + \mu'_g + \nu'_0 + \cdots + \nu'_h)})$.

According to the above analysis, the size of the search space is still large even we only search for small values of active bits and active nibbles for impossible

$(s+1)$-polytopic transitions with small value of parameter s. For example, if we search the $(1,1)$ input active bits and $(1,1)$ output active bits for the impossible 3-polytopic transition of a block cipher whose block size is 64, the number of pairs of input and output s-differences is $\binom{\binom{64}{1}}{2} \times \binom{\binom{64}{1}}{2} = 4064256 \approx 2^{22}$. Thus, we propose the following step by step strategy, which is quite helpful to search the impossible $(s+1)$-polytopic transitions when the search space is too large.

Step by step strategy. The core of this strategy is to search the impossible $(s+1)$-polytopic($s \geq 2$) transition based on the result of the impossible s-polytopic transition. To be specific, for a block cipher $E \in BC(n, m, l)$, if we know that $(\boldsymbol{\alpha}^{n,s-1}, \boldsymbol{\beta}^{n,s-1})$ is an impossible s-polytopic transition, then we search the impossible $(s+1)$-polytopic($s \geq 2$) transition in the set

$$\{(\alpha_0, \ldots, \alpha_{s-2}, \alpha) \times (\beta_0, \ldots, \beta_{s-2}, \beta) | \text{the active bits (nibbles) of } \alpha \text{ and } \beta \text{ is } u \text{ and } v \text{ respectively}\},$$

where u and v are the predetermined values.

5 Applications to Impossible Differentials from the Aspect of Cryptanalysis

In this section, we apply our method to various block ciphers, including the block cipher GIFT64 [2], the key-dependent permutation (or the key-dependent S-box) based block cipher PRINTcipher [15], the large S-boxes based block cipher MISTY1 [20], and the variable rotation based block cipher RC5 [24]. Only concise descriptions of those block ciphers are specified here. For more details, please refer to their coresponding references. All the experiments in this paper are conducted on this platform: Intel(R) Xeon(R) CPU E5-2650 v2 @2.60GHz, 64.00G RAM, 64-bit Windows 7 system. The source codes are available in https://github.com/HugeChaos/Impossible-differentials-and-impossible-polytopic-transitions.

5.1 GIFT64

GIFT64 was designed by Banik el at. [2], it is a 64-bit block cipher with 128-bit master key. Interestingly, its round key is 32-bit while it adopts the SPN structure.

Previous best result. In [2], they searched the impossible differentials by limiting the input difference activates only one of the first four S-boxes and the output difference activates only one S-box. The maximum number of rounds of impossible differentials they got in this search space is 6.

Advantage of our tool. Compared with the previous tools, our tools can search the impossible differentials taking into account the key schedule.

Configurations for the tool. Firstly, in the search space where the input and output difference activates only one S-box, the maximum number of rounds of the impossible differentials we got is also 6. Then, we try to find the 6-round

impossible differentials in which the contradiction cannot be detected by the previous method. To achieve this purpose, we randomly pick the input differences activate at most the right 16 bits and the output differences activate at most the i-th ($i \in \{0, 4, 8, 12, 17, 21, 25, 29, 34, 38, 42, 46, 51, 55, 59, 63\}$) bit. In this way, it allows at most the 0th, 4th, 8th and 12th S-box to be active in the 2nd round by propagating the input difference in the forward direction, and at most the 0th, 1st, 2nd and 3rd S-box to be active in the 5th round by propagating the output difference in the backward direction. After 65536 random tests, we find 3 6-round impossible differentials that the previous tools cannot detect.

Example of 6-round d-impossible differentials. One of the 6-round d-impossible differentials is

$$0x0000000000000600 \overset{6-round}{\nrightarrow} 0x0000004020000110.$$

Automatic verification for above example of impossible differential of GIFT64. Since this impossible differential cannot be detected by the propagation of difference, verifying this impossible differential by manual is difficult, we modify the verification algorithm in [10] and apply it to verify this impossible differential. The details of our verification are shown in the Full Version of our paper.

5.2 PRINTcipher

PRINTcipher [15] is proposed by Lars et al. at CHES 2010, consisting of two versions: PRINTcipher48 and PRINTcipher96. PRINTcipher48 is a block cipher with 48-bit block and 80-bit key. PRINTcipher96 is a block cipher with 96-bit block and 160-bit key.

