IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2020, No. 3, pp. 262–287. DOI:10.13154/tosc.v2020.i3.262-287

On the Usage of Deterministic (Related-Key) Truncated Differentials and Multidimensional Linear Approximations for SPN Ciphers

Ling Sun[1,2], David Gerault[3], Wei Wang[1,2] and Meiqin Wang[1,2]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[2] School of Cyber Science and Technology, Shandong University, Qingdao, China
[3] Nanyang Technological University, Singapore, Singapore
lingsun@sdu.edu.cn, dagerault@gmail.com, {weiwangsdu,mqwang}@sdu.edu.cn

Abstract. Among the few works realising the search of truncated differentials (TD) and multidimensional linear approximations (MDLA) holding for sure, the optimality of the distinguisher should be confirmed via an exhaustive search over all possible input differences/masks, which cannot be afforded when the internal state of the primitive has a considerable number of words. The incomplete search is also a long-term problem in the search of optimal impossible differential (ID) and zero-correlation linear approximation (ZCLA) since all available automatic tools operate under fixed input and output differences/masks, and testing all possible combinations of differences/masks is impracticable for now. In this paper, we start by introducing an automatic approach based on the constraint satisfaction problem for the exploration of deterministic TDs and MDLAs. Since we transform the exhaustive search into an inherent feature of the searching model, the issue of incomplete search is settled. This tool is applied to search for related-key (RK) TDs of AES-192, and a new related-key differential-linear (DL) distinguisher is identified with a TD with certainty. Due to the novel property of the distinguisher, the previous RK DL attack on AES-192 is improved. Also, the new distinguisher is explained from the viewpoint of differential-linear connectivity table (DLCT) and thus can be regarded as the first application of DLCT in the related-key attack scenario. As the second application of the tool, we propose a method to construct (RK) IDs and ZCLAs automatically. Benefiting from the control of the nonzero fixed differential pattern and the inherent feature of exhaustive search, the new searching scheme can discover longer distinguishers and hence possesses some superiorities over the previous methods. This technique is implemented with several primitives, and the provable security bounds of `SKINNY` and `Midori64` against impossible differential distinguishing attack are generalised.

Keywords: Truncated differential · Multidimensional linear approximation · Differential-linear attack · Impossible differential · Zero-correlation linear approximation

1 Introduction

Differential cryptanalysis [BS90] was introduced by Biham and Shamir in the early 1990s. For the target cipher, differential cryptanalysis studies how the input difference in the plaintext propagates to the output difference in the ciphertext. If a pair of input and output differences behaves differently from a random case, this pair can be used to create a distinguisher or even launch a key-recovery attack. Over almost the same period, Matsui [Mat93] proposed the linear cryptanalysis. With the biased linear relation among the

Received: 2019-11-23 Revised: 2020-06-01 Accepted: 2020-08-01 Published: 2020-09-28

plaintext, ciphertext, and secret key, the adversary can realise distinguishing and key-recovery attacks. As the most profound cryptanalytic approaches, these two methods showed significant effects on many of the subsequent methods.

As a generalisation of the differential cryptanalysis, the truncated differential (TD), for which only a part of the difference in the ciphertext is predicted, was proposed by Knudsen [Knu94]. The intuition of truncated differential lies in that determining the partial information of the difference is relatively easy, and this partial message also can be exploited to accomplish distinguishing and key-recovery attacks. This technique has been widely used in the analyses of various primitives [KB96, BKR97, KRW99]. One particular class of truncated differentials consists of differentials holding with probability one. This kind of differentials can be employed to assemble distinguishers of some combined attacks, such as differential-linear (DL) attack [LH94] and impossible differential (ID) attack [Knu98, BBS99]. A counterpart of the truncated differential in the field of linear cryptanalysis is the multidimensional linear approximation (MDLA) [BN14]. The strength of the multidimensional linear approximation is measured by its capacity. Similarly, those MDLAs with capacity being zero have an influence on the construction of zero-correlation linear approximations (ZCLA) for zero-correlation linear cryptanalysis [BW12, BLNW12, BR14], which is regarded as the dual of impossible differential cryptanalysis under linear attack setting.

Theoretically, $\mathcal{U}$-method [KHS+03] and UID-method [LLWG14], which are automated tools relying on the miss-in-the-middle approach for the search of truncated impossible differentials, can be utilised to develop deterministic TD/MDLA. In order to identify the optimal TD/MDLA, the program should be performed under all possible input differences/masks. However, when the internal state of the primitive has a considerable number of words ℓ, the exhaustive search cannot be afforded since its complexity is $\mathcal{O}(2^{\ell})$.

In the last decade, the automatic tools [SHW+14a, SHW+14b, KLT15, LWR16, SGL+17, AST+17, SWW18, GLMS18] for cryptanalysis obtained rapid development. Nevertheless, few works concentrated on the deterministic TD/MDLA. Although the automatic tools based on the mixed integer linear programming (MILP) [CJF+16, AST+17] and constraint programming (CP) [SGL+17] supported the search of impossible differential trail/zero-correlation linear characteristic, they mainly focused on the existence of trail under the fixed input and output differences/masks. Considering the computing power in existence does not allow us to exhaustively verify all combinations of input and output differences/masks, these approaches only check the trails with input and output differences/masks belonging to a predetermined set. A common choice for the set is the one containing all vectors with only one active cell or bit. Note that the tool in [AST+17] under arbitrary S-box mode targeting TD IDs also encounters the problem of the incomplete search when ℓ is relatively large as the complexity of the complete search is $\mathcal{O}(2^{2\ell})$, roughly.

For the lack of an automatic tool targeting deterministic TD/MDLA, we start by creating an automatic method for the search of TD/MDLA holding for sure. Since the feature of the exhaustive search is incorporated in the new model, we settle the long-term problem of incomplete search in the field of discovering truncated (impossible) differential and multidimensional (zero-correlation) linear approximation distinguishers.

Our Contributions.

An automatic tool for the search of deterministic (RK) TDs and MDLAs. In light of an absence of the automatic tool for the search of deterministic (RK) TDs and MDLAs, we propose a method based on constraint programming (CP) to fulfil this goal. The principle is to create a constraint satisfaction problem (CSP) depicting the cryptanalytic features that we are interested in and invoke some CP solvers to get the solution of the problem. The reason that we employ CP lies in the conciseness in the model-generation phase and the convenience in the model-verification phase. Note that the first step of

the tool with CP in [GLMS18] only realised a rough search for nondeterministic TDs, and the solution might be byte inconsistent, i.e., no actual trail follows the TD. While in our searching scheme, to trace the propagation of the differential pattern precisely, we introduce more variables to describe more subtle features of the internal state and generate a more comprehensive model. Thus, our model is promising in search of deterministic TDs. Besides, since we convert the exhaustive search into an inherent feature of the model, the long-term problem of inadequate search in the area of detecting deterministic TDs and MDLAs is settled.

Improved related-key differential-linear attack on AES-192. The tool targeting deterministic TDs is applied to search for RK TDs of AES-192. A new related-key differential-linear distinguisher is constructed with a RK TD holding for sure. Since the new distinguishing property enables us to guess fewer subkey bytes in the key-recovery attack, the previous attack result [ZZWF07] on seven rounds can be improved. To precisely evaluate the complexity, we adjust the statistical model in conventional linear cryptanalysis to this setting and formulate the relation among the data requirement, the success probability and the advantage of the attack. The theoretical statistical model is verified with random tests. With the new model, the time complexity of the attack is reduced from 2^{187} to $2^{170.5}$, and the data requirement is decreased, slightly. Moreover, we interpret the new distinguisher from the aspect of differential-linear connectivity table (DLCT) [BDKW19], which is intended to exploit the dependence between the two subciphers in the DL attack. Thus, the new distinguisher can be seen as the first application of the DLCT in the related-key attack scenario.

Constructing (RK) IDs with TDs and ZCLAs with MDLAs. As the second application of the automatic tool, we propose the $\mathcal{U}^*$-method relying on CP to construct (RK) IDs and ZCLAs. The $\mathcal{U}^*$-method is based on the miss-in-the-middle approach and serves as a basic version for the search of (RK) IDs and ZCLAs. Benefiting from the control of the nonzero fixed differential pattern and the inherent feature of exhaustive search, the $\mathcal{U}^*$-method can identify longer distinguishers and thus is superior over the $\mathcal{U}$-method [KHS+03] and UID-method [LLWG14]. We also invent an optimised version of the $\mathcal{U}^*$-method so that the contradictions recognised by the model are not limited to those located at the meeting point. Comparing to the system proposed by Wu and Wang [WW12], the optimised $\mathcal{U}^*$-method supports the exhaustive search even when the number of words regarding the objective is considerable, which illustrates its better performance. For instance, the number of runs to search for the optimal ID of Minalpher-P [STA+14] is reduced from 2^{128} to $2^{10.9}$. Given the practical computing power, the complexity of 2^{128} is out of reach anyhow, while our method spending a few thousands of minutes can be afforded. A comprehensive comparison of all tools targeting (RK) IDs for SPN ciphers is also comprised in the paper (see Table 1).

We acknowledge that some of the newly identified distinguishers are extensions with probability one of the ones that were published in previous literature. However, we think that the centre of the paper is more the new technique. Since the new tool can realise the exhaustive search in a surprisingly rapid manner, we hope it may play an essential role in the designing phase of new ciphers.

Provable security against ID distinguishing attack of `SKINNY` and `Midori64`. By invoking the automatic tool with MILP, Sasaki and Todo [ST17] claimed that the maximum length of impossible differential characteristics for `SKINNY`-64 is less than 12-round under the restriction that the input and output differences have one active nibble. With the aid of CP, we generalise this conclusion and prove that under the keyed (uniform) bijective S-box assumption, 13.5-round encryption of `SKINNY` is secure against impossible differential

distinguishing attack for arbitrary differential with nonzero input and output differences. Likewise, we update the security bound of `Midori64`.

Organisation of the paper.

In Sect. 2, we review fundamental conceptions in differential and linear cryptanalyses as well as the constraint satisfaction problem. Then, the method relying on CSP for the search of deterministic truncated differentials and multidimensional linear approximations is presented in Sect. 3. In Sect. 4, this approach is applied to search for related-key truncated differentials of AES-192. With the newly identified non-random property, the previous related-key differential-linear attack regarding AES-192 is improved. Sect. 5 demonstrates the second usage of the tool in Sect. 3. More specifically, we explain how it can be used to construct (related-key) impossible differentials and zero-correlation linear approximations. The applications on several designs updating the previous results can be found in Sect. 6. We conclude the paper in Sect. 7. Some details can be found in Supplementary Material, which is publicly available at https://github.com/Deterministic-TD-MDLA/auxiliary_material. The source codes can be obtained at the same web address.

