STP Models of Optimal Differential and Linear Trail for S-box Based Ciphers

Yu Liu[1,2], Huicong Liang[1], Muzhou Li[1], Luning Huang[1], Kai Hu[1], Chenhe Yang[1], and Meiqin Wang[1,3]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China
[2] School of Computer Engineering, Weifang University, Weifang 261061, China
mqwang@sdu.edu.cn

Abstract. Automatic tools have played an important role in designing new cryptographic primitives and evaluating the security of ciphers. Simple Theorem Prover constraint solver (STP) has been used to search for differential/linear trails of ciphers. This paper proposes general STP-based models searching for differential and linear trails with the optimal probability and correlation for S-box based ciphers. In order to get trails with the best probability or correlation for ciphers with arbitrary S-box, we give an efficient algorithm to describe probability or correlation of S-Box. Based on the algorithm we present a search model for optimal differential and linear trails, which is efficient for ciphers with S-Boxes whose DDTs/LATs contain entities not equal to the power of two. Meanwhile, the STP-based model for single-key impossible differentials considering key schedule is proposed, which traces the propagation of values from plaintext to ciphertext instead of propagations of differences. And we found that there is no 5-round AES-128 single-key truncated impossible differential considering key schedule, where input and output differences have only one active byte respectively. Finally, our proposed models are utilized to search for trails of bit-wise ciphers GIFT-128, DES, DESL and ICEBERG and word-wise ciphers ARIA, SM4 and SKINNY-128. As a result, improved results are presented in terms of the number of rounds or probabilities/correlations.

Keywords: STP · Differential trail· Linear trail· Bit-wise ciphers · Word-wise ciphers.

1 Introduction

Differential [9] and linear cryptanalysis [27] are the most popular cryptanalytic methods, which have been used to analyze numerous symmetric ciphers. The key point for these methods is to identify the differential or linear trail with high probability or correlation. To obtain better trails and escape from complicated manual work, automatic search tools have been widely used in cryptographic research. Early automatic tools are implemented from scratch in general purpose programming languages such as [6, 7, 28]. In recent years, automatic searching

tools are mainly based on Mixed-Integer Linear Programming (MILP), Boolean Satisfiability Problem (SAT)/Satisfiability Modulo Theories (SMT), and Constraint Programming (CP).

Table 1: Summary of Our Results on Bit-Oriented Ciphers

Cipher	Trail	Rounds	Probability/ Correlation	Reference
GIFT-128	Differential	9	$2^{-46.0}$	[4]
		9	$2^{-45.4}$ *	Sec. 6.1
		10	$2^{-49.4}$ *	Sec. 6.1
		11	$2^{-54.4}$ *	Sec. 6.1
		12	$2^{-60.4}$ *	Sec. 6.1
		13	$2^{-67.8}$ *	Sec. 6.1
		18	2^{-109}	[42]
		18	$2^{-103.4}$	Sec. 6.1
		21	$2^{-126.4}$	Sec. 6.1
ICEBERG	Linear	6	$2^{-30.1}$	[40]
		6	$2^{-30.0}$ *	Sec. 6.1
DES	RK Differential	4	$2^{-4.6}$	[7]
		4	$2^{-3.4}$ *	Sec. 6.1
		6	$2^{-12.9}$	[7]
		6	$2^{-12.2}$ *	Sec. 6.1
		7	$2^{-20.4}$	[7]
		7	$2^{-18.3}$	Sec. 6.1
DESL	RK Differential	4	$2^{-4.7}$	[7]
		4	$2^{-2.4}$ *	Sec. 6.1
		5	$2^{-7.2}$	[7]
		5	$2^{-5.6}$ *	Sec. 6.1
		6	$2^{-12.1}$	[7]
		6	$2^{-8.0}$ *	Sec. 6.1
		7	$2^{-20.0}$	[7]
		7	$2^{-12.2}$ *	Sec. 6.1
		8	$2^{-33.5}$	[38]
		8	$2^{-21.3}$	Sec. 6.1
		9	$2^{-41.9}$	[37]
		9	$2^{-31.5}$	Sec. 6.1
		10	$2^{-51.9}$	[37]
		10	$2^{-37.8}$	Sec. 6.1
		11	$< 2^{-31}$	[7]
		11	$2^{-51.7}$	Sec. 6.1

RK Differential: Related-Key Differential.
$*$: represents that the corresponding differential/linear trail is optimal.

MILP-based tools have been developed to automatically search for real differential and linear trails [13,29,38,39] and constructed to search for the impossible differentials and zero-correlation linear approximations [11, 32] for S-Box based

and ARX ciphers. These tools could express the whole Differential Distribution Table (DDT)/Linear Approximation Table (LAT) for the 4-bit S-Box [39]. For differential trails of ciphers with 8-bit S-Boxes, Abdelkhalek *et al.* presented MILP-based tools, which firstly search for word-wise truncated differential characteristics without describing the DDT of S-Box, and then with DDTs of S-Boxes find specific differential trails satisfying the identified truncated differential characteristics [1].

Table 2: Summary of Our Results on Word-Oriented Ciphers

Cipher	Trail	Rounds	Probability/ Correlation	Reference
ARIA	Linear	4	$2^{-49.2}$	[26]
		4	2^{-48} *	Sec. 6.2
		5	2^{-60}	[2]
		5	$2^{-52.6}$ *	Sec. 6.2
		6	2^{-72} *	Sec. 6.2
SM4	Differential	19	2^{-124}	[34]
		19	2^{-123}	Sec. 6.2
SKINNY-128	Impossible Differential	11	-	[5]
		12	-	Sec. 6.2

$*$: represents that the corresponding differential/linear trail is optimal.

Recently, the CP technique has been utilized to design a new general tool to search for (impossible) differential, (zero-correlation) linear trails and integral distinguishers [36]. Meanwhile, it has been applied to search for related-key differential trails for AES, Midori-64 and Midori-128 in [17, 18]. From [17, 18, 36], it is shown that CP-based tools are similar as MILP-based models that can also deal with ciphers with 8-bit S-Boxes, which also search for word-wise truncated differential characteristics without describing the DDT of S-Box first.

The SAT/SMT-based automatic tool uses SAT/SMT solvers to search for differential or linear trails by solving a SAT/SMT problem. The SAT/SMT solver has been used to design automatic searching tools to search for differential and linear trails [3, 21, 24, 30, 35].

For ciphers with 8-bit S-Boxes, it seems that the previous automatic search tools for differential and linear trails including MILP-based and CP-based models are efficient for word-wise ciphers. For bit-wise ciphers, there are so many truncated differential trails that it is infeasible to search for the optimal or long bit-based differential trails by verifying all those truncated differential trails. Another alternative method is to directly search for differential trails without pre-searching truncated differentials, which may reduce searching efficiency.

Besides, for ciphers whose DDTs/LATs contain entities not equal to the power of two (for convenience, we call this kind of DDT/LAT DDT*/LAT*), Abdelkhalek *et al.* [1] already have shown how to model the DDT* with probability, where they rounded the probability described as its negative log_2 at one

decimal place. And this method may miss some good differential trails. Although they claimed that they can increase the precision as much as they want, they have not considered how to set the precision to avoid missing the good differential trails. Searching for the optimal differential or linear trails is an important topic, *e.g.* Biryukov *et al.* have proposed the automatic search tool for the best different and linear trails of ARX ciphers at FSE'16 [8].

Since the problem of searching for differential/linear trails involves many XOR operations and STP [15] can model the XOR operation more easy than MILP, CP and SAT, we will construct STP-based automatic search models for optimal differential and linear trails, which will be suitable for ciphers even with DDT* or LAT*.

1.1 Our Contributions

In this paper, we aim to build automatic search models to search for optimal differential and linear trails for S-box based ciphers. Our contributions are described as follows.