Advantage of our tool. Previous tools cannot apply to PRINTcipher directly due to that they cannot handle the operation of key-dependent permutation. By making use of the conditional term, we propose the first modeling method to describe the propagation of state for key-dependent permutation: *ASSERT(y2@y1@y0 = (IF k1@k0 = 0bin11 THEN x0@x1@x2 ELSE (IF k1@k0 = 0bin10 THEN x2@x0@x1 ELSE (IF k1@k0 = 0bin01 THEN x1@x2@x0 ELSE x2@x1@x0 ENDIF) ENDIF) ENDIF));*
where $x2||x1||x0$ is the input variable, $y2||y1||y0$ is the output variable, and $k1||k0$ is the control key. This modeling method allows us to search the impossible differentials for PRINTcipher by considering the impact of all the details of key schedule. Besides, the PRINTcipher also can be regarded as the key-dependent S-box based block cipher, where the key-dependent S-box is consisted of the key-dependent permutation and the fixed S-box. We also propose the first modeling method to describe the propagation of state for key-dependent S-box directly, which is shown in the Full Version of our paper.

Configurations for the tool. By considering all the details of key schedule, we search the impossible differentials for PRINTcipher48 and PRINTcipher96 in the space where the input difference activates only one S-box in the first substitution layer and the output difference activates only one S-box in the last

substitution layer . Finally, we found 730 4-round d-impossible differentials for PRINTcipher48 and 234 5-round d-impossible differentials for PRINTcipher96 in total.

Example of d-impossible differentials of PRINTcipher. One of the 730 4-round d-impossible differentials of PRINTcipher48 is

$$0x000000000001 \overset{4-round}{\nrightarrow} 0x000000000008.$$

One of the 234 5-round d-impossible differentials of PRINTcipher96 is

$$0x000000000000000200000000 \overset{5-round}{\nrightarrow} 0x000000000000000000001000.$$

Manual verification for the above example of impossible differential of PRINTcipher. As the impossible differentials are detected by considering the key schedule, the verification is completely different from the previous impossible differentials. First, we have the following observation for the composition of key-dependent permutation and S-box.

Obsetvation 1 *Let $SP_k = S \circ P_k$, where S denotes the S-box of PRINTcipher and P_k denotes the key-dependent permutation. Then, $1 \xrightarrow{SP_0} \{1,3,5,7\}$, $1 \xrightarrow{SP_1} \{1,3,5,7\}$, $1 \xrightarrow{SP_2} \{2,3,6,7\}$, and $1 \xrightarrow{SP_3} \{4,5,6,7\}$. On the contrary, we have $\{1,3,5,7\} \xrightarrow{SP_0} 1$, $\{1,3,5,7\} \xrightarrow{SP_1} 1$, $\{2,3,6,7\} \xrightarrow{SP_2} 1$, and $\{4,5,6,7\} \xrightarrow{SP_3} 1$.*

Then, we verify the 4-round example of impossible differential of PRINTcipher48 in case that 0th or 5th S-box in the 3rd round is active. More details of the proof are given in the Full Version of our paper. The 5-round example of PRINTcipher96 can be verified similarly.

5.3 MISTY1

The block cipher MISTY1 was designed by Matsui [20]. It is a 64-bit block cipher which adopts the theory of provable security [23] against differential attack [5] and linear attack [19].

The result by Sasaki et al.'s method. Sasaki et al.'s method is the most advanced previous method to search the impossible differentials for block ciphers with large S-boxes. We employ this method to search the 1 input active bit and 1 output active bit impossible differentials by limiting the input difference activates only the right branch and the output difference activates only the left branch. After $32 \times 32 = 1024$ tests, the maximum number of rounds we got is 4 and a total of 28 4-round impossible differentials are found.

Advantage of our tool. Compared with previous tools, our tool is the first tool that can search the impossible differentials for large S-boxes based block ciphers taking into account the differential property of the S-boxes in the independent key setting.

Configurations for the Tool. We run our tool to search the i-impossible differentials in the search space as that by Sasaki et al.'s method. Finally, we found

902 4-round i-impossible differentials, and all the 4-round impossible differentials derived by Sasaki et al.'s method are detected by our tool.

List of 4-round i-impossible differentials. All the 4-round impossible differentials we found are shown in the Table 1, where $\mathbb{Z}_{32} = \{0, 1, \ldots, 31\}$ and $A = \{33, 35, 36, 46, 49, 50, 51, 52, 53, 57, 58, 62\}$.