2 Preliminary

2.1 Basics of Differential and Linear Cryptanalyses

A differential that predicts only parts of an n-bit difference is called a *truncated differential* (TD) [Knu94]. Since we focus on the TD holding for sure, instead of propagating the real difference, we only pay attention to the pattern of the difference. Let $\Delta X = (\Delta X_0, \Delta X_1, \ldots, \Delta X_{\ell-1})$ be the difference of the internal state X, where $\Delta X_i \in \mathbb{F}_{2^s}$, $n = \ell \cdot s$. The *differential pattern* $\Delta_X = (\Delta_{X_0}, \Delta_{X_1}, \ldots, \Delta_{X_{\ell-1}})$ of X is an ℓ-dimensional vector with each Δ_{X_i} being the linear combination[1] of the following four patterns of differences:

- zero differential pattern (Z): if $\Delta X_i = 0$;
- nonzero fixed differential pattern (N): if ΔX_i is non-zero and fixed;
- nonzero varied differential pattern (N*): if ΔX_i can be any value except zero;
- varied differential pattern (U): if ΔX_i can take any value.

Note that [KHS+03] is the first paper proposing these patterns of differences, and many subsequent works [WW12, LLWG14] followed these symbols.

With the iteration of the round function, the differential patterns of the inner states gradually lose information and will turn into a vector with all elements being U, eventually. The following lemmas are supplied to depict the propagations of differential patterns through three basic operations. Note that the certainties of all propagations enable us to construct deterministic TDs.

Lemma 1 (Branching)**.** *For the branching operation presented in Figure 1(a), the propagation of the differential pattern is* $\Delta_{Y_0} = \Delta_{Y_1} = \Delta_X$.

Lemma 2 (XOR)**.** *For the XOR operation illustrated in Figure 1(b), the correspondences among the differential pattern* Δ_Y *of the output branch and the two input patterns* Δ_{X_0} *and* Δ_{X_1} *are given in the following table.*

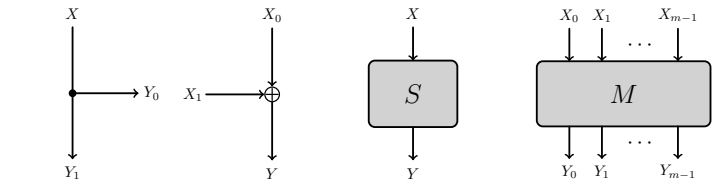
[1] In this paper, we only consider one kind of linear combination $\mathsf{N} \oplus \mathsf{N}^*$ comprising more than one pattern. Nevertheless, it is not included in our model for SPN ciphers. Please refer to Sect. 3 for an explanation.

Δ_Y		Δ_{X_1}				
		Z	N	N ⊕ N*	N*	U
Δ_{X_0}	Z	Z	N	N ⊕ N*	N*	U
	N	N	Z/N	N*/N⊕N*	N ⊕ N*	U
	N ⊕ N*	N ⊕ N*	N*/N⊕N*	U	U	U
	N*	N*	N ⊕ N*	U	U	U
	U	U	U	U	U	U

When the differential patterns of the two input branches are nonzero and fixed, the output pattern depends on the values of ΔX_0 and ΔX_1. If $\Delta X_0 = \Delta X_1$, the output pattern is Z*; otherwise,* $\Delta_Y =$ N.

Lemma 3 (S-box)**.** *For the s-bit S-box displayed in Figure 1(c), all the possible propagations from Δ_X to Δ_Y are listed in the following table.*

Δ_X	Z	N	N⊕N*	N*	U
Δ_Y	Z	N*	U	N*	U



(a) Branching. (b) XOR. (c) S-box. (d) MDS matrix.

Figure 1: Frequently-used operations.

Apart from the rules for the basic operations, we give the propagation of differential pattern for the Maximum Distance Separable (MDS) matrix, which is a frequently-used building block for the diffusion layer.

Lemma 4 (MDS matrix)**.** *As illustrated in Figure 1(d), suppose that M is an $m \times m$ MDS matrix. All the possible propagations from Δ_X to Δ_Y are given in the following table.*

Δ_X	(Z, Z, . . . , Z)	(Z, . . . , Z, N/N*, Z, . . . , Z)	*Remaining cases*
Δ_Y	(Z, Z, . . . , Z)	(N*, N*, . . . , N*)	(U, U, . . . , U)

In the area of linear cryptanalysis, an approximation is called a *multidimensional linear approximation* (MDLA) if only parts of input and output masks are known and fixed. With a multidimensional linear approximation $\mathbb{L}$, multidimensional linear cryptanalysis [JR94, BJV04, BCQ04, HCN09] intends to exploit the statistical behaviour of the correlations concerning all the linear approximations belonging to $\mathbb{L}$. Denote $\Gamma X = (\Gamma X_0, \Gamma X_1, \ldots, \Gamma X_{\ell-1})$ the linear mask of X, where $\Gamma X_i \in \mathbb{F}_{2^s}$ for $0 \leqslant i \leqslant \ell - 1$. The *linear pattern* $\Gamma_X = (\Gamma_{X_0}, \Gamma_{X_1}, \ldots, \Gamma_{X_{\ell-1}})$ of X is an ℓ-dimensional vector with each Γ_{X_i} being the linear combination of the following four patterns of linear masks:

- zero linear pattern (Z): if $\Gamma X_i = 0$;
- nonzero fixed linear pattern (N): if ΓX_i is nonzero and fixed;
- nonzero varied linear pattern (N*): if ΓX_i can be any value except zero;
- varied linear pattern (U): if ΓX_i can take any value.

The propagations of linear patterns are similar to those of differential patterns given in Lemma 1 - 4 since the propagations of linear masks and differences inside the SPN and Feistel ciphers are dual [SLG+16] to a certain extent. Thus, we do not expand on the rules of propagation under the linear setting.

2.2 Constraint Satisfaction Problem

The constraint satisfaction problem (CSP) is a kind of mathematical problems, in which the set of variables must satisfy a series of constraints.

Definition 1 (Constraint Satisfaction Problem, [SGL+17])**.** A constraint satisfaction problem is represented as a triple $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$.

- $\mathcal{X} = \{x_0, x_1, \ldots, x_{n-1}\}$ is a set of variables.
- $\mathcal{D} = \{\mathcal{D}(x_0), \mathcal{D}(x_1), \ldots, \mathcal{D}(x_{n-1})\}$ is a set of nonempty sets. $\mathcal{D}(x_i)$ specifies the domain of x_i.
- $\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{m-1}\}$ stands for a set of constraints. Each constraint $\mathcal{C}_j$ is a pair $\langle \mathcal{X}_j, \mathcal{R}_j \rangle$, where $\mathcal{X}_j$ is a subset of $\mathcal{X}$ composed of $k_j \triangleq |\mathcal{X}_j|$ variables, and $\mathcal{R}_j$ is a relation that limits the values of the k_j variables in $\mathcal{X}_j$ can take. These relations can be given intentionally as a formula (see Model 2), or extensionally as a set (see Model 4), or procedurally with an appropriate generating or recognising function (see Model 1, 3, and 5).

An evaluation fulfils the constraint $\mathcal{C}_j$ if the values assigned to the variables in $\mathcal{X}_j$ validate the relation $\mathcal{R}_j$. An evaluation is *consistent* if it does not violate any constraints. An evaluation is *complete* if it includes all variables in $\mathcal{X}$. An evaluation is a *solution* if it is consistent and complete.

The Boolean satisfiability problem (SAT) and the satisfiability modulo theories (SMT) can be viewed as individual cases of the CSP. Besides, the CSP can describe much harder cases, which may not be expressible with some of these relatively simpler instances.

The constraint programming (CP) is to search for solutions of the CSP. Although solving a CSP on a finite domain is an NP-complete problem, many CP solvers are available to solve problems of practical interest. In this paper, we use MiniZinc[2], which is a programming language used to describe decision problems over integers and real numbers, to format the issues of concern and optionally invoke CP solvers Gecode[3] and OSICBC[4] to solve the problems.

3 Finding Deterministic TDs and MDLAs

Typically, among all the possible differential patterns, Z, N, $\mathsf{N} \oplus \mathsf{N}^*$, and N^* contain effective information, with which we can realise distinguishing attacks. However, the model we are going to introduce only includes Z, N, and N^*. We give a few words to explain why ignoring the propagation of $\mathsf{N} \oplus \mathsf{N}^*$ for SPN ciphers does not prevent us from locating the optimal distinguisher. Note that after the nonlinear layer (see Lemma 3), N is replaced with N^*, and $\mathsf{N} \oplus \mathsf{N}^*$ turns into U, which indicates that after passing through at most two nonlinear layers, $\mathsf{N} \oplus \mathsf{N}^*$ will be absent from the pattern propagating phase of SPN ciphers under the *single-key attack scenario*. So, the optimal deterministic TDs covering more than two rounds do not predict the patterns of the output words as $\mathsf{N} \oplus \mathsf{N}^*$, and ignoring

[2] https://www.minizinc.org/software.html
[3] https://www.gecode.org/flatzinc.html
[4] https://projects.coin-or.org/Cbc

$\mathsf{N} \oplus \mathsf{N}^*$ is reasonable. Under the *related-key attack scenario*, $\mathsf{N} \oplus \mathsf{N}^*$ may reappear after more than two nonlinear layers for the influence of the subkey difference. Nonetheless, we note that the presence of $\mathsf{N} \oplus \mathsf{N}^*$ relies on the existence of N. Thus, if we notice that the pattern of a certain internal word is N, which may result from a known subkey difference, we can infer the survival of $\mathsf{N} \oplus \mathsf{N}^*$, manually. In this sense, employing the properties of the three patterns Z, N, and N^* for SPN ciphers is sufficient. So, the searching method in this section does not involve the pattern $\mathsf{N} \oplus \mathsf{N}^*$, and we regard $\mathsf{N} \oplus \mathsf{N}^*$ as U.

For Feistel ciphers, the propagation of $\mathsf{N} \oplus \mathsf{N}^*$ is of great importance. A generalisation of the searching method is proposed in Supplementary Material A.3, which fulfils the function of propagating $\mathsf{N} \oplus \mathsf{N}^*$ to a certain degree.

Now, we turn to the deterministic TDs and MDLAs finding method. The main idea is to convert the problem under consideration into a CSP $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$. In the following, after introducing the set $\mathcal{X}$ and the set $\mathcal{D}$, we expand upon the set $\mathcal{C}$ of constraints, which is split into two parts. The first part propagates the differential pattern, while the second one clarifies the searching scopes of the input and output patterns. The reason that we exploit the CSP is explained at the end of this section, briefly.