- **STP-based model for differential and linear trails with optimal probability and correlation.** The direct way is to compute the probability and correlation of trails by multiplying the entity of the DDT and LAT for each S-Box, which is inefficient for STP solver. More efficient method is to represent the differential probability by its negative log_2, which has been used in [1]. For DDT* and LAT*, they rounded the probability described as its negative log_2 at one decimal place, so some good differential trails may be missed. We approximate the negative log_2 of probability by a number to have n_f decimal place of accuracy. We shall simply call this n_f the precision of probability. How to set the precision n_f to avoid missing the good differential trails has not be considered. So we offer an efficient algorithm which can determine the precision n_f in order to ensure that we can obtain the optimal trial with the best probability or correlation. With our decided precision we propose STP-based models for differential and linear trails with optimal probability and correlation for S-Box based ciphers even with DDT*/LAT*. Note that our proposed algorithm to determine the precision can also be used in MILP model [1] to find optimal trails. This model is illustrated in Section 4.
- **STP-based model for single-key impossible differentials with key schedule.** Although the previous MILP-based model [11, 32] for single-key impossible differential has considered the differential property for specific S-Boxes, it omits the key schedule and the obtained impossible differentials always hold for any key schedule. The previous models are possible to wrongly regard some impossible differentials as possible differentials, because of losing constraints from the key schedule. We will construct an STP-based model to search for the (truncated) single-key impossible differential considering the key schedule, where we trace propagations of values from plaintext to ciphertext instead of propagations of differences. As we know, it is the

first time to give a searching method for impossible differentials under the key schedule. We present this technique in Section 5.

– **Applications to bit-oriented block ciphers GIFT-128, DES, DESL and ICEBERG.**

For GIFT-128, we obtained optimal differential trails for $9 \sim 13$ rounds. Moreover, we show that 25-round is sufficient to achieve a differential probability lower than 2^{-128}, while the designers originally expected that 26 rounds were required [4]. Meanwhile, we got a 21-round differential trail, which is the best one according to the number of rounds compared with previous public trails. And we identified improved 9-round and 20-round differential trails according to the probability better than ones in [4, 42].

For DES, we obtained improved 4-, 6- and 7-round related-key differential trails, where 4- and 6-round related-key differential trails are optimal.

For DESL, our identified differential trails from 4 to 7 rounds are optimal and we got improved related-key differential trails for $4 \sim 10$ rounds compared with results from the MILP-based method in [37, 38]. Meanwhile, it is the first time that we got an 11-round related-key differential trail for DESL with the probability $2^{-51.7}$.

For ICEBERG, we got the optimal 6-round linear trail with correlation $2^{-30.0}$.

We compare our identified differential and linear trails with those of previous results in Table 1.

– **Applications to word-oriented block ciphers AES, ARIA, SM4 and SKINNY-128.**

Using our model for single-key impossible differentials with key schedule, we searched for 5-round AES-128 truncated impossible differentials, where input and output differences have only one active byte respectively. As a result, we found that there is no such truncated impossible differential for 5-round AES-128. It is the first time to consider single-key truncated impossible differentials with key schedule for AES-128.

We found 4-, 5- and 6-round ARIA linear trails with optimal correlation 2^{-48}, $2^{-52.6}$ and 2^{-72}, respectively, while the previous best linear trail only covers 5 rounds with correlation 2^{-60} [26].

For SM4, we got the optimal 20-round linear trails which has the same correlation as the one in [25] and identified an improved 19-round differential trail with the probability 2^{-123}.

We got 12-round impossible differentials for SKINNY-128 without considering key schedule. Compared with the previous result given in the specification of SKINNY [5], the impossible differential we obtained has one more round.

We compare our identified (impossible) differential and linear trails with those of previous results in Table 2.

2 Previous SMT/SAT-Based Works for Differential/Linear Trails

SAT solvers are used to solve the Boolean satisfiability problems and are based on heuristic algorithms. SMT is the problem of deciding satisfiability of a logical formula, expressed in a combination of first-order theories [10]. SMT solvers are more powerful than SAT solvers since they work on an higher abstraction layer. Differences through the round functions as constraints that can be processed by SMT/SAT solvers. An advantage of using SMT over SAT for the modeling is that most SMT solvers support reasoning over bit-vectors which are commonly used in block cipher designs, especially when considering word-oriented ciphers. And there are lots of efficient SMT solvers which have been used in various software verification and analysis applications.

In this paper, we use the SMT solver STP which encodes the constraints with CVC, SMT-LIB1 and SMT-LIB2 languages. STP aims at solving constraints of bit-vectors and arrays [16], so it suits for S-Box based ciphers.

The STP-based model of searching for differential trails of ARX ciphers was firstly introduced by Mouha and Preneel [30]. Liu *et al.* [24] gave equations of the linear propagation for the modular addition and constructed the STP-based model to search for linear trials of ARX ciphers. Moreover Kölbl *et al.* [21] searched for linear trails for SIMON based on CryptoSMT [20] and STP [14]. Ankele and Kölbl [3] investigated the differential gap between single characteristics and differentials for 4-bit S-Box based ciphers by CryptoSMT, where they can search for the best differential trails for ciphers with S-Boxes whose DDTs/LATs only contain entities equal to the power of 2.

Besides SMT-based models for differential and linear trails, SAT-based models have also been given by Sun *et al.* [35], where they computed the accurate differential probability for LED64 and Midori64 using CryptoMiniSAT.

In all, searching models based on SAT/SMT solvers for S-Box based ciphers have been involved in [3] and [35], where they extract all valid/invalid differential trails to encode them in CNF and just restrict to S-Boxes whose entities in the DDT only take values equal to the power of 2. However, DDTs/LATs of many used S-Boxes contain entities not equal to 2^i such as S-Boxes used in CLEFIA, CRYPTON, and DES *etc.* Even for popular AES-like S-Boxes, their LATs also belong to this type. In this paper, we aim to construct SMT-based models for ciphers with DDT*s/LAT*s to search for **best** differential and linear trails.

3 Models for Differential/Linear Trails with the Minimal Number of Active S-Boxes

Although the SAT/SMT searching problem for differential trails with the minimal number of active S-Boxes have been well implemented in [3] and [35], we will reconstruct the model since the minimal number of active S-Boxes will be used in our model for optimal differential/linear trails.

3.1 Model for Differential Trails

Firstly, we describe equations of XOR and branching operations for differential propagation [28]. Fig. 1 illustrates differential and linear propagations of XOR and branching operations.

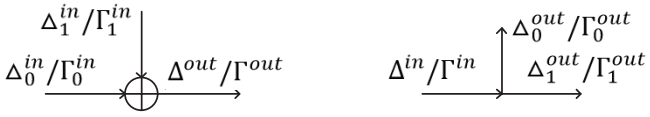

Fig. 1: XOR and Branching

Property 1 *For the XOR operation, denote Δ_0^{in}, Δ_1^{in} as input differences and denote Δ^{out} as the output difference, then the corresponding equation is $\Delta_0^{in} \oplus \Delta_1^{in} = \Delta^{out}$.*

Property 2 *For the branching operation, the input difference is written as Δ^{in} and output differences are written as Δ_0^{out} and Δ_1^{out}. The corresponding equation is $\Delta^{in} = \Delta_0^{out} = \Delta_1^{out}$.*

In order to search for differential trails with the minimal number of active S-Boxes, it is unnecessary to describe differential probabilities of S-Boxes in this model. We only focus on valid difference propagations through S-Boxes.

Property 3 *Assume that N S-Boxes are involved in the differential trail. For the i-th S-Box S_i, denote $\Delta_i^{in} \in \mathbb{F}_2^m$ and $\Delta_i^{out} \in \mathbb{F}_2^l$ as input and output differences, respectively. v_i represents the validity of the difference propagation, and could be written as*

$$v_i = \begin{cases} 0 \ (invalid) & if\ (\Delta_i^{in}, \Delta_i^{out}) \in \mathbb{S}_i^0,\ j = 0, \\ 1 \ (valid) & if\ (\Delta_i^{in}, \Delta_i^{out}) \in \mathbb{S}_i^j,\ 0 < j \leq 2^m, \end{cases}$$

where the set $\mathbb{S}_i^j$ $(1 \leq i \leq N)$ contains all pairs of input and output differences with the probability $j/2^m$.

To search for the valid differential trail, it is required to give functions $v_i = 1$ for all N S-Boxes. And the number of active S-Boxes needs setting an expected threshold.

If the differential for S-box $\alpha \rightarrow \beta$ is valid propagation, we describe it in CVC language,

$$\text{ASSERT}((\ \triangle_i^{in} = \alpha\ \&\&\ \triangle_i^{out} = \beta\ ) => \ (v_i = 1)),$$

otherwise, we present it in CVC language,

$$\text{ASSERT}((\ \triangle_i^{in} = \alpha\ \&\&\ \triangle_i^{out} = \beta\ ) => \ (v_i = 0)).$$

To avoid finding the trivial trail, set the input difference of the first round as non-zero in the single-key setting, while set the difference of master key as non-zero in the related-key setting. Algorithm 1 illustrates the whole process of searching for a differential trail with the expected number of active S-Boxes.

With the purpose of obtaining differential trails with the minimal number of active S-Boxes, we could proceed Algorithm 1 by adjusting the expected threshold repeatedly.