Table 1. 4-Round Impossible Differentials of MISTY-1

ID	ΔP	ΔC	Number
001	$e_i^{64}(i \in \mathbb{Z}_{32}/\{3, 12, 19, 28\})$	e_{32}^{64}	28
002	$e_i^{64}(i \in \mathbb{Z}_{32}/\{14, 30\})$	e_{34}^{64}	30
003	$e_i^{64}(i \in \mathbb{Z}_{32}/\{7, 23\})$	e_{37}^{64}	30
004	$e_i^{64}(i \in \{0, 9, 11, 12, 13, 14, 15, 16, 25, 27, 28, 29, 30, 31\})$	e_{38}^{64}	14
005	$e_i^{64}(i \in \{1, 4, 5, 6, 7, 10, 17, 20, 21, 22, 23, 26\}$	e_{43}^{64}	12
006	$e_i^{64}(i \in \{4, 5, 6, 7, 10, 20, 21, 22, 23, 26\})$	e_{44}^{64}	10
007	$e_i^{64}(i \in \{0, 3, 4, 5, 6, 7, 8, 10, 16, 19, 20, 21, 22, 23, 24, 26\}$	e_{45}^{64}	16
008	$e_i^{64}(i \in \mathbb{Z}_{32}/\{12, 28\})$	e_{48}^{64}	30
009	$e_i^{64}(i \in \mathbb{Z}_{32}/\{6, 22\})$	e_{54}^{64}	30
010	$e_i^{64}(i \in \mathbb{Z}_{32})$	$e_j^{64}(j \in A)$	384
011	$e_i^{64}(i \in \mathbb{Z}_{32}/\{12 + j, 28 + j\})$	$e_{55+j}^{64}(j \in \{0, 1\})$	60
012	$e_i^{64}(i \in \mathbb{Z}_{32}/\{11, 27\})$	$e_j^{64}(j \in \{47, 63\})$	60
013	$e_i^{64}(i \in \mathbb{Z}_{32}/\{11, 12, 13, 27, 28, 29\})$	$e_j^{64}(j \in \{59, 60, 61\})$	78
014	$e_i^{64}(i \in \mathbb{Z}_{32}/\{12 + j, 28 + j\})$	$e_{39+j}^{64}(j \in \{0, 1, 2, 3\})$	120

Manual verification for the 4-round i-impossible differentials $(e_i^{64}, e_{52}^{64})(i \in \mathbb{Z}_{32})$ of MISTY1. First, we study the property of the FL and FO function of MISTY1.

Obsetvation 2 *Let F denote the FL function of MISTY1, if the input difference is one of e_i^{32}, e_{i+16}^{32}, and $e_{i,i+16}^{32}$ $(0 \le i \le 15)$, all possible output difference of F is $\{e_i^{32}, e_{i+16}^{32}, e_{i,i+16}^{32}\}$. Moreover, all possible output difference of F^2 is also $\{e_i^{32}, e_{i+16}^{32}, e_{i,i+16}^{32}\}$, where F^2 denotes the composition of two FL function.*

Proposition 1. *Let F denote the FO function of MISTY1 and $\gamma_i (0 \le i \le 1)$ be the 16-bit variables, for $\forall(\gamma_1||\gamma_0) \in \{\beta | e_{20}^{32} \xrightarrow{F} \beta\}$, the weight of γ_1 must be greater than 1.*

Then, we verify the 4-round i-impossible differentials $(e_i^{64}, e_{52}^{64})(i \in \mathbb{Z}_{32})$ of MISTY1, which is finished in the Full Version of our paper.

5.4 RC5

RC5 is designed by Rivest in 1994 [24]. The block size of it can be 32, 64, or 128 bits. For each block size n, the version is denoted as RC5-$n(n = 32, 64, 128)$.

Advantage of our tool. The operation variable rotation highly depends on the value of state, which cannot be handled by the previous automatic search tools

for impossible differentials. In our model, by exploiting the modeling method we proposed in Section 4.1, we give the first automatic method for searching the impossible differentials of RC5.

Configurations of our tool. The key schedule of RC5 is very complex. Thus, we focus on searching i-impossible differentials. By observing the structure of RC5-n, the difference $e^n_{(i,i+\frac{n}{2})}$ propagates to the difference $e^n_{(i+\frac{n}{2})}$ after 0.5-round in the encryption direction. Thus, we search the i-impossible differentials for RC5-$n(n = 32, 64, 128)$ by limiting the input difference and output difference in the set $(e^n_{(i,i+\frac{n}{2})}, e^n_{(j)})(0 \le i \le \frac{n}{2} - 1, 0 \le j \le n - 1)$.

List of 2.5-round i-impossible differentials. As a result, our tool found 12 i-impossible differentials for RC5-32, 27 i-impossible differentials for RC5-64, and 58 i-impossible differentials for RC5-128. This is the first result of impossible differentials for RC5. All the results are shown in Table 2.

