IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2025, No. 1, pp. 444–470. DOI:10.46586/tosc.v2025.i1.444-470

Improved Search of Boomerang Distinguishers for Generalized Feistel and Application to WARP

Xinhao Zeng, Lin Tan and Hong Xu

Information Engineering University, Zhengzhou, China
feizao51666@163.com, tanlin100@163.com, xuhong0504@163.com

Abstract. Boomerang and rectangle cryptanalysis are powerful cryptanalytic techniques for security evaluation of block ciphers. Automated search for boomerang distinguishers is an important area of research. In FSE 2023, Hadipour et al. proposed a MILP model of searching boomerang distinguishers for Feistel structure, and applied their model to obtain the best known boomerang distinguishers to date for many generalized Feistel ciphers including WARP. In this paper, we focus on improving Hadipour et al.'s model for generalized Feistel structure and boomerang distinguishers on WARP. We show that a boomerang distinguisher with more active S-boxes may have a higher probability. It is caused by the semi-active S-boxes active only in one of the upper and lower differential trails, which are not considered in Hadipour et al.'s model. We classify the active S-boxes in the middle part E_m in more detail, according to the associated tables of DDT, DDT^2, FBCT and its variants in the computation formula of boomerang probability for E_m. Then, we propose an improved MILP model to search boomerang distinguishers for generalized Feistel structure. Applying our model to WARP, we find better boomerang distinguishers for all rounds than those found by Hadipour et al.'s model. For 15-round boomerang distinguisher on WARP, the probability is improved by a factor of $2^{5.78}$. For the longest 23-round boomerang distinguisher, the probability is improved by a factor of $2^{4.23}$, resulting in the best result presented on WARP so far. Exploiting the properties of two local structures and the probabilistic extension technique, we improve the 26-round rectangle attack and give the first 27-round rectangle attack on WARP, which extends the best previous rectangle attack by one round. Note that our findings do not threaten the security of WARP which iterates 41 rounds.

Keywords: Generalized Feistel · Boomerang · Rectangle attack · WARP · MILP

1 Introduction

Differential cryptanalysis proposed by Biham and Shamir [BS91] is one of the most powerful cryptanalytic approaches for assessing the security of block ciphers. It exploits some high-probability differentials to distinguish or recover the key of target ciphers. In many cases, it may be hard to find a long differential with high probability. In FSE 1999, Wagner [Wag99] proposed the boomerang attack, which connects two shorter differentials with high probability to get a longer distinguisher. Boomerang attack works in the adaptively chosen plaintext and ciphertext setting. To remove the requirement of the decryption oracle, Kelsey et al. [KKS00] proposed the amplified boomerang attack. Later, this attack was refined by Biham et al. [BDK01] and called the rectangle attack. The dependency between the two shorter differentials has a significant impact on the actual probability of the resulting boomerang distinguisher. Murphy [Mur11] provided examples where the two differential trails are incompatible. Biryukov and Khovratovich [BK09] showed that the probability of the boomerang distinguisher can be higher considering the

Received: 2024-09-01 Revised: 2024-11-22 Accepted: 2025-01-23 Published: 2025-03-07

boomerang switch. Later, Dunkelman et al. [DKS14] proposed the sandwich distinguisher to deal with the dependency between the upper and lower differentials. To formulate the probability of the middle part of a sandwich distinguisher, many tools such as the BCT [CHP+18] for SPN ciphers and the FBCT [BHL+20] for Feistel ciphers were proposed.

Automated search of boomerang distinguishers for block ciphers is an important area of research, and has seen significant advances over the past few years. There are mainly two types of Mixed-Integer Linear Programming (MILP) models for searching boomerang distinguishers. In [HBS21], Hadipour et al. proposed a MILP model to search boomerang distinguishers for SPN ciphers, taking the boomerang switching effect into account for multiple rounds. Delaune et al. [DDV20] proposed another MILP model of searching for boomerang distinguishers for SPN ciphers, which represents each S-box by 6 binary variables and handles the computation of probability for the middle part automatically. In FSE 2023, these two models were both extended to search boomerang distinguishers for Feistel structures and applied to the lightweight block cipher `WARP` [BBI+20]. Lallemand et al. [LMR22] adapted Delaune et al.'s model to search boomerang distinguishers for Feistel ciphers and gave the first 23-round boomerang distinguisher for `WARP`. Hadipour et al. [HNE22] threw the model of [HBS21] for SPN into Feistel structures, and provided an easy-to-use automatic tool to search sandwich distinguishers. They applied their tool to obtain the best known boomerang distinguishers for many generalized Feistel ciphers.

`WARP` is a 128-bit block cipher proposed by Banik et al. [BBI+20] as a lightweight alternative to `AES`. It is based on a generalized Feistel structure and provides 128-bit security in the single-key setting while achieving a small footprint. The designers of `WARP` provided the security analysis against differential, linear, impossible differential, integral, meet-in-the middle and invariant subspace attacks. The longest distinguisher they mentioned is a 21-round impossible differential distinguisher. Teh and Biryukov [TB22] gave a 23-round differential attack and a 24-round rectangle attack on `WARP` in the single-key setting, and also gave a 41-round differential attack in the related-key setting. Lallemand et al. [LMR22] gave the first 26-round rectangle attack on `WARP`. Hadipour et al. [HNE22] presented the best known boomerang distinguishers up to 23 rounds of `WARP`. Based on the monomial prediction technique, Hadipour et al. [HE22] gave integral attacks on up to 32 rounds of `WARP`. Sun et al. [SWW22] proposed a 33-round zero-correlation attack and improved the 41-round related-key differential attack. Shi et al. [SLLM24] proposed an impossible differential attack on 33-round `WARP`. Hadipour et al. [HDE24] gave the first differential-linear distinguishers up to 22 rounds of `WARP` via a boomerang perspective.

