A Deep Study of The Impossible Boomerang Distinguishers: New Construction Theory and Automatic Search Methods

Xichao Hu[1], Lin Jiao[1], Dengguo Feng[1], Yonglin Hao[1], Xinxin Gong[1], Yongqiang Li[2,3]

[1] State Key Laboratory of Cryptology, Beijing, China xchao_h@163.com
[2] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. The Impossible Boomerang Attack (IBA) has demonstrated remarkable power in the security evaluation of AES and other block ciphers. However, this method has not received sufficient attention in the field of symmetric cipher analysis. The existing search methods, namely $\mathcal{UB}$-method, ZWT-method, and BCL-method for Impossible Boomerang Distinguishers (IBDs)-the core of IBAs, exhibit limitations in terms of efficiency and applicability, very likely leading to the omission of critical attacks. Therefore, this paper delves into a comprehensive and systematic study on the construction theory and automatic search method of IBDs. Theoretically, we establish a new framework for constructing a series of IBDs from the aspects of differential propagation, state propagation, and generalized BCTs in the single-key setting. Furthermore, we rigorously prove the inclusion relations among these newly-defined IBDs, and result in critical conclusions indicating a type of tightest IBDs as well as several types of efficient IBDs as supplements. We extend the theory into related-key setting (RK-IBD) including two scenario: one involving two related-keys under arbitrary key schedules and another involving four related-keys under linear key schedules. Technically, we develop a general SAT-based automatic tool that enables us to search for IBDs for block ciphers including SPN, Feistel-network and ARX designs with arbitrary components under single/related-key settings. Additionally, we propose several effective strategies to improve the search process. As applications, we search for the (RK-)IBDs of 8 block ciphers and get their IBDs or RK-IBDs for the first time. In further comparison with impossible differentials (IDs), all IBDs are no worse than IDs. Moreover, we get 1 round more IBD on PRINTcipher48 than IDs in the single-key setting; 2-round more RK-IBDs on AES-128, 1-round more RK-IBDs on SPECK-32/64 (and SPECK-48/72, SPECK-48/96), and 2-round more RK-IBDs on SPECK-64/96 (and SPECK-64/128, SPECK-96/144, SPECK-128/192, SPECK-128/256) than RK-IDs in the two related keys setting respectively; 1, 1, 4, 2-round more RK-IBDs on DES, GIFT-64, CHAM-64/128 and CHAM-128/256 than RK-IDs respectively, as well as the full-round RK-IBDs on GOST in the four related keys setting.

Keywords: Impossible Boomerang Distinguishers · Single-Keys · Two Related-keys · Four Related-keys · SPN · Feistel · ARX.

1 Introduction

The differential attack, proposed by Biham and Shamir [1], is widely recognized as one of the most crucial methods for analyzing the security of block ciphers. Numerous cryptanalytic techniques have been developed based on it, including two well-known approaches: impossible differential attacks proposed by Biham et al. and Knudsen [2,3], and boomerang attacks simultaneously proposed by Wagner [4]. Undoubtedly, these two attacks have played a pivotal role in the security analysis of block ciphers [5,6,7,8,9]. The combination of these two attacks results in an impossible boomerang attack (IBA) proposed by J. Lu [10], which fundamentally relies on an impossible boomerang distinguisher (IBD) treating a block cipher E as two sub-ciphers $E_0 \circ E_1$ and employing two (or more) differentials for E_0 and E_1 each with a probability of 1 imposed on the non-zero XOR of the intermediate differences of these four differentials. In [10,11], the impossible boomerang attack was utilized to successfully break 6-round AES-128, 7-round AES-192 and 7-round AES-256 in a single key attack scenario, as well as 8-round AES-192 and 9-round AES-256 in a related-key attack scenario involving two keys, based on a 4-round IBD.

Automatic search methods based on certain mathematical problems, such as Boolean Satisfiability Problem (SAT)/Satisfiability Modulo Theories (SMT) problem [12,13,14], Mixed Integer Linear Programming (MILP) problem [15,16], and Constraint Programming (CP) problem [17,18], facilitates effective, thoughtful, and precise search for distinguishers. On searching for the distinguisher of impossible differentials (IDs), Cui et al. [19] proposed a MILP-based tool adapt to lightweight block ciphers considering all the propagation details; Sasaki and Todo [20] presented a further MILP-based tool adapt to SPN block ciphers by introducing arbitrary S-box (AS) mode, which treat large S-boxes as only permutations; Hu et al. [21] simultaneously presented a SAT/SMT-based tool by introducing the state propagation, allowing for consideration of the specific key schedule in the single-key scenario.

On constructing boomerang distinguishers (BDs), significant advancements have been made in recent years. The original boomerang attack postulated that the two sub-ciphers E_0 and E_1 were independent of each other, while Murphy [22] highlighted that two independently chosen characteristics might lead to a probability of zero for a right quartet of plaintext-ciphertext pairs. Furthermore, numerous improvements considering the dependence have been proposed, such as the middle round S-box trick [23], ladder switch, S-box switch and Feistel switch [7], which can be encapsulated within the framework of the sandwich attack proposed by Dunkelman et al. [8,24]. It divides the block cipher E into three parts $E_1 \circ E_m \circ E_0$, where E_0 and E_1 are covered by ordinary differential distinguishers, while E_m is subject to a small boomerang distinguisher that connects the two parts by specified input difference and output difference considering the

dependency between E_0 and E_1. Recently, new insights on what exactly happens in E_m have been investigated. At Eurocrypt 2018, Cid et al. [25] presented the Boomerang Connectivity Table (BCT), a tool facilitating the straightforward evaluation of BD's probability of E_m in the single-round scenario for SPN network. Subsequently, Wang et al. [26] proposed the Boomerang Difference Table (BDT) and its variant BDT', enabling systematic evaluation of boomerang switching effect in the multiple rounds involved scenario. Furthermore, in [27], Boukerrou et al. generalized the BCT and BDT to feistel network and proposed the concept of FBCT. Subsequently, Delaune et al. [28] proposed a CP-based method to search for BDs, and renamed BDT and BDT' as UBCT and LBCT for upper BCT and lower BCT, respectively. Additionally, they defined the EBCT for SPN network based on the definition of FBCT for Feistel-network . In [29], a SAT-based tool was presented for discovering BDs in ARX ciphers.

On searching for IBDs, a direct reflection of the security level of block ciphers, there are three existing methods currently.

- **$\mathcal{UB}$-method [30].** This method uses a miss-in-the-middle approach to construct IBDs. The core idea is to transform differential propagation into the manipulation of a matrix and seek contradictions by defining certain criteria. However, this method is unable to take into account the details of the S-box and linear layer as well as the key schedule.
- **ZWT-method [31].** This method is proposed to search for related-key IBDs (RK-IBDs) for SPN block ciphers. The core idea is to construct IBDs based on DBCT and differential characteristics with a probability of 1. Given that this method depends on DBCT, the linear layer of the block cipher must be byte- or nibble-based and sparse; otherwise, constructing and modeling DBCT becomes challenging. In parallel with our work, they also introduced the concepts of GUBCT, GLBCT, and GEBCT, as well as a method for constructing IBDs utilizing BCT and GBCT.
- **BCL-method [32].** This method is proposed by Bonnetain et al. to search for RK-IBDs for SPN block ciphers and Feistel-network block ciphers with quadratic round functions. The BCL-method shares the same reliance on DBCT as the ZWT-method. Moreover, Bonnetain et al. have also proposed a method for constructing IBDs based on BCT and FBCT. However, such construction necessitates a differential characteristic with a probability of 1.

To sum up, previous methods have the following limitations.

- **In the single-key setting.**
  - **Unable to take into account the details of operations for constructing IBDs with multiple rounds.** The $\mathcal{UB}$-method cannot take into account the details of operations. Moreover, both the ZWT-method and the BCL-method require a differential characteristic with a probability of 1, which is not feasible for multiple rounds.
  - **Unable to take into account the key schedule.** These method constructs the IBDs through the differential propagation only, thereby the impact of the key schedule is unable to be exploited since it is naturally counteracted in the single-key setting.
- **In the related-key setting.**

  - **The applicability is not universal.** The ZWT-method and the BCL-method can only be applied to SPN block ciphers with byte/nibble-based and sparse linear matrices, as well as Feistel-network block ciphers with quadratic round functions. However, these methods are not suitable for SPN block ciphers with MDS matrices (such as AES) and bit permutations (such as PRESENT), or Feistel-network block ciphers with non-quadratic round functions (such as DES).
  - **The related-key scenario is not comprehensively considered.** At present, automated methods usually search for RK-IBDs under four related keys by restricting the two key's differences of upper trail are same and the two key's differences of lower trail are same. Indeed, it is just sufficient to ensure that these four differences satisfy a certain linear relationship, which is a more generalized scenario that should be taken into account. Additionally, there is no discussion on search methods for block ciphers that adopt nonlinear key schedules.
- **In both the single-key and related-key setting.** The existing automatic search methods do not fully capture the essential definition of (RK)-IBDs. That is, there may be better (RK)-IBDs within the same search space.

Our contributions. Motivated by the strong threat posed by IBA method (e.g. powerful attacks on AES), as well as its significant lack of systematic theory and general search models, we initiate a comprehensive research work on its core, constructing IBDs, synchronously related to the theoretical development of boomerang attacks and impossible differential attacks.

Firstly, we establish a new theoretical framework for constructing IBDs. We propose a series of construction methods of (RK)-IBDs from different perspectives. Then, we prove the inclusion relationship among them and result in a completeness conclusion theoretically that is crucial for the subsequent research on (RK)-IBDs. We define two IBDs from the aspects of differential propagation.

T_0**-IBD:** the IBD regarding a bijective S-box as only a permutation.
T_1**-IBD:** the IBD constructed based on differential propagation purely, corresponding to the method proposed in [10].

Furthermore, we define two IBDs from the aspects of state propagation.

T_2**-IBD:** the IBD based on state propagation assuming independence of round keys .
T_3**-IBD:** the IBD based on state propagation considering the key schedule.

We also study the construction method based on the tables defined for constructing BDs, and define a series of IBDs based on the generalized BCTs.

T_P^S**-IBD:** the IBD constructed by a pre-defined propagation rule P based on a mixed use of generalized tables including UDDT, LDDT, GBCT [33], GUBCT, GLBCT, GEBCT for SPN block ciphers.
T_P^F**-IBD:** the IBD constructed by a pre-defined propagation rule P based on a mixed use of generalized tables including UDDT, LDDT, GFBCT, GFUBCT, GFLBCT, GEBCT for Feistel-network block ciphers.
T_C**-IBD:** the IBD constructed based on GEBCT merely for both SPN block ciphers and Feistel-network block ciphers in context with BD.

We further prove the inclusion relations between these newly-defined IBDs. Let S_{T_i} be the set containing all T_i-IBDs and S_{IBD} be the set corresponding to the essential definition of IBDs, and then we derive that

$$S_{T_0} \subseteq S_{T_1} \subseteq S_{T_P^S} \text{ (or } S_{T_P^F}) \subseteq S_{T_C} = S_{T_2} \subseteq S_{T_3} = S_{\text{IBD}}.$$

- For $0 \leq i \leq 3$, an r-round T_i-IBD is always an r-round T_{i+1}-IBD.
- T_C-IBD is equivalent with T_2-IBD for both SPN and Feistel-network block ciphers.
- Any construction method based on even mixed use of generalized DDTs and BCTs cannot be superior to T_C-IBD as well as T_3-IBD.
- The definition of T_3-IBD is equivalent to essential definition of IBD proposed in [10]. That is, the construction of T_3-IBD is the tightest method for constructing IBDs.

Therefore, we result in a theoretical conclusion of paramount importance.

- A rough estimation (lower bound) of the number of rounds of IBDs can be given based on T_0-IBD, and a precise evaluation (upper bound) of the number of rounds of IBDs can be given based on T_3-IBD.
- It is unnecessary to construct IBDs based on BCTs used in BDs.
- If the solving time permits, we should construct and search for the T_3-IBD; when it encounters the efficiency bottleneck of solvers for searching T_3-IBDs, T_0-IBDs, T_1-IBDs and T_2-IBDs can serve as a sufficient supplement and present an effective estimation of the number of rounds of IBDs.

Henceforth, our search models and applications are mainly based on T_0-IBDs, T_1-IBDs, T_2-IBDs and T_3-IBDs.

Secondly, we extend our study to the related-key setting. For encryption $E_{k_i}(x_i)$ with the master key $k_i (0 \leq i \leq 3)$ under the key differences $k_0 \oplus k_1 = \kappa_0, k_2 \oplus k_3 = \kappa_1, k_1 \oplus k_2 = \kappa_2, k_0 \oplus k_3 = \kappa_3$, where $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$, we define RT_j^i-IBDs for T_j-IBDs for $j = 0, 1, 3$ under i related-keys setting for $i = 2, 4$

- Under two related-keys: RK-IBDs under the key difference of $(\kappa, \kappa, 0, 0)$.
- Under four related-keys: RK-IBDs under the key difference of $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ on block ciphers with linear key schedule.

The above related-key setting scenarios allow us to search for RK-IBDs covering more rounds, which benefits either from the offsetting of state input/output differences within the initial/final rounds of RK-IBDs, or from the default/definitive differences of round keys.

Thirdly, we develop our general automatic search models to create efficient tools for searching for (RK-)IBDs based on our newly established theoretical framework. Specifically, we propose a SAT-based automatic method to search for T_j-IBDs for $0 \leq j \leq 3$ and T_j^i-IBDs for $i = 2, 4$ and $j = 0, 1, 3$. Additionally, we propose several search strategies including choosing the search space, speeding up the search based on the related-key differentials with a probability of 1, and verifying the (RK-)IBDs through computer-aid methods. **The method we propose is capable of surpassing all the limitations encountered in previous automatic search methods, as previously discussed.**

Finally, we apply our method to various block ciphers, including SPN, Feistel network and ARX designs. These selections cover the common classifications of

block ciphers and serve to verify the effectiveness of our approach. In the single-key setting, we apply our method on AES [34], a large S-box based block cipher that utilizes an MDS matrix; DES [35], a Feistel-network block cipher with non-bijective S-boxes; PRESENT-80 [36], a lightweight block cipher employing bit permutation; and PRINTcipher48 [37], which employs key-dependent permutation. In the two related-keys setting, we apply our method on AES; and SPECK [38], an ARX-based block cipher. In the four related-keys setting, we apply our method on DES; GIFT [39], a lightweight block cipher utilizing bit permutation; CHAM [40], an ARX-based block cipher; and GOST [41], a Feistel-network block cipher. The results are presented in Table 1. The findings presented here strongly indicate that our method outperforms the current three existing search methods for IBDs. Moreover, specific results suggest that IBDs offer an advantage over IDs. Considering the crucial role ID attacks play as a fundamental analysis technique, it is imperative to acknowledge and employ IBA seriously.

Outline. We introduce the notations and related work in Section 2. We establish our theoretical framework by presenting a series of IBD constructions and investigate their relationship in both the single-key setting (Section 3) and the related-key setting (Section 4). The automatic search method for (RK-)IBDs, along with core strategies, are detailed in Section 5. In Section 6, we applied our method to various block ciphers. We conclude this paper in Section 7.

2 Preliminaries

2.1 Notation

The primary notations used hereafter are detailed as follows.

- Let k and rk_i denote the master key and the i-th round key, respectively. The key schedule is denoted as KS, which generates $rk_i = \mathrm{KS}_i(k)$. Besides the generation of round keys, denote the state updating algorithm of KS be U_{KS}.
- Let $E_k^r(x)$ represent a r-round block cipher, encrypting the input $x \in \mathbb{F}_2^n$ under the master key $k \in \mathbb{F}_2^m$ to produce the output $y = E_k^r(x) \in \mathbb{F}_2^n$.
- Let $E_{i,rk_i}(x_i)$ represent the i-th round of $E_k^r(x)$, encrypting the input $x_i \in \mathbb{F}_2^n$ under the round key $rk_i \in \mathbb{F}_2^{m_i}$ to produce the output $x_{i+1} \in \mathbb{F}_2^n$. That is, $E_k^r(x) = E_{r-1,rk_{r-1}} \circ \cdots \circ E_{0,rk_0}(x)$. In unambiguous cases, $E_k^r(x)$ and $E_{i,rk_i}(x_i)$ are abbreviated as E (or E_k) and E_{i,rk_i}.
- Let S, SL, LL, and $AddKey$ denote an S-box, an S-box layer, a linear layer and a key-xor layer respectively.

2.2 Definitions

We revisit the definitions corresponding to boomerang attacks and impossible boomerang attacks.

Definition 1. *The basic definitions of differential analysis are as follows.*

Table 1: The IBDs on applications.

	Block cipher	Type	Round	Number	Time (hours)	Method	Illustration
Single-key setting	AES	1 ABT T_0-IBDs	4	61440	149.04	our	**First**: large S-boxes, details of linear layers
		1 ABT T_0-IBDs	5	none	203.44	our	
		IBDs	4	less	-	Manual [10]	
		1 ABT IBDs	2	none	1.6	$\mathcal{UB}^a$	
		1 ABT IBDs	2	61440	10.84	$\mathcal{UB}^b$	
		1 ABT IBDs	3	none	24.12	$\mathcal{UB}^b$	
		IDs	4	many	-	[34]	
		IDs	5	none	-	[42]	
	DES	1 Ab T_1-IBDs	7	1904	327.64	our	**First**: Feistel-network with arbitrary round functions
		1 Ab T_1-IBDs	8	none	372.38	our	
		1 Ab IDs	7	394	0.57	ST^a	
		1 Ab IDs	8	none	0.91	ST^a	
	PRESENT-80	1 AN T_2-IBDs	6	58	7.13	our	**First**: bit per- mutation
		1 AN T_3-IBDs	7	none	24.52	our	
		IDs	6	many	-	[43]	
		1 AN IDs	7	none	-	[21]	
	PRINTcipher48	**1 Ab T_3-IBDs**	**5**	**2**	**14.75**	**our**	**First**: key-dependent permutation
		1 Ab T_3-IBDs	6	none	40.07	our	
		1 Ab ID	4	many	-	[21]	**IBD better than ID by 1 more round**
		1 Ab ID	5	none	-	[21]	
Two related-keys setting	AES-128	**1 ABT RT_0^2-IBDs**	**5**	**768**	**14.44**	**our**	**First result**
		1 ABT RT_0^2-IBDs	6	none	18.68	our	for AES-128
		1 ABT RK-IDs	3	64	0.39	ST^a	**IBD better than ID by 2 more round**
		1 ABT RK-IDs	4	none	0.52	ST^a	
	SPECK-32/64	**RT_3^2-IBDs**	**8**	**377**	**0.18**	**our**	
		RT_3^2-IBDs	9	none	0.97	our	
	SPECK-48/72	**RT_3^2-IBDs**	**7**	**6**	**0.06**	**our**	
		RT_3^2-IBDs	8	none	0.26	our	
	SPECK-48/96	**RT_3^2-IBDs**	**8**	**6**	**0.09**	**our**	
		RT_3^2-IBDs	9	none	0.60	our	
	SPECK-64/96	**RT_3^2-IBDs**	**8**	**4**	**0.29**	**our**	**First**: modular additions
		RT_3^2-IBDs	9	none	0.60	our	
	SPECK-64/128	**RT_3^2-IBDs**	**9**	**4**	**0.28**	**our**	
		RT_3^2-IBDs	10	none	0.99	our	
	SPECK-96/144	**RT_3^2-IBDs**	**8**	**4**	**0.22**	**our**	
		RT_3^2-IBDs	9	none	0.65	our	
	SPECK-128/192	**RT_3^2-IBDs**	**8**	**4**	**0.33**	**our**	
		RT_3^2-IBDs	9	none	1.18	our	
	SPECK-128/256	**RT_3^2-IBDs**	**9**	**4**	**0.41**	**our**	
		RT_3^2-IBDs	10	none	1.78	our	
	SPECK-2w/4w ($w = 16, 24, 32, 64$)	RK-IDs	7	many	-	[44]	**IBD better than ID by 1 or 2 more round**
		RK-IDs	8	none	-	[44]	
	SPECK-2w/3w ($w = 24, 32, 48, 64$)	RK-IDs	6	many	-	[44]	
		RK-IDs	7	none	-	[44]	
Four related-keys setting	DES	**1 ABi RT_3^4-IBDs**	**9**	**14**	**137.68**	our	**First result**
		1 ABi RK-IDs	8	74	52.16	ST^a	**IBD better than ID by 1 more round**
		1 ABi RK-IDs	9	none	65.16	ST^a	
	GIFT-64	**1 AN RT_3^4-IBDs**	**13**	**48**	**0.51**	**our**	**First result**
		1 AN RT_3^4-IBDs	14	none	1.91	our	
		1 AN IDs	12	48	-	[44]	**IBD better than ID by 1 more round**
		1 AN IDs	13-16	none	-	[44]	
	GIFT-128	1 AN RT_3^4-IBDs	10	373	3.71	our	**First result**
		1 AN RT_3^4-IBDs	11	none	32.15	our	
		1 AN IDs	10	96	-	[44]	
		1 AN IDs	11-12	none	-	[44]	
	CHAM-64/128	**RT_3^4-IBDs**	**30**	**3**	**0.15**	**our**	**First result**
		RT_3^4-IBDs	31/32	none	0.22	our	
		IDs	26	many	-	[44]	**IBD better than ID by 4 more round**
		IDs	27	none	-	[44]	
	CHAM-64/128	**RT_3^4-IBDs**	**28**	**4**	**0.48**	**our**	**First result**
		RT_3^4-IBDs	29/30	none	0.63	our	
		IDs	26	many	-	[44]	**IBD better than ID by 2 more round**
		IDs	27	none	-	[44]	
	GOST-FB/PS	**RT_3^4-IBDs**	**full-round**	**2**	**0.08**	**our**	**First result: Full round**

$\mathcal{UB}$-method^b represents we add the MDS property to the $\mathcal{UB}$-method. The method with ^{a} means we implemented this method by ourselves. ABT: active byte truncated, Ab: active bit, AN: active nibble.

1. *For a function* $f: \mathbb{F}_2^m \times \mathbb{F}_2^n \to \mathbb{F}_2^n$*, the probability that an input difference* α *propagates to an output difference* β *under the key difference* κ *is given by* $P_{f,\kappa}(\alpha,\beta) = \#\{(k,x) \in \mathbb{F}_2^m \times \mathbb{F}_2^n \mid f(k,x) \oplus f(k \oplus \kappa, x \oplus \alpha) = \beta\}/2^{n+m}$*. If* $P_{f,\kappa}(\alpha,\beta) \neq 0$*, it is denoted as* $\alpha \overset{f,\kappa}{\to} \beta$*. Define* $\mathrm{DP}_{f,\kappa}(\alpha) = \{\beta | \alpha \overset{f,\kappa}{\to} \beta\}$*. Particularly, in the single-key setting,* $P_f(\alpha,\beta) = \#\{x \in \mathbb{F}_2^n \mid f(x) \oplus f(x \oplus \alpha) = \beta\}/2^n$*,* $\mathrm{DP}_f(\alpha) = \{\beta | \alpha \overset{f}{\to} \beta\}$*.*
2. *For a composite function* $f: \mathbb{F}_2^m \times \mathbb{F}_2^n \to \mathbb{F}_2^n$*, where* $f = f_{r-1} \circ \cdots \circ f_1 \circ f_0$*, an* r*-round related-key differential characteristic is defined as a series of differences* $\Omega = (\alpha_0, \ldots, \alpha_r)$ *under the key difference* $\Gamma = (\kappa_0, \ldots, \kappa_r)$*, where* $\alpha_i \overset{f_{i,\kappa_i}}{\to} \alpha_{i+1}$ *for* $0 \leq i \leq r-1$*, and the probability of* Ω *is given by* $P_{f,\Gamma}(\Omega) = \prod_{i=0}^{r-1} P_{f_i,\kappa_i}(\alpha_i, \alpha_{i+1})$*. Moreover, the probability of differential defined by* (α_0, α_r) *is given by* $P_{f,\Gamma}(\alpha_0, \alpha_r) = \sum_{\alpha_1,\ldots,\alpha_{r-1}} P_{f,\Gamma}(\Omega)$*. In the single-key setting,* Γ *is omit .*

Definition 2. *Given two differences* $\gamma \mathbb{F}_2^n, \theta \in \mathbb{F}_2^m$*, the DDT for an* $n \times m$*-bit function is defined as* $\mathrm{DDT}(\gamma,\theta) = \#\{x \in \mathbb{F}_2^n \mid f(x) \oplus f(x \oplus \gamma) = \theta\}$*.*

Definition 3 ([25]). *Given three differences* $\gamma, \theta, \delta \in \mathbb{F}_2^n$*, the BCT for an* n*-bit S-box are defined as* $\mathrm{BCT}(\gamma,\delta) = \#\left\{x \in \mathbb{F}_2^n \mid S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma\right\}$*.*

Definition 4. *Let* $E = E^1 \circ E^m \circ E^0$ *be an* r*-round SPN-network block cipher with* $r = r_0 + r_1 + 1$*, where* E^0*,* E^m *and* E^1 *denote the initial* r_0 *rounds, middle 1 round and final* r_1 *rounds of* E *respectively. Suppose* $\alpha \overset{E^0}{\to} \gamma$ *and* $\delta \overset{E^1}{\to} \beta$*, then the probability*

$$\Pr(E^{-1}(E(x) \oplus \beta) \oplus E^{-1}(E(x \oplus \alpha) \oplus \beta) = \alpha) = (P_{E^0}(\alpha,\gamma))^2 (P_{E^1}(\delta,\beta))^2 P_m,$$

where $P_m = \prod_{i=0}^{t} (\mathrm{BCT}(\gamma_i, \delta_i)/2^n)$*, assuming that there are* t *n-bit S-boxes in* E_m *with the input difference* γ_i *and output difference* δ_i*.*

To apply boomerang switch in multiple rounds, more tables have been proposed.

Definition 5 ([26,28]). *Given four differences* $\gamma, \theta, \lambda, \delta \in \mathbb{F}_2^n$*, the UBCT, LBCT and EBCT for an* n*-bit S-box are defined as*

$$\mathrm{UBCT}(\gamma,\theta,\delta) = \#\left\{ x \in \mathbb{F}_2^n \middle| \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\},$$

$$\mathrm{LBCT}(\gamma,\lambda,\delta) = \#\left\{ x \in \mathbb{F}_2^n \middle| \begin{array}{l} S(x) \oplus S(x \oplus \lambda) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\},$$

$$\mathrm{EBCT}(\gamma,\theta,\lambda,\delta) = \#\left\{ x \in \mathbb{F}_2^n \middle| \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S(x) \oplus S(x \oplus \lambda) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\}.$$

In addition to BCTs for SPN block ciphers, new tables are also defined for Feistel-network block ciphers.