Algorithm 1 Model for Differential Trails with the Expected Number of Active S-Boxes

Input: The number of rounds r, Expected threshold, Flag. /*If Flag = 1, search the trail in the single-key setting; If Flag = 0, search the related-key differential trail.*/
Output: A differential trail with the expected number of active S-Boxes.

for round $\leftarrow$ 1 to r **do**
    Represent equations about S-Boxes with Property 3.
    Illustrate equations about linear operations with Property 1,2.
if Flag = 1 **then**
    Describe equations, which set input differences as non-zero.
else
    Describe equations, which set differences of the master key as non-zero.
    List equations corresponding to the key schedule.
Give equations to set the number of S-Boxes less than the expected threshold.
Input all above equations into STP and solve them.

3.2 Model for Linear Trails

Firstly, we describe equations of XOR and branching operations for linear propagations [28].

Property 4 *Suppose that Γ_0^{in}, Γ_1^{in} are input masks and Γ^{out} is the output mask of the XOR operation. Then the equation is $\Gamma_0^{in} = \Gamma_1^{in} = \Gamma^{out}$.*

Property 5 *Suppose that Γ^{in} is the input mask, Γ_0^{out} and Γ_1^{out} are output masks of the branching operation. Then the equation is $\Gamma^{in} = \Gamma_0^{out} \oplus \Gamma_1^{out}$.*

Due to the duality between differential and linear propagations, the model for finding linear trails is similar to it for finding differential trails. We offer a simple description about equations for S-Boxes.

Property 6 *Assume that N S-Boxes are involved in the linear trail. For the i-th S-Box S_i $(0 \leq i < N)$, denote $\Gamma_i^{in} \in \mathbb{F}_2^m$ and $\Gamma_i^{out} \in \mathbb{F}_2^l$ as input and output*

masks, respectively. u_i *represents the validity of the linear propagation, and could be written as*

$$u_i = \begin{cases} 0\ (invalid) & if\ (\Gamma_i^{in}, \Gamma_i^{out}) \in \mathbb{T}_i^0,\ j = 0, \\ 1\ (valid) & if\ (\Gamma_i^{in}, \Gamma_i^{out}) \in \mathbb{T}_i^j,\ 0 < j \leq 2^{m-1}, \end{cases}$$

where the set $\mathbb{T}_i^j$ $(1 \leq i \leq N)$ *contains all pairs of input and output masks with the correlation* $\pm j/2^{m-1}$.

The function $u_i = 1$ needs writing in this model to obtain the valid linear trail. With the purpose of searching for linear trails with the minimal number of active S-Boxes, we could proceed the similar algorithm as Algorithm 1 and execute it repeatedly by changing the expected threshold.

4 Models for Differential/Linear Trails with Optimal Probability/Correlation

Models we give in Section 3 only focuses on finding trails with the optimal number of active S-Boxes. However, trails with optimal probability or correlation may not have the minimal number of active S-Boxes. Hence, we could miss some better trails. For example, for the 6-round related-key differential trail of DES, the minimal number of active S-Boxes is 4 , while actually the number of active S-Boxes is 5 for the trail with optimal probability.

In this section, models to search for differential and linear trails with optimal probability and correlation are presented, which are also suitable for ciphers with DDT*s/LAT*s. How to efficiently represent the probabilities and correlations of the differential and linear trails for such ciphers is the most important step in this model. The direct way is to compute the probability and correlation of trails by multiplying the entity of the DDT and LAT for each S-Box. However, from some experiments, we found that it is inefficient to compute the probability and correlation in this way by STP. These experiments will be presented in Section 4.4.

Note that Abdelkhalek *et al.* [1] already have shown how to model the DDT* with probability, where they rounded the probability described as its negative log_2 at one decimal place. And this method may miss some good differential trails. Although they claimed that they can increase the precision as much as they want, they have not considered how to set the precision to avoid missing the good differential trails.

In this section, we will give a method to decide the precision for probabilities and correlations in order to get the optimal trails.

4.1 Efficient Method for Computing the Probability

Suppose the total number of S-Boxes (including active and non-active S-Boxes) in the differential trail is N. And the number of S-Boxes with the probability

$j/2^m$ is N_j, where $2 \leq j \leq 2^m$ and j is even. It is obvious that $N = \sum_{j=2}^{2^m} N_j$. Under the Markov cipher assumption, the probability p of this differential trail is

$$p = \frac{(2^m)^{N_{2^m}} \times \ldots \times 6^{N_6} \times 4^{N_4} \times 2^{N_2}}{(2^m)^N},$$

which could be written as a general form

$$\begin{aligned} p =& f_p(N_{2^m}, N_{2^m-2}, \ldots, N_2) \\ =& (2^m/2^m)^{N_{2^m}} \times ((2^m-2)/2^m)^{N_{2^m-2}} \times \ldots \times (4/2^m)^{N_4} \times (2/2^m)^{N_2}. \end{aligned} \tag{1}$$

To compute the probability p, an approximation function G^* of $f_p(N_{2^m}, N_{2^m-2}, \ldots, N_2)$ is built.

First, compute the negative logarithm of the function f_p and obtain

$$\begin{aligned} g^*(N_{2^m}, N_{2^m-2}, \ldots, N_2) &= -\log_2 f_p(N_{2^m}, N_{2^m-2}, \ldots, N_2) \\ &= -\sum_{\substack{2 \leq j \leq 2^m \\ j \text{ is even}}} \sum_{\substack{k=1 \\ N_j \neq 0}}^{N_j} (\log_2 j - m). \end{aligned}$$

The function g^* is only composed of addition operations, so the efficiency of computations is improved. Obviously, the probability of the differential trail is equal to 2^{-g^*}. To avoid modelling the probability of a trail by multiplying the differential probabilities for each active S-box, the main idea here is to represent the differential probabilities by its negative log_2, and round the involved irrational values of the logarithms to the required precision such that the monotony is preserved.

It is the fact that the value of $\log_2 j$ is either an integer or an irrational decimal in the function g^*. We regard the situation that all values of $\log_2 j$ are integers as a special case. In this special case, the function g^* is computed by the addition of integers, which could be easily implemented by STP. Unfortunately, in most cases, not all $\log_2 j$ are integers. And STP is defined in the bit-vector theory, which only supports bit-vector variables and common bit-vector operations, such as XOR, AND, and Modular Addition *etc* [14]. Therefore STP's input language has no floating point datatypes or operations.

For coping with this situation, our strategy is to transform $\log_2 j$ into integers by multiplying 10^{n_f} to g^*, where n_f is a positive integer and called the precision of probability. Then take the integral part by the ceiling function to obtain the approximation function G^*, where

$$G^*(N_{2^m}, N_{2^m-2}, \ldots, N_2) = -\sum_{\substack{2 \leq j \leq 2^m \\ j \text{ is even}}} \sum_{\substack{k=1 \\ N_j \neq 0}}^{N_j} \lceil (\log_2 j - m) \times 10^{n_f} \rceil. \tag{2}$$

Due to the ceiling function, with the increase of the approximation function G^*, the probability will not always decrease, and vice versa. In this way, the

approximation function G^* could not be used to find the optimal trail with the maximal probability because of the possibility of missing better trails. In order to fix this problem, we have to elaborately choose the value of n_f in G^* to ensure that the following property holds.

Property 7 *For any two* $(2^{m-1}-1)$*-tuples* $\{N_{2^m}, N_{2^m-2}, \ldots, N_2\}$ *and* $\{N'_{2^m}, N'_{2^m-2}, \ldots, N'_2\}$*, if* $f_p(N_{2^m}, N_{2^m-2}, \ldots, N_2) > f_p(N'_{2^m}, N'_{2^m-2}, \ldots, N'_2)$*, then* $G^*(N_{2^m}, N_{2^m-2}, \ldots, N_2) < G^*(N'_{2^m}, N'_{2^m-2}, \ldots, N'_2)$.

We give Algorithm 2 to choose the value of n_f to make G^* satisfy Property 7.

In Algorithm 2, a list T_N is built first, which stores values of f_p and corresponding $(2^{m-1}-1)$-tuple $\{N_{2^m-2}, \ldots, N_4, N_2\}$. Then sort the list T_N by multiple keywords f_p, N_{2^m-2}, $\cdots$, N_4, N_2 in ascending order. Initialize n_f as 1. And compute values of the approximation function G^* with corresponding tuples in T_N as inputs. Check whether G^* satisfies Property 7 or not. If so, Algorithm 2 will return the value of n_f; otherwise, change the value of n_f to n_f+1 and recompute G^*. Let the amounts of value of non-zero entities in DDT expect 2^m be M and the number of loops for computing n_f is b. Usually $b < 20$. The time of step 2-6 is $\binom{M+N_s-1}{N_s}$ times of computations of f_p. The running time of step 8-17 is $2b \times \binom{M+N_s-1}{N_s}$ times of computations of G^*.