Our Contributions. In Hadipour et al.'s MILP model of searching boomerang distinguishers [HNE22], the objective function of searching truncated differential trails is substantially to minimize the total number of active S-boxes in E_0 and E_1, and the common active S-boxes in E_m. We show that Hadipour et al.'s model is not optimal when applied to the generalized Feistel structure. Taking `WARP` as an example, we present a 14-round boomerang distinguisher with more common active S-boxes in E_m, which has a higher probability than the 14-round boomerang distinguisher in [HNE22]. By analysis, we find that it is caused by the semi-active S-boxes in E_m, which are active only in one of the upper and lower differential trails. The semi-active S-boxes are not considered into the objective function in Hadipour et al.'s model. In the theoretical computation formula of the boomerang probability r for E_m, the semi-active S-boxes may lead to more DDTs involved. We classify the active S-boxes in E_m in more detail, according to the associated tables of DDT, DDT^2, FBCT and its variants in the computation of r, and propose an improved MILP model to search boomerang distinguishers for generalized Feistel structure. Applying our model to `WARP`, we find new 14 to 23 rounds boomerang distinguishers. Compared with all boomerang distinguishers on round-reduced `WARP` in [HNE22], our distinguishers have the higher probabilities. For 15-round boomerang distinguisher, we improve the probability by a factor of $2^{5.78}$. For the longest 23-round boomerang distinguisher, the

probability is improved by a factor of $2^{4.23}$. We present the best boomerang distinguishers on WARP so far, which are shown in Table 1. We also apply our model to TWINE [SMMK12] and Lblock-s [WZ11], and find better 14-round and 15-round boomerang distinguishers of these two ciphers. Based on our 21-round boomerang distinguisher for WARP, exploiting the properties of two local structures and the probabilistic extension technique proposed by Song et al. [SYC+24], we improve the rectangle attack on 26-round WARP and give the first rectangle attack on 27-round WARP, which improves the best previous rectangle attack on this cipher by one round. A summary of existing rectangle attacks on WARP is shown in Table 2. Our source codes for finding boomerang distinguishers on WARP are publicly available in the following Github repository: https://github.com/feizao51/GFS-model.

Comparisons with the Previous Models. Hadipour et al.'s model [HNE22] partitions the cipher into three parts and searches good truncated sandwich distinguishers by minimizing the number of active S-boxes. It encodes independently the propagation of truncated differentials in E_0 and E_1, keeping the propagation with probability 1 in E_m forward and backward. Delaune et al.'s model [DDV20] searches good truncated boomerang characteristics for the whole cipher without partition. It employs the framework of an MILP model to search for truncated differential characteristics. Delaune et al.'s model considers fully the impacts of different S-boxes on the probability of boomerang characteristics, by encoding each S-box with six variables related to DDT and BCT variants. But the time cost of the model [DDV20] is exponential in the number of rounds. It seems difficult to say which model is better. For the cipher WARP, the results in [HNE22] are better than those in [LMR22] which are obtained by adapting Delaune et al.'s model. Our model is based on Hadipour et al.'s model, and refine the modeling for the middle part E_m by the technique of [DDV20]. We consider the different impacts on the boomerang probability for the different types of active S-boxes in E_m and encode them by the table-related variables as [DDV20]. For the parts of E_0 and E_1, our model is the same as [HNE22], only counting the number of active S-boxes. In [DDV20], they need to generate a formula to compute the boomerang probability and use a CP model to consider the differential cluster, while we instantiate the differentials in E_0 and E_1 independently, and compute the probability of E_m experimentally. So, our model saves the execution time and considers the differential cluster more easily. Compared with the model of [HNE22], we classify the active S-boxes in E_m in more detail according to the associated tables of DDT, DDT^2, FBCT and its variants in the computation of connection probability. Then the MILP modeling for E_m is refined, and the objective function is improved with the weighted sum of different types of S-box variables. It provides the possibility of finding better boomerang distinguishers, which is demonstrated in round-reduced WARP as shown in Table 1.

Organization of the Paper. In Section 2, we recall boomerang and rectangle attacks, BCT and its variants, and the specification of WARP. In Section 3, we recall Hadipour et al.'s model for searching boomerang distinguishers, and show that the model is not optimal when applied to generalized Feistel structure. In Section 4, we propose an improved model to search boomerang distinguishers for generalized Feistel ciphers. In Section 5, we apply our model to WARP, TWINE and Lblock-s. In Section 6, we present an improved 26-round rectangle attack and the first 27-round rectangle attack on WARP. We conclude this paper in Section 7.

2 Preliminaries

2.1 Boomerang and Rectangle Attacks

The boomerang attack [Wag99] allows the adversary to connect two shorter differential paths to get a longer distinguisher. The target cipher E is split into two parts $E = E_1 \circ E_0$.