3.1 Initialising Variables

For each entry of the inner state $X = (X_0, X_1, \ldots, X_{\ell-1})$, we introduce an integer variable δ_{X_i} to stand for the differential pattern of X_i. The domain $\mathcal{D}(\delta_{X_i})$ of δ_{X_i} is $\{z \in \mathbb{Z} \mid 0 \leqslant z \leqslant 3\}$, and the correspondence between the differential pattern of X_i and δ_{X_i} is

$$\delta_{X_i} = \begin{cases} 0, & \text{if } \Delta_{X_i} = \mathsf{Z} \\ 1, & \text{if } \Delta_{X_i} = \mathsf{N} \\ 2, & \text{if } \Delta_{X_i} = \mathsf{N}^* \\ 3, & \text{if } \Delta_{X_i} = \mathsf{U} \end{cases}.$$

Note that the difference ΔX_i is known if $\Delta_{X_i} = \mathsf{N}$. To utilise the information of the nonzero fixed difference, we import another integer variable ζ_{X_i} for each X_i to represent the actual s-bit difference ΔX_i. The domain of ζ_{X_i} is $\mathcal{D}(\zeta_{X_i}) = \{z \in \mathbb{Z} \mid -2 \leqslant z \leqslant 2^s - 1\}$. The data range of ζ_{X_i} varies with the value of δ_{X_i}, that is,

$$\zeta_{X_i} \in \begin{cases} \{0\}, & \text{if } \delta_{X_i} = 0 \\ \{1, 2, \ldots, 2^s - 1\}, & \text{if } \delta_{X_i} = 1 \\ \{-1\}, & \text{if } \delta_{X_i} = 2 \\ \{-2\}, & \text{if } \delta_{X_i} = 3 \end{cases}.$$

If δ_{X_i} is equal to 0 or 1, the value of ζ_{X_i}, which records the value of ΔX_i, is useful; otherwise, the value of ζ_{X_i} does not have practical meaning, but this assignment method is convenient for us to create constraints in the CSP.

With $\mathcal{X}$ and $\mathcal{D}$ mentioned above, we study how to generate constraints in $\mathcal{C}$ according to our requirement. Suppose that $\mathcal{X}_j = \{x_0, x_1, \ldots, x_{k_j-1}\}$ is a set of variables, and $\mathcal{D}(\mathcal{X}_j) = \mathcal{D}(x_0) \times \mathcal{D}(x_1) \times \cdots \times \mathcal{D}(x_{k_j-1})$ is the domain of the k_j-tuple $x = \langle x_0, x_1, \ldots, x_{k_j-1} \rangle$. Note that one constraint implies one relation of variables, and the relation $\mathcal{R}_j$ on $\mathcal{X}_j$ restricts the data range of x to a set $\mathcal{V}(\mathcal{X}_j) \subseteq \mathcal{D}(\mathcal{X}_j)$ containing all valid values of x. With the knowledge of $\mathcal{V}(\mathcal{X}_j)$, we can apply one of the following two methods to generate the expression of $\mathcal{R}_j$.

- The *inclusion method* focuses on the vectors belonging to $\mathcal{V}(\mathcal{X}_j)$ and manages to describe the characters of these vectors (see Model 1, 2, 3, and 5).

- The *exclusion method* adopts an indirect strategy and attempts to remove the invalid vectors in $\mathcal{D}(\mathcal{X}_j)\backslash\mathcal{V}(\mathcal{X}_j)$ from $\mathcal{D}(\mathcal{X}_j)$ (see Model 4). Hence, this method pays more attention to the set $\mathcal{D}(\mathcal{X}_j)\backslash\mathcal{V}(\mathcal{X}_j)$.

According to our experience, the selection of the method depends on the sizes of the sets $\mathcal{V}(\mathcal{X}_j)$ and $\mathcal{D}(\mathcal{X}_j)\backslash\mathcal{V}(\mathcal{X}_j)$. Intuitively, the smaller the size of the set, the simpler the features of the vectors belonging to it. If $|\mathcal{V}(\mathcal{X}_j)| < |\mathcal{D}(\mathcal{X}_j)\backslash\mathcal{V}(\mathcal{X}_j)|$, we prefer to employ the inclusion method; otherwise, we turn to the exclusion method. When the size of the domain $\mathcal{D}(\mathcal{X}_j)$ is not very large, both of the two methods are optional since the sizes of the two subsets do not have significant differences.

To depict the features of a given set, we can use some logical and mathematical expressions. MiniZinc supports various descriptions, such as the primary logic operations '`and`($/\backslash$)', '`or`($\backslash/$)', and the conditional '`if-then-else-endif`' expression, with which we can flexibly formulate the relation. Please refer to https://www.minizinc.org/doc-2.2.3/en/index.html for more details.

The following model constructed with the inclusion method shows the constraint on the set of variables $\{\delta_{X_i}, \zeta_{X_i}\}$.

Model 1 (Relation between δ_{X_i} and ζ_{X_i})**.** *Adding the following expression on δ_{X_i} and ζ_{X_i} into $\mathcal{C}$ will ensure that ζ_{X_i} falls into the correct range.*

$$\begin{aligned}
&\texttt{if } \delta_{X_i} = 0 \texttt{ then } \zeta_{X_i} = 0\\
&\texttt{elseif } \delta_{X_i} = 1 \texttt{ then } \zeta_{X_i} > 0\\
&\texttt{elseif } \delta_{X_i} = 2 \texttt{ then } \zeta_{X_i} = -1\\
&\texttt{else } \zeta_{X_i} = -2 \texttt{ endif}
\end{aligned}$$

In the case of $\delta_{X_i} = 1$, we do not assign a fixed value to ζ_{X_i}. In the solving phase, if the CSP is solvable, the CP solver will return a suitable value for it.

3.2 Propagating Differential Patterns

To propagate the differential pattern across one round of encryption, we decompose the round function into multiple simple operations. In this subsection, we generate CSP models, i.e., constraints, for four frequently-used operations. The constraints corresponding to the round function are the composition of these basic models.

Model 2 (Branching)**.** *The following constraint restricts the pattern propagation for the Branching operation in Figure 1(a).*

$$\delta_{Y_0} = \delta_X \texttt{ and } \zeta_{Y_0} = \zeta_X \texttt{ and } \delta_{Y_1} = \delta_X \texttt{ and } \zeta_{Y_1} = \zeta_X$$

Model 3 (XOR)**.** *The following constraint specifies the propagation of the differential pattern for the XOR operation in Figure 1(b).*

$$\begin{aligned}
&\texttt{if } \delta_{X_0} + \delta_{X_1} > 2 \texttt{ then } \delta_Y = 3 \texttt{ and } \zeta_Y = -2\\
&\texttt{elseif } \delta_{X_0} + \delta_{X_1} = 1 \texttt{ then } \delta_Y = 1 \texttt{ and } \zeta_Y = \zeta_{X_0} + \zeta_{X_1}\\
&\texttt{elseif } \delta_{X_0} = \delta_{X_1} = 0 \texttt{ then } \delta_Y = 0 \texttt{ and } \zeta_Y = 0\\
&\texttt{elseif } \zeta_{X_0} + \zeta_{X_1} < 0 \texttt{ then } \delta_Y = 2 \texttt{ and } \zeta_Y = -1\\
&\texttt{elseif } \zeta_{X_0} = \zeta_{X_1} \texttt{ then } \delta_Y = 0 \texttt{ and } \zeta_Y = 0\\
&\texttt{else } \delta_Y = 1 \texttt{ and } \zeta_Y = \zeta_{X_0} \oplus \zeta_{X_1} \texttt{ endif}
\end{aligned}$$

Model 4 (S-box)**.** *The following constraint clarifies all the possible pattern propagations for the S-box in Figure 1(c).*

$$\delta_Y \neq 1 \texttt{ and } \delta_X + \delta_Y \in \{0, 3, 4, 6\} \texttt{ and } \delta_Y \geqslant \delta_X \texttt{ and } \delta_Y - \delta_X \leqslant 1$$

The constructions of Model 3 and Model 4 are based on the inclusion and exclusion methods, respectively. Please find more details in Supplementary Material A.1.

Model 5 (MDS matrix)**.** *The following constraint is sufficient to specify all possible propagations for the MDS matrix in Figure 1(d).*

$$\begin{aligned}
&\texttt{if } \sum_{i=0}^{m-1} \delta_{X_i} \equiv 0 \texttt{ then } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 0\\
&\texttt{elseif } \sum_{i=0}^{m-1} \delta_{X_i} \equiv 1 \texttt{ then } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 2\\
&\texttt{elseif} \sum_{i=0}^{m-1} \delta_{X_i} \equiv 2 \texttt{ and } \sum_{i=0}^{m-1} \zeta_{X_i} < 0 \texttt{ then } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 2\\
&\texttt{else } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 3 \texttt{ endif}
\end{aligned}$$

Model 5 is generated with the inclusion method. Since the construction is straightforward, we do not elaborate on it. For other word-oriented diffusion layers without using the MDS matrices, the propagation of the differential pattern can be established by iterating the models for Branching and XOR operations, sequentially.

Generally, the four frequently-used models are adequate for us to handle most of the primitives. When the round function of the objective involves novel designs which cannot be fully expressed with the above models, the users may build new models with the inclusion or exclusion method.

By organising these basic models, we can create constraints representing the pattern propagation across the cipher. For ciphers with word-oriented key schedules, the constraints about the key schedule can be integrated into the whole CSP so that we can realise the search of related-key truncated differentials.

3.3 Clarifying the Searching Scopes of the Input and Output Patterns

The searching scope of the input pattern. Previously, the usual way to set up the searching scope of the input pattern is to fix it as a predetermined value, and the format of the input pattern will influence the length of the TD. A general method is to select those patterns with only one active word. However, in some cases, the optimal trail does not originate from this kind of input patterns. An ideal solution for the optimal TD is an exhaustive search over all possible input patterns, which cannot be afforded when the internal state of the primitive has a considerable number of words. On the other side, identifying the input pattern resulting in a longer trail requires sophisticated experience.

To overcome this long-term problem, in the new model, we do not fix the format of the input pattern and only claim that the input difference is nonzero. Denote $(X_0^{r-1}, X_1^{r-1}, \ldots, X_{\ell-1}^{r-1})$ the input state of the r-th round ($r \geqslant 1$). We add the constraint $\sum_{i=0}^{\ell-1} \delta_{X_i^0} \neq 0$ into the CSP. Then, in the searching phase, the CP solver will automatically traverse all possible input patterns, and the exhaustive search turns into an inherent feature of our model.

Under the fixed input pattern searching mode, the program should be implemented for about 2^ℓ times. While with the new model, to ensure the existence of R-round TDs/MDLAs, at most, we invoke the searching program for $3 \cdot R \cdot \ell$ times. For instance, the number of runs[5] to search for the optimal ID of Minalpher-P introduced in Sect. 6 is reduced from 2^{128} to $2^{10.9} (\approx 2 \cdot 3 \cdot 5 \cdot 64)$.