This method to compute the probability of a differential trail is suitable for any automatic tools besides STP, especially for ones that seem not perform well for decimal operations or multiplication.

4.2 Model for Differential Trails

The complete model for differential trails is described in this subsection.

Property 8 *Assume that N S-Boxes are involved in the differential trail. For the i-th S-Box* S_i*, denote* $\Delta_i^{in} \in \mathbb{F}_2^m$ *and* $\Delta_i^{out} \in \mathbb{F}_2^l$ *as input and output differences, respectively.* v_i *is the flag variable to represent the validity of the difference propagation, where*

$$v_i = \begin{cases} 0\ (invalid) & \text{if } (\Delta_i^{in}, \Delta_i^{out}) \in \mathbb{S}_i^j,\ j = 0, \\ 1\ (valid) & \text{if } (\Delta_i^{in}, \Delta_i^{out}) \in \mathbb{S}_i^j,\ 0 < j \le 2^m. \end{cases}$$

The probability of S_i *can be represented by* p_i*, and* p_i *is defined as follows.*

$$p_i = \begin{cases} 0 & \text{if } \Delta_i^{in} = 0, \\ c_j^* & \text{if } (\Delta_i^{in}, \Delta_i^{out}) \in \mathbb{S}_i^j,\ 0 < j < 2^m, \end{cases}$$

where $c_j^* = -\lceil (\log_2 j - m) \times 10^{n_f} \rceil$ *and the set* $\mathbb{S}_i^j$ $(1 \le i \le N)$ *contains all pairs of input and output differences with the probability* $j/2^m$.

To keep the differential trail valid, the equation $v_i = 1$ needs to be appended. In addition, it is essential to set the value of approximation function $G^* =$

Algorithm 2 Algorithm to Calculate n_f for Given N_S and DDTs

Input: The number of active S-Boxes N_S, DDTs.
Output: $n_f \in \mathbb{Z}^+$.
Data:
T_N: a list storing 2^{m-1}-tuples $\{f_p, N_{2^m-2}, \ldots, N_4, N_2\}$;
V_{count}: the number of rows in the list T_N;
$f_p[i]$: the value of the i-th f_p in the list T_N;
$G^*[i]$: the value of the approximation function corresponding to the i-th $(2^{m-1}-1)$-tuples $\{N_{2^m-2}, \ldots, N_4, N_2\}$ in T_N.

1: $V_{count} \leftarrow 0$
2: **for** all possible values of $(2^{m-1}-1)$-tuples $\{N_{2^m-2}, \ldots, N_4, N_2\}$ **do**
3: $\quad f_p \leftarrow \frac{1}{2^{N_S \cdot pm}} \cdot \prod_{k=1}^{2^{m-1}-1} (2k)^{N_{2k}}$
4: $\quad$ insert $\{f_p, N_{2^m-2}, \ldots, N_4, N_2\}$ into T_N
5: $\quad V_{count}++$
6: sort T_N indexed by multiple keywords $f_p, N_{2^m-2}, \ldots, N_4, N_2$ in ascending order
7: $n_f \leftarrow 1$
8: $G^*[1] \leftarrow -\sum_{k=1}^{2^{m-1}-1}(N_{2k} \times \lceil(\log_2 2k - m) \times 10^{n_f}\rceil)$ corresponding to the 1-st $(2^{m-1}-1)$-tuple in T_N
9: **for** $i \leftarrow 2$ to V_{count} **do**
10: $\quad G^*[i] \leftarrow -\sum_{k=1}^{2^{m-1}-1}(N_{2k} \times \lceil(\log_2 2k - m) \times 10^{n_f}\rceil)$ corresponding to the i-th $(2^{m-1}-1)$-tuple in T_N
11: $\quad$ **if** $G^*[i] > G^*[i-1]$ **then**
12: $\quad\quad$ **continue**
13: $\quad$ **else if** $G^*[i] = G^*[i-1]$ and $f_p[i] = f_p[i-1]$ **then**
14: $\quad\quad$ **continue**
15: $\quad$ **else**
16: $\quad\quad n_f++$
17: $\quad\quad$ goto Line 8
return n_f

$\sum_{i=1}^{N} p_i$ no more than the expected threshold G^*_{th} and the number of active S-Boxes as N_S.

Further, when applying our model in the single-key setting, set the input difference of the first round as nonzero in order to avoid trivial trails. And set the difference of the master key as nonzero in the related-key setting.

Input all equations into STP and solve them. If it outputs a trail, we can gradually reduce the value of G^*_{th} and run STP again to get the differential trail with optimal probability.

We present Algorithm 3 to search for the differential trail with the expected probability. In addition, a general procedure to search for differential trails with optimal probability is depicted in Algorithm 4. In some cases, we can only obtain an optimized trail instead of the optimal one due to the unpractical time complexity of Algorithm 4.

Algorithm 3 Searching Algorithm for the Differential Trail with the Expected Probability

Input: The number of rounds r, Expected threshold G^*_{th}, N_S, Flag. /*If Flag=1, search the trail in the single-key setting; If Flag=0, search the related-key differential trail.*/

Output: A differential trail with the probability less than G^*_{th}.

1: **for** round $\leftarrow 1$ to r **do**
2: Represent equations about S-Boxes with Property 8.
3: Illustrate equations about linear operations with Property 1 and Property 2.
4: **if** Flag=1 **then**
5: Describe equations, which sets input differences as non-zero.
6: **else**
7: Describe equations, which sets differences of the master key as non-zero.
8: List equations corresponding to the key schedule.
9: Give equations to set the value of G^* less than the expected threshold G^*_{th}.
10: Give equations to set the number of active S-Boxes as N_S.
11: Input all above equations into STP and solve them.

4.3 Model for Linear Trails

In a similar way, the approximation function G^{**} for computing the absolute correlation of trails is given below.

$$G^{**}(N_{2^{m-1}}, N_{2^{m-1}-2}, \ldots, N_2) = -\sum_{\substack{2\le j\le 2^{m-1}\\ j \text{ is even}}} \sum_{\substack{k=1\\ N_j\neq 0}}^{N_j} \lceil (\log_2 j - m + 1) \times 10^{n_f} \rceil.$$

Meanwhile, the value of n_f in the approximation function G^{**} should be elaborately set due to the same reason as that in the function G^*.

Property 9 *Assume that N S-Boxes are involved in the linear trail. For the i-th S-Box S_i, denote input and output masks as $\Gamma_i^{in} \in \mathbb{F}_2^m$ and $\Gamma_i^{out} \in \mathbb{F}_2^l$, respectively. u_i is the flag variable to represent the validity of the linear mask propagation, where*

$$u_i = \begin{cases} 0\ (invalid) & \text{if } (\Gamma_i^{in}, \Gamma_i^{out}) \in \mathbb{T}_i^j,\ j = 0, \\ 1\ (valid) & \text{if } (\Gamma_i^{in}, \Gamma_i^{out}) \in \mathbb{T}_i^j,\ 0 < j \le 2^{m-1}. \end{cases}$$

The absolute correlation of S_i can be represented by C_i, and C_i is defined as follows.

$$C_i = \begin{cases} 0 & \text{if } \Gamma_i^{out} = 0, \\ c_j^{**} & \text{if } (\Gamma_i^{in}, \Gamma_i^{out}) \in \mathbb{T}_i^j,\ 0 < j < 2^{m-1}, \end{cases}$$

where $c_j^{**} = -\lceil (\log_2 j - m + 1) \times 10^{n_f} \rceil$ and the set $\mathbb{T}_i^j$ $(1 \le i \le N)$ contains all pairs of input and output masks with the absolute correlation $\pm j/2^{m-1}$.

Algorithm 4 General Searching Procedure for Differential Trails with Optimal Probability

Input: The number of rounds r, P_{max}
Output: A differential trail with optimal probability.
Data: P_{max} represents the maximal probability of all DDTs.