Table 1: Summary of boomerang distinguishers for WARP.

Rounds	Probability	Reference	Probability	Reference	Probability	Reference
14	$2^{-19.11}$	Sect. 5.1	$2^{-20.58}$	[HNE22]		
15	$2^{-22.80}$	Sect. 5.1	$2^{-28.58}$	[HNE22]		
16	$2^{-30.80}$	Sect. 5.1	$2^{-34.50}$	[HNE22]		
17	$2^{-38.80}$	Sect. 5.1				
18	$2^{-46.81}$	Sect. 5.1				
19	$2^{-58.81}$	Sect. 5.1				
20	$2^{-70.81}$	Sect. 5.1	$2^{-75.96}$	[HNE22]		
21	$2^{-79.36}$	Sect. 5.1	$2^{-84.55}$	[HNE22]	$2^{-121.11}$	[TB22]
22	$2^{-91.36}$	Sect. 5.1	$2^{-96.55}$	[HNE22]	2^{-108}	[LMR22]
23	$2^{-111.36}$	Sect. 5.1	$2^{-115.59}$	[HNE22]	2^{-124}	[LMR22]

Table 2: Summary of existing rectangle attacks on WARP.

Rounds	Time	Data	Memory	Reference
24	$2^{125.18}$	$2^{126.06}$	$2^{127.06}$	[TB22]
26	$2^{115.9}$	$2^{120.6}$	$2^{120.6}$	[LMR22]
26	$2^{111.5}$	$2^{106.18}$	$2^{106.18}$	Sect. 6.2
27	$2^{122.63}$	$2^{116.18}$	$2^{116.18}$	Sect. 6.3

For E_0, there is a differential $\Delta_1 \xrightarrow{E_0} \Delta_2$ with probability p, called the upper differential. For E_1, there is a differential $\nabla_2 \xrightarrow{E_1} \nabla_3$ with probability q, called the lower differential. Suppose the two differentials are independent, then we can combine them to form a boomerang distinguisher on E as shown in Figure 1. The probability of the boomerang distinguisher is estimated by

$$Pr(E^{-1}(E(x) \oplus \nabla_3) \oplus E^{-1}(E(x \oplus \Delta_1) \oplus \nabla_3) = \Delta_1) = p^2 q^2.$$

The process of distinguishing the cipher E from a random permutation by the boomerang distinguisher is as follows:

1. Randomly choose a plaintext pair (P_1, P_2) satisfying $P_1 \oplus P_2 = \Delta_1$, and query the encryption oracle to obtain the corresponding ciphertext pair (C_1, C_2).
2. Compute (C_3, C_4) by $C_3 = C_1 \oplus \nabla_3$, $C_4 = C_2 \oplus \nabla_3$, and query the decryption oracle to obtain the corresponding plaintext pair (P_3, P_4).
3. Check whether $P_3 \oplus P_4 = \Delta_1$ or not. If yes, a right quartet (P_1, P_2, P_3, P_4) is obtained, otherwise go back to Step 1.

For the target cipher E, the probability of obtaining a right quartet is p^2q^2, while it is 2^{-n} for a random permutation, where n is the block size of the cipher. In order to distinguish the cipher E from a random permutation, we need $p^2q^2 > 2^{-n}$, and the number of quartets is $\mathcal{O}(p^{-2}q^{-2})$. To formulate the dependency between the upper and lower differential trails, Dunkelman et al. [DKS14] proposed the sandwich distinguisher. The cipher E is split into three parts $E = E_1 \circ E_m \circ E_0$ as shown in Figure 1. The middle part E_m includes the dependency between the upper and lower differential trails, and E_0, E_1 are referred to as the outer parts of the sandwich distinguisher. The probability of the sandwich distinguisher is estimated by p^2q^2r, where r is the connection probability for E_m defined by

$$r = Pr(E_m^{-1}(E_m(x) \oplus \nabla_2) \oplus E_m^{-1}(E_m(x \oplus \Delta_2) \oplus \nabla_2) = \Delta_2).$$

A boomerang distinguisher can be converted into a rectangle distinguisher [BDK01] in the chosen plaintext setting. The distinguishing process is as follows:

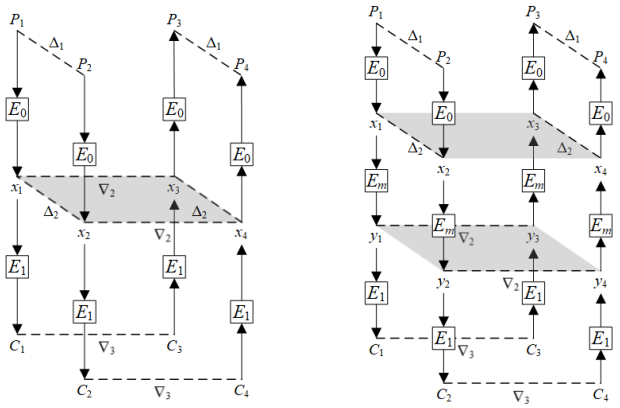

Figure 1: The boomerang distinguisher (left) and the sandwich distinguisher (right).

1. Randomly choose two plaintext pairs (P_1, P_2) and (P_3, P_4) satisfying $P_1 \oplus P_2 = P_3 \oplus P_4 = \Delta_1$, and query the encryption oracle to obtain the corresponding ciphertexts C_1, C_2, C_3, and C_4.