Definition 6 ([27]). *Given four differences* $\gamma, \theta, \lambda, \delta \in \mathbb{F}_2^n$*, the FBCT, BDT and F-BET for an* n*-bit S-box are defined as*

$$\mathrm{FBCT}(\gamma,\delta) = \#\left\{ x \in \mathbb{F}_2^n \middle| S(x) \oplus S(x \oplus \gamma) \oplus S(x \oplus \delta) \oplus S(x \oplus \gamma \oplus \delta) = 0 \right\},$$

$$\mathrm{BDT}(\gamma,\theta,\delta) = \#\left\{ x \in \mathbb{F}_2^n \middle| \begin{array}{l} S(x) \oplus S(x \oplus \gamma) \oplus S(x \oplus \delta) \oplus S(x \oplus \gamma \oplus \delta) = 0 \\ S(x) \oplus S(x \oplus \gamma) = \theta \end{array} \right\},$$

$$\mathrm{FBET}(\gamma,\theta,\delta,\lambda) = \#\left\{ x \in \mathbb{F}_2^n \middle| \begin{array}{l} S(x) \oplus S(x \oplus \gamma) \oplus S(x \oplus \delta) \oplus S(x \oplus \gamma \oplus \delta) = 0 \\ S(x) \oplus S(x \oplus \gamma) = \theta \\ S(x \oplus \gamma) \oplus S(x \oplus \gamma \oplus \delta) = \lambda \end{array} \right\}.$$

The properties present in one table have corresponding counterparts in the other tables. In [28], Delaune et al. proposed a method for establishing a BD with optimal probability using mixed tables.

The essential definition of IBD is defined as follows.

Definition 7 ([10]). *Given a block cipher $E : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$ under a key $k \in \mathbb{F}_2^m$, if for four differences $\alpha, \alpha', \beta, \beta'$, any pair of plaintexts (x_1, x_2) cannot satisfy*

$$E_k(x_1) \oplus E_k(x_2) = \beta,\ E_k(x_1 \oplus \alpha) \oplus E_k(x_2 \oplus \alpha') = \beta'$$

simultaneously, then $(\alpha, \alpha', \beta, \beta')$ is called an IBD for E_k, denoted by $(\alpha, \alpha') \nrightarrow (\beta, \beta')$.

An IBD constructed in the related-key setting is called an RK-IBD. Given an IBD or RK-IBD, an attacker can extend the number of rounds before and after the distinguisher to launch a key recovery attack, known as IBA. This attack poses new threats to block ciphers, particularly in certain scenarios where its impact surpasses that of IDs [11,31,32].

3 New Theory for Constructing IBDs in the Single-Key Setting

In this section, we establish a new theoretical framework for constructing IBDs from the aspects of both differential propagation and state propagation, as well as BCTs both for SPN and Feistel-network block ciphers, while proving the interrelationships among all construction methods. The proofs of theorems in this section are given in Appendix B.

3.1 Constructing IBDs from the aspect of differential propagation

We propose two IBD-definitions based on differential propagation. Firstly, we present two boomerang trails based on $\mathrm{DP}_f(\alpha)$ and its relaxing variant.

Definition 8. *Given an r-round block cipher $E = E^1 \circ E^0$, for two input differences α, α' and two output differences β, β',*

- *if there exist $\gamma \in \overline{\mathrm{DP}}_{E^0}(\alpha)$, $\gamma' \in \overline{\mathrm{DP}}_{E^0}(\alpha')$, $\delta \in \overline{\mathrm{DP}}_{(E^1)^{-1}}(\beta)$, and $\delta' \in \overline{\mathrm{DP}}_{(E^1)^{-1}}(\beta')$, such that $\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0$, then*

$$(\alpha, \alpha') \to \cdots \to \underbrace{(\gamma, \gamma')(\delta, \delta')}_{\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0} \to \cdots \to (\beta, \beta')$$

  is called an r-round T_0 boomerang trail, where $\overline{\mathrm{DP}}_f(\alpha)$ is a relaxing variant of $\mathrm{DP}_f(\alpha)$ by considering all the details of operations of f expect S-boxes.
- *if there exist $\gamma \in \mathrm{DP}_{E^0}(\alpha)$, $\gamma' \in \mathrm{DP}_{E^0}(\alpha')$, $\delta \in \mathrm{DP}_{(E^1)^{-1}}(\beta)$, and $\delta' \in \mathrm{DP}_{(E^1)^{-1}}(\beta')$, such that $\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0$, then*

$$(\alpha, \alpha') \to \cdots \to \underbrace{(\gamma, \gamma')(\delta, \delta')}_{\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0} \to \cdots \to (\beta, \beta')$$

  is called an r-round T_1 boomerang trail.

Accordingly, we present the following two IBD-construction methods: T_0-IBD and T_1-IBD. T_0-IBD is a new method, while T_1-IBD is a generation of the existing one proposed in [10]. Furthermore, we prove their inclusion relationship.

Construction 1 (T_0-IBD). *Given an r-round block cipher E, for two input differences α, α' and two output differences β, β', if there is no r-round T_0 boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_0-IBD.*

Construction 2 (T_1-IBD). *Given an r-round block cipher E, for two input differences α, α' and two output differences β, β', if there is no r-round T_1 boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_1-IBD.*

Theorem 1. *An r-round T_0-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_1-IBD.*

T_0-IBD is suitable for block ciphers with bijective S-boxes, as it treats the S-box only as a permutation. Consequently, T_0-IBD takes the advantage of efficiently searching and assessing a lower bound on the number of IBDs' rounds. T_1-IBD offers extensive applicability, within which the differential propagation rule of each component can be characterized. The examples of T_0-IBD and T_1-IBD are presented in Section 6.1, e.g. the applications to AES and DES respectively.

3.2 Constructing IBDs from the aspect of state propagation

Inspired by the concept proposed in [21] that utilizes the propagation of two states to construct IDs, we extend it to construct IBDs using the propagation of four states, which adapts to any block ciphers. Specifically, our method is able to take into account both scenarios of independent keys and key relations in the single-key setting.

Definition 9. *Let $E = E_{r-1,rk_{r-1}} \circ \cdots \circ E_{0,rk_0}(x)$ be an r-round block cipher. Given four differences $\alpha, \alpha', \beta, \beta'$, let $I = \{(x_0, x_1, x_2, x_3) \mid x_0 \oplus x_1 = \alpha, x_2 \oplus x_3 = \alpha'\}$ and $O = \{(y_0, y_1, y_2, y_3) \mid y_1 \oplus y_2 = \beta, y_0 \oplus y_3 = \beta'\}$. If there exist $(x_0^0, x_1^0, x_2^0, x_3^0) \in I$, $(x_0^r, x_1^r, x_2^r, x_3^r) \in O$ and independent round keys $(rk_0, \cdots, rk_{r-1})$, such that*

$$x_j^{i+1} = E_{i,rk_i}(x_j^i) \text{ for } 0 \le i \le r-1, 0 \le j \le 4,$$

then $(x_0^0, x_1^0, x_2^0, x_3^0) \to \cdots \to (x_0^r, x_1^r, x_2^r, x_3^r)$ is called an r-round T_2 boomerang trail.

This definition enables us to construct IBD in another way.

Construction 3 (T_2-IBD). *Given an r-round block cipher E, for two input differences α, α' and two output differences β, β', if there is no r-round T_2 boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_2-IBD.*

To consider the relationship of round keys in the single-key setting according to the key schedule, we further present the following definition.

Definition 10. *Let $E = E_{r-1,\mathrm{KS}_{r-1}(k)} \circ \cdots \circ E_{0,\mathrm{KS}_0(k)}(x)$ be an r-round block cipher with the key schedule* KS. *Given four differences $\alpha, \alpha', \beta, \beta'$, let $I = \{(x_0, x_1, x_2, x_3) \mid x_0 \oplus x_1 = \alpha, x_2 \oplus x_3 = \alpha'\}$ and $O = \{(y_0, y_1, y_2, y_3) \mid y_1 \oplus y_2 = \beta, y_0 \oplus y_3 = \beta'\}$. If there exist $(x_0^0, x_1^0, x_2^0, x_3^0) \in I$, $(x_0^r, x_1^r, x_2^r, x_3^r) \in O$ and an master key k such that*

$$x_j^{i+1} = E_{i,\mathrm{KS}_i(k)}(x_j^i) \text{ for } 0 \le i \le r-1, 0 \le j \le 4,$$

then $(x_0^0, x_1^0, x_2^0, x_3^0) \to \cdots \to (x_0^r, x_1^r, x_2^r, x_3^r)$ is called an r-round T_3 boomerang trail.

This definition allows us to consider the validity of round keys' compactness when constructing IBDs for the first time.

Construction 4 (T_3-IBD). *Given an r-round block cipher E, for two input differences α, α' and two output differences β, β', if there is no r-round T_3 boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_3-IBD.*

The inclusion relationship between T_2-IBD and T_3-IBD is direct according to their definitions.

Theorem 2. *An r-round T_2-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_3-IBD.*

The examples of T_2-IBD and T_3-IBD are presented in Section 6.1, e.g. the applications to PRESENT and PRINTcipher respectively.

Furthermore, based on the definitions of T_1-IBD and T_2-IBD, an r-round T_1-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is also an r-round T_2-IBD. However, it should be noted that these two definitions are not equivalent: e.g. the T_2-IBD in the application to PRESENT given in Section 6.1 is not an T_1-IBD.

We further prove that Construction 4 is the tightest method for constructing IBDs.

Theorem 3. *T_3-IBD is equivalent to the essential definition of IBDs given in Definition 7.*

At this point, we have the following inclusion relationships[4], which apply to any block cipher including SPN, Feistel-network and ARX designs.

Summary 1 *Let S_{T_i} denotes the set containing all T_i-IBDs, then we have*

$$(S_{T_0} \subseteq) S_{T_1} \subseteq S_{T_2} \subseteq S_{T_3}.$$

3.3 Constructing IBDs based on generalized BCTs

BCTs have been widely proposed in the BD construction of SPN and Feistel-network block ciphers. Theoretically, it is natural to extend BD to IBD and construct IBDs based on BCTs. Therefore, in this section, we discuss constructing IBDs based on generalized BCTs for SPN block ciphers and Feistel-network block ciphers.

For SPN block ciphers, the original boomerang attack assumes that two subciphers E^0 and E^1 are independent. However, this assumption may not hold true for two selected differential characteristics, as demonstrated in [22]. In other words, it is possible that an r-round T_1 boomerang trail does not exist at all. As a result, this method may overlook certain IBDs. This issue can be addressed by the concept of GBCT [33]. Here, for the purposes of this paper, the following definition is slightly different in expression from the original one.

Definition 11. *Given four differences $\mu, \mu', \varphi, \varphi' \in \mathbb{F}_2^n$, the GBCT for an n-bit S-box is defined as*

$$\mathrm{GBCT}(\mu, \mu', \varphi, \varphi') = \#\left\{ (x_0, x_1, x_2, x_3) \in \{0,1\}^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ S(x_1) \oplus S(x_2) = \varphi \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\}.$$

[4] The inclusion relationship for T_0-IBD is valid for block ciphers with bijective S-boxes.

Furthermore, as highlighted by Song et al. [45], it has been observed that the dependence can exert a significantly greater influence over multiple rounds, e.g. up to 6 rounds for SKINNY. While BCT cannot eliminate the incompatibility in multiple rounds, a serious of tables such as UBCT, LBCT, and EBCT are defined. We further introduce and generalize the concepts of these BCTs to IBDs.

Definition 12. *Given eight differences $\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi' \in \mathbb{F}_2^n$ where $\rho' = \mu \oplus \mu' \oplus \rho$, $\varphi' = \theta \oplus \theta' \oplus \varphi$, the GUBCT, GLBCT and GEBCT [5] for an n-bit S-box are defined as*

$$\mathrm{GUBCT}(\mu,\mu',\theta,\theta',\varphi,\varphi') = \#\left\{ (x_0,x_1,x_2,x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ S(x_0) \oplus S(x_1) = \theta \\ S(x_2) \oplus S(x_3) = \theta' \\ S(x_1) \oplus S(x_2) = \varphi \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\},$$

$$\mathrm{GLBCT}(\mu,\mu',\rho,\rho',\varphi,\varphi') = \#\left\{ (x_0,x_1,x_2,x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ x_1 \oplus x_2 = \rho \\ x_0 \oplus x_3 = \rho' \\ S(x_1) \oplus S(x_2) = \varphi \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\},$$

$$\mathrm{GEBCT}(\mu,\mu',\rho,\rho',\theta,\theta',\varphi,\varphi') = \#\left\{ (x_0,x_1,x_2,x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ x_1 \oplus x_2 = \rho \\ x_0 \oplus x_3 = \rho' \\ S(x_0) \oplus S(x_1) = \theta \\ S(x_2) \oplus S(x_3) = \theta' \\ S(x_1) \oplus S(x_2) = \varphi \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\}.$$

The above tables are supplemented with two additional notations for the sake of clarity:

$$\mathrm{UDDT}(\mu,\mu',\theta,\theta') = \#\left\{ (x_0,x_1,x_2,x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ S(x_0) \oplus S(x_1) = \theta \\ S(x_2) \oplus S(x_3) = \theta' \end{array} \right\},$$

$$\mathrm{LDDT}(\rho,\rho',\varphi,\varphi') = \#\left\{ (x_0,x_1,x_2,x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_1 \oplus x_2 = \rho \\ x_0 \oplus x_3 = \rho' \\ S(x_1) \oplus S(x_2) = \varphi \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\}.$$

A schematic diagram for these generalized BCTs is shown in Figure 1.

Theorem 4. UDDT, LDDT, GBCT, GUBCT, GLBCT, GEBCT *have the following relations:*

[5] In [31], the authors proposed the similar definitions and they did not use those tables to search for IBDs. Here, we emphasize that we have independent definitions.

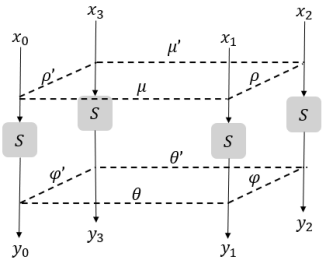

Fig. 1: The Generalized BCTs

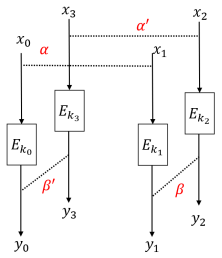

Fig. 2: The illustration of RK-IBD

1. $\exists \eta, \eta'$, *s.t.* $\mathrm{GBCT}(\mu, \mu', \varphi, \varphi') \subseteq \mathrm{UDDT}(\mu, \mu', \eta, \eta')$,
2. $\exists \omega, \omega'$, *s.t.* $\mathrm{GBCT}(\mu, \mu', \varphi, \varphi') \subseteq \mathrm{LDDT}(\omega, \omega', \varphi, \varphi')$,
3. $\mathrm{GUBCT}(\mu, \mu', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{UDDT}(\mu, \mu', \theta, \theta')$,
4. $\exists \omega, \omega'$, $\mathrm{GUBCT}(\mu, \mu', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{LDDT}(\omega, \omega', \varphi, \varphi')$,
5. $\exists \eta, \eta'$, *s.t.* $\mathrm{GLBCT}(\mu, \mu', \rho, \rho', \varphi, \varphi') \subseteq \mathrm{UDDT}(\mu, \mu', \eta, \eta')$,
6. $\mathrm{GLBCT}(\mu, \mu', \rho, \rho', \varphi, \varphi') \subseteq \mathrm{LDDT}(\rho, \rho', \varphi, \varphi')$,
7. $\mathrm{GEBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{UDDT}(\mu, \mu', \theta, \theta')$,
8. $\mathrm{GEBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{LDDT}(\rho, \rho', \varphi, \varphi')$,
9. $\mathrm{GEBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{GBCT}(\mu, \mu', \varphi, \varphi')$,
10. $\mathrm{GEBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{GUBCT}(\mu, \mu', \theta, \theta', \varphi, \varphi')$,
11. $\mathrm{GEBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') \subseteq \mathrm{GLBCT}(\mu, \mu', \rho, \rho', \varphi, \varphi')$.

We consider a hybrid use of DDT, GBCT, GUBCT, GLBCT, and GEBCT to construct IBDs, in a similar manner for achieving optimized BDs.

Definition 13. *Let E be a block cipher with t S-boxes $(S_0, \ldots, S_{t-1})$ in total. Define $\mathcal{AP}_E^S = \{(p_0, \ldots, p_{t-1}) | p_i \in \{\mathrm{UDDT}, \mathrm{LDDT}, \mathrm{GBCT}, \mathrm{GUBCT}, \mathrm{GLBCT}, \mathrm{GEBCT}\}\}$ as a set of propagation rules. Then $P = (p_0, \ldots, p_{t-1}) \in \mathcal{AP}_E^S$ denotes that the propagation rule through the i-th S-box follows p_i.*

Definition 14. *Let $E = E_{r-1,rk_{r-1}} \circ \cdots \circ E_{0,rk_0}(x)$ be an r-round block cipher. Let $P = (P_0, \ldots, P_{r-1})$ be a predefined propagation rule of E, where $P_i \in \mathcal{AP}_{E_{i,rk_i}}^S$ denotes a propagation rule of E_{i,rk_i} for $i \in \{0, \ldots, r-1\}$. Let $\epsilon_0^i, \epsilon_1^i, \epsilon_2^i, \epsilon_3^i$ be the four input differences and $\epsilon_0^{i+1}, \epsilon_1^{i+1}, \epsilon_2^{i+1}, \epsilon_3^{i+1}$ be the four output differences of E_{i,rk_i} for $i \in \{0, \ldots, r-1\}$. For two input differences α, α' and two output differences β, β' of the block cipher E, if there exists a trail*

$$(\epsilon_0^0 = \alpha, \epsilon_1^0, \epsilon_2^0 = \alpha', \epsilon_3^0) \xrightarrow{P_0} \cdots \xrightarrow{P_{r-1}} (\epsilon_0^r, \epsilon_1^r = \beta, \epsilon_2^r, \epsilon_3^r = \beta'),$$

then it is called an r-round T_P^S boomerang trail. Here, $\xrightarrow{P_i}$ represents that the propagation rule through S-boxes in E_{i,rk_i} follows P_i.

Accordingly, we have the following construction.

Construction 5 (T_P^S-IBD). *Given an r-round block cipher E and a predefined rule $P \in \mathcal{AP}_E^S$, for two input differences α, α' and two output differences β, β', if there is no r-round T_P^S boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_P^S-IBD.*

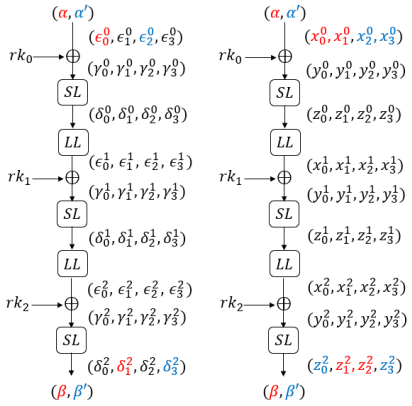

Fig. 3: The equivalence between T_C-IBD and T_2-IBD in SPN

T_P^S-IBD corresponds to a serious of IBDs for different predefined propagation rules P. T_1-IBD is a special example of T_P^S-IBD.

Theorem 5. *For any predefined rule $P \in \mathcal{AP}_E^S$, an r-round T_1-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_P^S-IBD.*

Theorem 4 demonstrates that GEBCT is a table belonging to all other tables. Thus we construct an IBD using only GEBCT.

Definition 15. *Let $E = E_{r-1,rk_{r-1}} \circ \cdots \circ E_{0,rk_0}(x)$ be an r-round block cipher. Let $\epsilon_0^i, \epsilon_1^i, \epsilon_2^i, \epsilon_3^i$ be the four input differences and $\epsilon_0^{i+1}, \epsilon_1^{i+1}, \epsilon_2^{i+1}, \epsilon_3^{i+1}$ be the four output differences of E_{i,rk_i} for $i \in \{0, \ldots, r-1\}$. For two input differences α, α' and two output differences β, β' of the block cipher E, if there exists a trail*

$$(\epsilon_0^0 = \alpha, \epsilon_1^0, \epsilon_2^0 = \alpha', \epsilon_3^0) \xrightarrow{GEBCT} \cdots \xrightarrow{GEBCT} (\epsilon_0^r, \epsilon_1^r = \beta, \epsilon_2^r, \epsilon_3^r = \beta'),$$

then it is called an r-round T_C boomerang trail.

Construction 6 (T_C-IBD). *Given an r-round block cipher E, for two input differences α, α' and two output differences β, β', if there is no r-round T_C boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_C-IBD.*

From the definitions, T_C-IBD is also a special case of T_P^S-IBD.

Theorem 6. *For any predefined rule $P \in \mathcal{AP}_E^S$, an r-round T_P^S-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_C-IBD.*

In addition to the above relations, we can prove the equivalency in Theorem 7 based on the schematic diagram shown in Figure 3.

Theorem 7. *Given an SPN block cipher, $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_C-IBD if and only if it is an r-round T_2-IBD.*

Similarly, for Feistel-network block ciphers, we explore the construction method of IBDs based on generalized BCTs in Appendix A.1. We also firstly generalize the definition of FBCT and FBDT, and then define GFBCT and GFUBCT/GFLBCT accordingly. Subsequently, we define the T_P^F-IBD and study the relations between T_P^F-IBD and T_i-IBD ($0 \leq i \leq 3$) as well as T_C-IBD. All in all, we get the following relations.

Summary 2 *Let S_{T_i} denotes the set containing all T_i-IBDs, then we have*

$$(S_{T_0} \subseteq) S_{T_1} \subseteq S_{T_P^S} \subseteq S_{T_C} = S_{T_2} \subseteq S_{T_3},\ (S_{T_0} \subseteq) S_{T_1} \subseteq S_{T_P^F} \subseteq S_{T_C} = S_{T_2} \subseteq S_{T_3}.$$

The inclusion relationship in Summary 2 and the equivalence in Theorem 3 indicate that we can provide a rough estimation (lower bound) of the number of rounds of IBDs based on T_0-IBDs, and a precise evaluation (upper bound) of the number of rounds of IBDs based on T_3-IBD. Furthermore, although constructing IBDs based on BCTs used in BDs seems like a reasonable technical approach, however, on the one hand, Summary 2 indicates that $S_{T_P^S}$ (resp. $S_{T_P^F}$) and S_{T_C} are not the tightest; on the other hand, the modeling method of searching IBDs based on BCTs is much more complex compared with $T_{0/1/2/3}$-IBDs. Therefore, through our theoretical study above, we have reached an important conclusion: it is unnecessary to construct IBDs based on BCTs used in BDs. This further paves the way for subsequent research on IBD constructions. Henceforth, our subsequent search models and applications are based on $T_{0/1/2/3}$-IBDs without using IBDs based on BCTs. As no approach for constructing IBDs surpasses Construction 4, searching for T_3-IBDs becomes imperative. Additionally, when it encounters the efficiency bottleneck of solvers for searching T_3-IBDs, $T_{0/1/2}$-IBDs can serve as a sufficient supplement and present an effective estimation of the number of rounds of IBDs.

4 New Theory for Constructing IBDs in the Related-Key Setting

Let k_0, k_1, k_2, k_3 be four master keys of an block cipher E. Then an r-round IBD $((\alpha, \alpha'), (\beta, \beta'))$ of E is equivalent to that there exists no solution of the system

$$x_1 = x_0 \oplus \alpha,\ x_3 = x_2 \oplus \alpha',\ E_{k_1}^r(x_1) \oplus E_{k_2}^r(x_2) = \beta,\ E_{k_0}^r(x_0) \oplus E_{k_3}^r(x_3) = \beta',$$

when $k_0 = k_1 = k_2 = k_3$ in the single-key setting as stated in Section 3. Furthermore, in the related-key setting, more relations between these keys can be controlled by attackers, which may lead greater power to IBAs. Specifically, we construct the RK-IBDs in the two related-keys setting and four related-keys setting subsequently. Additionally, the definition of $T_{0/1/3}$ boomerang trail given in Section 3 can naturally extend to the related-key setting.

Definition 16. *Given an r-round block cipher $E = E^1 \circ E^0$, for two input differences α, α', two output differences β, β', and four key differences κ_0, κ_1, κ_2 and $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$,*

- *if there exist* $\gamma \in \overline{\mathrm{DP}}_{E^0,\kappa_0}(\alpha)$, $\gamma' \in \overline{\mathrm{DP}}_{E^0,\kappa_1}(\alpha')$, $\delta \in \overline{\mathrm{DP}}_{(E^1)^{-1},\kappa_2}(\beta)$, *and* $\delta' \in \overline{\mathrm{DP}}_{(E^1)^{-1},\kappa_3}(\beta')$, *such that* $\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0$, *then*

$$(\alpha, \alpha') \to \cdots \to \underbrace{(\gamma, \gamma')(\delta, \delta')}_{\gamma\oplus\gamma'\oplus\delta\oplus\delta'=0} \to \cdots \to (\beta, \beta')$$

  is called an r*-round* T_0 *related-key boomerang trail, where* $\overline{\mathrm{DP}}_{f,\kappa}(\alpha)$ *is a relaxing variant of* $\mathrm{DP}_{f,\kappa}(\alpha)$ *by considering all the details of operations of* f *expect S-boxes.*
- *if there exist* $\gamma \in \mathrm{DP}_{E^0,\kappa_0}(\alpha)$, $\gamma' \in \mathrm{DP}_{E^0,\kappa_1}(\alpha')$, $\delta \in \mathrm{DP}_{(E^1)^{-1},\kappa_2}(\beta)$, *and* $\delta' \in \mathrm{DP}_{(E^1)^{-1},\kappa_3}(\beta')$, *such that* $\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0$, *then*

$$(\alpha, \alpha') \to \cdots \to \underbrace{(\gamma, \gamma')(\delta, \delta')}_{\gamma\oplus\gamma'\oplus\delta\oplus\delta'=0} \to \cdots \to (\beta, \beta')$$

  is called an r*-round* T_1 *related-key boomerang trail.*

Definition 17. *Let* $E = E_{r-1,\mathrm{KS}_{r-1}(k)} \circ \cdots \circ E_{0,\mathrm{KS}_0(k)}(x)$ *be an* r*-round block cipher with the key schedule* KS. *Given two input differences* α, α', *two output differences* β, β', *and four key differences* κ_0, κ_1, κ_2 *and* $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$, *let* $I = \{(x_0, x_1, x_2, x_3) \mid x_0 \oplus x_1 = \alpha, x_2 \oplus x_3 = \alpha'\}$ *and* $O = \{(y_0, y_1, y_2, y_3) \mid y_1 \oplus y_2 = \beta, y_0 \oplus y_3 = \beta'\}$. *If there exist* $(x_0^0, x_1^0, x_2^0, x_3^0) \in I$, $(x_0^r, x_1^r, x_2^r, x_3^r) \in O$ *and master keys* k_j *such that*

$$x_j^{i+1} = E_{i,\mathrm{KS}_i(k_j)}(x_j^i) \text{ for } 0 \le i \le r-1, 0 \le j \le 3,$$

under the key differences $k_0 \oplus k_1 = \kappa_0, k_2 \oplus k_3 = \kappa_1, k_1 \oplus k_2 = \kappa_2, k_0 \oplus k_3 = \kappa_3$, *then* $(x_0^0, x_1^0, x_2^0, x_3^0) \to \cdots \to (x_0^r, x_1^r, x_2^r, x_3^r)$ *is called an* r*-round* T_3 *related-key boomerang trail.*

The RK-IBDs in the two related-keys setting was first proposed by J. Lu [11], which only focused on constructing RK-IBDs based on differential propagation rather than state propagation. Thus we provide the following new definitions.