1: Proceed Algorithm 1 with STP to obtain the differential trail with the minimal number of active S-Boxes N_s.
2: Compute the probability of this differential trail, denoted as p.
3: Store this trail in list L.
4: $t \leftarrow 0$
5: Set the number of active S-Boxes as $N_s + t$, and then compute the value of n_f with Algorithm 2.
6: Gradually reduce the value of G^*_{th} and proceed Algorithm 3 to find the optimal trail with $N_s + t$ active S-Boxes.
7: Compute the probability of this differential trail denoted as p'.
8: **if** $p' > p$ **then**
9: $p \leftarrow p'$
10: Use this trail update L.
11: $t++$
12: **if** $(P_{max})^{N_S+t} > p$ **then**
13: **goto** Line 5
14: **else**
15: **return** L and p

To keep the linear trail valid, the equation $u_i = 1$ needs to be included. Meanwhile, equations about setting the value of approximation function $G^{**} = \sum_{i=1}^{N} C_i$ no more than the expected threshold G^{**}_{th} are required as well. The number of active S-Boxes is set as N_S.

In order to avoid trivial trails, it is essential to set the input mask of the first round as non-zero.

Proceed the searching process similar with Algorithm 3 and Algorithm 4, then the linear trail with optimal or optimized absolute correlation could be obtained.

4.4 Efficiency of Approximation Functions and Multiplication

As we described before, we could compute the probability/correlation of trails by two methods. One is to multiply entities of DDTs/LATs, and the other is using the approximation function G^*/G^{**}. We did some experiments on ARIA and DES on one PC with four Intel(R) Core(TM) i5 CPUs (3.20GHz) to compare the running time between those two methods.

Results are shown in Table 3. It is clear to see that the process of computing by the approximation function is much more efficient. And for multiplication, it will be impossible to gain a trail for long rounds.

Table 3: Comparison of Running Time

Cipher	Trail	Round	Time (mins)	
			G^*/G^{**}	Multiplication
ARIA	Linear	3	21	1241
		5	221	terminated after 430 minutes $*$
DES	Related-key differential	6	235	>6720

$*$: STP printed the error message: memory manager can't handle the load.

5 Models for Impossible Differentials under Key-Schedule

A direct way to construct STP-based models of searching for impossible differentials and zero-correlation linear trails is to use equations for S-Boxes in Property 3 and Property 6. And we specify input and output differences or masks for ciphers. Based on these equations, we let STP solve the model. If STP reveals that there is no solution, it means that the differential or linear trail with the given input and output differences or masks is an impossible differential or zero-correlation linear trail.

Although the above model for impossible differential has considered the differential property for specific S-Boxes, it assumes that round keys are independent and uniformly random. However, this assumption doesn't hold due to the existence of the key schedule. In original models for searching for single-key impossible differential, the key schedule is omitted and the obtained impossible differentials always hold for any key schedule. The original models are possible to wrongly regard some impossible differentials as possible differentials, because of losing constraints from the key schedule.

Related work about impossible differentials under the key schedule has been involved in [41], where it is proved that there is no truncated impossible differential for more than 4-round AES-256 considering the key schedule. However, their method cannot be used for AES-128 and AES-192. The interesting question is whether there is a truncated impossible differential for 5-round AES-128 or AES-192 considering the key schedule.

How to search for impossible differentials under the key schedule has not been considered as we know. In this section, we will build an STP-based model to search for (truncated) impossible differentials under the key schedule. In this model, we have to trace propagations of values from plaintext to ciphertext instead of propagations of differences. Therefore, original models couldn't be applied.

Fig. 2 illustrates the idea of this model. Suppose that E_K is an encryption algorithm comprised of r rounds, which includes the addition of round keys, linear operations and S-Boxes in each round. And K is the master key. The round key of E_K is produced by the key schedule. For plaintexts P_1 and P_2,

corresponding ciphertexts are $C_1 = E_K(P_1)$ and $C_2 = E_K(P_2)$. Denote Δ^{in} as the plaintext difference and Δ^{out} as the ciphertext difference of E_K.

Equations about the addition of the round key and S-Boxes are illustrated as follows.

Property 10 *Denote X_1 and X_2 as the input and output of the addition of round key K_i. Then we have $X_2 = X_1 + K_i$, where $+$ is either the XOR operation or the modular addition.*

Property 11 *Denote Y_1 and Y_2 as values of the input and output of the S-Box S_i. Then we have $Y_2 = S_i(Y_1)$.*

First, set Δ^{in} and Δ^{out} as specific values. For any plaintext pair $(P_1, P_2 = P_1 \oplus \Delta^{in})$ and any master key K, the corresponding ciphertext pair is (C_1, C_2). If we can find one plaintext-ciphertext pair satisfying the given difference values Δ^{in} and Δ^{out} under a master key value, it means that we identify a valid differential pair, and the differential with the input difference Δ^{in} and the output difference Δ^{out} is possible. If $C_1 \oplus C_2 \neq \Delta^{out}$ for all possible values of P_1 and K, an impossible differential is obtained.

Describe equations from the encryption algorithm with Property 10 and Property 11, then input all required equations in this model into STP and let it solve. If STP returns *unsatisfiable*, it shows that the differential pair $(\Delta^{in}, \Delta^{out})$ is impossible. Otherwise, the solved plaintext-ciphertext pair together with the value of K is a valid pair for the differential pair $(\Delta^{in}, \Delta^{out})$ under K. To identify the impossible differential, values of Δ^{in} and Δ^{out} should be changed and put all equations into STP again until STP returns *unsatisfiable*.

The complete model is presented in Algorithm 5. Note that, this model could be used to find truncated impossible differentials or related-key impossible differentials as well.

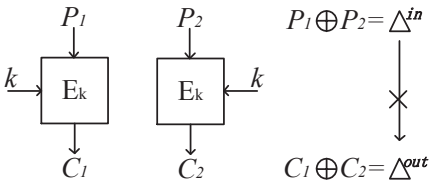

Fig. 2: Impossible Differential with Key Schedule

6 Applications

In this section, we apply our models to some S-Box based ciphers including bit-oriented ciphers such as GIFT-128, DES, DESL and ICEBERG and word-oriented ones such as AES, SM4, ARIA and SKINNY-128. Compared with previous results, we obtain improved differential and linear trails according to the

Algorithm 5 Model for Impossible Differential under Key Schedule

Input: The number of rounds r, Plaintext difference Δ^{in}, Ciphertext difference Δ^{out}.
Output: A possible differential trail or *unsatisfiable*.
/* If STP outputs invalid, an impossible differential could be obtained. */
1: Represent equations, which compute round keys by the key schedule.
2: $P_1 \oplus P_2 = \Delta^{in}$.
3: $C_1 \oplus C_2 = \Delta^{out}$.
4: **for** $i \leftarrow 1$ to r **do**
5: Illustrate equations about the addition of round keys with Property 10.
6: Describe equations about linear operations with Property 1 and Property 2.
7: Give equations about S-Boxes with Property 11.
8: Input all above equations into STP and solve them.

number of rounds or the probability and correlation. In some cases, we obtain the differential or linear trail with optimal probability or correlation. Our running environment is a cluster of computers with two Intel(R) Xeon(R) E5-2690 CPU (2.60GHz, 128G memory, 24 cores).

6.1 Applications to Bit-Oriented Block Ciphers

Differential Trails for GIFT-128. GIFT is a lightweight SPN block cipher proposed at CHES'17 [4], which has two versions GIFT-64 and GIFT-128. The substitution layer is made up of 4-bit S-Boxes and GIFT is a bit-oriented algorithm. We focus on GIFT-128, whose block size and key size are 128-bit. The specification [4] presented a 9-round differential trail with the probability $2^{-46.0}$ and pointed out that the probability of 26-round differential trails cannot be larger than 2^{-127}. In [42], an 18-round differential trail with the probability 2^{-109} has been identified.

We obtained optimal differential trails for $9 \sim 13$ rounds. And our identified 9-round differential trail with the probability $2^{-45.4}$ is better than the one in the specification [4]. Meanwhile, we got a 21-round differential trail with the probability $2^{-126.4}$, which is the longest differential trail until now. We found an improved 18-round differential trail with the probability $2^{-103.4}$, which is better than the one in [42]. Moreover, we give a tighter security bound of GIFT-128 against differential cryptanalysis. The probabilities of our optimal 12-round and 13-round differential trails are $2^{-60.4}$ and $2^{-67.8}$, respectively, which ensures that 25-round is sufficient to achieve a differential probability lower than 2^{-128}, while the designers originally expected that 26 rounds were required [4]. Details of 12-, 13- and 21-round differential trails are illustrated in Table 5, 6 and 7.