2. Check whether $C_1 \oplus C_3 = C_2 \oplus C_4 = \nabla_3$ or not. If yes, a right quartet (P_1, P_2, P_3, P_4) is obtained, otherwise go back to Step 1.

For the target cipher E, the probability that a right quartet is obtained is $2^{-n}p^2q^2r$, while it is 2^{-2n} for a random permutation. So it is needed that $p^2q^2r > 2^{-n}$, and the number of quartets we need is $\mathcal{O}(2^n p^{-2} q^{-2} r^{-1})$. According to [Sel08], the success probability of a rectangle attack with a b-bit advantage is evaluated by

$$P_s = \Phi\left(\frac{\sqrt{sS_N} - \Phi^{-1}(1 - 2^{-b})}{\sqrt{S_N + 1}}\right),$$

where $S_N = p^2q^2r/2^{-n}$ is the signal-to-noise ratio, $\Phi(\cdot)$ is the standard normal distribution function and s is the expected number of right quartets.

2.2 Boomerang Connectivity Table and Its Variants

Researchers have studied how to compute the probability r of the middle part E_m. First, Cid et al. [CHP+18] introduced the tool of boomerang connectivity table (BCT) to deal with only one S-box layer in E_m for SPN block ciphers. Wang et al. [WP19] extended the BCT and proposed the tool of boomerang difference table (BDT) for E_m with multiple rounds. Later, other variants of BCT are also proposed including the UBCT (upper BCT), LBCT (lower BCT) and EBCT (extended BCT) in [DDV20].

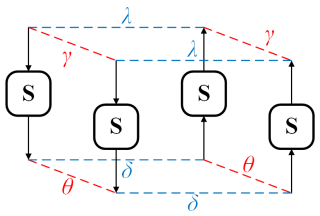

Figure 2: The parameters of BCT.

Definition 1. (BCT [CHP+18]). Let S be a permutation of $\mathbb{F}_2^n$, The BCT of S is a two-dimensional table defined by

$$\mathrm{BCT}(\gamma,\delta) = \#\{x \in \mathbb{F}_2^n | S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma\},$$

where $\gamma, \delta \in \mathbb{F}_2^n$.

Definition 2. (UBCT, LBCT and EBCT [DDV20]). Let S be a permutation of $\mathbb{F}_2^n$, the UBCT, LBCT and EBCT of S are defined respectively as

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

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

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

where $\gamma, \theta, \delta, \lambda \in \mathbb{F}_2^n$.

In [BHL+20], Boukerrou et al. extended the BCT framework for SPN block ciphers to propose the Feistel boomerang connectivity table (FBCT) for Feistel structure. To formulate the boomerang switch over multiple rounds, the UFBCT (upper FBCT), LFBCT (lower FBCT) and FBET for Feistel structure are defined as follows, which are analogous to the UBCT, LBCT and EBCT for SPN structure, respectively.

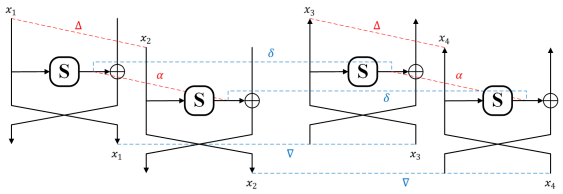

Figure 3: The parameters of FBCT.

Definition 3. (FBCT [BHL+20]). Let S be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$, the FBCT of S is a two-dimensional table defined by

$$\mathrm{FBCT}(\Delta,\nabla) = \#\{x \in \mathbb{F}_2^n | S(x) \oplus S(x \oplus \Delta) \oplus S(x \oplus \nabla) \oplus S(x \oplus \Delta \oplus \nabla) = 0\},$$

where $\Delta, \nabla \in \mathbb{F}_2^n$.

Definition 4. (UFBCT, LFBCT and FBET [BHL+20]). Let S be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$, the UFBCT, LFBCT and FBET of S are defined respectively as

$$\mathrm{UFBCT}(\Delta,\nabla,\alpha) = \#\left\{ x \in \mathbb{F}_2^n \middle| \begin{array}{l} S(x) \oplus S(x \oplus \Delta) \oplus S(x \oplus \nabla) \oplus S(x \oplus \Delta \oplus \nabla) = 0, \\ S(x) \oplus S(x \oplus \Delta) = \alpha \end{array} \right\},$$

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

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

where $\Delta, \nabla \in \mathbb{F}_2^n$ and $\alpha, \delta \in \mathbb{F}_2^m$.