Construction 7 (RT_i^2-IBD). *Given an* r*-round block cipher* E, *for two input differences* α, α' *and two output differences* β, β', *if there is no* r*-round* T_i *related-key boomerang trail under the key differences* $(\kappa, \kappa, 0, 0)$ *where* $k_0 \oplus k_1 = \kappa, k_2 \oplus k_3 = \kappa, k_1 \oplus k_2 = 0, k_0 \oplus k_3 = 0$, *then* $((\alpha, \alpha'), (\beta, \beta'))$ *is a two related-keys' RK-IBD, called an* r*-round* RT_i^2*-IBD, for* $i = 0, 1, 3$.

Accordingly, the RK-IBDs in the four related-keys setting are defined as follows.

Construction 8 (RT_i^4-IBD). *Given an* r*-round block cipher* E, *for two input differences* α, α' *and two output differences* β, β', *if there is no* r*-round* T_i *related-key boomerang trail under the key differences* $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ *where* $k_0 \oplus k_1 = \kappa_0$, $k_2 \oplus k_3 = \kappa_1$, $k_1 \oplus k_2 = \kappa_2$, $k_0 \oplus k_3 = \kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$, *then* $((\alpha, \alpha'), (\beta, \beta'))$ *is a four related-keys' RK-IBD, called an* r*-round* RT_i^4*-IBD, for* $i = 0, 1, 3$.

It is worth noting that when the key schedule is linear, the key differences $k_0 \oplus k_1 = \kappa_0$, $k_2 \oplus k_3 = \kappa_1$, $k_1 \oplus k_2 = \kappa_2$, implies

$$\begin{cases} \mathrm{KS}_i(k_0) \oplus \mathrm{KS}_i(k_1) = \mathrm{KS}_i(\kappa_0), \mathrm{KS}_i(k_2) \oplus \mathrm{KS}_i(k_3) = \mathrm{KS}_i(\kappa_1), \\ \mathrm{KS}_j(k_1) \oplus \mathrm{KS}_j(k_2) = \mathrm{KS}_j(\kappa_2), \mathrm{KS}_j(k_0) \oplus \mathrm{KS}_j(k_3) = \mathrm{KS}_j(\kappa_0 \oplus \kappa_1 \oplus \kappa_2), \end{cases}$$

for any round index i, j. That is, the differences of the round keys are all determined.

Table 2: Modeling the differential propagation through `Xor`, `Copy`, `KeyAdd`.

Operation	Input Diff	Output Diff	Modeling Method
`Copy`	$\alpha \in \mathbb{F}_2$	$\beta_0, \beta_1 \in \mathbb{F}_2$	$\beta_0 = \alpha, \beta_1 = \alpha$
`Xor`	$\alpha_0, \alpha_1 \in \mathbb{F}_2$	$\beta \in \mathbb{F}_2$	$\beta = \alpha_0 \oplus \alpha_1$
`KeyAdd`	$\alpha \in \mathbb{F}_2$	$\beta \in \mathbb{F}_2$	$\beta = \alpha$

Discussions on RK-IBDs based on Generalized BCTs: As in the single-key setting, we can also construct the RK-IBDs based on the G(U/L/E)BCT for SPN block ciphers and GF(U/L/E)BCT for Feistel-network block ciphers. With the same technique as in the single-key setting, in the same related-key setting, these RK-IBDs are superior to RT_1^i-IBD but inferior to RT_3^i-IBD, for $i = 2, 4$ corresponding to the two related-keys setting and four related-keys setting respectively. For the same reason as stated Section 3, we mainly focus on the RT_j^i-IBD with $i = 2, 4$ and $j = 0, 1, 3$.

5 New Automatic Search Methods for (RK-)IBDs

In this section, we present our search method for (RK-)IBDs from the aspect of the differential propagation and state propagation respectively, and further propose key search strategies.

5.1 Searching for (RK-)IBDs from the aspect of differential propagation

To search for $(R)T_0$-IBDs and $(R)T_1$-IBDs from the aspect of differential propagation, firstly we propose a SAT-based method[6] to model the differential propagation through common operations such as `Xor`, `Copy`, `KeyAdd`, `MatrixMultiply` and `S-box`, as well as the arbitrary S-box (`AS`) proposed in [20] by treating the S-box as only a permutation.

Model 1. *The method for modeling the differential propagation through the operations* `Xor`*,* `Copy`*,* `KeyAdd` *is presented in Table 2.*

Model 2. *For the operation* `MatrixMultiply` $M = (m_{i,j})_{u \times v}$*, let* α_i *for* $0 \leq i \leq v-1$ *and* β_i *for* $0 \leq i \leq u-1$ *be the input and output differences of* M*, it holds* $\beta_i = \oplus_{j=0}^{v-1} m_{i,j}\alpha_j$*. Thus the differential propagation through* `MatrixMultiply` *can be expressed according to that of* `Xor`*.*

Model 3. *For the operation* `S-box` S*, let* α_i *for* $0 \leq i \leq v-1$ *and* β_i *for* $0 \leq i \leq u-1$ *be the input and output differences of* S*. As the possible values of* α_i *for* $0 \leq i \leq v-1$ *and* β_i *for* $0 \leq i \leq u-1$ *are restricted by the* DDT *of* S*, the differential propagation*

[6] The SAT problem [46] involves determining the satisfiability of a given Boolean function. A typical framework of SAT-based automatic search methods is to convert the search for a distinguisher into a SAT problem, , subsequently solving this problem by leveraging existing solvers. In this paper, we employ STP (https://stp.github.io) and CryptoMiniSat (https://github.com/msoos/cryptominisat) as backends.

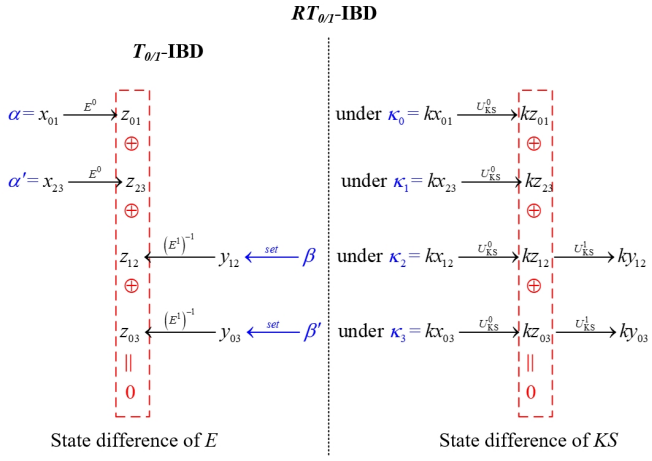

Fig. 4: The search model for $(R)T_0$-IBD or $(R)T_1$-IBD
†Here $x_{ij} = x_i \oplus x_j$, denoting the difference variable of state variables x_i and x_j.

through S can be expressed as a set of logic expressions with the help of some third party tool, such as Logic Friday[7].

Model 4. *For the operation* `AS` S*, let α_i and β_i for $0 \le i \le v-1$ be the input and output differences of S, then the following transitions are impossible:* $(0,1), \cdots, (0, 2^v - 1)$, $(1,0), \cdots, (2^v - 1, 0)$, *which can be removed by the boolean expressions:*

$$\alpha_{v-1}||\cdots||\alpha_0||\neg\beta_i = 1,\ \ \neg\alpha_i||\beta_{v-1}||\cdots||\beta_0 = 1 \quad \textit{for } 0 \le i \le v-1.$$

Furthermore, based on above modeling method, given an r-round block cipher $E = E^1 \circ E^0$, and its KS state updating algorithm $U_{\mathrm{KS}} = U^1_{\mathrm{KS}} \circ U^0_{\mathrm{KS}}$, we can construct a model to determine whether a given $\mathcal{D} = ((\alpha, \alpha'), (\beta, \beta'))$ qualifies as an r-round $(R)T_0$-IBD or $(R)T_1$-IBD under key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ by describing the constraints of the differential propagation shown in Figure 4; if the SAT-solver returns "no solution", then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD. Particularly in the single-key setting, all four key differences are set to zero. The corresponding algorithm is given in Algorithm 1 in Appendix C.

5.2 Searching for (RK-)IBDs from the aspect of state propagation

To search for T_2-IBDs and $(R)T_3$-IBDs from the aspect of state propagation, firstly we revisit the method for modeling the state propagation via each operation proposed in [21].

[7] Logic Friday (`http://sontrak.com/`) can be used to derive the minimum (or as small as possible in a reasonable time) product-of-sum representation of a given Boolean function from its truth table. This representation can be converted to a set of logic expressions equivalently. For detailed usage instructions, please refer to [47].

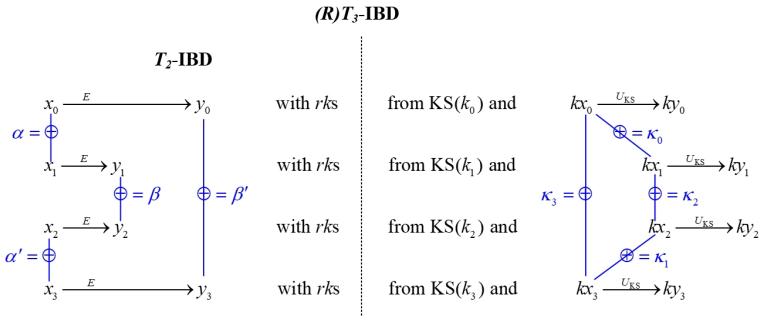

Fig. 5: The search model for T_2-IBD or $(R)T_3$-IBD

Model 5. *The method for modeling state propagation through the operations* `Xor`, `Copy`, `MatrixMultiply` *is identical to that of Model 1 and Model 2. The method for modeling state propagation through operation* `KeyAdd` *is the same as that of* `Xor`.

Model 6. *For the operation* `S-box` S, *let* α_i *for* $0 \le i \le v-1$ *and* β_i *for* $0 \le i \le u-1$ *be the input and output states of* S. *As the possible values of* α_i *for* $0 \le i \le v-1$ *and* β_i *for* $0 \le i \le u-1$ *is restricted by the truth table of* S, *the state propagation through* S *can be expressed as a set of logic expressions with the help of the third party tool Logic Friday.*

Furthermore, based on above modeling method, given an r-round block cipher E, and its KS state updating algorithm U_{KS}, we can construct a model to determine whether a given $\mathcal{D} = ((\alpha, \alpha'), (\beta, \beta'))$ qualifies as an r-round T_2-IBD or $(R)T_3$-IBD under key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ by describing the constraints of the differential propagation shown in Figure 5; if the SAT-solver returns "no solution", then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD. Particularly in the single-key setting, all four key differences are set to zero. The corresponding algorithm is given in Algorithm 2 in Appendix C.

5.3 The search strategies for (RK-)IBDs

Based on Algorithm 1 and Algorithm 2, we can efficiently search for (RK-)IBDs within a given search space. Let n be the block size, s be the S-box size, then there are $t = n/s$ S-boxes in SL. For $x = (x_{n-1}, \ldots, x_0) \in \mathbb{F}_2^n$, $wt(x) = \oplus_{i=0}^{m} x_i$, $\overline{x} = (x_{n-1}|\cdots|x_{m\times(t-1)}, \cdots, x_{m-1}|\cdots|x_0)$, where '|' denotes bitwise-or. Similar to ID, we mainly focus on searching for the three following types of (RK-)IBDs.

Type-1. l_i input active bits and l_o output active bits IBDs: an (RK-)IBD $\mathcal{D} = ((\alpha, \alpha'), (\beta, \beta'))$ with $wt_2(\alpha) = wt_2(\alpha') = l_i$ and $wt_2(\beta) = wt_2(\beta') = l_o$. Particularly, when $l_i = l_o$, it is called an l_i active bits (RK-)IBD.

Type-2. l_i input active nibbles (resp. bytes) and l_o output active nibbles (resp. bytes) IBDs: an (RK-)IBD $\mathcal{D} = ((\alpha, \alpha'), (\beta, \beta'))$ with $wt(\overline{\alpha}) = wt(\overline{\alpha'}) = l_i$ and $wt(\overline{\beta}) = wt(\overline{\beta'}) = l_o$. Particularly, when $l_i = l_o$, it is called an l_i active nibbles (resp. bytes) (RK-)IBD.

Type-3. l_i input active nibbles (resp. bytes) and l_o output active nibbles (resp. bytes) truncated IBDs: Given $a, a', b, b' \in \mathbb{F}_2^t$ with $wt(a) = wt(a') = l_i$ and $wt(b) = wt(b') = l_o$, if for $\forall \mathcal{D} \in \{((\alpha, \alpha'), (\beta, \beta')) \mid \overline{\alpha} = a, \overline{\alpha'} = a', \overline{\beta} = b, \overline{\beta'} = b'\}$, $\mathcal{D}$ is an (RK-)IBD, then (a, a', b, b') is called an l_i input active nibbles (resp. bytes) and l_o output active nibbles (resp. bytes) truncated IBD. Particularly, when $l_i = l_o$, it is called an l_i active nibbles/bytes truncated (RK-)IBD. To search for Type-3 (RK-)IBD, we can simply modify Algorithm 1 and Algorithm 2 by adding the relations between the bitwise input differences and output differences with (a, a') and (b, b').

For RK-IBDs, the selection of key differentials involves greater technicality. We discuss the selection strategy comprehensively considering the input-output differences and the key differences in two cases.

In the two related-keys setting. We set the two input differences with $\alpha = \alpha'$. A direct strategy is to set the key differences κ that can eliminate the input difference α, thereby allowing for several initial rounds of E without any differences. Subsequently, for each (α, κ) pair, we search for the output differences (β, β') such that $((\alpha, \alpha), (\beta, \beta'))$ is a r-round RT_i^2-IBD for $i = 0, 1, 3$ under key differences $(\kappa, \kappa, 0, 0)$. The search space of $((\alpha, \alpha), (\beta, \beta'))$ is discussed as above. This allows the difference to propagate through the non-linear operations in a few initial rounds with a probability of 1, thus thereby achieving the objective of obtaining an RK-IBD covering more rounds.

An advanced strategy involves utilizing related-key differentials with a probability of 1. That is, given a block cipher $E = E^1 \circ E^0$, we control the differences of round keys to eliminate the state differences before entering nonlinear operations in E_0. In details, we firstly search for an r_0-round related-key differential (α, γ, κ) with a probability of 1, i.e., $P_{E^0,\kappa}(\alpha, \gamma) = 1$. When searching, it is crucial to ensure that the key difference does not undergo the nonlinear operations of KS. Subsequently, we search for (β, β') such that $((\gamma, \gamma), (\beta, \beta'))$ is an r_1-round RT_i^2-IBD of E^1 for $i = 0, 1, 3$ under the key differences $(\kappa, \kappa, 0, 0)$. The search space of $((\gamma, \gamma), (\beta, \beta'))$ is also discussed as above. Consequently, $((\alpha, \alpha), (\beta, \beta'))$ is an $(r_0 + r_1)$-round RK-IBD under the key differences $(\kappa, \kappa, 0, 0)$.

In the four related-keys setting. Given a block cipher $E = E^2 \circ E^1 \circ E^0$, the search strategy for RT_i^4-IBD for $i = 0, 1, 3$ under the key differences κ_0, κ_1, κ_2 and $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$ are as follows. Firstly, we search for 2 r_0-round related-key differentials $(\alpha, \gamma, \kappa_0)$ of E^0 and $(\alpha', \gamma', \kappa_1)$ of E^2, and 2 r_2-round related-key differentials $(\beta, \delta, \kappa_2)$ and $(\beta', \delta', \kappa_3)$ with a probability of 1, i.e.,

$$P_{E^0,\kappa_0}(\alpha, \gamma) = 1, P_{E^0,\kappa_1}(\alpha', \gamma') = 1, P_{(E^2)^{-1},\kappa_2}(\beta, \delta) = 1, P_{(E^2)^{-1},\kappa_3}(\beta', \delta') = 1. \tag{1}$$

However,it proves to be challenging in practice when directly searching for Equation (1) while satisfying relation $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$. To address this issue, the linear key schedule offers definitive differences of round keys as $\mathrm{KS}_i(\kappa_j)$ for $i = 0, \ldots, r-1$ and $j = 0, 1, 2, 3$, which subsequently aids in managing the elimination of state differences in E^0. Therefore, in the four related-keys setting, we focus exclusively on the block ciphers with linear key schedule.

Subsequently, we verify whether $((\gamma, \gamma'), (\delta, \delta'))$ is an r_1-round RT_i^4-IBD for $i = 0, 1, 3$ under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ with $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$; if so, then $((\alpha, \alpha'), (\beta, \beta'))$ is an $(r_0 + r_1 + r_2)$-round RT_i^4-IBD for $i = 0, 1, 3$ under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ with $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$.

Here, we also present some general techniques.

Searching strategy: When the search for (RK-)IBDs relying solely on state propagation is extremely time-consuming, we can employ differential propagation (with no longer a probability of 1) for a certain number of rounds in the beginning or end of the distinguishers, that is, employing a mixture of state propagation and differential propagation. An example is given in Section 6.1 for searching for RK-IBDs on DES.

Verifying strategy: To compensate for the limitations of manual verification, we employ some techniques of computer-aided verification of (RK-)IBDs given in Appendix F, where the computer can play the roles of detecting the contradiction location and traversing all possible trails for disproving the (RK-)IBDs.

Comparison of RK-IBDs and RK-IDs. A related-key ID (RK-ID) is a pair of difference (α, β) such that the input difference α cannot propagate to the output difference β under the key difference κ. In comparison to RK-IDs, RK-IBDs present advantages in two folds. Firstly, regarding the search for the distinguisher itself and considering the two related-key setting illustrated in Figure 6, in the latter part of the distinguisher i.e. after the contradiction, the key difference for RK-IBD is 0. In contrast, the key difference for RK-ID has undergone extensive diffusion through the KS algorithm (particularly with respect to nonlinear KS algorithms), making it challenging to control or eliminate. Consequently, RK-IBDs often accommodates more rounds than RK-IDs. Secondly, regarding the extension of the distinguisher, the search strategies mentioned above for RK-IBDs can also be applied to RK-IDs. However, for block ciphers with linear key schedules, it is necessary to extend both E^0 and E^2 under the same key difference within RK-IDs, whereas greater flexibility exists in terms of key differences within RK-IBDs as show in Figure 7. In other words, equality is no longer required; only the relation $\kappa_3 = \kappa_0 \oplus \kappa_1 \oplus \kappa_2$ needs to be satisfied. Consequently, it becomes possible to extend more rounds within RK-IBDs than within RK-IDs.

6 Applications of (RK-)IBDs

In this section, we apply our method to search for (RK-)IBDs on 8 block ciphers involving 3 main structures as follows:

SPN: AES (large S-box), PRESENT-80 (bit-permutation), PRINTcipher (Keyed permutation), GIFT;
Feistel: DES (non-injective S-box), GOST;
ARX: SPECK, CHAM.

Specifically, we search for IBDs on AES, PRESENT-80, PRINTcipher and DES, as well as search for RK-IBDs on AES, SPECK with nonlinear KS, and DES,

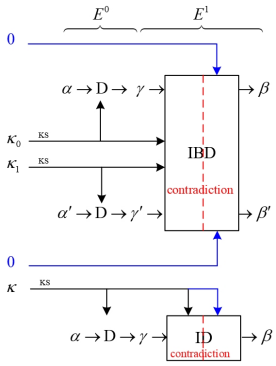

Fig. 6: A schematic in the two related-keys setting

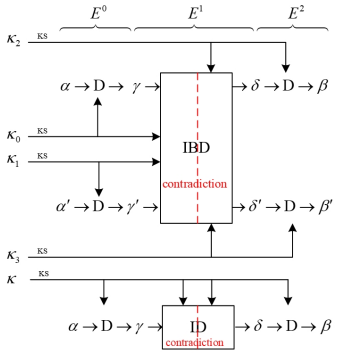

Fig. 7: A schematic in the four related-keys setting

†Here D denotes a related-key differential with a probability of 1.

GOST, GIFT, CHAM with linear KS. ***This is the first time that RK-IBDs has been provided for other block ciphers except AES.*** We present brief descriptions of these block ciphers in Appendix D, examples of searched IBDs in Appendix E, and their verifications in Appendix F.

The experiments in this paper were conducted on the AMD(R) @2.60GHz platform with 80.00G RAM, running a 64-bit Ubuntu18.04 system. All timing is attributed to the spent on searching for a distinguisher using a single core.

6.1 Applications of IBDs in the single-key setting

-AES [34] (Description: Appendix D.1; Distinguisher Example 1)

Configurations. As AES is built with 8-bit S-boxes with excellent cryptographic properties, making the search for T_i-IBDs ($i = 1, 2, 3$) extremely time-consuming. Therefore, our focus shifts to searching for T_0-IBDs that treat the S-boxes as a permutation by Algorithm 1. This allows us to evaluate truncated IBDs only (Type-3). Consequently, we search for 1 active byte truncated T_0-IBDs on AES, in a search space size of $16^4 = 65536$.

Results. We establish the absence of 5-round 1 active byte T_0-IBDs, requiring about 203.44 hours. Thus, we turn to search for such 4-round 1 active byte T_0-IBDs, and result in all 61440 such IBDs in about 149.04 hours.

Comparison. (1) Currently, the only result for IBDs on AES is some 4-round IBDs derived by manual derivation [10]. In contrast, our method enables the search for a large number of 4-round IBDs automatically. (2) Among existing automatic search methods, only $\mathcal{UB}$-method [30] can be used to search for IBDs on AES. However, this original $\mathcal{UB}$-method treats the Mixcolumn operation merely as a permutation. We implement this method and find that it returns truncated IBDs within above search space no more than 2 rounds. Subsequently, we extend this method by incorporating the constraints on the branch number of the MDS matrix. As a result, it returns 61440 3-round 1 active byte T_0-IBDs but fails to find any such IBDs beyond 3 rounds. The primary reason

these tools do not achieve the same level of performance as we do is that ***our approach is the first to enable the search for IBDs of block ciphers with large S-boxes considering the details of linear layers***. (3) On AES, the maximum number of rounds of IDs is 4 [34], and no 5-round ID exists as proven in [42]. Thus, the number of rounds of IBDs we found is the same as the maximum number of rounds of IDs.

-DES [35] (Description: Appendix D.2; Distinguisher Example 2)

Configurations. Our experimental results indicate that searching for T_2-IBDs or T_3-IBDs for more than 7 rounds is excessively time-consuming. Therefore, our focus lies in searching for 1 active bit T_1-IBDs (Type-1) by Algorithm 1. As DES employs the Feistel network, we restrict two input differences to be activated only in the left branch and two output differences to be activated only in the right branch. This choice enables us to propagate the difference forward and backward through one round with a probability of 1. Additionally, either two input differences or two output differences are required to be identical. The size of the search space is $2 \times 32^3 = 2^{16}$.

Results. We establish the absence of 8-round 1 active bit T_1-IBDs within above search space, requiring about 372.38 hours. Thus, we turn to search for the 7-round 1 active bit T_1-IBDs, and result in 1904 such IBDs in around 327.64 hours.

Comparison. (1) Currently, BCL-method [32] is the only approach on IBDs for Feistel-network block ciphers. However, this approach is limited to quadratic round functions and is not applicable to DES. Consequently, ***our approach is the first to enable the search for IBDs of block ciphers that employ the Feistel network with arbitrary round functions***. (2) To compare with IDs, we employ ST-method [20] to search IDs by constraining input difference and output difference to 1 active bit difference. As a result, we find 7-round IDs and ascertain the absence of 8-round IDs within such search spaces. Consequently, the number of rounds of IBDs obtained aligns with the maximum known number of rounds for IDs.

-PRESENT-80 [36] (Description: Appendix D.3; Distinguisher Example 3)

Configurations. We search for 1 active nibble T_3-IBDs (Type-2) directly by Algorithm 2, with the restriction that only the first S-box of the input two and output two differences are active. The size of the search space is $15^4 = 50625$.

Results. We establish the absence of 7-round T_3-IBDs in above search space, requiring about 24.52 hours. Thus, we turn to search for 6-round 1 active nibble T_3-IBDs, and result in 58 such IBDs in about 7.13 hours. To investigate the inclusion relationships given in Summary 1 academically, we test whether these T_3-IBDs are T_i-IBDs ($0 \leq i \leq 2$), and result in that all of these T_3-IBDs are T_2-IBDs, but not T_1-IBDs.

Comparison. (1) Since PRESENT utilizes bit permutation, the modeling of its D-BCT, which includes four 4-bit S-boxes, is extremely challenging. Consequently, all existing methods are not applicable for searching for IBDs on PRESENT. ***Our approach is the first to enable the search for IBDs of block ciphers that employ bit permutation.*** (2) To our knowledge, the maximum number of rounds of IDs is 6 [43], and there are no 7-round 1 bit active nibble IDs [21]. Therefore, the number of rounds of IBDs obtained is identical to the maximum number of rounds of IDs.

-PRINTcipher48 [37] (Description: Appendix D.4; Distinguisher Example 4)

Configurations. Since PRINTcipher48 employs a key-dependent permutation, we search for 1 active bit T_3-IBDs (Type-1) directly by Algorithm 2. In particular, either two input differences or two output differences are restricted to be the same. The size of the search space is $2 \times 48^3 = 221184$.
Results. We establish the absence of 6-round T_3-IBDs in above search space, requiring about 40.07 hours. Thus, we turn to search for 5-round 1 active bit T_3-IBDs, and result in 2 such IBDs in about 14.75 hours. To investigate the inclusion relationships given in Summary 1 academically, we remove the relationship between round keys to test whether those T_3-IBDs are T_2-IBDs, and the result is that they are not.
Comparison. (1) Since PRINTcipher48 employs the key-dependent permutation, all existing methods are not applicable to searching for its IBDs. ***Our approach is the first to enable the search for IBDs of block ciphers that employ key-dependent permutation.*** (2) Hu et al. [21] proposed the 4-round IDs and demonstrated that there are no 5-round IDs even considering the details of the key schedule. Consequently, ***the number of rounds of IBDs is one round more than that of IDs for PRINTcipher48.***

6.2 Applications of RK-IBDs in the related-key setting

-AES-128 [34] (Description: Appendix D.1; Distinguisher Example 5)

Configurations. Since AES employs a non-linear KS, we search for RT_0^2-IBDs by Algorithm 1. Similar to the single-key setting, we focus on 1 active byte RT_0^2-IBDs (Type-3) with input-output differences $((\alpha, \alpha), (\beta, \beta'))$ where α, β, β' are 1 active byte truncated differences under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3) = (\alpha, \alpha, 0, 0)$. The size of the search space is $16^3 = 2^{12}$.
Results. First, we establish the absence of 6-round RT_0^2-IBDs in above search space, requiring about 18.68 hours. Thus, we turn to search for the 5-round 1 active byte RT_0^2-IBDs, and result in 768 such IBDs in about 14.44 hours.
Comparison. (1) In [10], J. Lu manually derived some 6-round RK-IBDs for AES-192 and AES-256. However, there have been no results of RK-IBDs for AES-128 until now. ***Our method presents the first result of 5-round RK-IBDs for AES-128.*** (2) As discussed in the single-key setting, none of the existing methods can match the effectiveness of our approach to search for RK-IBDs on AES, since they cannot consider the details of linear layers . (3) To compared with the RK-IDs, we use the AS mode in the ST-method [20] to search for RK-IDs. Specifically, we search for r-round RK-ID with the input-output difference (α, β) where α, β are 1 active byte truncated differences under the key differences α. As a result, we find 3-round RK-IDs ascertain the absence of such 4-round RK-IDs. Consequently, ***the number of rounds of RK-IBDs is 2 rounds more than that of RK-IDs for AES-128***.