Table 2. 2.5-Round i-impossible Differentials of RC5

Block Size	ΔP	ΔC	Number
32	$e^{32}_{(i,i+16)}(4 \le i \le 15)$	$e^{32}_{(15)}$	12
64	$e^{64}_{(i,i+32)}(5 \le i \le 31)$	$e^{64}_{(31)}$	27
128	$e^{128}_{(i,i+64)}(6 \le i \le 63)$	$e^{128}_{(63)}$	58

Manual verification for the i-impossible differential $(e^n_{(\frac{n}{2}-1,n-1)}, e^n_{(\frac{n}{2})-1})$ of RC5-n. First, we study the relation of a pair of input values and a pair of output values for the operation variable rotation, and have that the parity of $W(z \oplus w)$ is the same as $W(x \oplus u)$, where $z = x \lll y, w = u \lll v$, $x, y, z, u, v, w \in \mathbb{F}_2^m$. Then, we verify the 2.5-round i-impossible differential $(e^{32}_{(15,31)}, e^{32}_{(15)})$ of RC5-32, $(e^{64}_{(31,63)}, e^{64}_{(31)})$ of RC5-64, and $(e^{128}_{(63,127)}, e^{128}_{(63)})$ of RC5-128 together. The details of our manual process are shown in the Full Version of our paper.

6 Applications to Impossible Differentials from the Aspect of Design

In this section, we apply our tool to evaluate the security of lightweight block ciphers against the d-impossible differentials directly. For block ciphers with large S-boxes, we propose the three phases technique and inside value technique, which improve the security evaluation efficiency against the impossible differentials.

Three phases technique. For a block cipher, proving that all the input differences in Λ and output differences in Θ are the r-round possible differentials may be time-consuming. To overcome this dilemma, we pick two sets Φ and Ψ satisfied: for $\forall \alpha \in \Lambda$, there exists $\alpha_0 \in \Phi$ such that α can propagate to α_0 after r_1 rounds in the forward direction, and for $\forall \beta \in \Theta$, there exists $\beta_0 \in \Psi$ such that β can propagate to β_0 after r_2 rounds in the backward direction.

In this way, we just need to prove all the difference of the Φ and Ψ are the $(r - r_1 - r_2)$-round possible differentials.

Inside value technique. For a block cipher, proving (α, β) is an r-round i-possible (resp. d-possible) differential directly may be time-consuming. To solve this problem, we prove that $(0, \alpha)$ and $(0, \beta)$ is an i-possible (resp. d-possible) 2-polygon instead. Our experimental results show that this technique speeds up our proof process.

6.1 Direct Application to GIFT64, PRESENT, Midori64, PRINTcipher48, and PRINTcipher96

By exploiting our tool, we prove that, in the search space where the input difference activates only one S-box in the first substitution and the output difference activates only one S-box in the last substitution, there exists no 7-round, 7-round, 6-round, 5-round, and 6-round impossible differential for GIFT64, PRESENT, Midori64, PRINTcipher48, and PRINTcipher96 even considering the details of the key schedule.

6.2 Three Phases Technique: Apply to AES-128

AES-128 is the most famous standard block cipher designed by Vincent Rijmen and Joan Daemen [11]. It is a 128-bit block cipher with 128-bit key. AES-128 adopts the SPN structure. Its 128-bit internal state s can be represented as a 4×4 matrix of bytes $s_{i,j} \in \mathbb{F}_2^8$ $(0 \leq i, j \leq 3)$, each values in the finite fields $\mathbb{F}_2^8$. For more details of AES, please refer to [11].

Previous result. Wang el at. [30] have proved that there exists no 5-round 1 input active word and 1 output active word impossible differentials for AES-128 without the last MC operation even considering all the details of the S-box in the key independent setting. But, the influence of the key schedule for the impossible differentials about AES-128 is still unknown.

Our method. Determine whether a pair of input and output differences is the 5-round impossible differential by considering all the details of the relations of the round keys is very time-consuming. To resolve this issue, we adopt the three phases technique to finish our proof. First, according to the following two observations and further the propositions by studying the differential property of the S-box of AES, we propagate the input difference one round in the forward direction and the output difference two rounds in the backward direction. Then, we run our algorithm to show that those differences after the propagation can be connected through two rounds of AES even considering the relation of 3-round keys.

Obsetvation 3 *Let S denote the S-box of AES, define $DDT_{in}(\beta) = \{\alpha | \exists x \in \mathbb{F}_2^8, s.t. S(x) \oplus S(x \oplus \alpha) = \beta\}$, then we have $DDT_{in}(0x01) \cup DDT_{in}(0x02) \cup DDT_{in}(0xec) = \mathbb{F}_2^8$.*

Obsetvation 4 *Let S denote the S-box of AES, define $DDT_{out}(\alpha) = \{\beta | \exists x \in \mathbb{F}_2^8, s.t. \beta = S(x) \oplus S(x \oplus \alpha)\}$, then we have $DDT_{out}(0x01) \cup DDT_{out}(0x02) \cup DDT_{out}(0xf7) = \mathbb{F}_2^8$. Moreover, we have*

$$\{0x0d, 0x1a, 0xff\} = \{0x0d \times 0x01, 0x0d \times 0x02, 0x0d \times 0xf7\} \in DDT_{out}(0x01),$$
$$\{0x0b, 0x16, 0xfb\} = \{0x0b \times 0x01, 0x0b \times 0x02, 0x0b \times 0xf7\} \in DDT_{out}(0x03),$$
$$\{0x09, 0x12, 0x0e\} = \{0x09 \times 0x01, 0x09 \times 0x02, 0x09 \times 0xf7\} \in DDT_{out}(0x06),$$
$$\{0x0e, 0x1c, 0xfd\} = \{0x0e \times 0x01, 0x0e \times 0x02, 0x0e \times 0xf7\} \in DDT_{out}(0x09).$$