Proposition 1. *For a function S from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$, we have*

$$\sum_{\nabla\in\mathbb{F}_2^n} \mathrm{UFBCT}(\Delta,\nabla,\alpha) = \mathrm{DDT}^2(\Delta,\alpha)$$
$$\sum_{\Delta\in\mathbb{F}_2^n} \mathrm{LFBCT}(\Delta,\nabla,\delta) = \mathrm{DDT}^2(\nabla,\delta)$$

where DDT *is the differential distribution table of S.*

Proof. We prove the first equation as follows, and the other equation can be proved similarly.

$$\begin{aligned}
&\sum_{\nabla\in\mathbb{F}_2^n} \mathrm{UFBCT}(\Delta,\nabla,\alpha)\\
&= \sum_{\nabla\in\mathbb{F}_2^n} \#\left\{ x\in\mathbb{F}_2^n \,\middle|\, \begin{array}{l} S(x)\oplus S(x\oplus\Delta)\oplus S(x\oplus\nabla)\oplus S(x\oplus\Delta\oplus\nabla)=0,\\ S(x)\oplus S(x\oplus\Delta)=\alpha \end{array} \right\}\\
&= \sum_{\nabla\in\mathbb{F}_2^n} \#\left\{ x\in\mathbb{F}_2^n \,\middle|\, \begin{array}{l} \alpha\oplus S(x\oplus\nabla)\oplus S(x\oplus\Delta\oplus\nabla)=0,\\ S(x)\oplus S(x\oplus\Delta)=\alpha \end{array} \right\}\\
&= \sum_{\nabla\in\mathbb{F}_2^n} \#\left\{ (x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^n \,\middle|\, \begin{array}{l} S(x)\oplus S(x\oplus\Delta)=\alpha,\\ S(y)\oplus S(y\oplus\Delta)=\alpha,\\ y=x\oplus\nabla \end{array} \right\}\\
&= \#\left\{ (x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^n \,\middle|\, \begin{array}{l} S(x)\oplus S(x\oplus\Delta)=\alpha,\\ S(y)\oplus S(y\oplus\Delta)=\alpha \end{array} \right\}\\
&= \mathrm{DDT}^2(\Delta,\alpha).
\end{aligned}$$

□

2.3 Specification of WARP

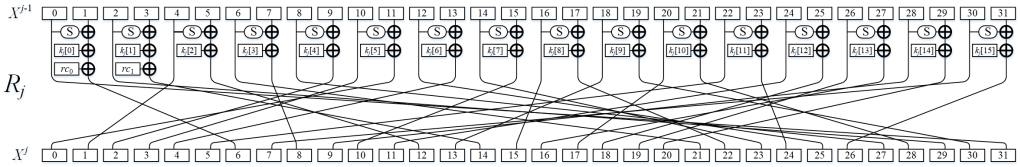

Figure 4: The round function of WARP.

WARP [BBI+20] is a lightweight block cipher with 128-bit block and 128-bit key. Employing a 32-branch generalized Feistel structure, WARP aims at providing 128-bit security in the single-key setting while achieving a small footprint. It performs 40 full rounds as represented in Figure 4 plus one partial round (without nibble permutation) to produce a 128-bit ciphertext. The input state of WARP in the j-th round can be represented as $X^{j-1} = X_0^{j-1}||\cdots||X_{31}^{j-1}$, where X_i^{j-1} are 4-bit nibbles, $0\le i\le 31$, $1\le j\le 41$. The 128-bit master key K is split into two 64-bit halves, $K = k_0||k_1$, and each half is used alternatively as the round key starting with k_0. The round function of WARP applies the same 4-bit S-box to each nibble with an even index X_{2i}^{j-1} and XORs the result to X_{2i+1}^{j-1} followed by a round key addition. Two constants are added to X_1^{j-1} and X_3^{j-1}.

Afterwards, a permutation π is applied to the nibbles of the state. We refer to the design paper [BBI+20] for a full specification. Split k_i into 16 nibbles $k_i = k_i[0]||\cdots||k_i[15]$, $i = 1, 2$. Denote by S_i^j the i-th S-box in the j-th round, $0 \le i \le 15$, $1 \le j \le 41$. Denote by ΔX^j and ∇X^j the differences at X^j in the upper and lower trails, and denote by ΔX_i^j and ∇X_i^j the differences at X_i^j in the upper and lower trails respectively, $0 \le i \le 31$, $1 \le j \le 41$.

Table 3: S-box S of WARP.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	c	a	d	3	e	b	f	7	8	9	1	5	0	2	4	6

Table 4: Nibble permutation π of WARP.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(x)$	31	6	29	14	1	12	21	8	27	2	3	0	25	4	23	10
x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\pi(x)$	15	22	13	30	17	28	5	24	11	18	19	16	9	20	7	26

3 Hadipour et al.'s Boomerang Search Model for Feistel Structure

3.1 Hadipour et al.'s Model from SPN to Feistel

In [HBS21], Hadipour et al. proposed a MILP model to search boomerang distinguishers for SPN ciphers. Throwing the model of [HBS21] into Feistel structure, Hadipour et al. [HNE22] proposed a modified model and applied it to many generalized Feistel ciphers including WARP. They gave the best known boomerang distinguisher for 23-round WARP with probability $2^{-115.59}$. The model in [HNE22] consists of the following four steps.

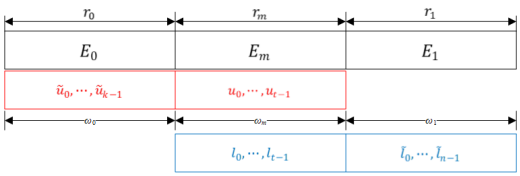

Figure 5: Framework of the boomerang search model in [HNE22].