[5] We know that the comparison is not entirely fair since the new model spends much time. However, given the practical computing power, the complexity of 2^{128} is out of reach anyhow, while our method consuming a few thousands of minutes can be afforded.

When it comes to the search under the related-key setting, the initial condition should be updated, accordingly.

The searching scope of the output pattern. With the iteration of the round function, the differential patterns of the internal states gradually lose information. The output differential patterns that we are interested in are Z, N and N* since they carry useful information, which can be exploited in various attacks. Thus, the searching scope of the output pattern is set to assert that a certain word follows one of the three functional differential patterns. To be specific, after adding the assertion $\delta_{X_i^r} = 0$ (resp. $1, 2$) into the CSP, the solver searches for TDs with ΔX_i^r being zero (resp. nonzero and fixed, any value except zero).

Since the propagations of differences and linear masks are dual [SLG+16], the method for the search of truncated differentials can be adjusted to search for multidimensional linear approximations, naturally, and we omit it for the space limitation.

3.4 The Reason to Use the CP

We do not claim that the CP is the unique method which can accomplish the search of (RK) TDs and MDLAs. The first reason we employ the CP lies in its conciseness. Since it supports high-level descriptions, the construction of the model is comparatively transparent. To demonstrate this advantage, we try to reconstruct Model 1 with the MILP and SAT methods, respectively. Please refer to Supplementary Material A.2 for more details. Besides, since the CP model is more readable than the MILP and SAT models, it is convenient for us to locate mistakes in the model.

At last, we remind the readers that, for a given operation, the way to generate the CP model is not unique, and we only provide one option.

4 Related-Key Differential-Linear Attack on AES-192

Differential-linear (DL) cryptanalysis [LH94] attempts to create a long distinguisher by connecting a short differential trail and a short linear approximation for primitives that are immune to differential and linear attacks. It is a combined attack and can be regarded as linear cryptanalysis under the chosen plaintext attack scenario. Related-key differential-linear cryptanalysis introduces a difference in the master key and manages to exploit potential weaknesses in the encryption and key schedule, simultaneously.

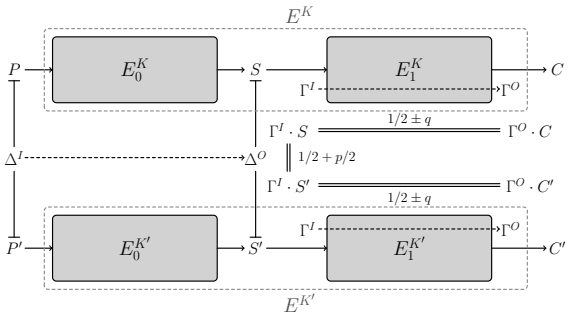


Figure 2: Related-key differential-linear distinguisher.

As in Figure 2, let E^K be a cipher with the secret key K that can be decomposed into a cascade, i.e., $E^K = E_1^K \circ E_0^K$. Denote the plaintexts as P, P', the ciphertexts as C, C', and the intermediate states between E_0 and E_1 as S, S', respectively. Suppose that

we have a related-key differential $(\Delta^I, \Delta^K) \xrightarrow{p} \Delta^O$ for E_0 with probability p and a linear approximation $\Gamma^I \xrightarrow{\pm q} \Gamma^O$ for E_1 with bias $\pm q$. To launch the distinguishing attack, the attacker sieves the plaintext pair (P, P') respectively encrypted with K and K' satisfying $P \oplus P' = \Delta^I$, $K \oplus K' = \Delta^K$ and checks whether the corresponding ciphertext pair (C, C') validates the equation $\Gamma^O \cdot C \oplus \Gamma^O \cdot C' = 0$. Denote the overall bias of the DL distinguisher as $\mathcal{E}_{\Delta^I,\Gamma^O}$. Under some randomness assumptions [BDKW19], we have $\mathcal{E}_{\Delta^I,\Gamma^O} \approx 2pq^2$.

While many subsequent works of the DL attack endeavoured to formalise basic assumptions [BLN17] and explored its generalisation [LGZL09, Lu15], the dependence between E_0 and E_1 is seldom to be studied. Enlightened by the boomerang connectivity table [CHP+18] intended to exploit the dependence between the subciphers in the boomerang attack [Wag99], Bar-On et al. [BDKW19] proposed the conception of differential-linear connectivity table (DLCT) to handle the dependence in the DL attack. With a more accurate expression for the overall bias of the DL distinguisher, the deviations of the overall biases in several instantiations [DIK08, DEMS15] get reasonable explanations.

In this section, we revisit the related-key differential-linear attack on AES-192 [ZZWF07]. Since the 5-round RK DL distinguisher in [ZZWF07] (see Figure 3) relies on a 4-round RK TD with probability one for the first subcipher, we try to implement the method in Sect. 3 to search for better RK TDs for AES-192 so that we can construct better related-key DL distinguishers. After the complete search with the model in Sect. 3 traversing all possible combinations of differences for the plaintext and master key, we find that the length of the optimal TD cannot be extended due to the well-designed diffusion layer. Nevertheless, we discover another non-trivial property in the DL distinguisher[6], which can be verified with fewer internal bytes. With this new feature, we improve the complexity of the previous key-recovery attack. The theoretical result is confirmed with random tests. Furthermore, the new distinguisher is explained from the aspect of DLCT.

4.1 Improved RK DL Attack on AES-192

The AES algorithm [DR02] encrypts 128-bit data blocks and accepts 128, 192, and 256-bit keys, which are written as AES-128, AES-192, and AES-256, respectively. The internal state is arranged in a 4×4 byte matrix, and the round function updates the state by applying the four operations `SubBytes` (`SB`), `ShiftRows` (`SR`), `MixColumns` (`MC`), and `AddRoundKey` (`AK`), sequentially. We denote the input of round i as x_i^I and use x_i^S, x_i^R, x_i^M, x_i^O to represent the internal values after the applications of `SB`, `SR`, `MC`, `AK` operations of round i, respectively. Let k_i be the subkey of the i-th round. In some cases, we interchange the order of `MC` and `AK` and denote the equivalent subkey of k_i as $wk_i \triangleq \texttt{MC}^{-1}(k_i)$. The internal state after the `AddRoundKey` operation with the equivalent subkey wk_i is signified as x_i^W. For a state x_*^*, $x_*^*[i,j]$ is referred to as the byte in the i-th row and j-th column, where $0 \leqslant i, j \leqslant 3$.

Previous distinguishing property. As in Figure 3, the differences ΔP and Δk_0 cancel each other out so that zero difference in x_1^I is maintained after two rounds of encryption. With the propagation of the difference, $\Delta x_4^R[0,0] \triangleq \widetilde{\delta}$ is nonzero, although its value is unknown. The following `MixColumns` operation guarantees $\Delta x_4^M[1,0] = \Delta x_4^M[2,0] = \widetilde{\delta}$. In [ZZWF07], the authors appended a 1-round linear approximation to the 4-round related-key truncated differential. To make sure that $\Delta x_5^I \cdot \Gamma x_5^I = 0$ holds with probability one, they set $\Gamma x_5^I[1,0] = \Gamma x_5^I[2,0] = \widetilde{\lambda} \neq 0$, $\Gamma x_5^I[i,j] = 0$ for the remaining bytes. Then, with experimental verifications, the authors observed that for each nonzero 8-bit linear mask λ, the bias of the linear equation

$$\lambda \cdot (\Delta x_5^W[1,3] \oplus \Delta x_5^W[2,2]) = 0 \tag{1}$$

[6] We do not claim the new DL distinguisher is optimal.

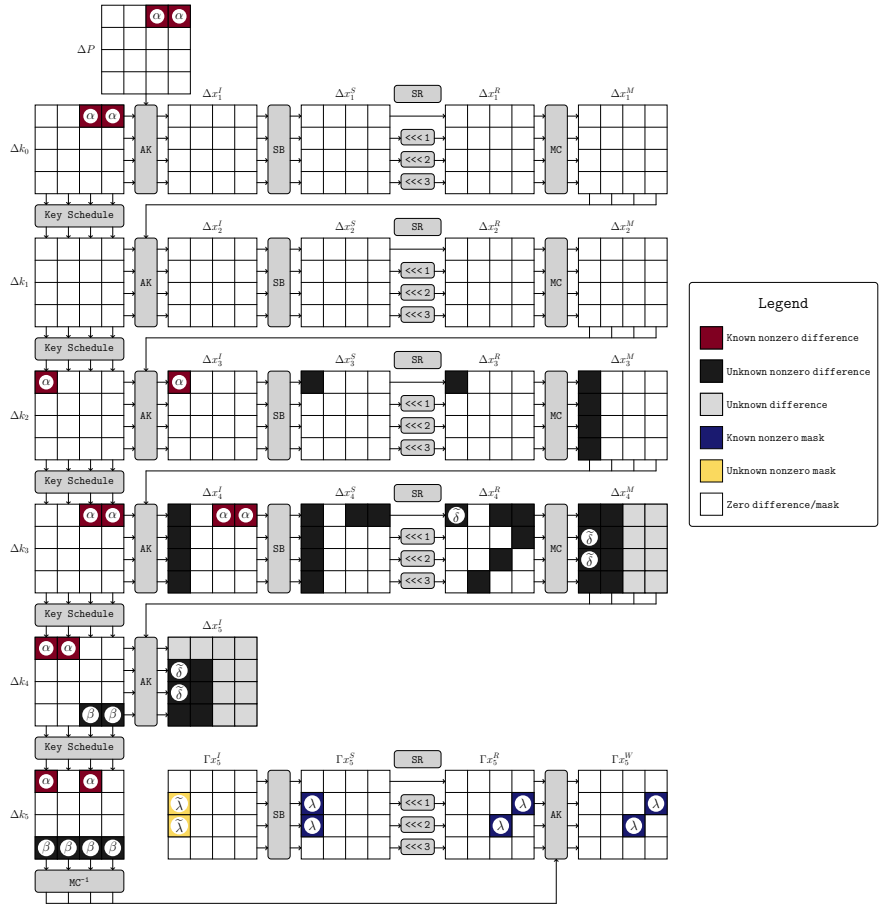


Figure 3: 5-round related-key differential-linear distinguisher [ZZWF07].

is about 2^{-9}. Based on this distinguisher, they proposed a 7-round key-recovery attack with 2^{22} chosen plaintexts. Note that the distinguishing property should be confirmed with the two bytes $x_5^W[1,3]$ and $x_5^W[2,2]$.