Proposition 2. *Let $F_1 = MC \circ SR \circ SB \circ ARK$, any difference $D_\alpha^{i,j}$ $(0 \le i \le 3, 0 \le j \le 3, \alpha \in \mathbb{F}_2^8/\{0\})$ can propagate to at least one of the differences of $MC \circ SR(D_{0x01}^{i,j})$, $MC \circ SR(D_{0x02}^{i,j})$, and $MC \circ SR(D_{0xec}^{i,j})$ through F_1.*

Proposition 3. *Let $F_2 = ARK \circ SR \circ SB \circ ARK \circ MC \circ SR \circ SB$ and*

$$P = \begin{pmatrix} 0x09 & 0x03 & 0x01 & 0x06 \\ 0x06 & 0x09 & 0x03 & 0x01 \\ 0x01 & 0x06 & 0x09 & 0x03 \\ 0x03 & 0x01 & 0x06 & 0x09 \end{pmatrix}.$$

Let $k = (j + i) \bmod 4$. Then, for any difference $D_\alpha^{i,j}$ $(0 \le i \le 3, 0 \le j \le 3, \alpha \in \mathbb{F}_2^8/\{0\})$, the difference $G_{i,j} := D_{P_{0,i}}^{0,k} + D_{P_{1,i}}^{1,(k+1)mod4} + D_{P_{2,i}}^{2,(k+2)mod4} + D_{P_{3,i}}^{3,(k+3)mod4}$ can propagate to it through F_2.

Proof. Let Q be the inverse matrix of the MDS used in AES[10]. According to Observation 4, for $\forall z \in \{0x01, 0x02, 0x7f\}$, we have $G_{i,j} \xrightarrow{SR \circ SB} D_{Q_{0,i} \times z}^{0,k} + D_{Q_{1,i} \times z}^{1,k} + D_{Q_{2,i} \times z}^{2,k} + D_{Q_{3,i} \times z}^{3,k}$, since the S-box is applied to each byte of the state in parallel in the SB operation. Then based on the definition of Q, we have $MC(D_{Q_{0,i} \times z}^{0,k} + D_{Q_{1,i} \times z}^{1,k} + D_{Q_{2,i} \times z}^{2,k} + D_{Q_{3,i} \times z}^{3,k}) = D_z^{i,k}$. According to Observation 4, for any difference $D_\alpha^{i,j}$ $(0 \le i \le 3, 0 \le j \le 3, \alpha \in \mathbb{F}_2^8/\{0\})$, at least one of $D_{0x01}^{i,k}, D_{0x02}^{i,k}$, and $D_{0x7f}^{i,k}$ can propagate to it through $SR \circ SB$. Thus, for any difference $D_\alpha^{i,j}$ $(0 \le i \le 3, 0 \le j \le 3, \alpha \in \mathbb{F}_2^8/\{0\})$, the difference $G_{i,j}$ can propagate to it through F_2. □

Our experiment. Let $F_3 = ARK \circ (MC \circ SR \circ SB \circ ARK)^2$. For $0 \le i, j, s, t \le 3$, by considering the relations of K_1, K_2, and K_3 according to the key schedule, we run our tool to determine whether all the differences of $MC \circ SR(D_{0x01}^{i,j})$, $MC \circ SR(D_{0x02}^{i,j})$, and $MC \circ SR(D_{0xec}^{i,j})$ can propagate to $G_{s,t}$ through F_3. After a total of $16 \times 16 \times 3 = 768$ tests, our result shows that all the differences of $MC \circ SR(D_{0x01}^{i,j})$, $MC \circ SR(D_{0x02}^{i,j})$, and $MC \circ SR(D_{0xec}^{i,j})$ can propagate to $G_{s,t}$ through F_3 in our setting, which leads to the following theorem.

[10]
$$Q = \begin{pmatrix} 0x0e & 0x0b & 0x0d & 0x09 \\ 0x09 & 0x0e & 0x0b & 0x0d \\ 0x0d & 0x09 & 0x0e & 0x0b \\ 0x0b & 0x0d & 0x09 & 0x0e \end{pmatrix}.$$

Theorem 6. *For* 5*-round AES-*128 *without the last MC operation, there exists no 1 input active word and 1 output active word impossible differentials by considering the relations of* K_1*,* K_2*, and* K_3*.*

6.3 Combination of Three Phases Technique and Inside Value Technique: Application to MISTY1

Previous result. Since MISTY1 adopts the 7-bit and 9-bit S-boxes, no automatic search tool could be used to evaluate its security taking account into the differential property of S-boxes so far.