Related-Key Differential Trails for DES and DESL. DES [31] is a 16-round Feistel network cipher with block size 64 bits and key size 56 bits. The DESL [23], based on the classical DES, was proposed by Leander *et al.* The main difference between DES and DESL lies in the round function. DESL uses

eight same S-boxes instead of eight different ones in DES and omits the initial permutation and its inverse. In [28], Matsui introduced a practical algorithm for deriving the best differential trail of DES. But the search algorithm developed by Matsui only targeted at the single-key setting. Then in [7], Biryukov and Nikolić presented an algorithm to find reduced-round related-key differential characteristics from 3 to 16 rounds for DES and DESL. They found the best related-key trails of DESL for up to 7 rounds and upper bounds for more than 7 rounds. In [39], Sun *et al.* found a related-key differential trail for 8-round DESL with probability $2^{-33.5}$. In [37], they also provided related-key differential trials for 9-round and 10-round DESL with probability $2^{-41.9}$ and $2^{-51.9}$, respectively.

We choose different positions of the starting round to find the optimized related-key differential trails for DES and DESL. We obtained improved related-key differential trails for 4-, 6- and 7-round DES depicted in Table 8. Moreover, we found that 4-round and 6-round related-key differential trails for DES are optimal.

For DESL, we attained improved related-key differential trails for $4 \sim 10$ rounds compared with previous results in [7, 37, 39]. And an 11-round related-key differential trail of DESL was found with probability $2^{-51.7}$. Note that we show that our identified trails from 4 to 7 rounds are optimal. We list our trails in Table 9.

The running time about searching for trails of DES is several minutes.

Linear Trails for ICEBERG. ICEBERG, proposed at FSE 2004, is a block cipher with block size 64 bits and key size 128 bits [33]. Sun *et al.* identified a linear approximation with correlation $2^{-30.1}$ and presented key recovery attacks to 7-round ICEBERG [40].

We regard the non-linear layer as eight same 8-bit S-Boxes. At first, by the model in Section 3, we gained that the minimal number of active S-Boxes for 6-round linear trails of ICEBERG is 12. The we use the model in Section 4.3 to find the linear trails for ICEBERG. Finally, we identified the linear trail for 6-round ICEBERG with optimal correlation $2^{-30.0}$ listed in Table 10, which is slightly better than the previous result in [40].

6.2 Applications to Word-Oriented Block Ciphers

Truncated Impossible Differentials for AES-128 under Key Schedule. The existence of 5-round truncated impossible differentials of AES-128 under key schedule is an unsolved problem. It is unpractical to solve this problem by searching for all truncated impossible differentials for AES-128 under key schedule using the model in Section 5, because the total number of truncated differentials to be checked is 2^{32}. Intuitively, truncated differentials with only one active byte for input and output differences respectively are more possible to be impossible differentials. Thus, we focus on searching 5-round AES-128 truncated impossible differentials with key schedule, where input and output differences

have only one active byte respectively. As a result, there is no truncated impossible differential satisfying the above condition. The average running time for checking one truncated differential is around 480 minutes.

Moreover, we found that 5-round truncated impossible differentials, which have the input difference with any number of active bytes and the output difference with 16 active bytes, don't exit.

Linear Trails for ARIA. ARIA, an iterative 128-bit block cipher, follows the SPN structure [22] and it has been adopted by some famous standard protocols such as IETF (RFC 5794), SSL/TLS (RFC 6209) and PKCS #11. For reduced-round ARIA, we could see that the linear attack is better than the differential one from previous attacks. So we focus on searching for linear trails here. The best known linear trail covers 4-round ARIA with correlation $2^{-49.15}$, which was given by Liu *et al.* in [26]. And Abdelkhalek *et al.* presented a 5-round linear hull with correlation $2^{-57.5}$, where the best linear trail has correlation 2^{-60} [2].

Firstly, we used the model in Section 3 to obtain that the minimal number of active S-Boxes is 17 for 5-round ARIA. Then by setting the number of active S-Boxes as 17, we can get $n_f = 7$ using the similar algorithm as Algorithm 2. According to the expected value of the correlation, we adjusted the corresponding value of G^{**} and used the model in Section 4.3 to find linear trails for ARIA. Finally, we identified the linear trail for 5-round ARIA with optimal correlation $2^{-52.6}$ listed in Table 4. And the running time of obtaining this trial is 485 minutes.

In a similar way, we found the optimal linear trail for 4-round ARIA with correlation 2^{-48} and the optimal 6-round ARIA linear trail with correlation 2^{-72} shown in the Table 4.

Differential/Linear Trails for SM4. SM4 [12], underlying the block cipher used in WAPI standard, the Chinese national standard for WLAN, is a 128-bit block cipher with unbalanced generalized Feistel structure. Su *et al.* [34] presented a family of about 2^{14} differential characteristics for 19-round SM4 and an attack on 23-round SM4, where the best differential characteristics has probability 2^{-124}.

The model in Section 4.2 is applied to find differential trails for SM4. Finally, we get the improved differential trail for 19-round SM4 with probability 2^{-123}. Meanwhile, we searched for the linear trail of SM4 and found that the correlation of the best 20-round linear trail is 2^{-60} same as the one in [25].

Impossible Differential Trails for SKINNY without Key Schedule. SKINNY is a family of lightweight block ciphers designed by Beierle *et al.* [5], which adopts the substitution-permutation network and TWEAKEY framework [19]. According to the block size, there are SKINNY-64 and SKINNY-128. In [5], 16 11-round impossible truncated differential characteristics in the single-key setting have been found by MILP for SKINNY-64 and SKINNY-128.

We make use of the method described in the first paragraph in Section 5 and get 3072 12-round impossible differentials for SKINNY-128 in single-key setting. And one of them is

$$(0,0,0,0,0,0,0,0,0,0,0,0,\delta,0,0,0) \nrightarrow (0,0,\beta,0,0,0,0,0,0,0,0,\beta,0,0,0,\beta,0),$$

where δ and β are arbitrary non-zero unfixed differences. We get one more round impossible differential for SKINNY-128 compared with that given in the specification of SKINNY [5].

7 Conclusions

In this paper, STP-based models are constructed to search for optimal differential and linear trails for S-box based ciphers, which are suitable for ciphers with DDT* and LAT*. Moreover, our models give an efficient algorithm to decide the precision of probabilities/correlations in order to not lose the better trails for ciphers with DDT*/LAT*. In this way, we can get the optimal trails more efficiently. In general, it is difficult to search for optimal long-round trails due to the limited computing resource. For some long-round ciphers, we can not get the optimal trails, but our computed precision should be helpful to get an improved trail. Meanwhile, models to find single-key impossible differentials with the key schedule are proposed and applied to 5-round AES. Our models have been applied on several S-Box based block ciphers, and obtained improved results. Till now, there are many automatic search models based on SAT, SMT, MILT and CP. It is very difficult to compare their efficiency because some models are suitable for some ciphers and other models are appropriate for other ciphers. In this paper, we just provide an additional automatic search model, but we can not ensure that it has advantage for any cipher.

References

1. A. Abdelkhalek, Y. Sasaki, Y. Todo, M. Tolba and A. M. Youssef. MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics. IACR Transactions on Symmetric Cryptology. VOLUME 2017, ISSUE 4, pp. 99-129.
2. A. Abdelkhalek, M. Tolba and A. M. Youssef. Improved Linear Cryptanalysis of Round-Reduced ARIA. In: M. Bishop, A. Nascimento (eds) ISC 2016. LNCS 9866, pp. 18-34. Springer.
3. R. Ankele and S. Kölbl. Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis. Cryptology ePrint Archive, Report 2018/689, (2018). `https://eprint.iacr.org/2018/689.pdf`.
4. S. Banik, S.K. Pandey, T. Peyrin, Y. Sasaki, S.M. Sim and Y. Todo. GIFT: A Small Present. In: Fischer W., Homma N. (eds) CHES 2017. LNCS 10529, pp. 321-345. Springer.
5. C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich and S. M. Sim. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In: M. Robshaw, J. Katz (eds) CRYPTO 2016. LNCS 9815, pp. 123-153. Springer.