-SPECK [38] (Description: Appendix D.5; Distinguisher Example 6)

Configurations. Since SPECK employs a non-linear KS, we search for the RT_3^2-IBDs by adopting the advanced strategy which utilizes related-key differentials with a probability of 1, as described in Section 5.3. Specifically, we first search for the r_0

related-key differential with a probability of 1, i.e. (α, γ, κ). Then we search for the β and β' such that $((\gamma, \gamma), (\beta, \beta))$ is an r_1-round RT_3^2-IBDs under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3) = (\kappa, \kappa, 0, 0)$, where $\beta, \beta' \in \Lambda = \{(\mu, \mu), (0, \mu \ggg_b), (\mu, \mu \oplus (\mu \ggg_b))\}$. In this context, $\ggg_b$ represents circular shift right operation by b bits, where $b = 2$ for SPECK-32 and $b = 3$ for other versions; μ is an $n/2$-bit value for SPECK-n, with its most significant bit being 1 and all other bits being 0. This choice allows us to go through the modulo addition operation in the last round with a probability of 1. Then we obtain $r = r_0 + r_1$-round such RT_3^2-IBDs. The size of the search space is 9.

Results. First, we find SPECK-$2w/4w(w = 16, 24, 32, 64)$ has 1 4-round related-key differential characteristic with a probability of 1 and SPECK-$2w/3w(w = 24, 32, 48, 64)$ has 1 3-round related-key differential characteristic with a probability of 1, within a very short time. Then, we run Algorithm 2 to search the RT_3^2-IBDs. The results are presented in Table 3.

Comparison. (1) Since SPECK employs the operation modular addition, all existing methods are not applicable to searching for IBDs of it. ***our approach is the first to enable the search for IBDs of block ciphers that employ modular additions.*** (2) To our knowledge, the maximum number of rounds of RK-IDs on SPECK-$2w/4w(w = 16, 24, 32, 64)$ is 7, as well as that on SPECK-$2w/3w(w = 24, 32, 48, 64)$ is 6 [44]. Thus, ***the number of rounds of RK-IBDs is one or two rounds more than that of RK-IDs for each version of SPECK.***

Table 3: The RK-IBDs of SPECK in the two related-keys setting.

Block cipher	Round (r)	Number	Time (hours)	Compared with IDs
SPECK-32/64	8	6	0.18	1 round more
	9	none	0.97	-
SPECK-48/72	7	6	0.06	1 round more
	8	none	0.26	-
SPECK-48/96	8	6	0.09	1 round more
	9	none	0.60	-
SPECK-64/96	8	4	0.29	2 rounds more
	9	none	0.60	-
SPECK-64/128	9	4	0.28	2 rounds more
	10	none	0.99	-
SPECK-96/144	8	4	0.22	2 rounds more
	9	none	0.65	-
SPECK-128/192	8	4	0.33	2 rounds more
	9	none	1.18	-
SPECK-128/256	9	4	0.41	2 rounds more
	10	none	1.78	-

-DES [35] (Description: Appendix D.2; Distinguisher Example 7)

Configurations. Since DES employs a linear KS, we search for r-round RT_3^4-IBDs by adopting the strategy that utilizes related-key differentials with a probability of 1. Specifically, let F denoted the expand function of DES, we search for the input differences (α_L, α_R) and (α'_L, α'_R) and output differences (β_L, β_R) and (β'_L, β'_R) such that they are r-round RK-IBDs under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$, where $\alpha_L = \alpha'_L = \beta_R = \beta'_R = 0$, α_R, α'_R and β_L are 1 bit active differences, and $\beta'_L = \alpha_R \oplus \alpha'_R \oplus \beta_L$, $KS_{rb}(\kappa_0) \oplus F(\alpha_R) = 0$, $KS_{rb}(\kappa_1) \oplus F(\alpha'_R) = 0$, $KS_{rb+r}(\kappa_2) \oplus F(\beta_L) = 0$, and $\kappa_3 = \kappa_0 \oplus$

$\kappa_1 \oplus \kappa_2$, where rb denotes the beginning round of the distinguisher and is set as $rb = 0$ without loss of generality. Then $KS_{rb+r}(\kappa_3) \oplus F(\beta'_L) = 0$ for the linear key schedule. This choice allows the input differences propagate 2 rounds in the forward direction and the output differences propagate 2 rounds in the backward direction with a probability of 1. Thus, we only need to verify whether $(((\alpha_L, \alpha_R), (\alpha'_L, \alpha'_R)), ((\beta_L, \beta_R), (\beta'_L, \beta'_R)))$ is $(r-4)$-round RK-IBD under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ by Algorithm 2. However, direct search is still extremely time-consuming when $r \geq 9$. To address this issue, we propagate the differences (α_L, α_R) and (α'_L, α'_R) one round at round $r_b + 3$, an convert it to determine the $(r-5)$-round RT_3^4-IBDs.

Results. We get 14 9-round RK-IBDs in around 137.68 hours.

Comparison. (1) As discussed in the single-key setting, the effectiveness of our approach is unparalleled as it enables the search for IBDs of block ciphers employing Feistel structure with arbitrary round functions, a capability not found in any existing method. (2) In comparison with RK-IDs, we also make use of the ST-method [20], which takes into account the details of the propagation of differences to search for RK-IDs. Specifically, we search for the RK-ID $((\theta_L, \theta_R), (\eta_L, \eta_R))$ under the key difference κ', where θ and η are 1 active bit differences and $KS_{rb}(\kappa') \oplus F(\theta_R) = 0$ accordingly. Specifically, we just obtain 8-round RK-IDs and no 9-round RK-IDs exists in above search space. Thus, ***the number of rounds of RK-IBDs is 1 round more than that of RK-IDs on DES***.

-GIFT [39] (Description: Appendix D.6; Distinguisher Example 8)

Configurations. Since GIFT employs a linear KS, we search for r-round RK_3^4-IBDs by adopting the strategy that utilizes related-key differentials with a probability of 1 by Algorithm 2. Specifically, for GIFT-64, we search for RK_3^4-IBDs under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3) = (\lambda, \lambda, \eta, \eta)$ by setting the input differences $(\alpha, \alpha') = (KS_{rb}(\lambda), KS_{rb}(\lambda))$ and output differences $(\beta, \beta') = (KS_{rb+r}(\eta), KS_{rb+r}(\eta))$, where rb denotes the beginning round of the distinguisher and is set as $rb = 0$ without loss of generality. This choice allows the input differences propagate 4 rounds in the forward direction and the output differences propagate 4 rounds in the backward direction with a probability of 1. Thus, we only need to verify whether $((0,0),(0,0))$ is $(r-8)$-round RK-IBD under the key differences $(\lambda, \lambda, \eta, \eta)$. Moreover, we set the value of λ such that only one S-box is activated at round $(rb+5)$, and the value of η such that only one S-box is activated at round $(rb+r-5)$, which facilitates the search for distinguishers with a large number of rounds. The size of the search space is $(16 \times 3)^2 = 2034$. We apply the same method to GIFT-128 and the size of the search space is $(32 \times 3)^2 = 9216$.

Results. The results obtained are presented in Table 4.

Comparison. (1) Similar to PRESENT, GIFT utilizes bit permutation, making it challenging to search for RK-IBDs using previous methods. (2)In comparison with IDs, Hu et al. [44] proposed 48 12-round RK-IDs and proved that no more round distinguisher exists for GIFT-64, as well as proposed 96 10-round RK-IDs and proved that no more round distinguisher exists for GIFT-128. To our knowledge, these are currently the best results achieved for RK-IDs on GIFT. Thus, ***the number of rounds of RK-IBDs is 1 round more than that of RK-IDs on GIFT-64.*** And for GIFT-128, the number of rounds of RK-IBDs we obtained is the same as the maximum number of rounds of RK-IDs; however, there are more instances of RK-IBDs compared

Table 4: The RK-IBDs of GIFT in the four related-keys setting.

Block cipher	Round(r)	Number	Time(hours)	Compared with IDs
GIFT-64	13	48	0.51	1 round more
	14	none	1.91	-
GIFT-128	10	373	3.71	same rounds, distinguishers more
	11	none	32.15	-

to those RK-IDs.

-CHAM [40] (Description: Appendix D.7; Distinguisher Example 9)

Configurations. Since CHAM employs a linear KS, we search for the RT_3^4-IBDs by adopting a strategy that utilizes related-key differentials with a probability of 1. Without loss of generality, we assume that the distinguisher starts from round 0. We search for (α_0, α_{r_0}), an r_0-round related-key differential characteristic with a probability of 1 under the key difference μ, and $(\alpha_{r_0+r_1+r_2}, \alpha_{r_0+r_1})$, an r_2-round related-key differential characteristic with a probability of 1 under the key difference ν, where α_i denotes the difference at round i. Subsequently, we verify whether $((\alpha_{r_0}, \alpha_{r_0}), (\alpha_{r_0+r_1}, \alpha_{r_0+r_1}))$ is an r_1-round RT_3^4-IBD under the key differences $\kappa_0 = \kappa_1 = \mu, \kappa_2 = \kappa_3 = \nu$ by Algorithm 2; if so, then $((\alpha_0, \alpha_0), (\alpha_{r_0+r_1+r_2}, \alpha_{r_0+r_1+r_2}))$ is an $r = r_0 + r_1 + r_2$-round RK-IBD under the above key difference.
Results. The results obtained are presented in Table 5.
Comparison. (1) To date, no existing methods have been able to find the RK-IBDs on CHAM due to its adoption of modular additions. (2) In comparison with RK-IDs, Hu et al. [44] proposed 48 12-round RK-IDs and proved that no more-round distinguisher exists for GIFT-64, as well as proposed 96 10-round RK-IDs and proved that no more-round distinguisher exists for GIFT-128. To our knowledge, the maximum number of rounds of RK-IDs of CHAM-64/128 and CHAM-128/256 are all 26 [44]. Thus, ***the number of rounds of RK-IBDs is 4 rounds and 2 rounds more than that of RK-IDs of CHAM-64/128 and CHAM-128/256 respectively.***

Table 5: The RK-IBDs of CHAM in the four related-keys setting.

Block cipher	Round(r)	Number	Time(hours)	Compared with IDs
CHAM-64/128	30	3	0.15	4 round more
	31/32	none	0.22	-
CHAM-128/256	28	4	0.48	2 round more
	29/30	none	0.63	-

-GOST [41] (Description: Appendix D.8; Distinguisher Example 10)

Configurations. Since GOST employs a linear KS, we search for RT_3^4-IBDs by adopting the strategy that utilizes related-key differentials with a probability of 1 by Algorithm 2. Specifically, according the KS, the key difference κ_i can be write as $\kappa_i = \kappa_{i,7}||\ldots||\kappa_{i,0}$, where $\kappa_{i,j}(0 \leq j \leq 7)$ is a 32-bit value. As shown in Figure 28, Figure 29, Figure 30 and Figure 31, GOST has 24-round related-key differential characteristics

and 7-round related-key differential characteristics with a probability of 1. To make good use of this property, we position the distinguisher in rounds from 23 to 25 and search for the value of $\kappa_{2,7}$ with $\kappa_{0,7} = 0x800000000, \kappa_{1,7} = 0x000000000, \kappa_{3,7} = \kappa_{0,7} \oplus \kappa_{1,7} \oplus \kappa_{2,7}$, such that $(((0x00000000, 0x80000000), (0x80000000, 0x00000000)), ((0x800 00000, 0x00000000), (0x00000000, 0x80000000)))$ is an 2-round RT_3^4-IBDs. This enables us to extend the distinguisher to the full rounds. Specifically, we impose a restriction that only 1 bit is active in $\kappa_{2,7}$. The size of the search space is 32.

Results. We found two 2-round RT_3^4-IBDs for both GOST-FB and GOST-PS within 5 minutes. Specifically, it requires $\kappa_{2,7} = 0x40000000$ or $\kappa_{2,7} = 0x20000000$. These two RT_3^4-IBDs can be extended to full-round distinguishers for both GOST-FB and GOST-PS.

Comparison. (1) To date, no existing methods have been able to find the RK-IBDs on GOST due to its adoption of modular additions. To our knowledge, ***this is the first full-round RK-IBDs on GOST***. (2) Since our distinguisher is full-round, we do not compare it with other distinguishers anymore.

7 Conclusion and Future Work

In this paper, we explore the construction theory of IBDs in both single-key and related-key settings. Additionally, we develop a SAT-based tool with novel strategies to automatically search for IBDs on various block ciphers, including SPN, Feistel-network, and ARX designs. The results obtained for the first time demonstrate that our approach overcomes all limitations of current search methods for IBDs and further reveals that the number of rounds of an IBD is more than that of an ID in many block ciphers. Consequently, resistance against IBA becomes a crucial consideration in block cipher design.

It should be noted that our work only focuses on the search method for basic IBDs with two input differences and two output differences. A more generalized boomerang distinguisher can involve multiple input differences and output differences; however, this aspect remains to be explored in future research.

References

1. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.
2. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999.
3. Lars R. Knudsen. Deal - a 128-bit block cipher. *Complexity*, 1998.
4. David A. Wagner. The boomerang attack. In Lars R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.

5. Patrick Derbez and Pierre-Alain Fouque. Automatic search of meet-in-the-middle and impossible differential attacks. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 157–184. Springer, 2016.
6. Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder. Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 128–157. Springer, 2023.
7. Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.
8. Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2010.
9. Jongsung Kim, Seokhie Hong, Bart Preneel, Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key boomerang and rectangle attacks: Theory and experimental analysis. *IEEE Trans. Inf. Theory*, 58(7):4948–4966, 2012.
10. Jiqiang Lu. Cryptanalysis of block ciphers. *mat.uniroma3.it*.
11. Jiqiang Lu. The (related-key) impossible boomerang attack and its application to the AES block cipher. *Des. Codes Cryptogr.*, 60(2):123–143, 2011.
12. Nicolas T. Courtois and Gregory V. Bard. Algebraic cryptanalysis of the data encryption standard. In Steven D. Galbraith, editor, *Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings*, volume 4887 of *Lecture Notes in Computer Science*, pages 152–169. Springer, 2007.
13. Abdel Alim Kamal and Amr M. Youssef. Applications of SAT solvers to AES key recovery from decayed key schedule images. In Reijo Savola, Masaru Takesue, Rainer Falk, and Manuela Popescu, editors, *Fourth International Conference on Emerging Security Information Systems and Technologies, SECURWARE 2010, Venice, Italy, July 18-25, 2010*, pages 216–220. IEEE Computer Society, 2010.
14. Nicky Mouha and Bart Preneel. Towards finding optimal differential characteristics for arx: Application to salsa20. Cryptology ePrint Archive, Paper 2013/328, 2013. https://eprint.iacr.org/2013/328.
15. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

16. Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 158–178. Springer, 2014.
17. David Gérault, Marine Minier, and Christine Solnon. Constraint programming models for chosen key differential cryptanalysis. In Michel Rueher, editor, *Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings*, volume 9892 of *Lecture Notes in Computer Science*, pages 584–601. Springer, 2016.
18. Siwei Sun, David Gérault, 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.
19. Tingting Cui, Keting Jia, Kai Fu, Shiyao Chen, and Meiqin Wang. New automatic search tool for impossible differentials and zero-correlation linear approximations. *IACR Cryptol. ePrint Arch.*, page 689, 2016.
20. Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysist aspects - revealing structural properties of several ciphers. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 185–215, 2017.
21. Xichao Hu, Yongqiang Li, Lin Jiao, Shizhu Tian, and Mingsheng Wang. Mind the propagation of states - new automatic search tool for impossible differentials and impossible polytopic transitions. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 415–445. Springer, 2020.
22. Sean Murphy. The return of the cryptographic boomerang. *IEEE Trans. Inf. Theory*, 57(4):2517–2521, 2011.
23. Alex Biryukov, Christophe De Cannière, and Gustaf Dellkrantz. Cryptanalysis of SAFER++. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2003.
24. Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. *J. Cryptol.*, 27(4):824–849, 2014.
25. Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, *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*, volume 10821 of *Lecture Notes in Computer Science*, pages 683–714. Springer, 2018.

26. Haoyang Wang and Thomas Peyrin. Boomerang switch in multiple rounds. application to AES variants and deoxys. *IACR Trans. Symmetric Cryptol.*, 2019(1):142–169, 2019.
27. Hamid Boukerrou, Paul Huynh, Virginie Lallemand, Bimal Mandal, and Marine Minier. On the feistel counterpart of the boomerang connectivity table introduction and analysis of the FBCT. *IACR Trans. Symmetric Cryptol.*, 2020(1):331–362, 2020.
28. Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. Catching the fastest boomerangs application to SKINNY. *IACR Trans. Symmetric Cryptol.*, 2020(4):104–129, 2020.
29. Dachao Wang, Baocang Wang, and Siwei Sun. Sat-aided automatic search of boomerang distinguishers for ARX ciphers. *IACR Trans. Symmetric Cryptol.*, 2023(1):152–191, 2023.
30. Jiali Choy and Huihui Yap. Impossible boomerang attack for block cipher structures. In Tsuyoshi Takagi and Masahiro Mambo, editors, *Advances in Information and Computer Security, 4th International Workshop on Security, IWSEC 2009, Toyama, Japan, October 28-30, 2009, Proceedings*, volume 5824 of *Lecture Notes in Computer Science*, pages 22–37. Springer, 2009.
31. Jianing Zhang, Haoyang Wang, and Deng Tang. Impossible boomerang attacks revisited applications to deoxys-bc, joltik-bc and SKINNY. *IACR Trans. Symmetric Cryptol.*, 2024(2):254–295, 2024.
32. Xavier Bonnetain, Margarita Cordero, Virginie Lallemand, Marine Minier, and María Naya-Plasencia. On impossible boomerang attacks application to simon and skinnyee. *IACR Trans. Symmetric Cryptol.*, 2024(2):222–253, 2024.
33. Chenmeng Li, Baofeng Wu, and Dongdai Lin. Generalized boomerang connectivity table and improved cryptanalysis of GIFT. In Yi Deng and Moti Yung, editors, *Information Security and Cryptology - 18th International Conference, Inscrypt 2022, Beijing, China, December 11-13, 2022, Revised Selected Papers*, volume 13837 of *Lecture Notes in Computer Science*, pages 213–233. Springer, 2022.
34. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.
35. U.S. Department of Commerce National Bureau of Standards. Data encryption standard. *Federal Information Processing Standards Publication 46*, 1977.
36. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
37. Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. Printcipher: A block cipher for ic-printing. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.
38. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015*, pages 175:1–175:6. ACM, 2015.

39. Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.
40. Dongyoung Roh, Bonwook Koo, Younghoon Jung, Ilwoong Jeong, Donggeon Lee, Daesung Kwon, and Woo-Hwan Kim. Revised version of block cipher CHAM. In Jae Hong Seo, editor, *Information Security and Cryptology - ICISC 2019 - 22nd International Conference, Seoul, South Korea, December 4-6, 2019, Revised Selected Papers*, volume 11975 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2019.
41. Vasily Dolmatov. GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms. RFC 5830, March 2010.
42. Qian Wang and Chenhui Jin. Upper bound of the length of truncated impossible differentials for AES. *Des. Codes Cryptogr.*, 86(7):1541–1552, 2018.
43. Hosein Hadipour, Simon Gerhalter, Sadegh Sadeghi, and Maria Eichlseder. Improved search for integral, impossible differential and zero-correlation attacks application to ascon, forkskinny, skinny, mantis, PRESENT and qarmav2. *IACR Trans. Symmetric Cryptol.*, 2024(1):234–325, 2024.
44. Xichao Hu, Yongqiang Li, Lin Jiao, Shizhu Tian, and Mingsheng Wang. Mind the propagation of states new automatic search tool for impossible differentials and impossible polytopic transitions (full version). *IACR Cryptol. ePrint Arch.*, page 1093, 2020.
45. Ling Song, Xianrui Qin, and Lei Hu. Boomerang connectivity table revisited. application to SKINNY and AES. *IACR Trans. Symmetric Cryptol.*, 2019(1):118–141, 2019.
46. Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971.
47. 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.

A Constructing IBDs Based on Generalized BCTs for Other Structure Block Ciphers

A.1 Constructing IBDs based on generalized BCTs for Feistel-network block ciphers

The incompatibility of boomerang distinguishers resulting from the dependence in Feistel-network block ciphers was observed by Boukerrou et al. [27]. To address this problem, they extended the BCT and BDT to Feistel-network and proposed the concepts of FBCT, FBDT, and FBET. For constructing IBDs, we generalize the definition of FBCT and FBDT, and define GFBCT and GFUBCT/GFLBCT accordingly. Additionally, the generalized table of FEBCT is, in fact, defined identically to that of GEBCT.

Definition 18. *Given nine differences $\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi' \in \mathbb{F}_2^n$ where $\rho' = \mu \oplus \mu' \oplus \rho$, $\varphi' = \theta \oplus \theta' \oplus \varphi$), the GFBCT, GFUBCT and GFLBCT for an n-bit S-box is defined as*

$$\mathrm{GFBCT}(\mu, \mu', \rho, \rho', \eta) = \#\left\{ (x_0, x_1, x_2, x_3) \in \{0,1\}^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ x_1 \oplus x_2 = \rho \\ x_2 \oplus x_3 = \rho' \\ \oplus_{i=0}^{3} S(x_i) = \eta \end{array} \right\},$$

$$\mathrm{GFUBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \eta) = \#\left\{ (x_0, x_1, x_2, x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ x_1 \oplus x_2 = \rho \\ x_2 \oplus x_3 = \rho' \\ \oplus_{i=0}^{3} S(x_i) = \eta \\ S(x_0) \oplus S(x_1) = \theta \\ S(x_2) \oplus S(x_3) = \theta' \end{array} \right\},$$

$$\mathrm{GFLBCT}(\mu, \mu', \rho, \rho', \varphi, \varphi', \eta) = \#\left\{ (x_0, x_1, x_2, x_3) \in \mathbb{F}_2^{4n} \middle| \begin{array}{l} x_0 \oplus x_1 = \mu \\ x_2 \oplus x_3 = \mu' \\ x_1 \oplus x_2 = \rho \\ x_2 \oplus x_3 = \rho' \\ \oplus_{i=0}^{3} S(x_i) = \eta \\ S(x_1) \oplus S(x_2) = \varphi \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\}.$$

A schematic diagram for these generalized BCTs is shown in Figure 8. To il-

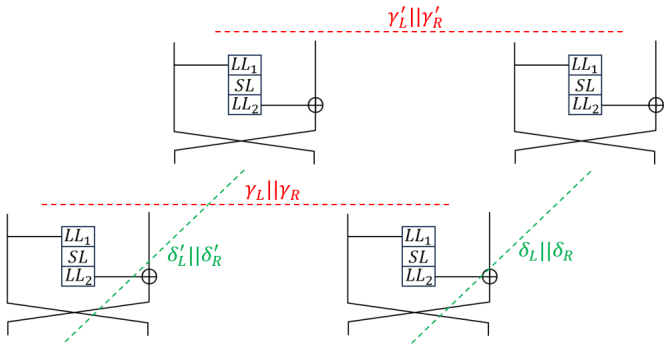

Fig. 8: The illustration of differential propagation rule through S-boxes in E based on GFBCT

lustrate the relations of these generalized tables, we introduce some notations accordingly.

Notation 1. *For an n-bit Feistel-network block cipher E, let $x = (x_L \| x_R) \in \mathbb{F}_2^n$ be E's state, where x_L (resp. x_R) denotes the state of E's left (resp. right) branch. Let $\gamma, \gamma', \eta, \eta' \in \mathbb{F}_2^n$ be four input differences and $\omega, \omega', \delta, \delta', \in \mathbb{F}_2^n$ be four output differences of E. Then,*

1. $(\gamma,\gamma') \xrightarrow{GFBCT} (\delta,\delta')$ *represents that the propagation rule through S-boxes follows GFBCT. Specifically, for an* E *as shown in Figure 8, let* $\gamma = \gamma_L||\gamma_R$, $\gamma' = \gamma'_L||\gamma'_R$, $\delta = \delta_L||\delta_R$, $\delta' = \delta'_L||\delta'_R$, *and* $LL_1(\gamma_L) = (a_{t-1},\ldots,a_0)$, $LL_1(\gamma_{L'}) = (a'_{t-1},\ldots,a'_0)$, $LL_1(\delta_R) = (b_{t-1},\ldots,b_0)$, $LL_1(\delta'_R) = (b'_{t-1},\ldots,b'_0)$, $LL_2^{-1}(\gamma_R \oplus \gamma'_R \oplus \delta_L \oplus \delta'_L) = (c_{t-1},\ldots,c_0)$, *where* LL_i *for* $i = 1,2$ *are linear layers and there are* t *S-boxes in* SL. *Thus* a_i, a'_i, b_i, b'_i *and* c_i *are the differences corresponding to GFBCT of the* i*-th S-box for* $i = 0,\ldots,t-1$. *Then,* $(\gamma,\gamma') \xrightarrow{GFBCT} (\delta,\delta')$ *is equivalent to that there exists* $i \in \{0,\ldots,t-1\}$ *such that* $\mathrm{GFBCT}(a_i, a'_i, b_i, b'_i, c_i) \neq 0$.
2. $(\gamma,\gamma') \xrightarrow{GFUBCT} (\omega,\omega',\delta,\delta')$ *represents that the propagation rule through S-boxes follows GFUBCT, which is equivalent to that* $(\gamma,\gamma') \xrightarrow{GFBCT} (\delta,\delta')$ *and* $(\gamma,\gamma') \xrightarrow{UDDT} (\omega,\omega')$.
3. $(\gamma,\gamma',\eta,\eta') \xrightarrow{GFLBCT} (\delta,\delta')$ *represents that the propagation rule through S-boxes follows GFLBCT, which is equivalent to that* $(\gamma,\gamma') \xrightarrow{GFBCT} (\delta,\delta')$ *and* $(\eta,\eta') \xrightarrow{LDDT} (\delta,\delta')$.
4. $(\gamma,\gamma',\eta,\eta') \xrightarrow{GEBCT} (\omega,\omega',\delta,\delta')$ *represents that the propagation rule through S-boxes follows GEBCT, which is equivalent to that* $(\gamma,\gamma') \xrightarrow{GFUBCT} (\omega,\omega',\delta,\delta')$ *and* $(\gamma,\gamma',\eta,\eta') \xrightarrow{GFLBCT} (\delta,\delta')$.

Theorem 8. *The table* UDDT, LDDT, GFBCT, GFUBCT, GFLBCT, GEBCT *have the following relations:*

1. *If* $(\gamma,\gamma') \xrightarrow{GFBCT} (\delta,\delta')$, *then* $\exists \eta,\eta',\omega,\omega'$, *s.t.* $(\gamma,\gamma') \xrightarrow{UDDT} (\omega,\omega')$ *and* $(\eta,\eta') \xrightarrow{LDDT} (\delta,\delta')$.
2. *If* $(\gamma,\gamma') \xrightarrow{GFUBCT} (\omega,\omega',\delta,\delta')$, *then* $\exists \eta,\eta'$, *s.t.* $(\gamma,\gamma') \xrightarrow{UDDT} (\omega,\omega')$ *and* $(\eta,\eta') \xrightarrow{LDDT} (\delta,\delta')$.
3. *If* $(\gamma,\gamma',\eta,\eta') \xrightarrow{GFLBCT} (\delta,\delta')$, *then* $\exists \omega,\omega'$, *s.t.* $(\gamma,\gamma') \xrightarrow{UDDT} (\omega,\omega')$ *and* $(\eta,\eta') \xrightarrow{LDDT} (\delta,\delta')$.
4. *If* $(\gamma,\gamma',\eta,\eta') \xrightarrow{GEBCT} (\omega,\omega',\delta,\delta')$, *then* $(\gamma,\gamma') \xrightarrow{UDDT} (\omega,\omega')$ *and* $(\eta,\eta') \xrightarrow{LDDT} (\delta,\delta')$, $(\gamma,\gamma') \xrightarrow{GFUBCT} (\omega,\omega',\delta,\delta')$ *and* $(\gamma,\gamma',\eta,\eta') \xrightarrow{GFLBCT} (\delta,\delta')$.