Our approach. We combine the three phases technique and inside value technique to accelerate our tool in this part. Denote $\beta_0||\alpha_0$ be the 1 input active bit difference and $\beta_5||\alpha_5$ be the 1 output active bit difference, and $FO_{(KI,KO)}$ be the FO function, where KI and KO are the secret keys in the FO function. Let

$$\beta_1||\alpha_1 = \begin{cases} e^{64}_{i+32}, \text{ if } (\beta_0||\alpha_0) = e^{64}_i (0 \le i \le 31), \\ (FO_{0,0}(0) \oplus FO_{0,0}(e^{32}_{i-32}))||e^{32}_{i-32}, \text{ if } \beta_0||\alpha_0) = e^{64}_i (32 \le i \le 63). \end{cases}$$

$$\beta_4||\alpha_4 = \begin{cases} e^{32}_i||(FO_{0,0}(0) \oplus FO_{0,0}(e^{32}_i))e^{64}_{i+32}, \text{ if } (\beta_5||\alpha_5) = e^{64}_i (0 \le i \le 31), \\ e^{64}_{i-32}, \text{ if } \beta_5||\alpha_5) = e^{64}_i (32 \le i \le 63). \end{cases}$$

That is, we propagate the difference $\beta_0||\alpha_0$ through one round to $\beta_1||\alpha_1$ in the forward direction and the difference $\beta_5||\alpha_5$ through one round to $\beta_4||\alpha_4$ in the backward direction. Then, we prove that $(0, \beta_1||\alpha_1)$ and $(0, \beta_4||\alpha_4)$ is the i-possible 2-polygons.

Our experiment. We run our tool to determine whether the input 2-polygons $(0, \beta_1||\alpha_1)$ and the output 2-polygons $(0, \beta_4||\alpha_4)$ are the i-possible 2-polygons for 3 rounds MISTY1. After a total of $64 \times 64 = 4096$ tests, our result shows that all the input 2-polygons $(0, \beta_1||\alpha_1)$ and the output 2-polygons $(0, \beta_4||\alpha_4)$ are the i-possible 2-polygons for 3-round MISTY1, which leads to the following theorem.

Theorem 7. *For* 5*-round MISTY*1 *in which the FL layers were placed at the even rounds, there exists no 1 input active bit and 1 output active bit impossible differentials in the key independent setting.*

7 Applications to Impossible $(s+1)$-polytopic $(s \ge 2)$ Transitions

In this section, we run our tool to search the impossible $(s+1)$-polytopic$(s \ge 2)$ transitions for PRINTcipher, GIFT64, PRESENT, and RC5. All the contradictions of the distinguishers in this section can be detected by our verification algorithm, the details are shown in the Full Version of our paper in the supplementary materials. First, for S-boxes based block ciphers, we define some search spaces for the input and output s-differences.

Search space$_1$: In this space, the input 2-difference (b_1, b_2) is the $(1, 1)$ active bit which only activates the two right S-boxes in the first round, and the output 2-difference (e_1, e_2) is the $(1, 1)$ active bit.

Search space$_2$: In this space, the input 2-difference (b_1, b_2) is the $(1, 1)$ input active bit which only activates the first right S-box in the first round and the 2-difference (e_1, e_2) is the $(1, 1)$ output active bit which activates the same S-box in the last round.

Search space$_i$($i = 3, 4$): In this space, the input 3-difference is of pattern $(b_1, b_2, b_1 \oplus b_2)$ and the output 3-difference is of pattern $(e_1, e_2, e_1 \oplus e_2)$, where (b_1, b_2) and (e_1, e_2) are in Search space$_{i-2}$.

7.1 The *d*-impossible polytopic transitions of PRINTcipher

In this part, we show our method to search the impossible 3-polytopic transitions and impossible 4-polytopic transitions for PRINTcipher48 and PRINTcipher96 by considering all the details of the key schedule. Besides, we also study the influence of the Xor key and control key for the d-impossible 3-polytopic transitions of PRINTcipher48.

For the d-impossible 3-polytopic transitions of PRINTcipher48, we search such distinguishers in the Search space$_1$. After a total of $\binom{6}{1}_2 \times \binom{48}{1}_2 = 16920$ tests, the maximum number of rounds of d-impossible 3-polytopic transitions in this search space is 6, and a total of 1471 6-round d-impossible 3-polytopic transitions are found. One of them is

$$(0x000000000001, 0x000000010000) \overset{6-round}{\nrightarrow} (0x000000000002, 0x000000000200).$$

Impact of the constraints of the Xor keys. In our search above, we restrict the Xor keys and control keys according to the key schedule. To investigate the impact of the constraints of the Xor keys, we further release the constraints of the Xor keys and keep the constraints of the control keys. Then, we run our tool to search the 6-round impossible 3-polytopic transitions in Search space$_1$. Finally, we get 1448 6-round impossible 3-polytopic transitions. This result shows that, the constraint of the Xor keys leads to more contradictions for constructing the impossible 3-polytopic transitions.