New distinguishing property. The new distinguishing property only depends on one output byte of the TD with a nonzero difference. We take $\Delta x_5^I[1,0]$ as an example. It can be observed that the difference of this byte stems from nonzero difference $\Delta x_3^I[0,0]$, and the difference is propagated as follows,

$$\Delta x_3^I[0,0] = \alpha \xrightarrow[S]{} \Delta x_3^S[0,0],\ \left(\texttt{0x02}\cdot\Delta x_3^S[0,0]\right) \xrightarrow[S]{} \widetilde{\delta},\ \widetilde{\delta} \xrightarrow[S]{} \Delta x_5^S[1,0].$$

Since there are about 128 possible output differences under one fixed input difference of the S-box, it is reasonable to assume that $\Delta x_5^S[1,0]$ takes all 255 nonzero values with equal probability[7]. Thus, for any nonzero mask λ, the linear equation $\lambda\cdot\Delta x_5^S[1,0]=0$ holds with probability 127/255. The absolute value[8] of the bias for the linear equation

$$\lambda\cdot\Delta x_5^W[1,3]=0 \tag{2}$$

[7]The intuitive explanation and experimental verification of this artificial randomness property are provided in Supplementary Material B.1

[8]The sign of the bias is related to the value of $\texttt{MC}^{-1}(\Delta k_5)[1,3]$.

is about $\varepsilon = 2^{-8.99}$, which is a nonrandom property and allows us to launch a distinguishing attack. Considering the complexity of the distinguishing attack is influenced by the overall bias of the DL distinguisher, and the biases of Eq. (1) and Eq. (2) are almost the same, the complexity of the distinguishing attack with the new property basically remains unchanged. However, the complexity of the key-recovery attack drops because less key bytes get involved in the key-recovery phase.

Improved key-recovery attack. Given N pairs of plaintexts, we use a counter Σ to record the number of times that Eq. (2) is fulfilled. With the Central Limit Theorem, the statistic $|\Sigma/N - 0.5|$ follows the normal distribution $\mathcal{N}(\varepsilon, 1/4N)$ when the oracle corresponds to the real cipher; otherwise, $|\Sigma/N - 0.5|$ follows the folded normal distribution $\overline{\mathcal{N}}(0, 1/4N)$. These theoretical distributions are checked with random tests, and the results can be found in Supplementary Material B.2.

With the distributions of the statistic, the complexity of the distinguishing attack can be accurately quantified by the statistical hypothesis testing method. To be specific, we set the threshold as τ. The null assumption that the oracle is an encryption algorithm will be accepted if $|\Sigma/N - 0.5| > \tau$. Denote α_0 (resp. α_1) the probability that we wrongfully discard the cipher (resp. accept the random permutation). With the method in [BN17], the complexity of the distinguisher is $N = \frac{(q_{1-\alpha_0}+q_{1-\alpha_1/2})^2}{4\varepsilon^2}$, where $q_{1-\alpha_0}$ and $q_{1-\alpha_1/2}$ are the quantiles of the standard normal distribution evaluated at $1-\alpha_0$ and $1-\alpha_1/2$. We set $\alpha_0 = 2^{-4.32}$, $\alpha_1 = 2^{-8}$ and compute the success probability $P_S \approx 95\%$ and $N \approx 2^{20.3}$. In summary, the key-recovery attack requires $2^{21.3}$ chosen plaintexts, and the time complexity is reduced from 2^{187} to $2^{170.5}$. The detailed attack procedure can be found in Supplementary Material B.3

4.2 Explanation of the New Distinguisher with DLCT

From the viewpoint of DLCT [BDKW19], we decompose the 5-round encryption covered by the distinguisher as $E = E_m \circ E_0'$, where E_0' covers the first four rounds, and E_m is referred to as the fifth round. With the property of DLCT [CKW19, Nyb19], we know that

$$\sum_{\widetilde{\delta}\in\mathbb{F}_2^8\setminus\{0\}} \mathrm{DLCT}_S\left(\widetilde{\delta}, \lambda\right) = -128 \text{ for all } \lambda \neq 0,$$

holds for all 8-bit bijective S-boxes. According to Eq. (5) in [BDKW19], for any nonzero mask λ, the overall bias of the DL distinguisher should be computed as

$$\begin{aligned}\mathcal{E}_{\Delta^I,\Gamma^O} &= \left(\sum_{\Delta\in\mathbb{F}_2^n} \Pr\left[\left(\Delta^I, \Delta^K\right) \xrightarrow[E_0']{} \Delta\right] \cdot \frac{\mathrm{DLCT}_{E_m}\left(\Delta, \Gamma^O\right) + 2^{n-1}}{2^n}\right) - \frac{1}{2} \\ &= \left(\sum_{\widetilde{\delta}\in\mathbb{F}_2^8\setminus\{0\}} \frac{1}{255} \cdot \frac{\mathrm{DLCT}_S\left(\widetilde{\delta}, \lambda\right) + 2^7}{2^8}\right) - \frac{1}{2} \approx -2^{-8.99},\end{aligned} \tag{3}$$

where

$$\mathrm{DLCT}_{E_m}(\Delta, \Gamma^O) \triangleq \left|\left\{x \in \mathbb{F}_2^n \;\middle|\; \Gamma^O \cdot E_m(S) = \Gamma^O \cdot E_m(S \oplus \Delta)\right\}\right| - 2^{n-1}.$$

Thus, the new distinguisher is explained from the aspect of DLCT[9] and can be regarded as the first application of DLCT in the related-key attack scenario.

[9] We remind the readers that Eq. (3) is based on the randomness assumption upon $\Delta x_5^S[1, 0]$.

More related-key truncated differentials with probability one. Although the length of the optimal RK TD cannot be extended with the method in Sect. 3, we get three more distinguishers apart from the one in [ZZWF07]. Please find in Supplementary Material B.4 for more information.

5 Constructing IDs with TDs and ZCLAs with MDLAs

In this section, we illustrate how to use the method in Sect. 3 to construct impossible differentials and zero-correlation linear approximations for SPN ciphers. We will start with a basic version, which applies the miss-in-the-middle approach. Then, the basic tool is advanced so that the solver may recognise more categories of contradictions. At last, a comprehensive comparison of all available searching tools targeting (related-key) impossible differentials for SPN ciphers is supplemented. Since the search of impossible differentials and zero-correlation linear approximations can be carried out similarly, we only elaborate on the search of impossible differentials and leave the search of zero-correlation linear approximations as a trivial generalisation.

5.1 Basic Tool Relying on Miss-in-the-Middle Approach

The basic method is motivated by the $\mathcal{U}$-method [KHS+03]. With the miss-in-the-middle approach [Bir05], after constructing two deterministic truncated differentials $\Delta^{I_1} \xrightarrow[R_1\text{-round}]{} \Delta^{O_1}$ and $\Delta^{O_2} \xleftarrow[R_2\text{-round}]{} \Delta^{I_2}$ in opposite directions, we check the compatibility of the two output differential patterns Δ^{O_1} and Δ^{O_2}. If there exists at least one word (e.g., the i-th word) such that the corresponding difference of the pattern $\Delta^{O_1}[i]$ cannot be interpreted by $\Delta^{O_2}[i]$, $\Delta^{I_1} \nrightarrow \Delta^{I_2}$ forms an $(R_1 + R_2)$-round impossible differential distinguisher. The position of the internal state that the two truncated differentials intersect with each other is called the *meeting point*.

Aside from the way to implement the search, the distinction between our tool and the $\mathcal{U}$-method is the set of differential patterns applied to yield contradictions. The $\mathcal{U}$-method considers the set $\mathcal{U} = \{\mathsf{Z}, \mathsf{N}, \mathsf{N} \oplus \mathsf{N}^*, \mathsf{N}^*\}$, while we take the smaller set $\mathcal{U}^* = \{\mathsf{Z}, \mathsf{N}, \mathsf{N}^*\}$. In this sense, we name our approach the $\mathcal{U}^*$*-method*. At first sight, our tool is weaker than the $\mathcal{U}$-method since $\mathcal{U}^* \subsetneq \mathcal{U}$ implies that the contradictions recognised by the $\mathcal{U}^*$-method only occupy a part of those identified by the $\mathcal{U}$-method. However, as we mentioned at the beginning of Sect. 3, for any input differential pattern, after passing through at most two nonlinear layers, $\mathsf{N} \oplus \mathsf{N}^*$ disappears from the pattern propagating phase of SPN ciphers under the single-key setting. Under the related-key attack scenario, since the survival of $\mathsf{N} \oplus \mathsf{N}^*$ can be inferred from the existence of N, the contradictions related to the pattern $\mathsf{N} \oplus \mathsf{N}^*$ can be derived after identifying the position of N that results in this nonzero varied pattern. This procedure can be finished by fine-tuning the searching program. Therefore, the $\mathcal{U}^*$-method has almost the same performance as the $\mathcal{U}$-method regarding SPN ciphers.

With a similar approach as in the $\mathcal{U}$-method, for the element X in $\mathcal{U}^*$, we define its *auxiliary set* $\overline{\mathsf{X}} \subset \mathcal{U}^*$ as $\overline{\mathsf{Z}} = \{\mathsf{N}, \mathsf{N}^*\}$, $\overline{\mathsf{N}} = \{\mathsf{Z}, \mathsf{N}\}$, $\overline{\mathsf{N}^*} = \{\mathsf{Z}\}$. Note that the corresponding differences of the patterns in $\overline{\mathsf{X}}$ cannot be interpreted by X. The two patterns X and Y are said to be *compatible* with each other if $\mathsf{Y} \notin \overline{\mathsf{X}}$. Given an input difference α, let $\Delta_\alpha^i = (\Delta_\alpha^i[0], \Delta_\alpha^i[1], \ldots, \Delta_\alpha^i[\ell - 1])$ be the differential pattern after i rounds of encryption. For the output difference β, we denote $\Delta_\beta^j = (\Delta_\beta^j[0], \Delta_\beta^j[1], \ldots, \Delta_\beta^j[\ell - 1])$ the differential pattern after j rounds of decryption. With these notations, the maximum number of rounds regarding impossible differential distinguishers identified by the $\mathcal{U}^*$-method is summarised in the following proposition.