1. Partition the target cipher E into three parts: E_0, E_m and E_1, containing r_0, r_m and r_1 rounds, respectively. Generate two MILP models with independent variables to encode the propagation of truncated upper and lower differential trails through $r_0 + r_m$ and $r_m + r_1$ rounds, respectively. For the upper trail, encode the propagation of the truncated differential in a standard way over E_0, and the propagation forward with probability one in E_m. Similarly, encode the propagation of the truncated differential in a standard way over E_1, and the propagation backward with probability one in E_m. Define binary variables to indicate whether the S-boxes are active in the upper and lower truncated trails. Denote by $\tilde{u}_0, \ldots, \tilde{u}_{k-1}$ and $\tilde{l}_0, \ldots, \tilde{l}_{n-1}$ the activity of S-boxes in E_0 and E_1, respectively. Denote by $u_0, \ldots, u_{t-1}$ and $l_0, \ldots, l_{t-1}$ the activity of S-boxes over E_m in the upper and lower trails, respectively. Denote $s_0, \ldots, s_{t-1}$ to indicate whether the S-boxes in E_m are both active in the upper and

lower trails. Clearly, $s_i = 1$ if and only if $u_i = l_i = 1$, $0 \le i \le t-1$. Let the constants ω_0, ω_m and ω_1 be the costs of the active S-boxes in E_0, E_m and E_1, respectively. Then construct the MILP model to search truncated boomerang trails with the objective function:

$$\min \sum_{i=0}^{k-1} \omega_0 \cdot \tilde{u}_i + \sum_{i=0}^{t-1} \omega_m \cdot s_i + \sum_{i=0}^{n-1} \omega_1 \cdot \tilde{l}_i.$$

2. Instantiate the discovered truncated differential trails in Step 1. Look for the best concrete differential characteristics in E_0 and E_1 using a bit-wise MILP model. If there is no differential characteristic satisfying the derived truncated trails, go back to Step 1 and try again for another truncated trail. After deriving the concrete differential characteristics, compute the cluster effect of the differential characteristics with the same input and output differences for E_0 and E_1. That is, compute the probabilities $p = Pr(\Delta_1 \xrightarrow{E_0} \Delta_2)$ and $q = Pr(\nabla_2 \xrightarrow{E_1} \nabla_3)$.

3. After instantiating the differentials in E_0 and E_1, compute experimentally the boomerang probability r for E_m:

$$r = Pr\left(E_m^{-1}(E_m(x) \oplus \nabla_2) \oplus E_m^{-1}(E_m(x \oplus \Delta_2) \oplus \nabla_2) = \Delta_2\right).$$

   The amount of experimental data can be calculated based on the theoretical estimation of r by the tool of FBCTs. If $r = 0$, it means the upper and lower differential trails are incompatible. If so, go back to Step 1 and repeat the process.

4. Compute the entire probability p^2q^2r of the discovered boomerang distinguisher and check it. For obtaining a more accurate estimate of the boomerang probability for the cipher E, the boundaries of the middle part E_m may need to be adjusted and repeat the process above.

3.2 Non-Optimality of Hadipour et al.'s Model for Generalized Feistel

In Hadipour et al.'s model, the objective function of searching truncated boomerang trails is substantially to minimize the total number of active S-boxes in E_0 and E_1, and the common active S-boxes in E_m. We will show that it is not optimal when applied to the generalized Feistel structure, taking 14-round WARP as an example. There are two boomerang distinguishers on 14-round WARP shown in Appendix A, where Distinguisher I in Figure 12 was given by Hadipour et al.'s model in [HNE22] and Distinguisher II in Figure 13 is found by our model. Both of the two distinguishers have the same number of rounds for E_0, E_1 and E_m, i.e., $r_0 = r_1 = 2$ and $r_m = 10$, and have the same probabilities $p = 2^{-4}$ for E_0 and $q = 2^{-4}$ for E_1. The input and output differences of E and E_m are shown in Table 5. For the middle part E_m, Distinguisher I has 3 common active S-boxes, while Distinguisher II has 4 common active S-boxes. The experimental probability of r for E_m of Distinguisher I is $r_{\mathrm{I}} = 2^{-4.58}$ given in [HNE22]. The experimental probability of r for E_m of Distinguisher II is $r_{\mathrm{II}} = 2^{-3.11}$, which is computed with an experimental data of 2^{16}. This example shows that a distinguisher with more active S-boxes may have a higher probability.

To analyze the reason, we present the theoretical computation formulas of the probability r over E_m for the two distinguishers as follows, using the tool of tables in Section 2.2. Here $A = a =$ 0xa are known, and B, C, D, E, b, c, d, e are the intermediate variables.

$$\begin{aligned} r_{\mathrm{I}} = 2^{-40} \sum_{B,C,D,E,b,c,d,e} & \mathrm{FBCT}(A,e)\mathrm{DDT}(a,e)\mathrm{FBCT}(C,d)\mathrm{DDT}(A,B)\mathrm{DDT}(B,C) \\ & \mathrm{DDT}(b,c)\mathrm{DDT}(c,d)\mathrm{LFBCT}(E,a,b)\mathrm{DDT}(B,D)\mathrm{DDT}(D,E), \end{aligned}$$

Table 5: Specification of two boomerang distinguishers on 14-round WARP.