Impact of the constraints of the control keys. Similarly, we keep the constraints of the Xor keys and release the constraints of the control keys over again. Then, we run our tool to search the 6-round impossible 3-polytopic transitions in Search space$_1$. Finally, we found that there exists no 6-round impossible 3-polytopic transitions in such search space. This result shows that the constraints of the control keys have a very significant impact on constructing the impossible 3-polytopic transitions.

Those two results show that, both the Xor keys and control keys may have influences on the results of impossible $(s+1)$-polytopic transitions. Thus, in the search of impossible $(s+1)$-polytopic transitions, we should consider the details of key schedule as much as possible if the time cost permits.

For the d-impossible 4-polytopic transitions of PRINTcipher48, we search such distinguishers in Search space$_3$. Finally, we found one 7-round d-impossible 4-polytopic transition of PRINTcipher48 as follows and stop our tool due to the limitation of search time.

$$(0x000000000001, 0x000000010000, 0x000000010001) \overset{7-round}{\nrightarrow} (0x000000000001, 0x000000000200, 0x000000000201).$$

For the d-impossible 3-polytopic transitions of PRINTcipher96, we search such distinguishers in Search space$_1$. Finally, we find one 7-round d-impossible 3-polytopic transition of PRINTcipher96 as follows and stop our tool due to the limitation of search time.

$$(0x000000000000000000000001, 0x000000000000000100000000) \overset{7-round}{\nrightarrow} (0x000000000000000000000001, 0x000000000000000008000000)$$

For the d-impossible 4-polytopic transitions of PRINTcipher96, we search such distinguishers in Search space$_3$. Finally, we find one 8-round d-impossible 4-polytopic transition of PRINTcipher96 as follows (as the left 48-bit of each value are 0, we only show the right 48 bits here) and stop our tool due to the limitation of search time.

$$(0x000000000001, 0x000100000000, 0x000100000001) \overset{8-round}{\nrightarrow} (0x000000000001, 0x000000000200, 0x000000000201).$$

7.2 The 7-round d-impossible 3-polytopic transition of GIFT64

For GIFT64, we search the d-impossible 3-polytopic transitions in Search space$_2$ Finally, we find one 7-round d-impossible 3-polytopic transition as follows and stop our tool due to the limitation of search time.

$$(0x0000000000000001, 0x0000000000000002) \overset{7-round}{\nrightarrow} (0x0000000000000001, 0x0000000000000008).$$

7.3 The 7-round i-impossible 4-polytopic transition of PRESENT

For the i-impossible 4-polytopic transitions of PRESENT, we search such distinguishers in Search space$_4$. Finally, we find one 7-round d-impossible 4-polytopic transition of PRESENT as follows and stop our tool due to the limitation of search time.

$$(0x0000000000000001, 0x0000000000000002, 0x0000000000000003) \overset{7-round}{\nrightarrow} (0x0000000000000001, 0x0000000000010000, 0x0000000000010001).$$

7.4 The 3-round i-impossible 3-polytopic transition of RC5-32 and RC5-64

In this subsection, we show our method for searching the i-impossible 3-polytopic transition of RC5-32 and RC5-64 by adopting the step by step strategy.

For RC5-32, since $(0x80008000, 0x00008000)$ is the 2.5-round impossible differential, we search the i-impossible 3-polytopic transitions by limiting the input 2-difference (b_1, b_2) in the set $\{(0x80008000, e^{32}_{i,i+16})|0 \leq i \leq 15\}$ and the output 2-difference (e_1, e_2) in the set $\{(0x00008000, e^{32}_i)|0 \leq i \leq 31\}$. Finally, we find 108 3-round i-impossible 3-polytopic transitions and result in that there exists no 3.5-round i-impossible 3-polytopic transitions in such search space. One of the transitions is

$$(0x80008000, 0x00100010) \overset{3-round}{\nrightarrow} (0x80000000, 0x00200000).$$

By adopting the same method for RC5-32, we find one 3-round i-impossible 3-polytopic transition as follows.

$$(0x8000000080000000, 0x0000002000000020) \overset{3-round}{\nrightarrow} (0x8000000000000000, 0x0000004000000000).$$

8 Conclusion

In this paper, we redefine the impossible differentials and impossible $(s+1)$-polytopic transitions based on the notation of s-polygon, and design a unity SAT-based automatic tool to search them. We apply our tool to various block ciphers. These results show that our tool can not only be used to search the distinguishers by considering the key schedule in the single-key setting, but also make the most of the inside property of large S-boxes or variable rotation for several typical classes of block ciphers.