Proposition 1. *For all possible nonzero input differences α, the maximum number of encryption rounds such that the differential pattern of the i-th subblock in the internal state*

follows the pattern X *is denoted as*

$$\mathcal{E}(\mathsf{X})[i] = \max_{\alpha \neq 0} \left\{ r \geqslant 0 \;\middle|\; \Delta_\alpha^r[i] = \mathsf{X} \right\}.$$

Similarly, for all possible nonzero output differences β*, the maximum number of decryption rounds so that the differential pattern of the* i*-th subblock in the internal state satisfies the pattern* X *is denoted as*

$$\mathcal{D}(\mathsf{X})[i] = \max_{\beta \neq 0} \left\{ r \geqslant 0 \;\middle|\; \Delta_\beta^r[i] = \mathsf{X} \right\}.$$

The maximum number of rounds for impossible differential distinguishers that can be discovered with the $\mathcal{U}^*$*-method is*

$$\mathcal{L} = \max_{0 \leqslant i \leqslant \ell - 1} \left\{ \mathcal{E}(\mathsf{X})[i] + \mathcal{D}(\mathsf{Y})[i] \;\middle|\; \mathsf{X} \in \mathcal{U}^*, \mathsf{Y} \in \overline{\mathsf{X}} \right\}.$$

Algorithm 1 Basic method for the search of impossible differential distinguishers

Require: CSPs $\mathtt{TD}^{\mathtt{cipher}}_{\langle \mathtt{Forward} \rangle}$ and $\mathtt{TD}^{\mathtt{cipher}}_{\langle \mathtt{Backward} \rangle}$ for the primitive `cipher`
Ensure: The length $\mathcal{L}$ of the optimal impossible differential distinguisher

for all $0 \leqslant i \leqslant \ell - 1, \mathsf{X} \in \mathcal{U}^*$ **do**
  $\mathcal{E}(\mathsf{X})[i] = 0$, $\mathcal{D}(\mathsf{X})[i] = 0$
end for
for all PROPAGATION $\in$ {FORWARD, BACKWARD} **do**
  for all $\mathsf{X} \in \mathcal{U}^*$ **do**
    for $i = 0; i < \ell; i++$ **do**
      $r = 1$, $flag =$ true
      while $flag \equiv$ true **do**
        $flag =$ PROPAGATION$(r, \Delta_\alpha^r[i] = \mathsf{X})$
        $r = r + 1$
      end while
      if PROPAGATION $\equiv$ FORWARD **then**
        $\mathcal{E}(\mathsf{X})[i] = r - 1$
      else
        $\mathcal{D}(\mathsf{X})[i] = r - 1$
      end if
    end for
  end for
end for
$\mathcal{L} = \max \left\{ \mathcal{E}(\mathsf{X})[i] + \mathcal{D}(\mathsf{Y})[i] \;\middle|\; 0 \leqslant i \leqslant \ell - 1, \mathsf{X} \in \mathcal{U}^*, \mathsf{Y} \in \overline{\mathsf{X}} \right\}$
return $\mathcal{L}$

function FORWARD(r, obj)
  if $\mathtt{TD}^{\mathtt{cipher}}_{\langle \mathtt{Forward} \rangle}(r, obj)$ solvable **then**
    return true
  else
    return false
  end if
end function

function BACKWARD(r, obj)
  if $\mathtt{TD}^{\mathtt{cipher}}_{\langle \mathtt{Backward} \rangle}(r, obj)$ solvable **then**
    return true
  else
    return false
  end if
end function

A complete description of the $\mathcal{U}^*$-method can be found in Algorithm 1. With the method in Sect. 3, for a specific primitive `cipher`, we generate two CSPs $\mathtt{TD}^{\mathtt{cipher}}_{\langle \mathtt{Forward} \rangle}$ and $\mathtt{TD}^{\mathtt{cipher}}_{\langle \mathtt{Backward} \rangle}$, which search for truncated differentials in the forward and backward directions,

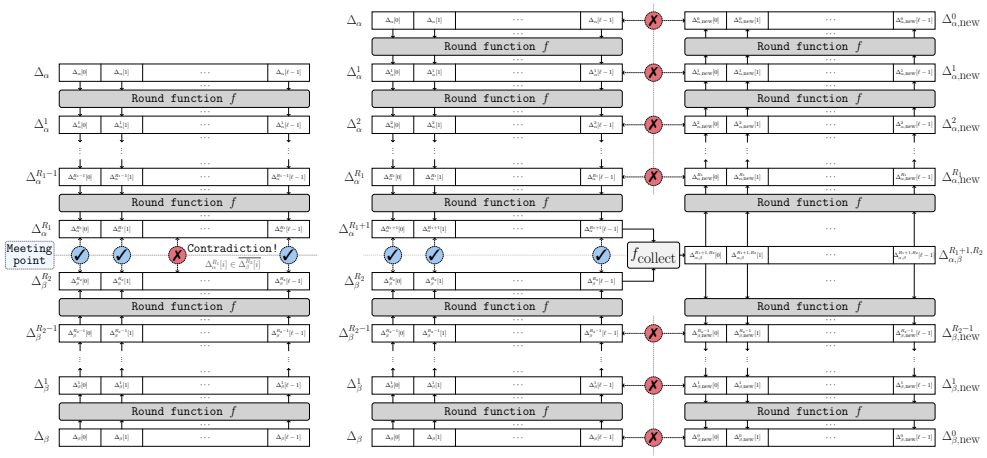
respectively. The values of $\mathcal{E}(\mathsf{X})[i]$ and $\mathcal{D}(\mathsf{X})[i]$ can be determined by invoking the CP solver under different parameter settings. Then, we acquire the length $\mathcal{L}$ of the optimal impossible differential distinguisher.

5.2 Optimising IDs and ZCLAs Obtained with Algorithm 1

As discussed in [WW12], the $\mathcal{U}$-method only centres on the contradictions at the meeting point (see Figure 4(a)). However, in some cases, although the two differential patterns at the meeting point are compatible with each other, the given input and output differences still encompass inconsistencies. Based on this observation, we intend to generalise the basic approach so that we can detect impossible differentials with contradictions belonging to the category illustrated in Figure 4(b). Before we look into the details of the optimising method, we introduce the definition of *message collecting function*, which is used to unify information of two compatible differential patterns.

Definition 2 (Message Collecting Function)**.** The message collecting function f_{collect} is a function over two differential patterns Δ_X and Δ_Y with $\Delta_Y \notin \overline{\Delta_X}$. The output $f_{\text{collect}}(\Delta_X, \Delta_Y)$ is a differential pattern, and its evaluations regarding two inputs Δ_X and Δ_Y are specified in the following table. The position of the token '$\times$' corresponds to the case where (Δ_X, Δ_Y) does not fall into the domain of f_{collect}.

$f_{\text{collect}}(\Delta_X, \Delta_Y)$		Δ_Y			
		Z	N	N*	U
Δ_X	Z	Z	$\times$	$\times$	Z
	N	$\times$	N	N	N
	N*	$\times$	N	N*	N*
	U	Z	N	N*	U



(a) Type-I contradiction. (b) Type-II contradiction.

Figure 4: Different categories of contradictions.

Now, we consider the possibility of extending the optimal trail found with Algorithm 1. Suppose that $\alpha \nrightarrow \beta$ is an $R(= R_1 + R_2)$-round impossible differential distinguisher returned by Algorithm 1. Denote Δ_α and Δ_β the input and output differential patterns, respectively. The miss-in-the-middle approach implies that the two patterns $\Delta_\alpha^{R_1}$ and $\Delta_\beta^{R_2}$ at the meeting point contradict with each other. We wonder whether $\alpha \nrightarrow \beta$ constitutes an $(R+1)$-round impossible differential.

As in Figure 4(b), we extend the R_1-round subcipher by one round and claim that $\Delta_\alpha^{R_1+1}$ and $\Delta_\beta^{R_2}$ are compatible since Algorithm 1 never leaves out the contradiction at the meeting point. Then, we apply f_{collect} to the two patterns $\Delta_\alpha^{R_1+1}$ and $\Delta_\beta^{R_2}$ and denote the output pattern as $\Delta_{\alpha,\beta}^{R_1+1,R_2}$, i.e., $\Delta_{\alpha,\beta}^{R_1+1,R_2}[i] = f_{\text{collect}}(\Delta_\alpha^{R_1+1}[i], \Delta_\beta^{R_2}[i])$ for all $0 \leqslant i \leqslant \ell - 1$. Since the message collecting function unifies the information of $\Delta_\alpha^{R_1+1}$ and $\Delta_\beta^{R_2}$, $\Delta_{\alpha,\beta}^{R_1+1,R_2}$ should contain the information of the input pattern Δ_α and the output pattern Δ_β, simultaneously. Then, we propagate $\Delta_{\alpha,\beta}^{R_1+1,R_2}$ in the forward and backward directions, respectively, and use $\Delta_{\alpha,\text{new}}^{R_1+1-r}$ (resp. $\Delta_{\beta,\text{new}}^{R_2-r}$) to represent the pattern after r rounds of decryption (resp. encryption). Note that $\Delta_{\alpha,\text{new}}^{r_1}$ carries the information originating from the output pattern Δ_β, and $\Delta_{\beta,\text{new}}^{r_2}$ keeps the information deriving from the input pattern Δ_α. Therefore, the inconsistencies between $\Delta_{\alpha,\text{new}}^{r_1}$ and $\Delta_\alpha^{r_1}$ and (or) $\Delta_{\beta,\text{new}}^{r_2}$ and $\Delta_\beta^{r_2}$ somehow exhibit the inconsistency between Δ_α and Δ_β. If $\Delta_\alpha \to \Delta_\beta$ is a possible TD over $(R+1)$ rounds of encryption, then

$$\Delta_{\alpha,\text{new}}^{r_1}[i] \notin \overline{\Delta_\alpha^{r_1}[i]}, \Delta_{\beta,\text{new}}^{r_2}[i] \notin \overline{\Delta_\beta^{r_2}[i]} \text{ for all } 0 \leqslant r_1 \leqslant R_1, 0 \leqslant r_2 < R_2, 0 \leqslant i \leqslant \ell - 1. \quad (4)$$

In contrast, if there is at least one assertion in Eq. (4) is not satisfied, the consistency between Δ_α and Δ_β is broken, and $\alpha \nrightarrow \beta$ turns out to be an $(R+1)$-round impossible differential.

Likewise, we can extend the R_2-round subcipher by one round and explore the feasibility of optimising the trail. These procedures can be converted into CSPs. With this optimising method, we extend some 10.5-round zero-correlation linear approximations of `SKINNY` to 11.5-round ones. For more details, please refer to Sect. 6.1, Supplementary Material C and attached source codes.

5.3 Comparison of All Tools Targeting (RK) IDs of SPN Ciphers

The first automated tool for the search of TD IDs was proposed by Kim et al. [KHS+03], which was based on the miss-in-the-middle approach [Bir05] and was known as the $\mathcal{U}$-method (①). The round function of the block cipher structure is replaced with matrix representation, and the propagation of difference inside the cipher is transformed into matrix multiplication. After generating two TDs with fixed input differential pattern in reversed directions, the compatibility of the two output differential patterns is analysed. An impossible differential is encountered if inconsistencies are spotted at the meeting point. Later, Luo et al. [LLWG14] introduced the UID-method (②), which generalised the $\mathcal{U}$-method by cancelling the 1-property matrix[10] requirement on the characteristic matrix of the round function and incorporating more kinds of inconsistencies. After formulating an equation system depicting differential propagation inside the objective primitive, Wu and Wang (③) [WW12] put forward a novel tool targeting truncated impossible differentials for word-oriented block ciphers. This tool further generalises the $\mathcal{U}$-method and UID-method. It allows us to narrow the gap between the optimal IDs obtained with previous programmed methods and the best ones relying on sophisticated cryptanalytic experience, although it does not improve the lengths of IDs for existing block ciphers.