Distinguisher I [HNE22]	
ΔX^0	ΔX^2
0x0000000000000a00000000a500000000	0x0a000000000000000000000000000000
∇X^{12}	∇X^{14}
0x000a000000000000000000000000000	0x0000f0a0000000000000000f000000
Distinguisher II	
ΔX^0	ΔX^2
0x000a000000000000ff00000000000000	0x00000000000000000000000a00000000
∇X^{12}	∇X^{14}
0x00a000000000000000000000000000000	0x000000000070000000000a00500000000

$$r_{\mathrm{II}} = 2^{-24} \sum_{B,b} \mathrm{FBCT}^3(B,a)\mathrm{DDT}(A,B)\mathrm{DDT}^2(a,b).$$

In Distinguisher I, for S_3^3 the input difference in the upper trail is A, and the input difference e in the lower trail is propagated from the difference a at X_4^{12} through S_{10}^5, so it is computed by $\mathrm{FBCT}(A,e)\mathrm{DDT}(a,e)$. For S_{14}^8, the input difference C in the upper trail is propagated from A at X_1^2 through S_{14}^6 and S_{10}^7, and in the lower trail the input difference d is propagated from b at X_{19}^{11} which is the output of common active S-box S_9^{11} passing S_{13}^{10} and S_4^9, so it is computed by $\mathrm{FBCT}(C,d)\mathrm{DDT}(A,B)\mathrm{DDT}(B,C)\mathrm{DDT}(b,c)\mathrm{DDT}(c,d)$. For S_9^{11}, the input difference E in the upper trail is propagated from B at X_{20}^6 which is the output of S_{14}^6 through S_{11}^9 and S_{12}^{10}, and the input difference of common active S-box S_{14}^8 in the lower trail comes from the output difference of S_9^{11}, so it is computed by $\mathrm{LFBCT}(E,a,b)\mathrm{DDT}(B,D)\mathrm{DDT}(D,E)$. In Distinguisher II, the input differences of S_9^5, S_2^7 and S_8^9 in the upper trail are all B propagated from A at X_{22}^2 through S_{12}^4, and their output differences in the lower trail are all a coming from X_2^{13}, so for the three active S-boxes it can be computed by $\mathrm{FBCT}^3(B,a)\mathrm{DDT}(A,B)$. For the fourth common active S-box S_{14}^{11}, the input difference is free in the upper trail and non-zero in the lower trail, so it is computed by $\mathrm{DDT}^2(a,b)$ from Proposition 1.

From the two formulas above, we can see that the former involves more DDTs than the latter when summing up the probabilities over all boomerang characteristics. It is caused by the S-boxes active only in one of the upper and lower differential trails, which are not considered by Hadipour et al.'s model.

4 Improved Model of Searching Boomerang Distinguishers for Generalized Feistel

In this section, we classify the active S-boxes in the middle part E_m according to the tables used in computation of the probability r, including DDT, DDT^2, FBCT and its variants, and propose an improved MILP model to search boomerang distinguishers for generalized Feistel structure. For the target cipher, we suppose that round keys are independent and uniformly random, and the upper and lower differentials are independent.

1. **Full-active S-box**: An S-box in E_m which has non-zero differences both in the upper and lower differential trails. It is different from the common active S-box in Hadipour et al.'s model, which includes the case of zero difference caused by XORing two equal non-zero differences. In computation of the probability r, for full-active S-boxes, the tables of FBCT, UFBCT, LFBCT or FBET are needed.

2. **Semi-active S-box**: An S-box in E_m which has non-zero difference only in one of the upper and lower differential trails. A semi-active S-box is called by **constrained**

S-box, if its output difference affects the input differences of other active S-boxes in E_m. For example, in Figure 6, suppose the differences γ_i and δ_i are non-zero, $i = 1, 2, 3$. Then S_0 and S_2 are semi-active S-boxes, and S_1 is a full-active S-box. In the upper trail, the output difference γ_2 of S_0 equals the input difference of S_1, so S_0 is a constrained S-box. In the lower trail, the output difference δ_2 of S_2 equals the input difference of S_1, so S_2 is also a constrained S-box. When computing the probability of a certain boomerang trail, one of the FBCT, UFBCT, LFBCT and FBET for full-active S-boxes and the corresponding DDT of the related constrained S-boxes are needed.

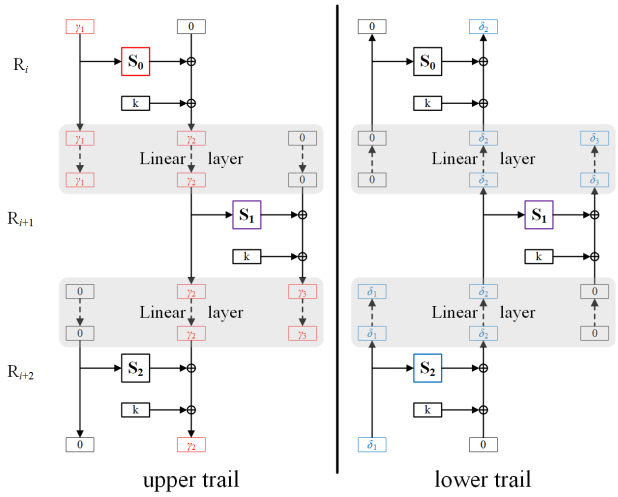

Figure 6: An example of constrained S-box.