Moreover, we derive an interesting result that, with the increase of the parameter s, the number of rounds in which the impossible $(s+1)$-polytopic transition exists also increases. Although due to the limitations of computing power, we can only search the impossible $(s+1)$-polytopic transition with a small value of s. But, the result indicates a challenge clearly that the impossible $(s+1)$-polytopic transition may bring threats for block ciphers with the development of the solver of the SAT and the computing power, and it is better to resist this kind of cryptanalysis in a theoretical way of cipher design.

Acknowledgements. We are very grateful to the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (No.61772517,61902030,61772516), Beijing Municipal Science & Technology Commission (No.Z191100007119004), and Youth Innovation Promotion Association CAS.

References

1. Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015.
2. Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.
3. Adnan Baysal and Sühap Sahin. Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*, volume 9542 of *Lecture Notes in Computer Science*, pages 58–76. Springer, 2015.
4. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999.
5. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.
6. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *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*, pages 450–466. Springer, 2007.
7. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.
8. Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 179–199. Springer, 2014.
9. Jiazhe Chen, Meiqin Wang, and Bart Preneel. Impossible differential cryptanalysis of the lightweight block ciphers tea, XTEA and HIGHT. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *Progress in Cryptology - AFRICACRYPT 2012, Ifrance, Morocco, July 10-12, 2012. Proceedings*, volume 7374 of *Lecture Notes in Computer Science*, pages 117–137. Springer, 2012.

10. Tingting Cui, Keting Jia, Kai Fu, Shiyao Chen, and Meiqin Wang. New automatic search tool for impossible differentials and zero-correlation linear approximations. *IACR Cryptology ePrint Archive*, 2016:689, 2016.
11. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.
12. Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 268–288. Springer, 2016.
13. Jongsung Kim, Seokhie Hong, Jaechul Sung, Changhoon Lee, and Sangjin Lee. Impossible differential cryptanalysis for block cipher structures. In Thomas Johansson and Subhamoy Maitra, editors, *Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings*, volume 2904 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 2003.
14. Lars Knudsen. Deal - a 128-bit block cipher. In *NIST AES Proposal*, 1998.
15. Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. Printcipher: A block cipher for ic-printing. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.
16. Fukang Liu, Takanori Isobe, and Willi Meier. Automatic verification of differential characteristics: Application to reduced gimli. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 219–248. Springer, 2020.
17. Yiyuan Luo, Xuejia Lai, Zhongming Wu, and Guang Gong. A unified method for finding impossible differentials of block cipher structures. *Inf. Sci.*, 263:211–220, 2014.
18. Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved impossible differential cryptanalysis of 7-round AES-128. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010, Hyderabad, India, December 12-15, 2010. Proceedings*, volume 6498 of *Lecture Notes in Computer Science*, pages 282–291. Springer, 2010.
19. Mitsuru Matsui. 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*, pages 386–397. Springer, 1993.
20. Mitsuru Matsui. New block encryption algorithm MISTY. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 1997.
21. Ilya Mironov and Lintao Zhang. Applications of SAT solvers to cryptanalysis of hash functions. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2006.

22. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *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*, pages 57–76. Springer, 2011.
23. Kaisa Nyberg and Lars R. Knudsen. Provable security against a differential attack. *J. Cryptology*, 8(1):27–37, 1995.
24. Ronald L. Rivest. The RC5 encryption algorithm. In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 86–96. Springer, 1994.
25. Ronald L. Rivest, Matthew J. B. Robshaw, and Yiqun Lisa Yin. RC6 as the AES. In *The Third Advanced Encryption Standard Candidate Conference, April 13-14, 2000, New York, New York, USA*, pages 337–342. National Institute of Standards and Technology,, 2000.
26. Sadegh Sadeghi, Vincent Rijmen, and Nasour Bagheri. Proposing an milp-based method for the experimental verification of difference trails. *IACR Cryptol. ePrint Arch.*, 2020:632, 2020.
27. Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 185–215, 2017.
28. Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *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*, pages 158–178. Springer, 2014.
29. Tyge Tiessen. Polytopic cryptanalysis. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 214–239. Springer, 2016.
30. Qian Wang and Chenhui Jin. Upper bound of the length of truncated impossible differentials for AES. *Des. Codes Cryptogr.*, 86(7):1541–1552, 2018.
31. Shengbao Wu and Mingsheng Wang. Automatic search of truncated impossible differentials for word-oriented block ciphers. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, volume 7668 of *Lecture Notes in Computer Science*, pages 283–302. Springer, 2012.