6. A. Biryukov and I. Nikolić. Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. In: H. Gilbert (ed) EUROCRYPT 2010. LNCS 6110, pp. 322-344. Springer.
7. A. Biryukov and I. Nikolić. Search for Related-Key Differential Characteristics in DES-Like Ciphers. In: A. Joux (ed) FSE 2011. LNCS 6733, pp. 18-34. Springer.
8. A. Biryukov, V. Velichkov and Y. Le. Corre. Automatic Search for the Best Trails in ARX: Application to Block Cipher Speck. In: Peyrin T. (ed) FSE 2016. LNCS 9783, pp. 289-310. Springer.
9. E. Biham and A. Shamir. Differential Cryptanalysis of DES-Like Cryptosystems. In: A.J. Menezes, S.A. Vanstone (eds) CRYPTO 1990. LNCS 537, pp. 2-21. Springer.
10. R. Brummayer and A. Biere. Boolector: An efficient SMT solver for bit-vectors and arrays. In: S. Kowalewski, A. Philippou (eds) TACAS 2009. LNCS 5505, pp. 174-177. Springer.
11. T. Cui, K. Jia, K. Fu, S. Chen and M. Wang. New Automatic Search Tool for Impossible Differentials and Zero-Correlation Linear Approximations. Cryptology ePrint Archive, Report 2016/689, (2016). `https://eprint.iacr.org/2016/689.pdf`.
12. W. Diffie and G. Ledin (translators). SMS4 Encryption Algorithm for Wireless Networks. Cryptology ePrint Archive, Report 2008/329, (2008). `http://eprint.iacr.org/2008/329.pdf`.
13. K. Fu, M. Wang, Y. Guo, S. Sun and L. Hu. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In: T. Peyrin (ed) FSE 2016. LNCS 9783, pp. 268-288. Springer.
14. V. Ganesh and D. L. Dill. A Decision Procedure for Bit-Vectors and Arrays. CAV 2007. LNCS 4590, pp. 519C531. Springer.
15. V. Ganesh, T. Hansen, M. Soos, D. Liew and R. Govostes. STP. (2014). `https://stp.github.io/`
16. V. Ganesh, T. Hansen, M. Soos, D. Liew and R. Govostes. STP Constraint Solver. (2014). `https://github.com/stp/stp`.
17. D. Gérault and P. Lafourcade. Related-Key Cryptanalysis of Midori. In: O. Dunkelman, S. Sanadhya (eds) INDOCRYPT 2016. LNCS 10095, pp. 287-304. Springer.
18. D. Gérault, M. Minier and C. Solnon. Constraint Programming Models for Chosen Key Differential Cryptanalysis. In: M. Rueher (ed) CP 2016. LNCS 9892, pp. 584-601. Springer.
19. J. Jean, I. Nikolić and T. Peyrin. Tweaks and keys for block ciphers: the TWEAKEY framework. In: P. Sarkar, T. Iwata (eds) ASIACRYPT 2014. LNCS 8874, pp. 274-288. Springer.
20. S. Kölbl. CryptoSMT: An Easy to Use Tool for Cryptanalysis of Symmetric Primitives. (2015). `https://github.com/kste/cryptosmt`.
21. S. Kölbl, G. Leander and T. Tiessen. Observations on the SIMON Block Cipher Family. In: R. Gennaro, M. Robshaw (eds) CRYPTO 2015. LNCS 9215, pp. 161-185. Springer.
22. D. Kwon, J. Kim, S. Park, S. H. Sung, Y. Sohn, J. H. Song, Y. Yeom, E-J. Yoon, S. Lee, J. Lee, S. Chee, D. Han and J. Hong. New Block Cipher: ARIA. In: JI. Lim, DH. Lee (eds) ICISC 2003. LNCS 2971, pp. 432-445. Springer.
23. G. Leander, C. Paar, A. Poschmann and K. Schramm. New Lightweight DES Variants. In: A. Biryukov (ed) FSE 2007. LNCS 4593, pp. 196-210. Springer.

24. Y. Liu, Q. Wang and V. Rijmen. Automatic Search of Linear Trails in ARX with Applications to SPECK and Chaskey. In: M. Manulis, AR. Sadeghi and S. Schneider (eds) ACNS 2016. LNCS 9696, pp. 485-499. Springer.
25. Y. Liu, H. Liang, W. Wang and M. Wang. New Linear Cryptanalysis of Chinese Commercial Block Cipher Standard SM4. Security and Communication Networks, 2017.
26. Z. Liu, D. Gu, Y. Liu, J. Li and W. Li. Linear Cryptanalysis of ARIA Block Cipher. In: S. Qing, W. Susilo, G. Wang D. Liu (eds) ICICS 2011. LNCS 7043, pp. 242-254. Springer.
27. M. Matsui. Linear Cryptanalysis Method for DES Cipher. In: T. Helleseth (ed) EUROCRYPT 1993, LNCS 765, pp. 386-397. Springer.
28. M. Matsui. On Correlation between the Order of S-Boxes and the Strength of DES. In: De Santis, A. (ed) EUROCRYPT 1994. LNCS 950, pp. 366C375. Springer. (1995)
29. N. Mouha, Q. Wang, D. Gu and B. Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In: C. Wu, M. Yung, D. Lin (eds) Inscrypt 2011, LNCS 7537, pp. 57-76. Springer.
30. N. Mouha and B. Preneel. Towards Finding Optimal Differential Characteristics for ARX: Application to Salsa20. Cryptology ePrint Archive, Report 2013/328, (2013). `https://eprint.iacr.org/2013/328.pdf`.
31. National Bureau of Standards, Data Encryption Standard, FIPS-Pub. 46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.
32. Y. Sasaki and Y. Todo. New Impossible Differential Search Tool from Design and Cryptanalysis Aspects. In: JS. Coron, J. Nielsen (eds) EUROCRYPT 2017. LNCS 10212, pp. 185-215. Springer.
33. F. Standaert, G. Piret, G. Rouvroy, J. Quisquater and J-D. Legat. ICEBERG: An Involutional Cipher Efficient for Block Encryption in Reconfigurable Hardware. In: B. Roy, W. Meier (eds) FSE 2004. LNCS 3017, pp. 279-298. Springer.
34. B. Su, W. Wu and W Zhang. Security of the SMS4 Block Cipher against Differential Cryptanalysis. Journal of Computer Science and Technology, Volume 26, Issue 1, pp. 130C138. (2011).
35. L. Sun, W. Wang and M. Wang. More Accurate Differential Properties of LED64 and Midori64. IACR Transactions on Symmetric Cryptology 2018, to appear.
36. S. Sun, D. Gerault, P. Lafourcade, Q. Yang, Y. Todo, K. Qiao and L. Hu. Analysis of AES, SKINNY and Others with Constraint Programming. IACR Transactions on Symmetric Cryptology, 2017.
37. S. Sun, L. Hu, K. Qiao, X. Ma, J. Shan and L. Song. Improvement on the Method for Automatic Differential Analysis and its Application to Two Lightweight Block Ciphers DESL and LBlock-s. In: K. Tanaka, Y. Suga (eds) IWSEC 2015. LNCS 9241, pp. 97-111. Springer.
38. S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma and L. Song. Automatic Security Evaluation and (Related-Key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES (L) and Other Bit-Oriented Block Ciphers. In: P. Sarkar, T. Iwata, (eds) ASIACRYPT 2014. LNCS 8873, pp. 158-178. Springer.
39. S. Sun, L. Hu, M. Wang, P. Wang, K. Qiao, X. Ma, D. Shi, L. Song and K. Fu. Towards Finding the Best Characteristics of Some Bit-Oriented Block Ciphers and Automatic Enumeration of (Related-Key) Differential and Linear Characteristics with Predefined Properties. Cryptology ePrint Archive, Report 2014/747, (2014). `https://eprint.iacr.org/2014/747.pdf`.

40. Y. Sun. Linear Cryptanalysis of Light-Weight Block Cipher ICEBERG. Advances in Electronic Commerce, Web Application and Communication. pp. 529-532. Springer.
41. Q. Wang and C. Jin. Upper bound of the length of truncated impossible differentials for AES. Designs, Codes and Cryptography, 2017, pp. 1-12.
42. B. Zhu, X. Dong and H. Yu. MILP-based Differential Attack on Round-reduced GIFT. Cryptology ePrint Archive, Report 2018/390, (2018). `http://eprint.iacr.org/2018/390`.