In the last decade, with the introduction of automatic tools for cryptanalysis based on the MILP [SHW+14a, SHW+14b, AST+17], SAT/SMT [KLT15, LWR16, SWW18] and CP [SGL+17, GLMS18], the constructions of distinguishers in differential, linear as well as integral cryptanalyses become much more convenient. These tools significantly reduce the workload of cryptographers in designing and analysing phases of cryptographic primitives. Based on the automatic tool for differential cryptanalysis [SHW+14b], two independent works [CJF+16, ST17] studied its applicability for the search of impossible differentials. The

[10] If the number of '1' in each column of the encryption or decryption characteristic matrix is zero or one, the matrix is called a 1-property matrix. See [KHS+03, LLWG14] for the definition of the matrix.

main idea is adding (in)equations to fix the input and output differences in the automatic model [SHW+14b] and verifying the solvability of the corresponding MILP problem with some well-developed solvers. If the problem is unsolvable, an impossible differential is discovered. Moreover, Sasaki and Todo (④) [ST17] provided a deeper understanding of ID from design and cryptanalysis aspects, while Cui et al. [CJF+16] mainly focused on the utility for ARX structure. In terms of CP, Sun et al. (⑤) [SGL+17] realised the search of (related-key) impossible differential characteristics, which is also adapted from a differential characteristic searching scheme by fixing the input and output differences with specific values. So, this method is also faced with the problem of incomplete search and is only performed under input and output differences with low Hamming weight, typically.

The advantage of our tool over all of the previous methods rests in that it supports an exhaustive search for all possible combinations of input and output differences as the input and output differential patterns are not predetermined in our model. For the tools ① ② ③ ④ holding the function of finding TD IDs, the exhaustive search with complexity $\mathcal{O}(2^{2\ell})$ is afforded only when the number ℓ of words in the internal state is not very large. As a result, for the case of Minalpher-P [STA+14] with $\ell = 64$ in Sect. 6, the tools ① ② ③ ④ only perform some inadequate searches under input and output differences satisfying some specific patterns, while our method achieves a complete search.

Compared to the programmed tools ① ② ③ aiming at TD IDs, the superiority of our system originates from the control of the pattern N. As we mentioned previously, if the differential pattern of an internal word X_i is N, its actual difference ΔX_i is recorded in ζ_{X_i}. Accompanied by the nature of exhaustive search, the new method can identify possible cancellations among two or more nonzero fixed patterns during the first few rounds when the message in the pattern N is not destroyed by some certain operations in the round function. Consequently, we can get longer impossible differential distinguishers.

Theoretically, considering the kinds of inconsistencies detected by the tool, the performance of ② is better than that of ① owing to the refined manipulation of the pattern $\mathsf{N} \oplus \mathsf{N}^*$. However, when the objective is restricted to SPN ciphers, these two methods behave almost the same since $\mathsf{N} \oplus \mathsf{N}^*$ disappears from the pattern propagation phase after passing through at most two nonlinear layers. In this sense, the $\mathcal{U}^*$-method works better than ① and ② concerning its quality of complete search. The method ③ is preferred over ② because it is capable of identifying Type-II contradiction in Figure 4(b), which is also in the range of the optimised $\mathcal{U}^*$-method. Along with the inherent feature of exhaustive search, the optimised $\mathcal{U}^*$-method attains more excellent performance than ③.

The tools ④ (in 'specific S-box mode') and ⑤ depict the differential propagation in primitives at the bit-level. They can catch any contradictions since the inconsistencies in the system are automatically recognised by MILP and CP solvers. Therefore, when the input and output differences are limited to vectors with low Hamming weight, these tools indeed maximise the number of rounds for impossible differential characteristics and thus may outperform the optimised $\mathcal{U}^*$-method. However, the optimality of the distinguisher cannot be guaranteed for the incompetence of exhaustive search. In addition, only ⑤ and the (optimised) $\mathcal{U}^*$-method support the search of distinguishers under the related-key attack scenario. Please find in Table 1 a comparison of all tools targeting (RK) IDs for SPN ciphers.

6 Finding (RK) IDs and ZCLAs with the CP Method

In this section, we apply the method in Sect. 5 to search for (related-key) impossible differentials and zero-correlation linear approximations of several SPN ciphers, including `SKINNY`, `Midori64`, and Minalpher-P. The source codes of the searching programs can be found at https://github.com/Deterministic-TD-MDLA/auxiliary_material. All tests are implemented with one processor Intel® Xeon® Gold 5118 CPU @ 2.30GHz. For

Table 1: *Top*: Comparison of all tools targeting (RK) IDs for SPN ciphers. *Bottom*: Explanations of properties.

	Properties							
Method	*P1: 1-property*	*P2: DDT*	*P3: truncated*	*P4: 8-bit S-box*	*P5: fixed*	*P6: exhaustive*	*P7: RK ID*	Ref.
$\mathcal{U}$-method ①	★		★	★	★	✯		[KHS+03]
UID-method ②			★	★	★	✯		[LLWG14]
Wu and Wang ③			★	★	★	✯		[WW12]
Sasaki and Todo ④		★	★	★	★	✯		[ST17]
Sun et al. ⑤		★			★		★	[SGL+17]
(Optimised) $\mathcal{U}^*$-method			★	★		★	★	Sect.5

Properties	Explanations
P1: 1-property	The encryption and decryption characteristic matrices of the block cipher structure must be 1-property matrices.
P2: DDT	The method can take differential distribution table (DDT) of the S-box into consideration and provides a more accurate description for the differential propagation of the S-box.
P3: truncated	The method can search for truncated IDs. In other words, the resulting impossible differential distinguisher is valid for any bijective S-boxes.
P4: 8-bit S-box	The method supports the search for ciphers with 8-bit S-box.
P5: fixed	The search is implemented under the fixed input and output differences.
P6: exhaustive	The method can exhaustively check all possible combinations of input and output patterns. The symbol ✯ indicates that the exhaustive search is capable only when the number ℓ of words in the internal state is not very large.
P7: RK ID	The method can be used to search for impossible differential distinguishers under the related-key attack scenario.

`SKINNY` and `Midori64`, the CSPs are solved with the solver Gecode, and all programs finish in several seconds. For Minalpher-P, we use the solver OSICBC, and it takes several minutes to return the result due to the considerable state size.

6.1 Applications to `SKINNY`

`SKINNY` [BJK+16] is a family of tweakable block cipher proposed at CRYPTO 2016. It has 64-bit and 128-bit versions, and the internal state is viewed as a 4×4 array of cells in both versions. The construction of `SKINNY` is based on the tweakey framework [JNP14], and the tweakey size t can be n, $2n$, or $3n$, where n stands for the block size. The encryption algorithm with n-bit block and t-bit tweakey is written as `SKINNY`-n-t. Please find in Supplementary Material C.1 and [BJK+16] for more details about `SKINNY`.

6.1.1 Impossible Differentials

Previous cryptanalysis. The designers [BJK+16] investigated all impossible differentials with one active cell in both input and output differences and proposed an 11-round distinguisher. Later, under the assumption that each subkey is chosen independently and uniformly at random, Sasaki and Todo [ST17] proved that the maximum number of rounds for impossible differential trails with one active nibble in both input and output differences for `SKINNY`-64 is less than 12.

12.5-round impossible differentials with the optimised $\mathcal{U}^*$-method. Since the search of truncated differentials only employs the bijectivity of the S-box, the impossible differentials

for SKINNY-64-$*$ also operate for SKINNY-128-$*$. Thus, we only execute the program for SKINNY-64-$*$. Besides, to propagate the difference for more rounds, we remove the first SubCells operation in the forward direction, which is also adopted in [SMB18]. After invoking Algorithm 1, we know that $\mathcal{L} = 12.5$ and obtain twelve 12.5-round impossible differentials[11]. The output of Algorithm 1 and the concrete distinguishers can be found in Supplementary Material C.2.

In the following theorem, we reconsider the provable security of SKINNY against impossible differentials and update the security bound under a certain assumption. Please find the proof of the theorem in Supplementary Material C.3.

Theorem 1. *Under the keyed (uniform) bijective S-box assumption,* 13.5*-round encryption of SKINNY is secure against impossible differentials with arbitrary nonzero input and output differences.*

6.1.2 Related-Tweakey Impossible Differentials for SKINNY-n-n

In the related-tweakey setting, we only consider the case of $t = n$ since the tweakey schedules of $t = 2n$ and $t = 3n$ involve bit-wise operations.

Previous cryptanalysis. With the MILP and CSP methods, Liu et al. [LGS17] and Sun et al. [SGL+17] proposed 12-round related-tweakey impossible differentials for SKINNY-n-n, independently. Both of these methods restrict that the input, output and tweakey differences have at most one active cell. Motivated by the observation that the positions of active cells in the input, output and tweakey influence the key-recovery attack, Sadeghi et al. [SMB18] revisited the construction of related-tweakey impossible differentials for SKINNY-n-n and proposed several 12.5-round distinguishers. Likewise, the input, output and tweakey differences are restricted to the states having at most one active cell.

New distinguishers with the $\mathcal{U}^*$-method. We apply the $\mathcal{U}^*$-method in the related-tweakey setting. The experimental results show that the optimal distinguisher covers 12.5-round. Apart from the six distinguishers mentioned in [SMB18], much more distinguishers are detected since the search is not implemented under fixed differences.

Further, we prove the following theorem, which claims that 13.5-round SKINNY-n-n is secure against related-tweakey impossible differentials under certain assumptions. The proof can be found in Supplementary Material C.4.

Theorem 2. 13.5*-round SKINNY-n-n is secure against related-tweakey impossible differentials with arbitrary nonzero input and output differences under the following assumptions:*

- *the S-box satisfies keyed (uniform) bijective assumption;*
- *the difference of tweakey only has one active cell.*

6.1.3 11.5-Round Zero-Correlation Linear Approximations

Under the restriction that the input and output masks have one active cell, Sadeghi et al. [SMB18] created 10-round zero-correlation linear approximations, manually. We adjust Algorithm 1 to the ZCLA searching mode and notice that the optimal zero-correlation linear approximations obtained with the $\mathcal{U}^*$-method achieve 10.5-round. Then, the optimised $\mathcal{U}^*$-method is utilised to check the possibility of extending the distinguishers at hand. At last, we derive sixteen 11.5-round zero-correlation linear approximations[12] for SKINNY, which can be found in Supplementary Material C.5.