We also consider a hybrid use of UDDT, LDDT, GFBCT, GFUBCT, GFLBCT and GEBCT to construct IBDs.

Definition 19. *Let* E *be a block cipher with* t *S-boxes* $(S_0,\ldots,S_{t-1})$ *in total. Define* $\mathcal{AP}_E^F = \{(p_0,\ldots,p_{t-1}) | p_i \in \{\mathrm{UDDT}, \mathrm{LDDT}, \mathrm{GFBCT}, \mathrm{GFUBCT}, \mathrm{GFLBCT}, \mathrm{GEBCT}\}\}$ *as a set of propagation rules. Then* $P = (p_0,\ldots,p_{t-1}) \in \mathcal{AP}_E^F$, *denotes that the propagation rule through the* i*-th S-box follows* p_i.

Definition 20. *Let* $E = E_{r-1,rk_{r-1}} \circ \cdots \circ E_{0,rk_0}(x)$ *be an* r*-round block cipher. Let* $P = (P_0,\ldots,P_{r-1})$ *be a predefined propagation rule of* E, *where* $P_i \in \mathcal{AP}^F_{E_{i,rk_i}}$ *denotes a propagation rule of* E_{i,rk_i} *for* $i \in \{0,\ldots,r-1\}$. *Let* $\epsilon_0^i, \epsilon_1^i, \epsilon_2^i, \epsilon_3^i$ *be the four input differences and* $\epsilon_0^{i+1}, \epsilon_1^{i+1}, \epsilon_2^{i+1}, \epsilon_3^{i+1}$ *be the four output differences of the round function* E_{i,rk_i} *for* $i \in \{0,\ldots,r-1\}$. *For two input differences* α,α' *and two output differences* β,β' *of the block cipher* E, *if there exists a trail*

$$(\epsilon_0^0 = \alpha, \epsilon_1^0, \epsilon_2^0 = \alpha', \epsilon_3^0) \xrightarrow{P_0} \cdots \xrightarrow{P_{r-1}} (\epsilon_0^r, \epsilon_1^r = \beta, \epsilon_2^r, \epsilon_3^r = \beta'),$$

then it is called an r*-round* T_P^F *boomerang trail. Here,* $\xrightarrow{P_i}$ *represents that the propagation rule through S-boxes in* E_{i,rk_i} *follows* P_i.

Accordingly, we have the following construction.

Construction 9 (T_P^F-IBD). *Given an r-round block cipher E and a predefined rule $P \in \mathcal{AP}_E^F$, for two input differences α, α' and two output differences β, β', if there is no r-round T_P^F boomerang trail, then $((\alpha, \alpha'), (\beta, \beta'))$ is an IBD, called an r-round T_P^F-IBD.*

T_1-IBD is a special example of T_P^S-IBD.

Theorem 9. *For any predefined rule $P \in \mathcal{AP}_E^F$, an r-round T_1-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_P^F-IBD.*

Theorem 8 demonstrates that GEBCT in Feistel-network block ciphers has a similar status in SPN block ciphers. Additionally, the definition of T_C-IBD is also applicable to Feistel-network block ciphers, Furthermore, T_C-IBD is also a special case of T_P^F-IBD.

Theorem 10. *For any predefined rule $P \in \mathcal{AP}_E^F$, an r-round T_P^F-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_C-IBD.*

In addition to the above relationship, we can prove that the definition of T_C-IBD is equivalent with that of T_2-IBD within Feistel-network block ciphers. A schematic diagram is shown in Figure 10.

Theorem 11. *Given an Feistel-network block cipher, $((\alpha, \alpha'), (\beta, \beta'))$ is an r-round T_C-IBD if and only if it is an r-round T_2-IBD.*

B Proofs

Theorem 1

Proof (proof by contradiction). If an r-round T_0-IBD $((\alpha, \alpha'), (\beta, \beta'))$ is not an r-round T_1-IBD, there must exist one r-round T_1 boomerang trail:

$$(\alpha, \alpha') \to \cdots \to \underbrace{(\gamma, \gamma')(\delta, \delta')}_{\gamma \oplus \gamma' \oplus \delta \oplus \delta' = 0} \to \cdots \to (\beta, \beta'),$$

which is an r-round T_0 boomerang trail. Thus, $((\alpha, \alpha'), (\beta, \beta'))$ is neither an r-round T_0-IBD. □

Theorem 2

Proof (proof by contradiction). According to the definitions, an r-round T_2 boomerang trail is also an r-round T_3 boomerang trail. □

Theorem 3

Proof. (Definition 7⇒ Construction 4) Let $((\alpha, \alpha'), (\beta, \beta'))$ be an r-round IBD, then any pair of plaintexts (x_0, x_3) cannot simultaneously satisfy $E_k(x_0) \oplus E_k(x_3) = \beta$ and $E_k(x_0 \oplus \alpha) \oplus E_k(x_3 \oplus \alpha') = \beta'$. If $((\alpha, \alpha'), (\beta, \beta'))$ is not an r-round T_3-IBD. Let $x_0^0 = x_0$, $x_1^0 = x_0 \oplus \alpha$, $x_3^0 = x_3$, $x_2^0 = x_3 \oplus \alpha'$, there exist an r-round T_3 boomerang trail $(x_0^0, x_1^0, x_2^0, x_3^0) \to \cdots \to (x_0^r, x_1^r, x_2^r, x_3^r)$, where $x_1^r \oplus x_2^r = \beta$ and $x_0^r \oplus x_3^r = \beta'$. Thus $E_k(x_0) \oplus E_k(x_3) = \beta$ and $E_k(x_0 \oplus \alpha) \oplus E_k(x_3 \oplus \alpha') = \beta'$, which is a contradiction.

(Construction 4⇒ Definition 7) Let $((\alpha, \alpha'), (\beta, \beta'))$ be an r-round T_3-IBD. then there is not any r-round T_3 boomerang trail $(x_0^0, x_1^0, x_2^0, x_3^0) \to \cdots \to (x_0^r, x_1^r, x_2^r, x_3^r)$. Thus, any pair of (x_0^0, x_3^0) cannot simultaneously meet $E_k(x_0^0) \oplus E_k(x_3^0) = \beta$ and $E_k(x_0^0 \oplus \alpha) \oplus E_k(x_3^0 \oplus \alpha') = \beta'$, which is according with Definition 7. □

Theorem 5

Proof (proof by contradiction). If $\exists P \in \mathcal{AP}_E^S$ and an r-round T_1-IBD $((\alpha,\alpha'),(\beta,\beta'))$ such that it is not an r-round T_P^S-IBD, there must exist at least one r-round T_P^S boomerang trail: $(\epsilon_0^0=\alpha,\epsilon_1^0,\epsilon_2^0=\alpha',\epsilon_3^0) \xrightarrow{P_0} \cdots \xrightarrow{P_{r-1}} (\epsilon_0^r,\epsilon_1^r=\beta,\epsilon_2^r,\epsilon_3^r=\beta')$. Based on the relations of tables in Theorem 4, it is also an r-round T_1-IBD boomerang trail. Thus, $((\alpha,\alpha'),(\beta,\beta'))$ is neither an r-round T_P^S-*IBD*. □

Theorem 6

Proof (proof by contradiction). If an r-round T_P^S-IBD $((\alpha,\alpha'),(\beta,\beta'))$ is not an r-round T_C-*IBD*, there must exist at least one r-round T_C boomerang trail: $(\epsilon_0^0,\epsilon_1^0,\epsilon_2^0,\epsilon_3^0) \xrightarrow{GEBCT} \cdots \xrightarrow{GEBCT} (\epsilon_0^{r_0},\epsilon_1^{r_0},\epsilon_2^{r_0},\epsilon_3^{r_0}) \xrightarrow{GEBCT} (\epsilon_0^{r_0+1},\epsilon_1^{r_0+1},\epsilon_2^{r_0+1},\epsilon_3^{r_0+1}) \xrightarrow{GEBCT} \cdots \xrightarrow{GEBCT} (\epsilon_0^r,\epsilon_1^r,\epsilon_2^r,\epsilon_3^r)$. Based on the relations of tables in Theorem 4, it is also an r-round T_P^S boomerang trail. Thus, $((\alpha,\alpha'),(\beta,\beta'))$ is neither an r-round T_C-*IBD*. □

Theorem 7

Proof. This is equivalent to prove that $(\epsilon_0^0,\epsilon_1^0,\epsilon_2^0,\epsilon_3^0) \overset{\text{AddKey}}{\rightarrow} (\gamma_0^0,\gamma_1^0,\gamma_2^0,\gamma_3^0) \overset{\text{GEBCT}}{\rightarrow} (\delta_0^0,\delta_1^0,\delta_2^0,\delta_3^0) \overset{\text{LL}}{\rightarrow} (\epsilon_0^1,\epsilon_1^1,\epsilon_2^1,\epsilon_3^1) \overset{\text{AddKey}}{\rightarrow} \cdots \overset{\text{GEBCT}}{\rightarrow} (\delta_0^{r-1},\delta_1^{r-1},\delta_2^{r-1},\delta_3^{r-1})$ is an r-round T_C boomerang trail if and only if $(x_0^0,x_1^0,x_2^0,x_3^0) \overset{\text{AddKey}}{\rightarrow} (y_0^0,y_1^0,y_2^0,y_3^0) \overset{\text{SL}}{\rightarrow} (z_0^0,z_1^0,z_2^0,z_3^0) \overset{\text{LL}}{\rightarrow} (x_0^1,x_1^1,x_2^1,x_3^1) \overset{\text{AddKey}}{\rightarrow} \cdots \overset{\text{SL}}{\rightarrow} (z_0^{r-1},z_1^{r-1},z_2^{r-1},z_3^{r-1})$ is an r-round T_2 boomerang trail, where $\alpha=\epsilon_0^0,\alpha'=\epsilon_2^0,\beta=\epsilon_1^{r-1},\beta'=\epsilon_3^{r-1}$, and $\alpha=x_0^0\oplus x_1^0,\alpha'=x_2^0\oplus x_3^0$, $\beta=z_1^{r-1}\oplus z_2^{r-1}$ and $\beta'=z_0^{r-1}\oplus z_3^{r-1}$. In particular, we prove this in the case of $r=3$. The other cases can be proved analogously. Suppose $(\epsilon_0^0,\epsilon_1^0,\epsilon_2^0,\epsilon_3^0) \overset{\text{AddKey}}{\rightarrow} (\gamma_0^0,\gamma_1^0,\gamma_2^0,\gamma_3^0) \overset{\text{GEBCT}}{\rightarrow} (\delta_0^0,\delta_1^0,\delta_2^0,\delta_3^0) \overset{\text{LL}}{\rightarrow} (\epsilon_0^1,\epsilon_1^1,\epsilon_2^1,\epsilon_3^1) \overset{\text{AddKey}}{\rightarrow} \cdots \overset{\text{GEBCT}}{\rightarrow} (\delta_0^2,\delta_1^2,\delta_2^2,\delta_3^2)$ is an 3-round T_C boomerang trail. Since $(\gamma_0^i,\gamma_1^i,\gamma_2^i,\gamma_3^i) \overset{\text{SL, GEBCT}}{\rightarrow} (\delta_0^i,\delta_1^i,\delta_2^i,\delta_3^i)$, there exists $(y_0^i,y_1^i,y_2^i,y_3^i)$ and $(z_0^i,z_1^i,z_2^i,z_3^i)$, such that

$$y_0^i\oplus y_1^i=\gamma_0^i, y_1^i\oplus y_2^i=\gamma_1^i, y_2^i\oplus y_3^i=\gamma_2^i, y_0^i\oplus y_3^i=\gamma_3^i,$$
$$z_0^i\oplus z_1^i=\delta_0^i, z_1^i\oplus z_2^i=\delta_1^i, z_2^i\oplus z_3^i=\delta_2^i, z_0^i\oplus z_3^i=\delta_3^i.$$

Let $x_i^0=y_i^0\oplus rk_0$ $(0\le i\le 3)$ and $rk_j=LL(z_x^{j-1})\oplus y_i^j$ $(0\le i\le 3, j=1,2)$, then $(x_0^0,x_1^0,x_2^0,x_3^0) \overset{\text{AddKey}}{\rightarrow} (y_0^0,y_1^0,y_2^0,y_3^0) \overset{\text{SL}}{\rightarrow} (z_0^0,z_1^0,z_2^0,z_3^0) \overset{\text{LL}}{\rightarrow} (x_0^1,x_1^1,x_2^1,x_3^1) \overset{\text{AddKey}}{\rightarrow} \cdots \overset{\text{SL}}{\rightarrow} (z_0^{r-1},z_1^{r-1},z_2^{r-1},z_3^{r-1})$ is an 3-round T_2 boomerang trail. The above process is invertible. □

Theorem 8

Proof. As shown in Figure 9, let l_i be the linear function of $LL_i (i=1,2)$, $\gamma=\gamma_L||\gamma_R$, $\gamma'=\gamma'_L||\gamma'_R$, $\delta=\delta_L||\delta_R$, and $\delta'=\delta'_L||\delta'_R$. $(a_{t-1},\ldots,a_0)=l_1(\gamma_L)$, $(a'_{t-1},\ldots,a'_0)=l_1(\gamma_{L'})$, $(b_{t-1},\ldots,b_0)=l_1(\delta_R)$, $(b'_{t-1},\ldots,b'_0)=l_1(\delta'_R)$, and $(c_{t-1},\ldots,c_0)=l_2^{-1}(\gamma_R\oplus\gamma'_R\oplus\delta_L\oplus\delta'_L)$, where a_i,a'_i,b_i,b'_i and c_i are the input or the output difference of the S-box in SL and t is the number of the S-boxes in SL. For the input differences a_i and a'_i, according to the definition of GFBCT, there exist output differences d_i and d'_i, such that a_i propagates to d_i and a_i propagates to d_i, and $d_i\oplus d'_i=c_i (0\le i\le t-1)$. Let $\lambda=l_2((d_{t-1},\ldots,d_0))$ and $\lambda'=l_2((d'_{t-1},\ldots,d'_0))$, then $\gamma_R\oplus\gamma'_R\oplus\delta_L\oplus\delta'_L=\lambda\oplus\lambda'$. Therefore, there exist λ and λ' with $\gamma_R\oplus\gamma'_R\oplus\delta_L\oplus\delta'_L=\lambda\oplus\lambda'$, such that the difference γ can propagate to the difference $(\gamma_R\oplus\lambda)||\gamma_L$, and the difference γ' can propagate to the difference $(\gamma'_R\oplus\lambda')||\gamma'_L$. That is, $(\gamma,\gamma') \xrightarrow{UDDT} ((\gamma_R\oplus\lambda)||\gamma_L,\gamma'_R\oplus\lambda')||\gamma'_L)$. For other relations, we can prove them similarly.

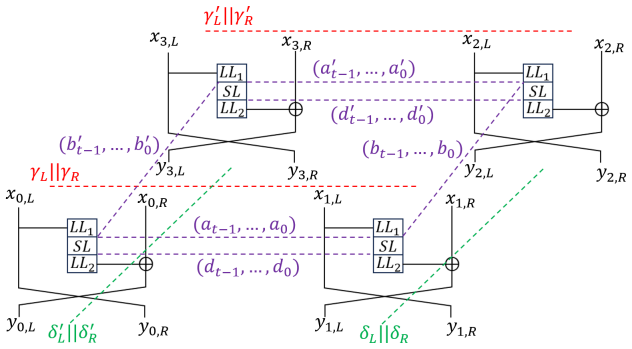

Fig. 9: The relation between GFBCT and UDDT

Theorem 9

Proof (proof by contradiction). If $\exists P \in \mathcal{AP}_E^F$ and an r-round T_1-IBD $((\alpha,\alpha'),(\beta,\beta'))$ such that it is not an r-round T_P^F-IBD, there must exist at least one r-round T_P^F boomerang trail: $(\epsilon_0^0=\alpha,\epsilon_1^0,\epsilon_2^0=\alpha',\epsilon_3^0) \xrightarrow{P_0} \cdots \xrightarrow{P_{r-1}} (\epsilon_0^r,\epsilon_1^r=\beta,\epsilon_2^r,\epsilon_3^r=\beta')$. Based on the Theorem 8, the T_P^F boomerang trail is also an r-round T_1-IBD boomerang trail. Thus, $((\alpha,\alpha'),(\beta,\beta'))$ is neither an r-round T_P^F-*IBD*. □

Theorem 10

Proof (proof by contradiction). If an r-round T_P^F-IBD $((\alpha,\alpha'),(\beta,\beta'))$ is not an r-round T_C-*IBD*, there must exist at least one r-round T_C boomerang trail: $(\epsilon_0^0,\epsilon_1^0,\epsilon_2^0,\epsilon_3^0) \xrightarrow{GEBCT} \cdots \xrightarrow{GEBCT} (\epsilon_0^{r_0},\epsilon_1^{r_0},\epsilon_2^{r_0},\epsilon_3^{r_0}) \xrightarrow{GEBCT} (\epsilon_0^{r_0+1},\epsilon_1^{r_0+1},\epsilon_2^{r_0+1},\epsilon_3^{r_0+1}) \xrightarrow{GEBCT} \cdots \xrightarrow{GEBCT} (\epsilon_0^r,\epsilon_1^r,\epsilon_2^r,\epsilon_3^r)$. Based on the Theorem 8, it is also an r-round T_P^S boomerang trail. Thus, $((\alpha,\alpha'),(\beta,\beta'))$ is neither an r-round T_C-*IBD*. □

Theorem 11

Proof. This is equivalent to prove that $((\epsilon_0^0,\epsilon_1^0,\epsilon_2^0,\epsilon_3^0),(\omega_0^0,\omega_1^0,\omega_2^0,\omega_3^0)) \overset{\text{GEBCT}}{\to} \cdots \overset{\text{GEBCT}}{\to} \cdots \overset{\text{GEBCT}}{\to} ((\epsilon_0^r,\epsilon_1^r,\epsilon_2^r,\epsilon_3^r),(\omega_0^r,\omega_1^r,\omega_2^r,\omega_3^r))$ is an r-round T_C boomerang trail if and only if $((x_0^0,x_1^0,x_2^0,x_3^0),(y_0^0,y_1^0,y_2^0,y_3^0)) \to \cdots \to \cdots \to ((x_0^r,x_1^r,x_2^r,x_3^r),(y_0^r,y_1^r,y_2^r,y_3^r))$ is an r-round T_2 boomerang trail, where $(\alpha_L,\alpha_R)=(\epsilon_0^0,\omega_0^0),(\alpha'_L,\alpha'_R)=(\epsilon_2^0,\omega_2^0),(\beta_L,\beta_R)=(\epsilon_1^r,\omega_1^r)$, and $(\beta'_L,\beta'_R)=(\epsilon_3^r,\omega_3^r)$, and $(\alpha_L,\alpha_R)=(x_0^0\oplus x_1^0, y_0^0\oplus y_1^0),(\alpha'_L,\alpha'_R)=(x_2^0\oplus x_3^0, y_2^0\oplus y_3^0),(\beta_L,\beta_R)=(x_0^r\oplus x_2^r, y_0^r\oplus y_2^r)$, and $(\beta'_L,\beta'_R)=(x_1^r\oplus x_3^r, y_1^r\oplus y_3^r)$. In particular, we prove this in the case of $r=3$. The other cases can be proved analogously. Suppose $((\epsilon_0^0,\epsilon_1^0,\epsilon_2^0,\epsilon_3^0),(\omega_0^0,\omega_1^0,\omega_2^0,\omega_3^0)) \overset{\text{GEBCT}}{\to} \cdots \overset{\text{GEBCT}}{\to} \cdots \overset{\text{GEBCT}}{\to} ((\epsilon_0^3,\epsilon_1^3,\epsilon_2^3,\epsilon_3^3),(\omega_0^3,\omega_1^3,\omega_2^3,\omega_3^3))$ is an 3-round T_C boomerang trail. For $0 \le i \le 2$, since $(\gamma_0^i,\gamma_1^i,\gamma_2^i,\gamma_3^i) \overset{\text{SL, GEBCT}}{\to} (\delta_0^i,\delta_1^i,\delta_2^i,\delta_3^i)$, there exists $(u_0^i,u_1^i,u_2^i,u_3^i)$ and $(v_0^i,v_1^i,v_2^i,v_3^i)$, such that

$$u_0^i \oplus u_1^i = \gamma_0^i, u_1^i \oplus u_2^i = \gamma_1^i, u_2^i \oplus u_3^i = \gamma_2^i, u_0^i \oplus u_3^i = \gamma_3^i,$$
$$v_0^i \oplus v_1^i = \delta_0^i, v_1^i \oplus v_2^i = \delta_1^i, v_2^i \oplus v_3^i = \delta_2^i, v_0^i \oplus v_3^i = \delta_3^i.$$

Let l_j be the linear function of $LL_j (j=1,2)$, $(z_0^i,z_1^i,z_2^i,z_3^i) = l_1^{-1}(u_0^i,u_1^i,u_2^i,u_3^i)$, and $(w_0^i,w_1^i,w_2^i,w_3^i) = l_2(v_0^i,v_1^i,v_2^i,v_3^i)$. Then $z_0^i \oplus z_1^i = x_0^i \oplus x_1^i$, $z_1^i \oplus z_2^i = x_1^i \oplus x_2^i$, $z_2^i \oplus z_3^i = x_2^i \oplus x_3^i$, and $z_0^i \oplus z_3^i = x_0^i \oplus x_3^i$. Let $rk_i = x_0^i \oplus z_0^i$, then $rk_i = x_j^i \oplus z_j^i (0 \le j \le 3)$.

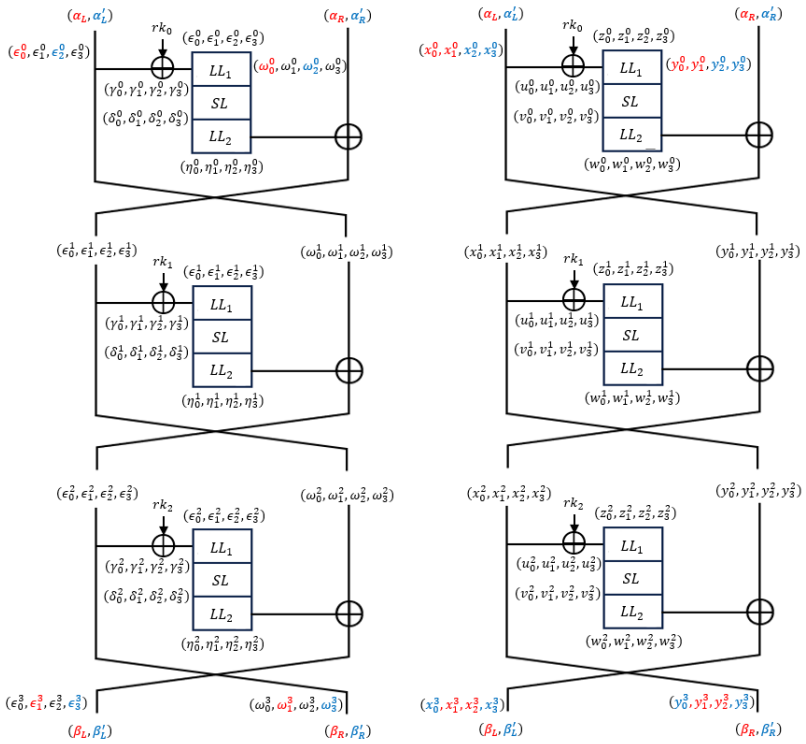

Fig. 10: The equivalence between T_C-IBD and T_2-IBD in Feistel-network

Let $x_j^{i+1} = w_j^i \oplus y_j^i (0 \le j \le 3)$, then $((x_0^0, x_1^0, x_2^0, x_3^0), (y_0^0, y_1^0, y_2^0, y_3^0)) \to \cdots \to \cdots \to ((x_0^3, x_1^3, x_2^3, x_3^3), (y_0^3, y_1^3, y_2^3, y_3^3))$ is an r-round T_2 boomerang trail. The above process is invertible. □

C The Algorithm of Searching for (RK-)IBDs

C.1 The algorithm of searching for the (RK-)IBDs from the aspect of differential propagation

A brief illustration to Algorithm 1 is provided as follows.

- As our experimental observations indicate that contradictions of IBDs may occur at different rounds; hence we have made r_u as a input parameter in Algorithm 1.
- Line 4-7: Function $BuildUpDP(r, x, z, kx, kz)$ forms the relations of differential propagation of $x \xrightarrow{E^0, kx} z$ and $kx \xrightarrow{U_{\mathrm{KS}}^0} kz$; Function $BuildLowDP(r, y, z, kx, kz, ky)$ forms the relations of differential propagation of $y \xrightarrow{(E^1)^{-1}, kx} z$, $kx \xrightarrow{U_{\mathrm{KS}}^0} kz$ and $kx \xrightarrow{U_{\mathrm{KS}}} ky$ according to the differential propagation rule, where x, z, y denote the variables for state difference of E, and kx, kz, ky denote the variables for state difference of KS.

Algorithm 1: Identifying $(R)T_0$-IBD or $(R)T_1$-IBD

Input: input differences (α, α'), output differences (β, β'), key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$, E's round number r, E_0's round number r_u

Output: Model C

1 Declare six variables of input differences $x_{01}, x_{23}, kx_{01}, kx_{23}, kx_{12}, kx_{03}$ and four variables of output differences $y_{12}, y_{03}, ky_{12}, ky_{03}$; Declare eight intermediate variables of differences $z_{01}, z_{23}, z_{12}, z_{03}$ and $kz_{01}, kz_{23}, kz_{12}, kz_{03}$

2 $C_0 = \mathit{BuildUpDP}(r_u, x_{01}, z_{01}, kx_{01}, kz_{01})$

3 $C_1 = \mathit{BuildUpDP}(r_u, x_{23}, z_{23}, kx_{23}, kz_{23})$

4 $C_2 = \mathit{BuildLowDP}(r - r_u, y_{12}, z_{12}, kx_{12}, kz_{12}, ky_{12})$

5 $C_2 = \mathit{BuildLowDP}(r - r_u, y_{03}, z_{03}, kx_{03}, kz_{03}, ky_{03})$

6 $C_4 = \mathit{DiffConnectBD}(z_{01}, z_{23}, z_{12}, z_{03})$

7 $C_5 = \mathit{DiffConnectKey}(kz_{01}, kz_{23}, kz_{12}, kz_{03})$

8 $C_6 = \mathit{SetDiffIn}(x_{01}, x_{23}, \alpha, \alpha')$

9 $C_7 = \mathit{SetDiffOut}(y_{12}, y_{03}, \beta, \beta')$

10 $C_8 = \mathit{SetDiffKey}(kx_{01}, kx_{23}, kx_{12}, kx_{03}, \kappa_0, \kappa_1, \kappa_2, \kappa_3)$

11 $C = [C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9]$

12 **return** C

- Line 8: Function $\mathit{DiffConnectBD}(z_{01}, z_{23}, z_{12}, z_{03})$ sets $z_{01} \oplus z_{23} \oplus z_{12} \oplus z_{03} = 0$ according to the definition of BD.
- Line 9: Function $\mathit{DiffConnectKey}(kz_{01}, kz_{23}, kz_{12}, kz_{03})$ sets $kz_{01} \oplus kz_{23} \oplus kz_{12} \oplus kz_{03} = 0$ according to a trivial elimination.
- Line 10-13: Function *SetDiffIn* and *SetDiffOut* both assign the value of the third parameter to the first parameter and the value of the fourth parameter to the second parameter.
- For T_0-IBD and T_1-IBD, the relations between variables for state difference of KS are all omitted.