A Our Identified Trails

Table 4: Linear Trails of ARIA

Linear Trail of 5-Round ARIA		
Round	Input mask	Correlation
1	0x000000000000000000AF000000000000	2^{-3}
2	0xEAEA000000EAEA00000000EAEA00EA00	$2^{-21.8}$
3	0x00000000000000000023000000000000	2^{-3}
4	0xEAEA000000EAEA00000000EAEA00EA00	$2^{-21.8}$
5	0x00000000000000000023000000000000	2^{-3}
6	0x6868000000686800000006868006800	–
Linear Trail of 4-Round ARIA		
Round	Input mask	Correlation
1	0x9000700071006F000000000000000000	2^{-12}
2	0x0007000700070007000000000000000	2^{-12}
3	0x0700070007000700000000000000000	2^{-12}
4	0x0012001200120012000000000000000	2^{-12}
5	0x56B2B9B2E4EF56EFEF5DB200EF00B25D	–

Table 5: Differential Trial of 12-Round GIFT-128

Round	Input difference	Probobility
1	0x0000000A00000CC00000000000000000	2^{-6}
2	0x00000000000000000000000001060000	2^{-5}
3	0x0000000000000000000000A000000000	2^{-2}
4	0x00000100000000000000000000000000	2^{-3}
5	0x00000000000000000800000000000000	2^{-2}
6	0x00002000000010000000000000000000	2^{-5}
7	0x04040000020200000000000000000000	2^{-8}
8	0x00000000505000000000000050500000	2^{-12}
9	0x0000000000000000000000000A000A00	2^{-4}
10	0x00000011000000000000000000000000	2^{-6}
11	0x000000000C0000000600000000000000	2^{-4}
12	0x00002000000000000000000000400000	$2^{-3.4}$
13	0x04000010020000000000004000000020	

Table 6: Differential Trial of 13-Round GIFT-128

Round	Input difference	Probobility
1	0x00000000000000000000000A00000C600	2^{-6}
2	0x00001060000000000000000000000000	2^{-5}
3	0x0000000000000000A00000000000000	2^{-2}
4	0x00000000000010000000000000000000	2^{-3}
5	0x00000000000000000000000000080000	2^{-2}
6	0x00000000000000000000002000000010	2^{-5}
7	0x00000000000000000000040400000202	2^{-8}
8	0x00000000000005050000000000000505	2^{-12}
9	0x0000000000000000000A000A00000000	2^{-4}
10	0x00000000000000000000000000001100	2^{-6}
11	0x000000060000000000000000000000C	2^{-4}
12	0x00000000000000000200000200000000	2^{-4}
13	0x00002000000004000000020000004000	$2^{-6.8}$
14	0x04020004020101020000000100040400	

Table 7: Differential Trial of 21-Round GIFT-128

Round	Input difference	Probobility
1	0x00000000906000000000000000000000	2^{-5}
2	0x0000000000000000000000000A00000	2^{-2}
3	0x00000010000000000000000000000000	2^{-3}
4	0x00000000800000000000000000000000	2^{-2}
5	0x00200000001000000000000000000000	2^{-5}
6	0x00000000000000004040000020200000	2^{-8}
7	0x00005050000000000000505000000000	2^{-12}
8	0x0000000000000000000000000A000A00	2^{-4}
9	0x00000000000000110000000000000000	2^{-6}
10	0x00080000000800000000000000000000	2^{-4}
11	0x00000000000000002020000010100000	2^{-10}
12	0x00005050000000000000505000000000	2^{-12}
13	0x0000000000000000000000000A000A00	2^{-4}
14	0x00000000000000110000000000000000	2^{-6}
15	0x00080000000800000000000000000000	2^{-4}
16	0x00000000000000002020000010100000	2^{-10}
17	0x00005050000000000000505000000000	2^{-12}
18	0x0000000000000000000000000A000A00	3^{-4}
19	0x00000000000000110000000000000000	4^{-6}
20	0x00000000000C0000000600000000000	5^{-4}
21	0x00000000004000000000200000000000	$6^{-3.4}$
22	0x00100400000000000040010000200000	

Table 8: Related-Key Differential Trails of DES

Related-Key Differential Trail of 4-Round DES				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
1	0xFFFDFFFF	0xFFFFFFFF	0xFFFFFFFFFFFF	1
2	0xFFFFFFFF	0xFFFDFFFF	0xFFFFFBFFFFFF	1
3	0xFFFDFFFF	0xFFFFFFFF	0xFFFFFFFFFFFF	1
4	0xFFFFFFFF	0xFFFDFFFF	0xFFFFDFFFFFFF	$2^{-3.4}$
5	0xFFFDFFFF	0xFFBFEFBF	–	–
Related-Key Differential Trail of 6-Round DES				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
0	0x84010140	0x00020000	0x000800000000	$2^{-4.4}$
1	0x00020000	0x00000000	0x000000000000	1
2	0x00000000	0x00020000	0x000004000000	1
3	0x00020000	0x00000000	0x000000000000	1
4	0x00000000	0x00020000	0x000020000000	$2^{-3.4}$
5	0x00020000	0x80000000	0x800000000000	$2^{-4.4}$
6	0x80000000	0x00028820	–	–
Related-Key Differential Trail of 7-Round DES				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
3	0xFDFEFFFB	0xFFDFFFFF	0xFFFDFFFFFFFF	$2^{-2.4}$
4	0xFFDFFFFF	0xFDFFFFFF	0xFFEFFFFFFFFF	2^{-5}
5	0xFDFFFFFF	0xFFDFFFFF	0xFFFFBFFFFFFF	$2^{-3.4}$
6	0xFFDFFFFF	0xFDFFFFFF	0xFFBFFFFFFFFF	1
7	0xFDFFFFFF	0xFFDFFFFF	0xFFFEFFFFFFFF	1
8	0xFFDFFFFF	0xFDFFFFFF	0xFDFFFFFFFFFF	$2^{-2.7}$
9	0xFDFFFFFF	0xFFDFFFFF	0xF7FFFFFFFFFF	$2^{-4.8}$
10	0xFFDFFFFF	0xF97F7EFF	–	–

Table 9: Related-Key Differential Trails of 4∼7-Round DESL

Related-Key Differential Trail of 4-Round DESL				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
1	0xFFFDFFFF	0xFFFFFFFF	0xFFFFFFFFFFFF	1
2	0xFFFFFFFF	0xFFFDFFFF	0xFFFFFBFFFFFF	1
3	0xFFFDFFFF	0xFFFFFFFF	0xFFFFFFFFFFFF	1
4	0xFFFFFFFF	0xFFFDFFFF	0xFFFFDFFFFFFF	$2^{-2.4}$
5	0xFFFDFFFF	0xFFFFFFBF	–	–
Related-Key Differential Trail of 5-Round DESL				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
6	0x00000000	0x00000020	0x000000000100	1
7	0x00000020	0x00000000	0x000000000000	1
8	0x00000000	0x00000020	0x000000000040	$2^{-3.4}$
9	0x00000020	0x00000400	0x000000008000	1
10	0x00000400	0x00000020	0x000000000400	$2^{-2.2}$
11	0x00000020	0x00000401	–	–
Related-Key Differential Trail of 6-Round DESL				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
5	0x00200028	0x00000000	0x000000020000	$2^{-2.4}$
6	0x00000000	0x00000020	0x000000000100	1
7	0x00000020	0x00000000	0x000000000000	1
8	0x00000000	0x00000020	0x000000000040	$2^{-3.4}$
9	0x00000020	0x00000400	0x000000008000	1
10	0x00000400	0x00000020	0x000000000400	$2^{-2.2}$
11	0x00000020	0x00000401	–	–
Related-Key Differential Trail of 7-Round DESL				
Round	ΔX_L^r	ΔX_R^r	ΔK^r	Probability
5	0xFFDFFFD7	0xFFFFFFFF	0xFFFFFFFDFFFF	$2^{-2.4}$
6	0xFFFFFFFF	0xFFFFFFDF	0xFFFFFFFFFEFF	1
7	0xFFFFFFDF	0xFFFFFFFF	0xFFFFFFFFFFFF	1
8	0xFFFFFFFF	0xFFFFFFDF	0xFFFFFFFFFFBF	$2^{-3.4}$
9	0xFFFFFFDF	0xFFFFFBFF	0xFFFFFFFF7FFF	1
10	0xFFFFFBFF	0xFFFFFFDF	0xFFFFFFFFFBFF	2^{-4}
11	0xFFFFFFDF	0xFFFFFFFF	0xFFFFFFF7FFFF	$2^{-2.4}$
12	0xFFFFFFFF	0xFEFBFF5F	–	–

Table 10: Linear Trial of 6-Round ICEBERG

Round	Input mask	Correlation
1	0x4761000000000047	$2^{-6.9}$
2	0x0000000000000004	$2^{-2.5}$
3	0x0040000000100040	$2^{-7.8}$
4	0x0000000000000040	$2^{-2.4}$
5	0x0040000000100040	$2^{-7.8}$
6	0x0000000000000040	$2^{-2.4}$