[11] The new IDs are extensions with probability one of the ones that were given by the designers.
[12] These new ZCLAs are extensions with probability one of the ones that were proposed in [SMB18].

6.2 Provable Security of Midori64 against ID

Previous cryptanalysis. The designer [BBI+15] estimated that the maximum number of rounds for impossible differential characteristics is 7-round based on the 3-round full diffusion property. Specifically, they proposed several 6-round impossible differentials with one active nibble in the input and output differences. Later, Shahmirzadi et al. [SAS+17] provided two 6.5-round impossible differentials, one of which was exploited to launch key-recovery attacks. With the tool based on the MILP method, Sasaki and Todo [ST17] showed that under the subkey uniform assumption, 7-round Midori64 is secure against impossible differentials with one active nibble in the input and output differences.

Provable security against ID. With Algorithm 1, we identify that the optimal impossible differential distinguisher detected with the $\mathcal{U}^*$-method attains 6.5-round. In total, we derive 480 6.5-round impossible differentials, which are summarised in Supplementary Material D.3. Furthermore, with the output of Algorithm 1, we show that 7.5-round Midori64 is secure against impossible differentials under a particular assumption.

Theorem 3. *Under the keyed (uniform) bijective S-box assumption, 7.5-round Midori64 is secure against impossible differentials with arbitrary nonzero input and output differences.*

6.3 8.5-Round Impossible Differentials of Minalpher-P

Previous cryptanalysis. The designer [STA+14] identified a 6.5-round truncated impossible differential for Minalpher-P with the $\mathcal{U}$-method. The meeting point is positioned between the MixColumns and SubNibbles operations. With the tool based on the MILP method, Sasaki and Todo [ST17] found 1152 7.5-round impossible differentials.

8.5-round impossible differentials with the $\mathcal{U}^*$-method. We apply the $\mathcal{U}^*$-method to Minalpher-P, and the meeting point is placed between the XorMatrix and MixColumns operations. In aggregate, 600 8.5-round impossible differentials are returned. Considering the Hamming weight of the input and output differences is closely linked with the complexity of the key-recovery attack, we sieve impossible differentials with lower Hamming weight. The test results reveal that the minimum Hamming weight is 9, and the number of distinguishers meeting this requirement is 160. Please find in Supplementary Material E for more information.

7 Conclusion

In this paper, we explore the usage of deterministic TDs and MDLAs. An automatic method based on the CSP is put forward to accomplish the search of deterministic TDs and MDLAs. Since the new tool realises an exhaustive search, the long-standing problem of inadequate search for the optimal TD and MDLA is settled. It is applied to search for RK TDs of AES-192. A novel RK DL distinguisher is established, and the previous RK DL attack on AES-192 is improved. As the second application of the tool, we propose the (optimised) $\mathcal{U}^*$-method, which enables us to construct (RK) IDs and ZCLAs, automatically. The new searching scheme accomplishes the exhaustive search with the CP and considers possible cancellations of more active words in the input and output differences/masks during the forward and (or) backward propagating processes. This technique is implemented with several primitives and discovers longer distinguishers. Moreover, the provable security bounds of SKINNY and Midori64 against ID distinguishing attack are generalised.

We acknowledge that some of the newly identified distinguishers are extensions with probability one of the ones that were published in previous literature. However, we think that the centre of the paper is more the new technique. Since the new tool can realise the

exhaustive search in a surprisingly rapid manner, we hope it may play an essential role in the designing phase of new ciphers.

Constructing a unified framework that combines the key-recovery approach with the new tool must be nice work. Nonetheless, estimating the cost of the key-recovery approach in ID attack is of high technicality, and we find it is hard to create a general model that works for a wide range of ciphers. So, we leave it as future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The research leading to these results has received funding from the National Key Research and Development Program of China (Grant No. 2018YFA0704702), the Major Scientific and Technological Innovation Project of Shandong Province, China (Grant No. 2019JZZY010133), and the Qingdao Postdoctor Application Research Project (Grant No. 61580070311101).

References

[AST+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP modeling for (large) S-boxes to optimize probability of differential characteristics. *IACR Trans. Symmetric Cryptol.*, 2017(4):99–129, 2017.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 411–436, 2015.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 12–23, 1999.

[BCQ04] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. On multiple linear approximations. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 1–22, 2004.

[BDKW19] Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. DLCT: A new tool for differential-linear cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, pages 313–342, 2019.

[Bir05] Alex Biryukov. Miss-in-the-middle attack. In *Encyclopedia of Cryptography and Security*. 2005.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology*

Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, pages 123–153, 2016.

[BJV04] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, pages 432–450, 2004.

[BKR97] Johan Borst, Lars R. Knudsen, and Vincent Rijmen. Two attacks on reduced IDEA. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, pages 1–13, 1997.

[BLN17] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. Differential-linear cryptanalysis revisited. *J. Cryptology*, 30(3):859–888, 2017.

[BLNW12] Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. Integral and multidimensional linear distinguishers with correlation zero. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 244–261. Springer, 2012.

[BN14] Céline Blondeau and Kaisa Nyberg. Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 165–182, 2014.

[BN17] Céline Blondeau and Kaisa Nyberg. Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. *Des. Codes Cryptography*, 82(1-2):319–349, 2017.

[BR14] Andrey Bogdanov and Vincent Rijmen. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *Des. Codes Cryptography*, 70(3):369–383, 2014.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO '90*, volume 537 of *LNCS*, pages 2–21. Springer, 1990.

[BW12] Andrey Bogdanov and Meiqin Wang. Zero correlation linear cryptanalysis with reduced data complexity. In *FSE 2012*, pages 29–48, 2012.

[CHP+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 683–714, 2018.

[CJF+16] 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.

[CKW19] Anne Canteaut, Lukas Kölsch, and Friedrich Wiemer. Observations on the DLCT and absolute indicators. *IACR Cryptol. ePrint Arch.*, 2019:848, 2019.

[DEMS15] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of Ascon. In *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, pages 371–387, 2015.

[DIK08] Orr Dunkelman, Sebastiaan Indesteege, and Nathan Keller. A differential-linear attack on 12-round Serpent. In *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings*, pages 308–321, 2008.

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

[GLMS18] David Gérault, Pascal Lafourcade, Marine Minier, and Christine Solnon. Revisiting AES related-key differential attacks with constraint programming. *Inf. Process. Lett.*, 139:24–29, 2018.

[HCN09] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. Multidimensional extension of Matsui's algorithm 2. In *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, pages 209–227, 2009.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, pages 274–288, 2014.

[JR94] Burton S. Kaliski Jr. and Matthew J. B. Robshaw. Linear cryptanalysis using multiple approximations. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, pages 26–39, 1994.

[KB96] Lars R. Knudsen and Thomas A. Berson. Truncated differentials of SAFER. In *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings*, pages 15–26, 1996.

[KHS+03] Jongsung Kim, Seokhie Hong, Jaechul Sung, Changhoon Lee, and Sangjin Lee. Impossible differential cryptanalysis for block cipher structures. In *Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings*, pages 82–96, 2003.

[KLT15] Stefan Kölbl, Gregor Leander, and Tyge Tiessen. Observations on the SIMON block cipher family. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 161–185, 2015.

[Knu94] Lars R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, pages 196–211, 1994.

[Knu98] Lars Knudsen. DEAL-a 128-bit block cipher. *complexity*, 258(2):216, 1998.

[KRW99] Lars R. Knudsen, Matthew J. B. Robshaw, and David A. Wagner. Truncated differentials and Skipjack. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 165–180, 1999.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song. Security analysis of SKINNY under related-tweakey settings (long paper). *IACR Trans. Symmetric Cryptol.*, 2017(3):37–72, 2017.

[LGZL09] Zhiqiang Liu, Dawu Gu, Jing Zhang, and Wei Li. Differential-multiple linear cryptanalysis. In *Information Security and Cryptology - 5th International Conference, Inscrypt 2009, Beijing, China, December 12-15, 2009. Revised Selected Papers*, pages 35–49, 2009.

[LH94] Susan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, pages 17–25, 1994.

[LLWG14] 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.

[Lu15] Jiqiang Lu. A methodology for differential-linear cryptanalysis and its applications. *Des. Codes Cryptography*, 77(1):11–48, 2015.

[LWR16] Yunwen Liu, Qingju Wang, and Vincent Rijmen. Automatic search of linear trails in ARX with applications to SPECK and Chaskey. In *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, pages 485–499, 2016.

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

[Nyb19] Kaisa Nyberg. The extended autocorrelation and boomerang tables and links between nonlinearity properties of vectorial boolean functions. *IACR Cryptol. ePrint Arch.*, 2019:1381, 2019.

[SAS+17] Aein Rezaei Shahmirzadi, Seyyed Arash Azimi, Mahmoud Salmasizadeh, Javad Mohajeri, and Mohammad Reza Aref. Impossible differential cryptanalysis of reduced-round Midori64 block cipher. In *14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology, ISCISC 2017, Shiraz, Iran, September 6-7, 2017*, pages 99–104, 2017.

[SGL+17] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of AES, SKINNY, and others with constraint programming. *IACR Trans. Symmetric Cryptol.*, 2017(1):281–306, 2017.

[SHW+14a] Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Report 2014/747, 2014. https://eprint.iacr.org/2014/747.

[SHW+14b] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, pages 158–178, 2014.

[SLG+16] Bing Sun, Meicheng Liu, Jian Guo, Vincent Rijmen, and Ruilin Li. Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis. In *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*, pages 196–213, 2016.

[SMB18] Sadegh Sadeghi, Tahereh Mohammadi, and Nasour Bagheri. Cryptanalysis of reduced round SKINNY block cipher. *IACR Transactions on Symmetric Cryptology*, 2018(3):124–162, Sep. 2018.

[ST17] Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In *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*, pages 185–215, 2017.

[STA+14] Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, and Shoichi Hirose. Minalpher v1. *CAESAR Round*, 1, 2014.

[SWW18] Ling Sun, Wei Wang, and Meiqin Wang. More accurate differential properties of LED64 and Midori64. *IACR Trans. Symmetric Cryptol.*, 2018(3):93–123, 2018.

[Wag99] David A. Wagner. The boomerang attack. In *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, pages 156–170, 1999.

[WW12] Shengbao Wu and Mingsheng Wang. Automatic search of truncated impossible differentials for word-oriented block ciphers. In *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, pages 283–302, 2012.

[ZZWF07] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-key differential-linear attacks on reduced AES-192. In *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, pages 73–85, 2007.