Finally, Algorithm 1 returns Model C to SAT solver and identifies an IBD if there exists no solution.

C.2 The algorithm of searching for the (RK-)IBDs from the aspect of state propagation.

A brief illustration to Algorithm 2 is provided as follows.

- Line 2-5: Function $\mathit{BuildSP}(r, x_j, y_j, kx_j, ky_j)$ forms the relations of state propagation of $U_{\mathrm{KS}}(kx_j) = ky_j$ with $rk_i^j = \mathrm{KS}_i(kx)$ for $i = 0, \ldots, r-1$, and $E_{r-1,rk_{r-1}} \circ \cdots \circ E_{0,rk_0}(x_j) = y_j$, according to the state propagation rule, where x_j, y_j denote the variables for state of E, and kx_j, kz_j, ky_j denote the variables for state of KS, for $j = 0, 1, 2, 3$.
- Line 6: Function $\mathit{BuildKeyRelation}(r, rk^0, rk^1, rk^2, rk^3)$ forms the relations of round keys. For T_2-IBD, $rk_i^0 = rk_i^1 = rk_i^2 = rk_i^3$ for $i = 0, \ldots, r-1$. For

Algorithm 2: Model for determining the T_2-IBD and $(R)T_3$-IBD

Input: input differences (α, α'), output differences (β, β'), key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$, E's round number r
Output: Model C

1 Declare eight input variables $x_0, x_1, x_2, x_3, kx_0, kx_1, kx_2, kx_3$ and eight output variables $y_0, y_1, y_2, y_3, ky_0, ky_1, ky_2, ky_3$
2 $C_0, rk_0 = BuildStatePropagation(r, x_0, y_0, kx_0, ky_0)$
3 $C_1, rk_1 = BuildStatePropagation(r, x_1, y_1, kx_1, ky_1)$
4 $C_2, rk_2 = BuildStatePropagation(r, x_2, y_2, kx_2, ky_2)$
5 $C_3, rk_3 = BuildStatePropagation(r, x_3, y_3, kx_3, ky_3)$
6 $C_4 = BuildKeyRelation(r, rk_0, rk_1, rk_2, rk_3)$
7 $C_5 = SetStateIn(x_0, x_1, x_2, x_3, \alpha, \alpha')$
8 $C_6 = SetStateOut(y_0, y_1, y_2, y_3, \beta, \beta')$
9 $C_7 = SetStateKey(kx_0, kx_1, kx_2, kx_3, \kappa_0, \kappa_1, \kappa_2, \kappa_3)$
10 $C = [C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8]$
11 **return** C

$(R)T_3$-IBD, $rk_i^j = \mathrm{KS}_i(kx_j)$ for $i = 0, \ldots, r-1$ and $j = 0, 1, 2, 3$. Especially for T_3-IBD, $kx_0 = kx_1 = kx_2 = kx_3$.

- Line 7: Function $SetStateIn(x_0, x_1, x_2, x_3, \alpha, \alpha')$ sets $x_0 \oplus x_1 = \alpha, x_2 \oplus x_3 = \alpha'$.
- Line 8: Function $SetStateOut(y_0, y_1, y_2, y_3, \beta, \beta')$ sets $y_1 \oplus y_2 = \beta, y_0 \oplus y_3 = \beta'$.
- Line 9: Function $SetStateKey(kx_0, kx_1, kx_2, kx_3, \kappa_0, \kappa_1, \kappa_2, \kappa_3)$ sets $kx_0 \oplus kx_1 = \kappa_0, kx_2 \oplus kx_3 = \kappa_1, kx_1 \oplus kx_2 = \kappa_2, kx_0 \oplus kx_3 = \kappa_3$.

Finally, Algorithm 2 returns Model C to SAT solver and identifies an IBD if there exists no solution.

D Specifications of Block Ciphers

Only brief descriptions of block ciphers for applications are given here. For more details, please refer to their corresponding references.

D.1 Specifications of AES

AES [34] is one of the most renowned block ciphers across the world. Its design philosophy has had a profound impact on block ciphers. AES is a 128 bits block cipher that supports key sizes of 128, 192, and 256 bits, and the S-box size is 8 bits. It is a SPN block cipher that employs the MDS matrix to achieve excellent diffusivity. The internal state is regarded as a square array of bytes as follows, where $s_i \in \mathbb{F}_2^8$ $(0 \leq i \leq 15)$.

$$S = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}.$$

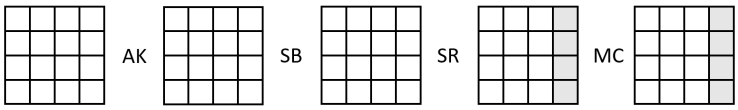

Fig. 11: One round of block cipher AES

One encryption round of AES is depicted in Figure 11, and it consists of the following four operations:

- AddRoundKey(AK): The 128-bit round key which is derived from the key schedule is XORed with the state.
- SubBytes(SB): Applying the -bit S-box to each byte in parallel to the cipher's internal state.
- ShiftRows(SR): The i-th rows ($0 \leq i \leq 3$) of the internal state is rotated by i bytes form right to left.
- Mix-Column(MC): Each column of the internal state is multiplied with the MDS matrix.

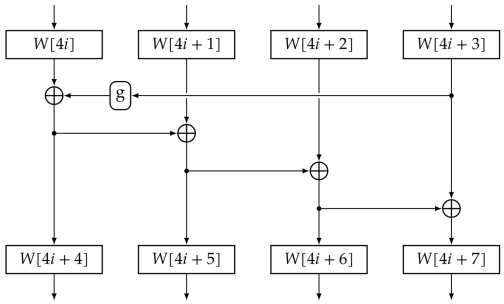

Fig. 12: The key schedule of AES-128

AES-128 is the block cipher AES with is 128 bits key. The key schedule of AES-128 is shown as Figure 12. The function g is a 32-bit to 32-bit function which consists of:

1. Perform a right rotation of the input by 1 byte.
2. Process all four bytes of this rotated input using the AES S-box.
3. Add a fixed round coefficient to the output of the first S-box.

D.2 Specifications of DES

DES [35] is one of the earliest block ciphers to gain widespread adoption. It was standardized for use in a variety of applications, thereby becoming a pioneer in bringing encryption to a broader range of users, including those in commercial and civilian sectors. DES is a 64-bit block cipher with a real key size of 56 bits. It employs eight distinct non-bijective S-boxes where the input of each S-box is 6 bits, and the output is 4 bits. DES adopts the Feistel network.

One round of DES is depicted in Figure 13, the round function acts on a 32-bit branch at a time and is composed of four stages:

- Expansion (EX): The 32-bit half-block is expanded to 48 bits through the expansion permutation by duplicating half of its bits.
- Key mixing: The result is XORed with a round key. Sixteen 48-bit round keys, one per round, are derived from the main key via the key schedule.
- Substitution (SL): The eight 6-bit chunks of the state are non-linearly transformed by eight distinct S-boxes. The output of each S-box is only 4 bits in length.
- Permutation (P): This is a fixed permutation of the output of the substitution layer, which guarantees diffusion.

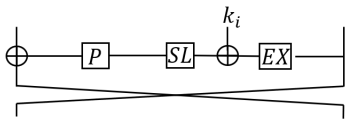

Fig. 13: One round of block cipher DES

The key schedule divides the 56 effective bits of the key into two 28-bit halves. The function responsible for partitioning the bits is named PC1. Each of these two halves is cyclically rotated by a fixed amount in each round. The amount of rotation is either one or two bits depending on the round. The sequence of rotation amounts is irregular. Specifically, in rounds 1, 2, 9, and 16, the rotation amount is one bit, while in all other rounds, it is two bits. From the two rotated 28-bit halves, 48 bits are selected, with 24 bits from each half, by using a fixed function called PC2 to form the round key.

D.3 Specifications of PRESENT-80

PRESENT [36] is a notable lightweight block cipher. It is extremely crucial for resource-constrained devices such as RFID tags and sensor nodes in the Internet of Things (IoT). As of now, it acts as a benchmark for new lightweight ciphers

in terms of security and efficiency. PRESENT-80 is one version of PRESENT. It has a block size of 64 bits, a key size of 80 bits, a S-box size of 4 bits. It is a SPN block cipher which makes use of the operation of bit permutation.

Table 6: The S-box of PRESENT

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S(x)$	12	5	6	11	9	0	10	13	3	14	15	8	4	7	1	2

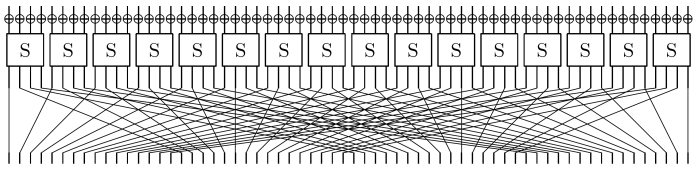

Fig. 14: One round of block cipher PRESENT.

One round of PRESENT is depicted in Figure 14, the round function of it involves an XOR with the round key, the application of a 4-bit S-box (as shown in Table 6) in parallel to the state and a bit permutation.

For the key schedule of PRESENT-80, the master key is stored in a register K and is represented as $\mathrm{k}_{79}\mathrm{k}_{78}\cdots\mathrm{k}_0$. At round i, the round key K_i consists of the 64 leftmost bits of the current content of the register $K_i = \mathrm{k}_{79}\mathrm{k}_{78}\ldots\mathrm{k}_{16}$. Once the round key is extracted, the register K is updated in the following way:

$$[\mathrm{k}_{79}\mathrm{k}_{78}\ldots\mathrm{k}_1\mathrm{k}_0] = [\mathrm{k}_{18}\mathrm{k}_{17}\ldots\mathrm{k}_{20}\mathrm{k}_{19}]$$
$$[\mathrm{k}_{79}\mathrm{k}_{78}\mathrm{k}_{77}\mathrm{k}_{76}] = S\,[\mathrm{k}_{79}\mathrm{k}_{78}\mathrm{k}_{77}\mathrm{k}_{76}]$$
$$[\mathrm{k}_{19}\mathrm{k}_{18}\mathrm{k}_{17}\mathrm{k}_{16}\mathrm{k}_{15}] = [\mathrm{k}_{19}\mathrm{k}_{18}\mathrm{k}_{17}\mathrm{k}_{16}\mathrm{k}_{15}] \oplus \mathrm{round\_counter}.$$

D.4 Specifications of PRINTcipher48

PRINTcipher [37] enjoys a prominent status in the realm of lightweight cryptography. It is elaborately designed for settings with intense resource constraints. To date, it has rendered substantial contributions to the exploration and development of security solutions specifically targeted at low-power devices. PRINTcipher48 is one version of PRINTcipher. It has a block size of 48 bits, a key size of 80 bits, a S-box size of 3 bits. It is a SPN block cipher which makes use of the operation of key-dependent bit permutation.

One round of PRINTcipher48 is shown in Figure 15, the round function of it involves a XOR with the round key, a bit permutation, a XOR with the round

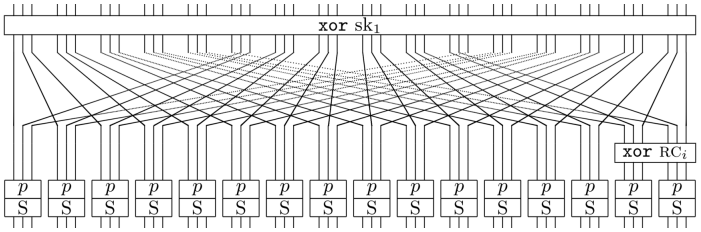

Fig. 15: One round of block cipher PRINTcipher48

constant, the key-dependent bit permutation and the application of a 3-bit S-box in parallel to the state.

The key schedule of PRINTcipher48 is rather simple, it uses the same key for all rounds.

D.5 Specifications of SPECK

SPECK [38] is an important player in the field of lightweight cryptography. It has emerged as a notable algorithm in the family of block ciphers. It has been recognized for its suitability for use in resource-constrained environments, which has given it a distinct place in modern cryptographic research and development. The SPECK is usually denoted as SPECK-n/m where n, m are block size and key size respectively in bits, and SPECK-n if the key length does not need to be specified, where the parameters n and m are shown in Table 7. SPECK is an add-rotate-xor (ARX) cipher with operations modular addition and so on.

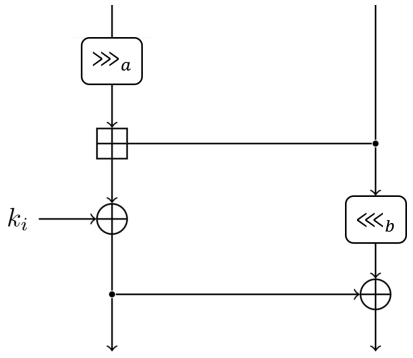

Fig. 16: One round of block cipher SPECK

block size n	key size m
32	64
48	72
	96
64	96
	128
96	96
	144
128	128
	192
	256

Table 7: The parameters n, m of SPECK.

One round of SPECK is shown in Figure 16, the round function of it involves a modular addition $\boxplus$, bitwise-xor $\oplus$, left circular shift $\lll$, and right circular shift $\ggg$, where $(a, b) = (7, 2)$ for SPECK-32 and $(a, b) = (8, 3)$ for other versions.

For the key schedule, the master key k is written as $k = (l_{t-2}, \ldots, l_0, k_0)$, where $t = 2m/n$. The k_i and l_i are defined by

$$l_{i+m-1} = (k_i + (l_i \ggg_a) \oplus i,$$
$$k_{i+1} = k_i \lll_b \oplus l_{i+m-1}.$$

The value k_i is the i-th round key.

D.6 Specifications of GIFT

GIFT [39] has solidly positioned itself as a significant constituent in the realm of lightweight cryptography. With its design focused on efficient resource usage, it stands apart from a plethora of cryptographic algorithms. It has garnered acclaim as a contemporary and superbly crafted block cipher and is being seriously considered for applications where both security and efficiency hold paramount significance. GIFT comes in two versions: GIFT-64 and GIFT-128. GIFT-64 is a 64-bit block cipher with a 128-bit master key. GIFT-128 is a 128-bit block cipher also with a 128-bit master key. For both of these versions, they are SPN block ciphers that utilize the operation of bit permutation.

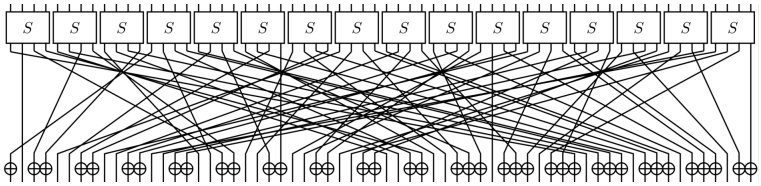

Fig. 17: One round of block cipher GIFT-64

One round of GIFT-64 is shown in Figure 17, the round function of it involves the application of a 4-bit S-box in parallel to the state, a bit permutation and an XOR with the round key. In particular, in i-th round, for the 64-bit state $s_j (0 \leq j \leq 63)$, the 32-bit round key $rk_i = u||v = u_{15} \ldots u_0||v_{15} \ldots v_0$ is xored to the state as $b_{4j+1} \leftarrow b_{4j+1} \oplus u_j, b_{4j} \leftarrow b_{4j} \oplus v_j, (0 \leq j \leq 15)$.

One round of GIFT-128 is similar to GIFT-64, the round function of it involves the application of a 4-bit S-box in parallel to the state, a bit permutation and an XOR with the round key. In particular, in i-th round, for the 128-bit state $s_j (0 \leq j \leq 128)$, the 64-bit round key $rk_i = u||v = u_{31} \ldots u_0||v_{31} \ldots v_0$ is xored to the state as $b_{4j+1} \leftarrow b_{4j+1} \oplus u_j, b_{4j+2} \leftarrow b_{4j+2} \oplus v_j, (0 \leq j \leq 31)$.

For both versions of GIFT, the 128-bit master key k is donated as $k = k_7||k_6||\ldots||k_1||k_0$, the key is updated as follows,

$$k_7 \, \|k_6\| \ldots \|k_1\| \, k_0 \leftarrow k_1 \ggg_2 \|k_0 \ggg_2\| \ldots \|k_3\| \, k_2,$$

where $\ggg_i$ is an i bits right rotation within a 16-bit word. For GIFT-64, the 32-bit round key $rk_i = u||v$ is derived as $u \leftarrow k_1$ and $v \leftarrow k_0$. For GIFT-128, the 64-bit round key $rk_i = u||v$ is derived as $u \leftarrow k_5||k_4$ and $v \leftarrow k_1||k_0$.

D.7 Specifications of CHAM

CHAM [40] is a family of block ciphers that is suitable for devices with limited resources, such as Internet of Things (IoT) devices and embedded systems. Its design has been optimized to have relatively low requirements for computational power, memory, and energy consumption. Each cipher in this family is denoted by CHAM-n/m, where n represents the block size and m represents the key size. Table 8 presents the list of ciphers within the family along with their parameters. Here, w denotes the bit length of a branch, and r represent the new number of rounds, respectively. CHAM adopts the 4-branch generalized Feistel-network with the operation modular addition.

Table 8: List of CHAM ciphers and their parameters.

Cipher	n	m	w	r
CHAM-64/128	64	128	16	88
CHAM-128/128	128	128	32	112
CHAM-128/256	128	256	32	120

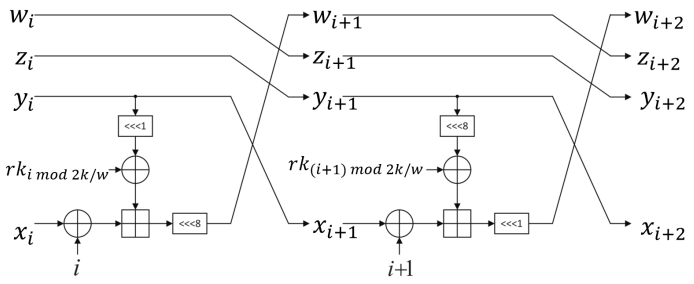

Fig. 18: Two consecutive rounds of block cipher CHAM beginning with the even i-th round

As shown in Figure 18, CHAM-n/k encrypts four w-bit words (x_0, y_0, z_0, w_0) to four w-bit words (x_r, y_r, z_r, w_r). To be more specific, in the i-th round $(0 \leq$

$i < r$)

$$(x_{i+1}, y_{i+1}, z_{i+1}, w_{i+1}) \leftarrow \left(y_i, z_i, w_i, \left((x_i \oplus i) \boxplus \left((y_i \lll \alpha_i) \oplus rk_{i \bmod 2k/w}\right)\right) \lll \beta_i\right),$$

where $\alpha_i = 1$ and $\beta_i = 8$ when $i \bmod 2 = 0$ and $\alpha_i = 8$ and $\beta_i = 1$ when $i \bmod 2 = 1$, and $rk_{i \bmod 2k/w}$ is the round key.

The key schedule of CHAM-n/k takes k/w secret keys $K[0], K[1], \cdots, K[k/w-1]$ and generates $2k/w$ w-bit round keys $rk_0, rk_1, \cdots, rk_{2k/w-1}$. The round keys are generated in the following way:

$$\begin{aligned} rk_i &\longleftarrow K[i] \oplus (K[i] \lll 1) \oplus (K[i] \lll 8), \\ rk_{(i+k/w)\oplus 1} &\longleftarrow K[i] \oplus (K[i] \lll 1) \oplus (K[i] \lll 11), \end{aligned}$$

where $0 \le i < k/w$.

D.8 Specifications of GOST

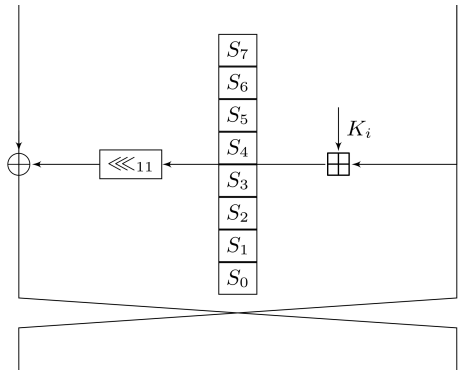

Fig. 19: One round of block cipher GOST

GOST 28147-89 has been used in a variety of applications in Russia and some other regions with historical or technological ties to Russia. It has been implemented in government and military communication systems, as well as in some financial and industrial applications where data security is of great importance. GOST is a block cipher with a 64-bit block size and a 256-bit key size. It consists of 32 Feistel rounds and adopts the operation modular addition. As depicted in Figure 19, the i-th round is defined as follows:

$$F_{K_i}(X_L, X_R) = (X_R, X_L \oplus \lll_{11}(S(X_R \boxplus K_i))),$$

where $\oplus$ denotes bit-wise XOR and $\boxplus$ denotes modular addition modulo 2^{32}, $\lll_{11}(A)$ denotes cyclic left-rotation of A by 11 bits for 32-bit word A, K_i

denotes the round key, and S is an S-box layer of eight 4 bits S-boxes, these S-boxes can be either public or secret and are not necessarily permutations.

In our work, we employ public S-boxes for automatic search. Particularly, we search for IBDs in two of the most renowned versions, namely GOST-FB and GOST-PS. GOST-FB indicates GOST that uses eight different S-boxes as employed by the Central Bank of the Russian Federation (as following S_0-S_7), and GOST-PS represents GOST with only the PRESENT S-box.

$$
\begin{aligned}
S_0 &= \{4, 10, 9, 2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3\} \\
S_1 &= \{14, 11, 4, 12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9\} \\
S_2 &= \{5, 8, 1, 13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11\} \\
S_3 &= \{7, 13, 10, 1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3\} \\
S_4 &= \{6, 12, 7, 1, 5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2\} \\
S_5 &= \{4, 11, 10, 0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14\} \\
S_6 &= \{13, 11, 4, 1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12\} \\
S_7 &= \{1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12\}
\end{aligned}
$$

The key schedule is extremely simple. The 256-bit key is divided into eight 32-bit subkeys $K_0, \ldots, K_7$. These subkeys are employed in this particular order three times during rounds $1 - 24$. In the last 8 rounds $25 - 32$, they are used in the reversed order of $K_7, \ldots, K_0$.

E Example of (RK-)IBDs

Distinguisher 1 (AES). $((\alpha, \alpha'), (\beta, \beta'))$ *is an IBD of* 4 *rounds AES-128 without the last SR and MC layer, where*

$$
\begin{cases}
\alpha = 0xuv000000000000000000000000000000 (0xuv \in \mathbb{F}_2^8/\{0\}), \\
\alpha' = 0xu'v'000000000000000000000000000000 (0xu'v' \in \mathbb{F}_2^8/\{0\}), \\
\beta = 0xpq000000000000000000000000000000 (0xpq \in \mathbb{F}_2^8/\{0\}), \\
\beta' = 0x00000000p'q'00000000000000000000, (0xp'q' \in \mathbb{F}_2^8/\{0\}).
\end{cases}
$$

Distinguisher 2 (DES). $((\alpha, \alpha'), (\beta, \beta'))$ *is an IBD of* 7*-round DES, where*

$$
\begin{cases}
\alpha = 0x4000000000000000, \alpha' = 0x4000000000000000, \\
\beta = 0x0000000040000000, \beta' = 0x0000000010000000.
\end{cases}
$$

Distinguisher 3 (PRESENT). $((\alpha, \alpha'), (\beta, \beta'))$ *is an IBD of* 6*-round PRESENT-80 without the last bit permutation, where*

$$
\begin{cases}
\alpha = 0x0000000000000001, \alpha' = 0x0000000000000001, \\
\beta = 0x0000000000000001, \beta' = 0x0000000000000005.
\end{cases}
$$

Distinguisher 4 (PRINTcipher48). $((\alpha, \alpha'), (\beta, \beta'))$ *is an IBD of* 5*-round PRINTcipher48, where*

$$
\begin{cases}
\alpha = 0x000080000000, \alpha' = 0x400000000000, \\
\beta = 0x000100000000, \beta' = 0x000100000000.
\end{cases}
$$

Distinguisher 5 (AES). $((\alpha,\alpha'),(\beta,\beta'))$ *is an IBD of* 5*-round AES-128 without the last MC layer under the key differences* $(\kappa_0,\kappa_1,\kappa_2,\kappa_3)=(\alpha,\alpha',0,0)$*, where*

$$\begin{cases}\alpha = 0x0000000000000000uv00000000000000(0xuv \in \mathbb{F}_2^8/\{0\}),\\ \alpha' = 0x0000000000000000u'v'00000000000000(0xu'v' \in \mathbb{F}_2^8/\{0\}),\\ \beta = 0x0000pq000000000000000000000000(0xpq \in \mathbb{F}_2^8/\{0\}),\\ \beta' = 0xp'q'000000000000000000000000000000, (0xp'q' \in \mathbb{F}_2^8/\{0\}).\end{cases}$$

Distinguisher 6 (SPECK). $((\alpha,\alpha),(\beta,\beta'))$ *is an IBD of* r*-round SPECK under the key differences* $(\kappa_0,\kappa_1,\kappa_2,\kappa_3)=(\kappa,\kappa,0,0)$*, where* r,κ,α,β *and* β' *are shown in Table 9.*

Distinguisher 7 (DES). $((\alpha,\alpha'),(\beta,\beta'))$ *is an IBD of* 9*-round DES under the key differences* $(\kappa_0,\kappa_1,\kappa_2,\kappa_3)$*, where*

$$\begin{cases}\alpha = 0x0000000002000000, \alpha' = 0x0000000004000000,\\ \beta = 0x0200000000000000, \beta' = 0x0400000000000000,\\ \kappa_0 = 0x02000000000000, \kappa_1 = 0x00010000000000,\\ \kappa_2 = 0x00000800000000, \kappa_0 = 0x02010800000000.\end{cases}$$

Distinguisher 8 (GIFT). $((\alpha,\alpha),(\beta,\beta))$ *is an IBD of* r*-round GIFT under the key differences* $(\kappa_0,\kappa_1,\kappa_2,\kappa_3)=(\eta,\eta,\theta,\theta)$*, where* r,α,β,η *and* θ *are shown in Table 10.*

Distinguisher 9 (CHAM). $((\alpha,\alpha),(\beta,\beta))$ *is an IBD of* r*-round CHAM under the key differences* $(\kappa_0,\kappa_1,\kappa_2,\kappa_3)=(\eta,\eta,\theta,\theta)$*, where* r,α,β,η *and* θ *are shown in Table 11.*

Distinguisher 10 (GOST). $((\alpha,\alpha),(\beta,\beta))$ *is an IBD of full-round GOST under the key differences* $\kappa_{i,j}(0\le i\le 3, 0\le j\le 7)$*, where*

$$\begin{cases}\alpha = 0x8000000000000000, \alpha' = 0x0000000080000000,\\ \beta = 0x0000000080000000, \beta' = 0x8000000000000000,\\ \kappa_{0,0} = 0x00000000, \kappa_{1,0} = 0x80000000, \kappa_{2,0} = 0x00000000, \kappa_{3,0} = 0x80000000,\\ \kappa_{0,1} = 0x80000000, \kappa_{1,1} = 0x00000000, \kappa_{2,1} = 0x80000000, \kappa_{3,1} = 0x00000000,\\ \kappa_{0,2} = 0x00000000, \kappa_{1,2} = 0x80000000, \kappa_{2,2} = 0x00000000, \kappa_{3,2} = 0x80000000,\\ \kappa_{0,3} = 0x80000000, \kappa_{1,3} = 0x00000000, \kappa_{2,3} = 0x80000000, \kappa_{3,3} = 0x00000000,\\ \kappa_{0,4} = 0x00000000, \kappa_{1,4} = 0x80000000, \kappa_{2,4} = 0x00000000, \kappa_{3,4} = 0x80000000,\\ \kappa_{0,5} = 0x80000000, \kappa_{1,5} = 0x00000000, \kappa_{2,5} = 0x80000000, \kappa_{3,5} = 0x00000000,\\ \kappa_{0,6} = 0x00000000, \kappa_{1,6} = 0x80000000, \kappa_{2,6} = 0x00000000, \kappa_{3,6} = 0x80000000,\\ \kappa_{0,7} = 0x80000000, \kappa_{1,7} = 0x00000000, \kappa_{2,7} = 0x40000000, \kappa_{3,7} = 0xc0000000.\end{cases}$$

F Verification of Examples of (RK-)IBDs

For an automated method, the correctness of the results stems from two aspects. Firstly, it is the correctness of the modeling approach. Secondly, it is the accuracy of the code implementation. In our work, we employ the same modeling method

Table 9: The example of RK-IBDs of SPECK in the two related-keys setting.

Block cipher	params	value
SPECK-32/64	r	8
	κ	$0x0040000000000000$
	α	$0x00000000$
	β	$0x80008002$
	β'	$0x80008000$
SPECK-48/72	r	7
	κ	$0x000080000000000000$
	α	$0x000000000000$
	β	$0x800000800004$
	β'	$0x800000800000$
SPECK-48/96	r	8
	κ	$0x000080000000000000000000$
	α	$0x000000000000$
	β	$0x800000800004$
	β'	$0x800000800000$
SPECK-64/96	r	8
	κ	$0x000000800000000000000000$
	α	$0x0000000000000000$
	β	$0x8000000080000004$
	β'	$0x8000000080000000$
SPECK-64/128	r	9
	κ	$0x000000800000000000000000000000000$
	α	$0x0000000000000000$
	β	$0x8000000080000004$
	β'	$0x8000000080000000$
SPECK-64/96	r	8
	κ	$0x000000000800000000000000000000000000$
	α	$0x000000000000000000000000$
	β	$0x800000000000800000000004$
	β'	$0x800000000000800000000000$
SPECK-128/192	r	8
	κ	$0x000000000000008000000000$
		$0x000000000000000000000000$
	α	$0x00000000000000000000000000000000$
	β	$0x80000000000000008000000000000004$
	β'	$0x80000000000000008000000000000000$
SPECK-128/256	r	9
	κ	$0x000000000000008000000000000000$
		$0x000000000000000000000000000000$
	α	$0x00000000000000000000000000000000$
	β	$0x80000000000000008000000000000004$
	β'	$0x80000000000000008000000000000000$

Table 10: The example of RK-IBDs of GIFT in the four related-keys setting.

Block cipher	params	value
GIFT-64	r	13
	α	$0x0001000000000000$
	β	$0x0000000000020000$
	η	$0x00000000000000000000000000001000$
	θ	$0x00000000000000000100000000000000$
GIFT-128	r	10
	α	$0x00000000000000000002000000000004$
	β	$0x00000000000000000400000000000000$
	η	$0x00000000000000010000000000001000$
	θ	$0x00000000000000000000000000008000$

Table 11: The example of RK-IBDs of CHAM in the four related-keys setting.

Block cipher	params	value
CHAM-64/128	r	30
	α	$0x0000000000000000$
	β	$0x0000000000000000$
	η	$0x00006020000000000000000000000000$
	θ	$0x6020c040000000000000000000000000$
CHAM-128/256	r	28
	α	$0x00000000000000000000000000000000$
	β	$0x00000000000000000000000000000000$
	η	$0x78081828000000000000000000000000$
		$0x00000000000000000000000000000000$
	θ	$0x0000000000000000078081828f0103050$
		$0x00000000000000000000000000000000$

and call the same set of code interfaces to automatically search for distinguishers. Therefore, verifying a portion of the distinguishers is sufficient to demonstrate the correctness of our results.

Manual derivation is a common means of verifying the correctness of results obtained through an automated method. However, in some cases, manual derivation is difficult and extremely time-consuming. Under such circumstances, we can make use of computer-aided verification.

In computer-aided verification, the computer can play two roles.

- Detect the location where the contradiction takes place. Take Algorithm 1 as an example, assume $\mathcal{D} = (\alpha, \alpha', \beta, \beta')$ is an IBD or an RK-IBD under key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$, we can implement this feature by simply modifying the Algorithm 1. In Line 6 of Algorithm 1, we implement the function *DiffConnectBD*$(z_{01}, z_{23}, z_{12}, z_{03})$ to set $z_{01} \oplus z_{23} \oplus z_{12} \oplus z_{03} = 0$, i.e. $z_{01,i} \oplus z_{23,i} \oplus z_{12,i} \oplus z_{03,i} = 0 (0 \le i \le n-1)$, where $(v_0, \ldots, v_{n-1})$ is the bit representation of $v \in \mathbb{F}_2^n$ and n represents the block size of a block cipher. To locate where the contradiction occurs, we design a new function *DiffConnectD*$(z_{01}, z_{23}, z_{12}, z_{03}, p)$ $(0 \le p \le n-1)$ to set $z_{01,i} \oplus z_{23,i} \oplus z_{12,i} \oplus z_{03,i} = 0 (0 \le i \le n-1, i \ne p)$. Then, if $\mathcal{D} = (\alpha, \alpha', \beta, \beta')$ is still an (RK-)IBD under the modified algorithm when $p = j$, j is a position unrelated to the contradiction. Using this method, we can find all the positions unrelated to the contradiction and then derive the contradiction from the remaining positions.
- Traverse all possible trails and disprove them. When the computing power of the computer permits, we can propagate the differences or states from the input to the middle round in the forward direction and propagate the differences or states from the output to the middle round in the backward direction. Then, we can use GBCT, GEBCT and so on to disprove all possible trails.

Now we present our verification of some examples of IBDs and RK-IBDs in Appendix E as follows.

Distinguisher 1

Verification (verify by contradiction). Assume (α, α') can propagate to (β, β'), as shown in Figure 20. For $X_0, X_1 = X_0 \oplus \alpha, X_2, X_3 = X_2 \oplus \alpha'$, and $Y_0, Y_1, Y_2 = Y_1 \oplus \beta, Y_3 = Y_0 \oplus \beta'$, let Z_i be the value obtained by encrypting X_i after 2 rounds without the last MC layer, and W_i be the value obtained by decrypting Y_i after 2 rounds. Then $Z_0 \oplus Z_1 = \gamma$, $Z_2 \oplus Z_3 = \gamma'$, $W_1 \oplus W_2 = \delta$, and $W_0 \oplus W_3 = \delta'$. On the one hand, $W_{0,0} \oplus W_{1,0} \oplus W_{2,0} \oplus W_{3,0} = \delta_0 \ne 0$ and $W_{0,1} \oplus W_{1,1} \oplus W_{2,1} \oplus W_{3,1} = 0$, since

$$W_{1,0} \oplus W_{2,0} = \delta_0 \ne 0, W_{0,0} \oplus W_{3,0} = 0,$$
$$W_{1,1} \oplus W_{2,1} = 0, W_{0,1} \oplus W_{3,1} = 0.$$

On the other hand, since

$$\begin{pmatrix} W_{0,0} \oplus W_{1,0} \oplus W_{2,0} \oplus W_{3,0} \\ W_{0,1} \oplus W_{1,1} \oplus W_{2,1} \oplus W_{3,1} \\ W_{0,2} \oplus W_{1,2} \oplus W_{2,2} \oplus W_{3,2} \\ W_{0,3} \oplus W_{1,3} \oplus W_{2,3} \oplus W_{3,3} \end{pmatrix} = M \cdot \begin{pmatrix} Z_{0,0} \oplus Z_{1,0} \oplus Z_{2,0} \oplus Z_{3,0} \\ Z_{0,1} \oplus Z_{1,1} \oplus Z_{2,1} \oplus Z_{3,1} \\ Z_{0,2} \oplus Z_{1,2} \oplus Z_{2,2} \oplus Z_{3,2} \\ Z_{0,3} \oplus Z_{1,3} \oplus Z_{2,3} \oplus Z_{3,3} \end{pmatrix} = M \cdot \begin{pmatrix} \gamma_0 \oplus \gamma_0' \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

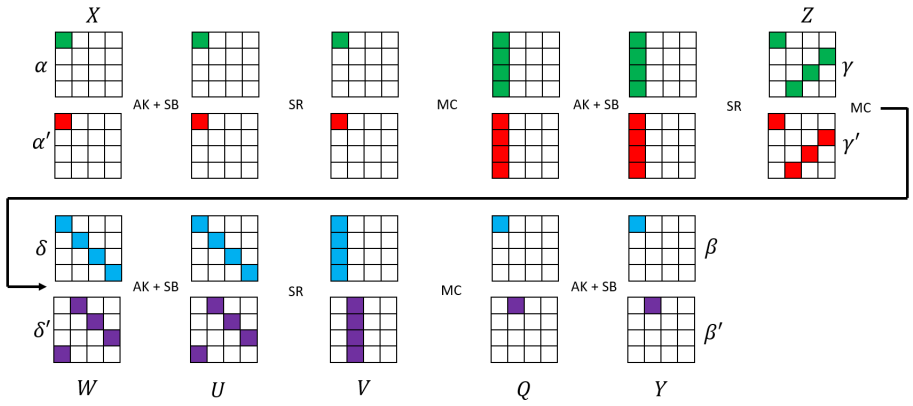

Fig. 20: One of 4-round IBDs of AES

$W_{0,0} \oplus W_{1,0} \oplus W_{2,0} \oplus W_{3,0} = 0$ and $W_{0,1} \oplus W_{1,1} \oplus W_{2,1} \oplus W_{3,1} = 0$, or $W_{0,0} \oplus W_{1,0} \oplus W_{2,0} \oplus W_{3,0} \neq 0$ and $W_{0,1} \oplus W_{1,1} \oplus W_{2,1} \oplus W_{3,1} \neq 0$. Thus there is a contradiction. □

Distinguisher 2

Verification (verify by contradiction). We propagate the input differences 3 rounds in the forward direction, and the output differences 4 round in the backward direction. The middle 5 rounds of the distinguisher are shown in Figure 21, where a and a' are the two differences propagated from the input differences, b and b' are the two differences propagated from the output differences, and (x_0, x_1, x_2, x_3) is the states of the right branch in round 4. Then $x_0 \oplus x_1 = P(a) \oplus 1$, $x_2 \oplus x_3 = P(a') \oplus 1$, and $x_1 \oplus x_2 = P(b) \oplus 1$, $x_0 \oplus x_3 = P(b') \oplus 3$, where P denotes the permutation of DES. Thus $b \oplus b' = a \oplus a' \oplus P^{-1}(3)$. Let $b' = (b'_0, \ldots, b'_7)$ where $(b'_i \in \mathbb{F}_2^4, 0 \leq i \leq 7)$. Then according to the permutation P and the values of a, a' and b, we have $b'_6 = b'_0 = 0$, which implies the bit 0 and bit 3 of the output of the second S-box in round 6 must be zero. However, the input difference of the S-box is $0x20$, the first and fourth bit of the output cannot be zero simultaneously. There is a contradiction.

Distinguisher 3

Our verification makes use of the definition of GEBCT. Thus we demonstrate some basic properties of the S-box of PRESENT in the view of such table. The analysis reveals some new properties of the S-box of PRESENT.

Property 1 (GEBCT). *Let $\mathcal{T}$ and $\mathcal{T}_{inv}$ be the* GEBCT *of S-box and the invertible S-box of PRESENT, then*

$$(1,1,*,*) \xrightarrow{\mathcal{T}} (1,1,0,0),(1,1,1,1),$$
$$(*,*,1,5) \xrightarrow{\mathcal{T}_{inv}} (1,0,1,0),(0,1,1,0) \xrightarrow{\mathcal{T}_{inv}} (1,0,1,0),(0,1,1,0),$$

where $(\mu, \mu', \rho, \rho') \xrightarrow{T} (\theta, \theta', \varphi, \varphi')$ means $T(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') \neq 0$ for $T \in \{\mathcal{T}, \mathcal{T}_{inv}\}$ and '$*$' represents arbitrary 4-bit value.

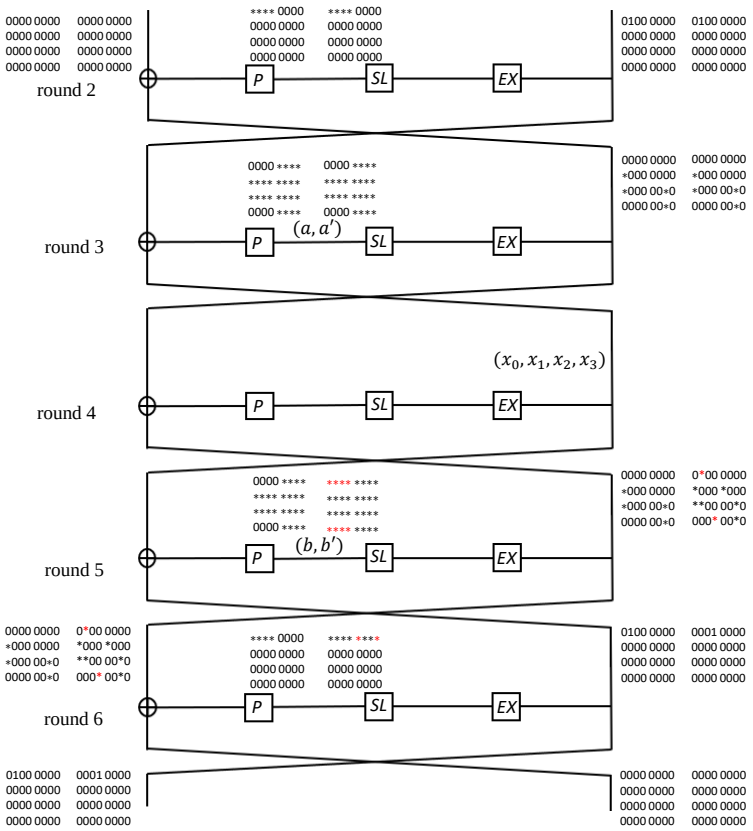

Fig. 21: The core of one 7-round IBDs of DES

Verification (verify by contradiction). As shown in Figure 22, let $(\theta_{i,0}, \theta'_{i,0}, \varphi_{i,0}, \varphi'_{i,0})$ be the bit 0 of the output of the S-boxes layer in the round i $(i = 0, 1, 2)$. According to Property 1, it holds that $A_0 = A_1 = A_2 = \{(1,1,0,0), (1,1,1,1)\}$, and $(\theta_{2,0}, \theta'_{2,0}, \varphi_{2,0}, \varphi'_{2,0}) \in A_2$.

Analogously, let $(\mu_{i,0}, \mu'_{i,0}, \rho_{i,0}, \rho'_{i,0})$ be the bit 0 of the input of the S-boxes layer in the round $i(i = 3, 4, 5)$. According to Property 1, it holds that $B_0 = B_1 = B_2 = \{(1,0,1,0), (0,1,1,0)\}$, and $(\mu_{3,0}, \mu'_{3,0}, \rho_{3,0}, \rho'_{3,0}) \in B_2$.

There is a contradiction, since $(\theta_{2,0}, \theta'_{2,0}, \varphi_{2,0}, \varphi'_{2,0}) = (\mu_{3,0}, \mu'_{3,0}, \rho_{3,0}, \rho'_{3,0})$, $\theta_{2,0} = 1, \theta'_{2,0} = 1$, and one value of $\mu_{3,0}$ and $\mu'_{3,0}$ is 0. □

Distinguisher 4

To conduct this verification, we first give some properties of the S-box from the perspectives of DDT and GBCT.

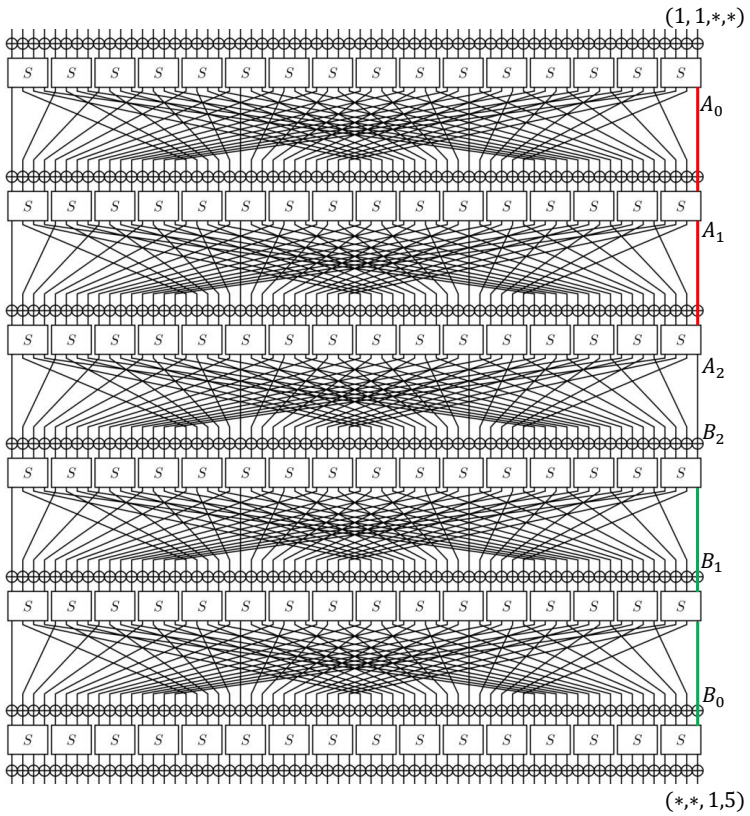

Fig. 22: One of 6-round IBDs of PRESENT-80

Property 2 (DDT). *Let $\mathcal{T}$ and $\mathcal{T}_{inv}$ be the* DDT *of S-box and the invertible S-box of PRINTcipher48, then*

$$001 \xrightarrow{\mathcal{T}} **1, 010 \xrightarrow{\mathcal{T}} *1*, 100 \xrightarrow{\mathcal{T}} 1**,$$
$$001 \xrightarrow{\mathcal{T}_{inv}} **1, 010 \xrightarrow{\mathcal{T}_{inv}} *1*, 100 \xrightarrow{\mathcal{T}_{inv}} 1**,$$

where '$*$' can be 0 *or* 1*, and $abc \xrightarrow{T} a'b'c'$ means $T(abc, a'b'c') \neq 0$ for $T \in \{\mathcal{T}, \mathcal{T}_{inv}\}$ and* 3*-bit values abc and $a'b'c'$.*

Property 3 (GBCT). *Let $\mathcal{T}$ be the* GBCT *of S-box of PRINTcipher48, then*

$$(\alpha, 0) \not\xrightarrow{\mathcal{T}} (\beta, \beta),$$
$$(\gamma, 0)) \not\xrightarrow{\mathcal{T}} (0, 0), (\gamma, \delta)) \not\xrightarrow{\mathcal{T}} (0, 0),$$
$$(1, 2) \xrightarrow{\mathcal{T}} (4, 4), (1, 4) \xrightarrow{\mathcal{T}} (2, 2), (2, 4) \xrightarrow{\mathcal{T}} (1, 1),$$

where the weight of both α *and* β *is* 1, $\gamma \neq 0$, *and* $\delta \neq \gamma$.

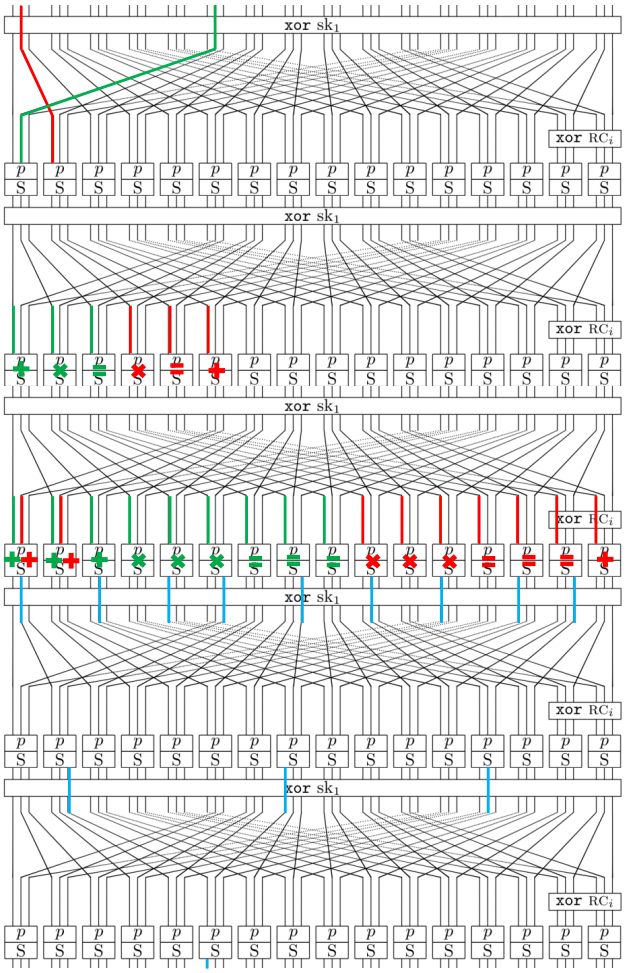

Fig. 23: One of 5-round IBDs of PRINTcipher48

Verification (verify by contradiction). We prove it as shown in Figure 23. In which, blocks of the same color and the same symbol indicate that they are affected by the same S-box. Let $(x_i^0, x_i^1, x_i^2, x_i^3)$ and $(y_i^0, y_i^1, y_i^2, y_i^3)$ be the four states before and after the key-dependent layer at the round i, and $(z_i^0, z_i^1, z_i^2, z_i^3)$ be the four states after the S-box layer at the round i, $x_{i,j}^t$, $y_{i,j}^t$ and $z_{i,j}^t$ be the value of j-th nibble of x_i^t, y_i^t and z_i^t, $x_{i,j,k}^t$, $y_{i,j,k}^t$ and $z_{i,j,k}^t$ be the k-th bit value of j-th nibble of x_i^t, y_i^t and $z_i^t (0 \leq k \leq 2, 0 \leq j \leq 15, 0 \leq t \leq 3)$.

We propagate the input two differences 3 rounds in the forward direction, and the output two differences 2 rounds in the backward direction. According to the Property 3,

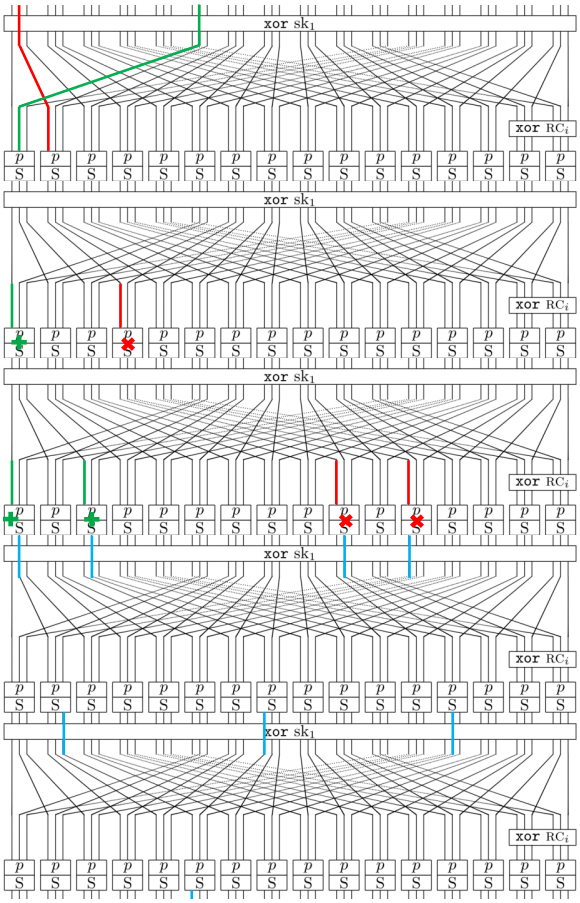

Fig. 24: One impossible propagation trail of 5-round IBD of PRINTcipher48

the input differences and output differences of S-box i ($i \in \{0, 3, 5, 7, 9, 12, 14\}$) in round 3 must be 0. Since $z^1_{5,10,2} \oplus z^2_{5,10,2} = 1$, and $z^0_{5,10,2} \oplus z^3_{5,10,2} = 1$, according the Property 2, $\exists k \in \{0, 1, 2\}$, such that $x^1_{5,10,j} \oplus x^2_{5,10,j} = 1$, and $x^0_{5,10,j} \oplus x^3_{5,10,j} = 1$, Similarly, $\exists (j, k) \in \{(1, 0), (2, 2), (4, 2), (6, 1), (8, 0), (10, 0), (11, 2), (13, 1), (15, 1)\}$, such that $z^1_{3,j,k} \oplus z^2_{3,j,k} = 1$, and $z^0_{3,j,k} \oplus z^3_{3,j,k} = 1$.

Now we continue to discuss in different cases. When the key-dependent permutation 15 transform the bit 1 to bit 2 and the the key-dependent permutation 14 transform the bit 2 to bit 2. As shown in Figure 24, $\exists (j, k) \in \{(4, 2), (6, 1), (13, 1), (15, 1)\}$, such that $z^1_{3,j,k} \oplus z^2_{3,j,k} = 1$, and $z^0_{3,j,k} \oplus z^3_{3,j,k} = 1$. If only $z^1_{3,15,1} \oplus z^2_{3,15,1} = 1$, and $z^0_{3,15,1} \oplus z^3_{3,15,1} = 1$, then the S-box 14 in round 4 is active, and the key-dependent permutation 14 transform the bit 2 to bit 0. This leads to a contradiction. Meanwhile, $x^0_{3,15,2} \oplus x^1_{3,15,2} = 0$, and $x^2_{3,15,2} \oplus x^3_{3,15,2} = 0$, since the key-dependent permutation 15 transform the bit 1 to bit 2 already and cannot transform the bit 2 to bit 2. Thus, $x^0_{3,13,2} \oplus x^1_{3,13,2} = 1$, and $x^2_{3,13,2} \oplus x^3_{3,13,2} = 0$ or $x^0_{3,13,2} \oplus x^1_{3,13,2} = 0$, and $x^2_{3,13,2} \oplus x^3_{3,13,2} =$

1. According to the Property 3, $z^1_{3,13,1} \oplus z^2_{3,13,1} = 1$, and $z^0_{3,13,1} \oplus z^3_{3,13,1} = 1$ cannot hold.

Now, only $\exists(j,k) \in \{(4,2),(6,1)\}$, such that $z^1_{3,j,k} \oplus z^2_{3,j,k} = 1$, and $z^0_{3,j,k} \oplus z^3_{3,j,k} = 1$. Assume $z^1_{3,4,2} \oplus z^2_{3,4,2} = 1$, and $z^0_{3,4,2} \oplus z^3_{3,4,2} = 1$, then

$$\begin{cases} z^1_{3,4,2} \oplus z^2_{3,4,2} = 1, z^0_{3,4,2} \oplus z^3_{3,4,2} = 1, \\ y^0_{3,4,2} \oplus y^1_{3,4,2} = 0, y^2_{3,4,2} \oplus y^3_{3,4,2} = 0, \\ z^1_{3,6,1} \oplus z^2_{3,6,1} = a, z^0_{3,6,1} \oplus z^3_{3,6,1} = a \oplus 1, \\ y^0_{3,6,1} \oplus y^1_{3,6,1} = 0, y^0_{3,6,2} \oplus y^3_{3,6,3} = 1, \\ z^1_{3,13,1} \oplus z^2_{3,13,1} = b, z^0_{3,13,1} \oplus z^3_{3,13,1} = b \oplus 1, \\ y^0_{3,13,1} \oplus y^1_{3,13,1} = 1, y^2_{3,13,1} \oplus y^3_{3,13,1} = 0, \end{cases}$$

must be hold, where a, b can be 0 or 1. Thus,

$$\begin{cases} x^1_{5,10} \oplus x^2_{5,10} = 1||a||b, x^0_{5,10} \oplus x^3_{5,10} = 1||a \oplus 1||b \oplus 1, \\ x^0_{5,10} \oplus x^1_{5,10} = c||0||1, x^2_{5,10} \oplus x^3_{5,10} = c||1||0, \end{cases}$$

where c can be 0 or 1. For any permutation of key-dependent permutation 10 in round 5, this is impossible according to the $GEBCT$ of S-box and $z^1_{5,10,2} \oplus z^2_{5,10,2} = 1$, and $z^0_{5,10,2} \oplus z^3_{5,10,2} = 1$. For other cases, we can verify similarly.

Distinguisher 5

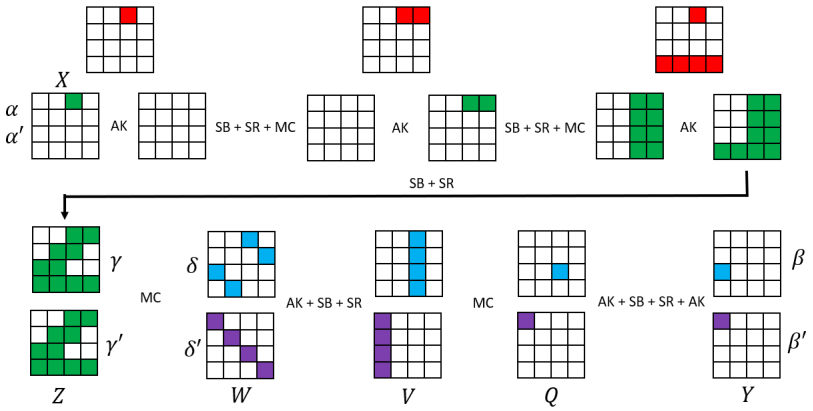

Fig. 25: One of 5-round RT^2_0-IBDs of AES-128

Verification (verify by contradiction). Assume (α, α') can propagate to (β, β'), as shown in Figure 25, for $X_0, X_1 = X_0 \oplus \alpha, X_2, X_3 = X_2 \oplus \alpha'$, and $Y_0, Y_1, Y_2 = Y_1 \oplus \beta, Y_3 = Y_0 \oplus \beta'$, let Z_i be the value obtained by encrypting X_i after 3 rounds without the last MC layer, and W_i be the value obtained by decrypting Y_i after 2 rounds. Then $Z_0 \oplus Z_1 = \gamma$, $Z_2 \oplus Z_3 = \gamma'$, $W_1 \oplus W_2 = \delta$, and $W_0 \oplus W_3 = \delta'$.

On the one hand, since

$$\begin{aligned}
&W_{1,0} \oplus W_{2,0} = 0, W_{0,0} \oplus W_{3,0} = \delta_0' \neq 0,\\
&W_{1,1} \oplus W_{2,1} = 0, W_{0,1} \oplus W_{3,1} = 0,\\
&W_{1,2} \oplus W_{2,2} = \delta_2 \neq 0, W_{0,2} \oplus W_{3,2} = 0,\\
&W_{1,3} \oplus W_{2,3} = 0, W_{0,3} \oplus W_{3,3} = 0.
\end{aligned}$$

we have $W_{0,0} \oplus W_{1,0} \oplus W_{2,0} \oplus W_{3,0} = \delta_0' \neq 0$, $W_{0,1} \oplus W_{1,1} \oplus W_{2,1} \oplus W_{3,1} = 0$, $W_{0,2} \oplus W_{1,2} \oplus W_{2,2} \oplus W_{3,2} = \delta_2 \neq 0$ and $W_{0,3} \oplus W_{1,3} \oplus W_{2,3} \oplus W_{3,3} = 0$.

On the other hand,

$$\begin{pmatrix} W_{0,0} \oplus W_{1,0} \oplus W_{2,0} \oplus W_{3,0} \\ W_{0,1} \oplus W_{1,1} \oplus W_{2,1} \oplus W_{3,1} \\ W_{0,2} \oplus W_{1,2} \oplus W_{2,2} \oplus W_{3,2} \\ W_{0,3} \oplus W_{1,3} \oplus W_{2,3} \oplus W_{3,3} \end{pmatrix} = M \cdot \begin{pmatrix} Z_{0,0} \oplus Z_{1,0} \oplus Z_{2,0} \oplus Z_{3,0} \\ Z_{0,1} \oplus Z_{1,1} \oplus Z_{2,1} \oplus Z_{3,1} \\ Z_{0,2} \oplus Z_{1,2} \oplus Z_{2,2} \oplus Z_{3,2} \\ Z_{0,3} \oplus Z_{1,3} \oplus Z_{2,3} \oplus Z_{3,3} \end{pmatrix} = M \cdot \begin{pmatrix} 0 \\ 0 \\ \gamma_2 \oplus \gamma_2' \\ \gamma_3 \oplus \gamma_3' \end{pmatrix}.$$

If $\gamma_2 \oplus \gamma_2' = \gamma_3 \oplus \gamma_3' = 0$, then $W_{0,i} \oplus W_{1,i} \oplus W_{2,i} \oplus W_{3,i} = 0 (0 \leq i \leq 3)$. There exists a contradiction. If $\gamma_2 \oplus \gamma_2' \neq 0$ or $\gamma_3 \oplus \gamma_3' \neq 0$, then according to the property of MDS matrix, at least three of $i (0 \leq i \leq 3)$ such that $W_{0,i} \oplus W_{1,i} \oplus W_{2,i} \oplus W_{3,i} = 0$. There exists a contradiction. □

Distinguisher 6

We choose the example of RK-IBD of SPECK-32/64 to verify.

Verification (verify by contradiction). We propagate the input differences 5 rounds in the forward direction, and the output differences 3 rounds in the backward direction. The differential propagation of the last 4-round is shown in Figure 26, where the $a_i, a_i', b_i, b_i', c_i, c_i'$ can be 0 or 1 and $b_{11}' = b_8' \oplus 1$.

On the one hand, since $x_{0,9} \oplus x_{1,9} = a_0 = y_{0,9} \oplus y_{1,9}$ and $x_{2,9} \oplus x_{3,9} = a_0' = y_{2,9} \oplus y_{3,9}$, $x_{1,9} \oplus x_{2,9} = 1$, $y_{1,9} \oplus y_{2,9} = 0$, $x_{0,9} \oplus x_{3,9} = c_2'$, $y_{0,9} \oplus y_{3,9} = 1$, we have $c_2' = 0$. Meanwhile, since $x_{0,8} \oplus x_{1,8} = 1$ and $x_{2,8} \oplus x_{3,8} = 1$, $x_{1,8} \oplus x_{2,8} = 0$, we have $x_{0,8} \oplus x_{0,8} = c_1' = 0$. Similarly, it holds that $c_0' = 0, b_{11}' = 1, b_7' = b_6' = 0$.

On the other hand, for the second modular addition in the right of the Figure 26, the least 3 bits of input of the modular addition is $(000, 100)$, thus $b_6' = 1$. This is a contradiction.

Distinguisher 8

We choose the example of RK-IBD of GIFT-64 to verify.

Verification (verify by contradiction). Let (γ, γ') be the difference that propagates (α, α') 4-round in the forward direction, and (δ, δ') be the difference that propagates (β, β') 4-round in the backward direction. Then, according the key schedule, (α, α') propagates to $(\gamma, \gamma') = (0, 0)$ under key differences (κ_0, κ_1) with probability 1, and (β, β') propagates to $(\delta, \delta') = (0, 0)$ under key differences (κ_2, κ_3) with a probability of 1. Now, we show that (γ, γ') cannot propagate to the output differences (δ, δ') under the key differences $(\kappa_0, \kappa_1, \kappa_2, \kappa_3)$ after 5 rounds of GIFT-64.

Let $(x_i^0, x_i^1, x_i^2, x_i^3)$ and $(y_i^0, y_i^1, y_i^2, y_i^3)$ be the four states before and after the S-box layer at the round i. Let $x_{i,j}^t$ and $y_{i,j}^t$ be the j-th nibble value of x_i^t and $y_i^t (0 \leq j \leq 15, 0 \leq t \leq 3)$. First, we remove the constraints nibble by nibble in round 6 in our SAT model to detect the necessary nibble for generating contradictions. The contradiction

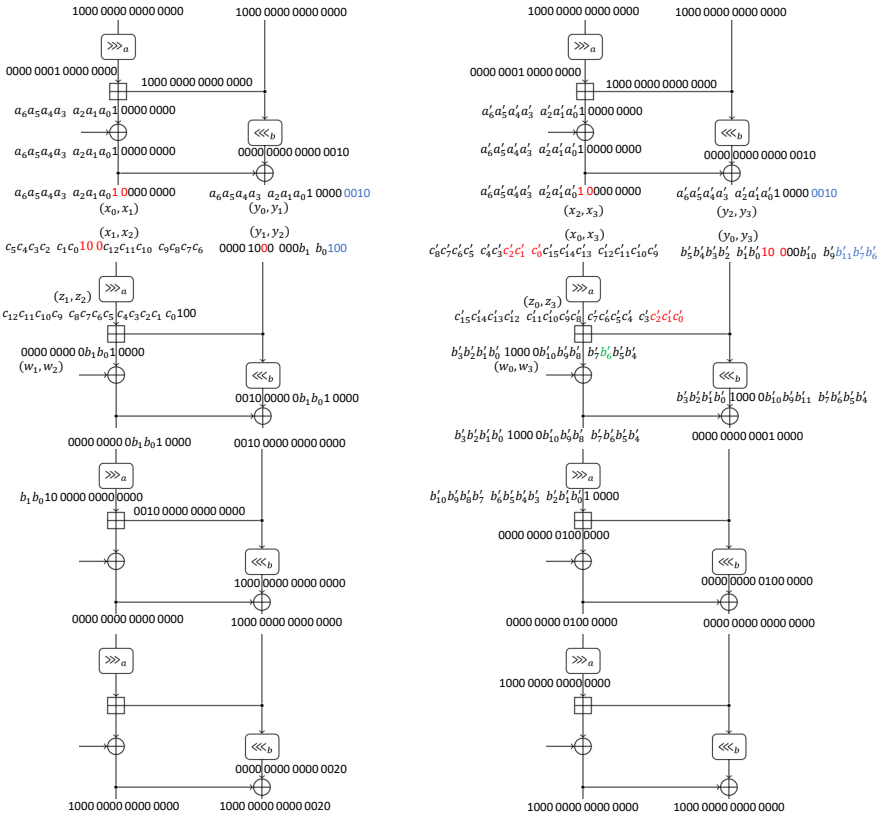

Fig. 26: The core of one 8-round RT_3^2-IBDs of SPECK-32/64

occurs in the nibble i ($i \in \{2, 3, 7, 15\}$). Then, we propagate the input two differences 2 rounds in the forward direction, and propagate the output two differences 2 rounds in the backward direction, and whether those differences can be connected according to the GBCT of the S-box, i.e. we check whether $(x_{6,j}^0 \oplus x_{6,j}^1, x_{6,j}^2 \oplus x_{6,j}^3) \xrightarrow{GBCT} (y_{6,j}^1 \oplus y_{6,j}^2, y_{6,j}^0 \oplus y_{6,j}^3)$ ($j \in \{2, 3, 7, 15\}$). Through a simple Python program, we can remove most of the differential propagations.

The remaining differential propagation is $((y_4^0 \oplus y_4^1, y_4^2 \oplus y_4^3), (x_8^1 \oplus x_8^2, x_8^0 \oplus x_8^3)$, where $y_{4,0}^0 \oplus y_{4,0}^1 \in \{8, 9, 10, 11\}$, $y_{4,0}^2 \oplus y_{4,0}^3 \in \{8, 9, 10, 11\}$, and $x_{8,8}^1 \oplus x_{8,8}^2 \in \{5, 6, 7, 12, 13, 15\}$, $x_{8,8}^0 \oplus x_{8,8}^3 \in \{5, 6, 7, 12, 13, 15\}$. Thus, the bit 3 of $y_4^0 \oplus y_4^1$ and $y_4^2 \oplus y_4^3$ must be 1, and the bit 34 of $x_8^1 \oplus x_8^2$ and $x_8^0 \oplus x_8^3$ must be 1. That is, the differential propagation in the Figure 27 must be hold, where κ_i^t denotes the key difference of κ_t in round i. For the S-box 3 of round 6, if $(x_{6,3}^0 \oplus x_{6,3}^1, x_{6,3}^2 \oplus x_{6,3}^3) = (1, 1) \xrightarrow{GBCT} (y_{6,3}^1 \oplus y_{6,3}^2, y_{6,3}^0 \oplus y_{6,3}^3) = (8, 8)$, then $x_{6,3}^1 \oplus x_{6,3}^2, x_{6,3}^0 \oplus x_{6,3}^3) = (1, 1)$. Thus, for the S-box 12 of round 5, it must

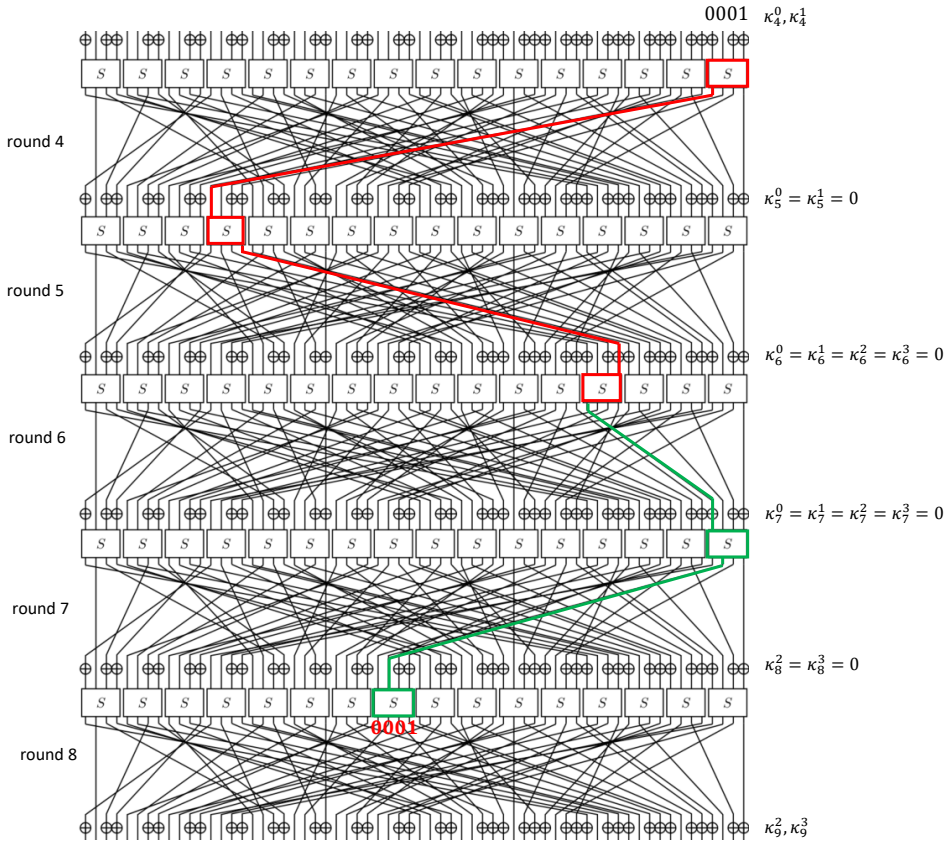

Fig. 27: The core of one 13-round RT_3^2-IBDs of GIFT-64

hold as follows:

$$\begin{cases} (x_{5,12}^0 \oplus x_{5,12}^1, x_{5,12}^2 \oplus x_{5,12}^3) = (8,8) \xrightarrow{GBCT} (y_{5,12}^1 \oplus y_{5,12}^2, y_{5,12}^0 \oplus y_{5,12}^3) = (1,1), \\ y_{5,12}^0 \oplus y_{5,12}^1 = 1, \\ y_{5,12}^2 \oplus y_{5,12}^3 = 1. \end{cases}$$

However, the above formula does not hold. Thus, for the remaining differential propagation, it still cannot hold. In conclusion, we have verified our distinguisher.

Distinguisher 10

Verification (verify by contradiction). As shown in Figure 28, Figure 29, Figure 30 and Figure 31. The input differences (α, α') can propagate to $((0x00000000, 0x80000000), (0x80000000, 0x00000000))$ with a probability of 1 after 23 rounds in the forward direction, and the output differences (β, β') can propagate to $((0x80000000, 0x00000000), (0x00000000, 0x80000000))$ with a probability of 1 after 7 rounds in the backward direction. Thus, we

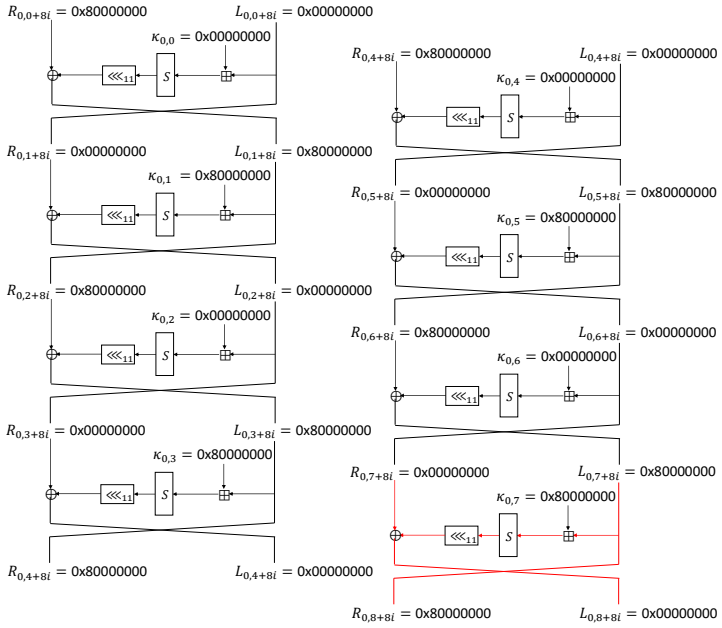

Fig. 28: The 24-round related-key differential of GOST by iterating above 3 times

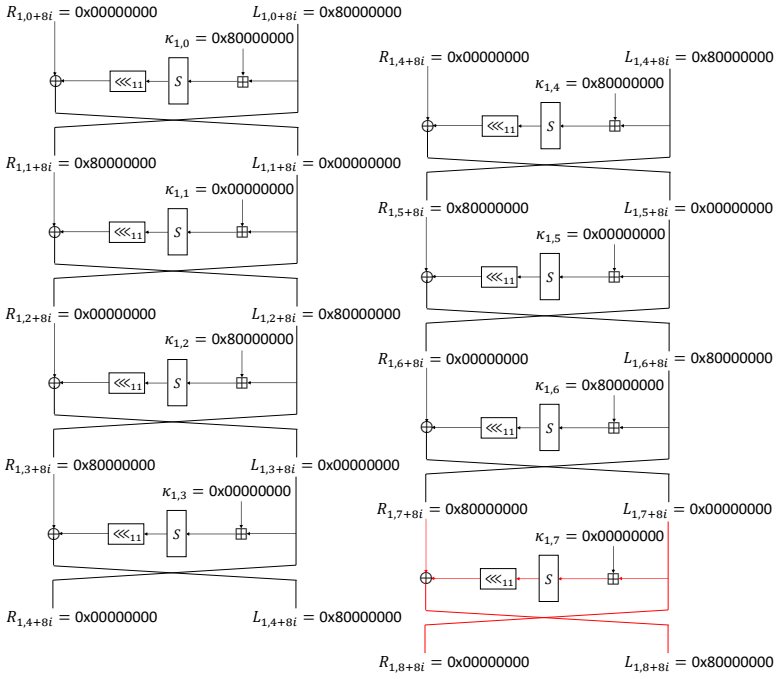

Fig. 29: The 24-round related-key differential of GOST by iterating above 3 times

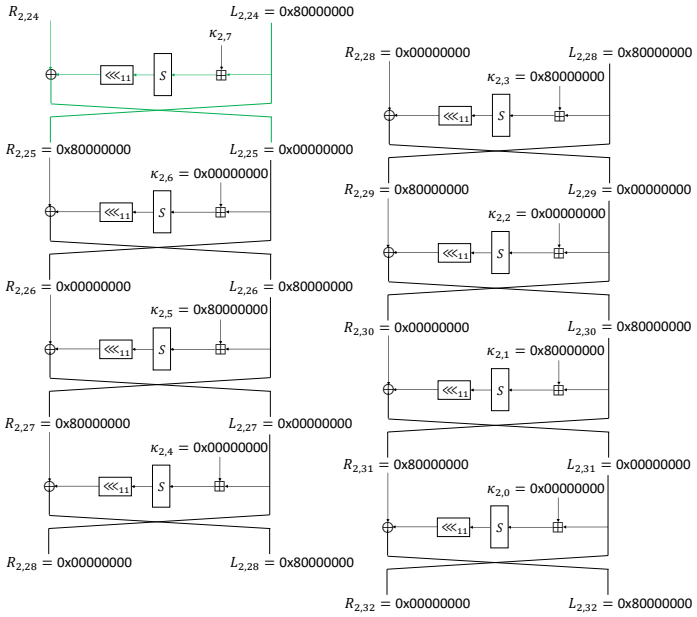

Fig. 30: The 7-round related-key differential of GOST

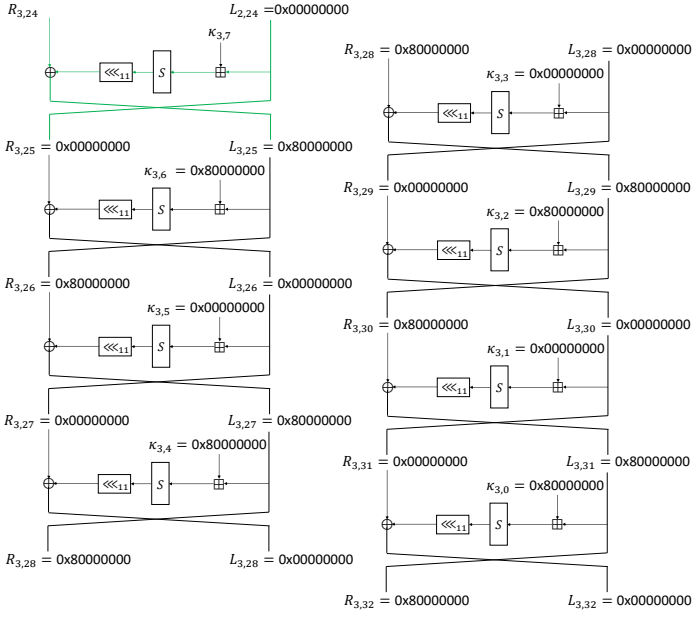

Fig. 31: The 7-round related-key differential of GOST

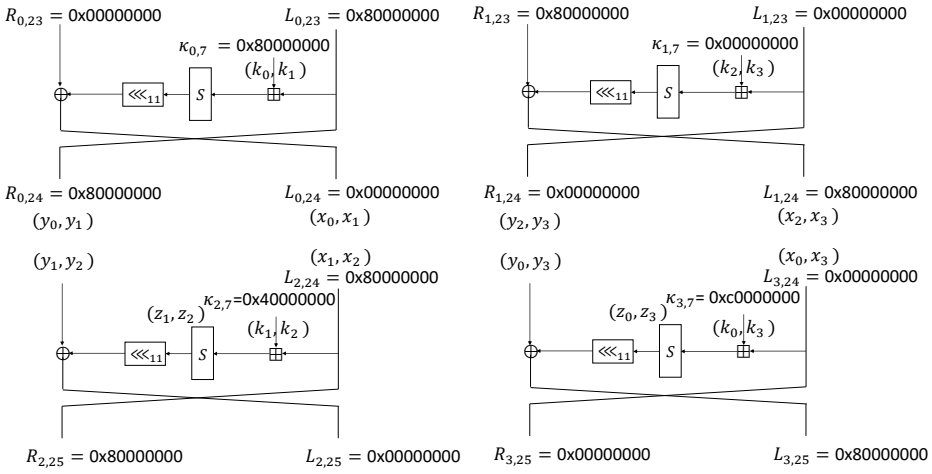

Fig. 32: The core of one full-round RK-IBDs of GOST

just need to verify $((0x00000000, 0x80000000), (0x80000000, 0x00000000)$ cannot propagate to $((0x800000\ 00, 0x00000000), (0x00000000, 0x80000000)$ after 2 rounds under the key differences $\kappa_{0,7}, \kappa_{1,7}, \kappa_{2,7}, \kappa_{3,7}$.

As shown in Figure 32, we have $x_1 = x_0, x_2 = x_0 \oplus 0x80000000$ and $x_3 = x_0$. On the one hand, since $y_0 \oplus y_1 \oplus y_2 \oplus y_3 = 0x80000000$, with the values of $L_{2,25}$ and $L_{3,25}$, we have $z_0 \oplus z_1 \oplus z_2 \oplus z_3 = 0$. On the other hand, we have $k_{1,7} = k_{0,7} \oplus 0x80000000$, $k_{2,7} = k_{3,7} = k_{0,7} \oplus 0xc0000000$. For an 32-bit value v, let v' be the most significant 4-bit, and S represents the S-box operate the most significant 4-bit. Then, we have

$$\begin{cases} z'_0 = S(x'_0 + k'_0), z'_1 = S(x'_0 + k'_0 \oplus 0x8), \\ z'_2 = S(x'_0 \oplus 0x8 + k'_0 \oplus 0xc), z'_3 = S(x'_0 + (k'_0 \oplus 0xc). \end{cases}$$

Then, $z'_0 \oplus z'_1 \oplus z'_2 \oplus z'_3 = 0$ cannot hold for S-box of both GOST-FB and GOST-PS. This is a contradiction.