3. **Free S-box**: An S-box in E_m, of which the input difference can take all possible values including zero. If an S-box has non-zero input difference in the upper trail while the input difference is free in the lower trail, then we say the S-box is **l-free**. If an S-box has non-zero input difference in the lower trail while the input difference is free in the upper trail, then we say the S-box is **u-free**. The free S-boxes appear after XORing two active cells, where the differences may be offset. For example, in Figure 7, suppose the differences γ_i and δ_i are non-zero, $i = 1, 2$, then the S-box S_0 is l-free. Because S_0 has non-zero input difference in the upper trail, while the input difference of S_0 is free in the lower trail. For a free S-box, suppose that the input differences take all possible values uniformly. By Proposition 1, the expected boomerang probability of an l-free S-box is

$$\frac{1}{2^n}\sum_{\nabla}\frac{1}{2^n}\mathrm{UFBCT}(\Delta,\nabla,\alpha)=\frac{1}{2^{2n}}\mathrm{DDT}^2(\Delta,\alpha),$$

and the expected boomerang probability of a u-free S-box is $\frac{1}{2^{2n}}\mathrm{DDT}^2(\nabla,\delta)$. Thus, for u-free or l-free S-boxes, DDT^2 will be used in computation of the probability r. We note that free S-boxes usually appear in the last few rounds in the upper trail forward or lower trail backward. So, for E_m covering many rounds, there are rarely S-boxes that are free in both trails. To simplify the modeling, we ignore this kind of S-boxes.

MILP Modeling. For the parts of E_0 and E_1, the modeling is same as Hadipour et al.'s model [HNE22]. For the middle part E_m of r_m rounds, denote by S_i^j the i-th S-box in the j-th round, $1 \le j \le r_m$, $0 \le i < n_s$, where n_s is the number of S-boxes in each round.

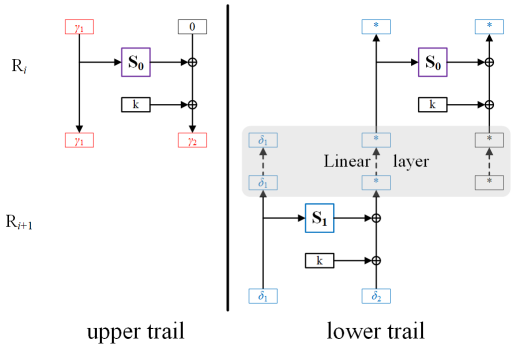

Figure 7: An example of l-free S-box.

Denote binary variables s_i^j to indicate whether S_i^j is common active in the upper and lower differential trails. Denote binary variables $isfreeu_i^j$ and $isfreel_i^j$ to indicate whether S_i^j is u-free and l-free, respectively. A free S-box is common active, so $isfreeu_i^j \leq s_i^j$, $isfreel_i^j \leq s_i^j$, and $isfreeu_i^j + isfreel_i^j \leq 1$. For the start of the upper and lower trails, set $isfreeu_i^1 = 0$ and $isfreel_i^{r_m} = 0$. Denote binary variables $isconu_i^j$ and $isconl_i^j$ to indicate whether S_i^j is a constrained S-box in the upper and lower trails, respectively. A constrained S-box is a semi-active S-box, so $isconu_i^j \leq 1 - s_i^j$ and $isconl_i^j \leq 1 - s_i^j$. For S-boxes S_i^j, we also define the following binary variables corresponding to the tables used in computation of the probability r.

1. If S_i^j is a full-active S-box, then $s_i^j = 1$, $isfreeu_i^j = 0$ and $isfreel_i^j = 0$. In the computation formula of r, the tables of FBCT, UFBCT, LFBCT or FBET may be used. For simplicity, we define the binary variables $isFBCT_i^j$ to indicate whether one of the four tables is used for S_i^j. Then we have the following constraint conditions:

$$\begin{cases} isFBCT_i^j \leq s_i^j, \\ isFBCT_i^j \geq s_i^j - isfreeu_i^j - isfreel_i^j, \\ isFBCT_i^j \leq 1 - isfreeu_i^j - isfreel_i^j. \end{cases}$$

2. If S_i^j is a free S-box, i.e., u-free or l-free, then the DDT^2 will be used in the computation formula of r. We define the binary variables $isDDT2_i^j$ to indicate whether the DDT^2 is used for S_i^j. Then we have the following constraint conditions:

$$\begin{cases} isDDT2_i^j \geq isfreeu_i^j, \\ isDDT2_i^j \geq isfreel_i^j, \\ isDDT2_i^j \leq isfreeu_i^j + isfreel_i^j. \end{cases}$$

3. If S_i^j is a constrained S-box, then its output difference will be treated as an intermediate variable which affects the input differences of the other active S-boxes. In the computation formula of r, the corresponding DDT will be involved. We define the binary variables $isDDT_i^j$ to indicate whether the DDT is used for S_i^j. Then we have $isDDT_i^j = isconu_i^j + isconl_i^j$.

Let the constants ω_{DDT}, ω_{FBCT} and ω_{DDT2} be the costs of three kinds of tables, respectively. Suppose the number of S-boxes involved in E_0 and E_1 are n_0 and n_1, respectively. Denote by $\tilde{u}_0, \ldots, \tilde{u}_{n_0-1}$ and $\tilde{l}_0, \ldots, \tilde{l}_{n_1-1}$ the activity of S-boxes in E_0 and

E_1, and let the constants ω_0 and ω_1 be the costs of the active S-boxes in E_0 and E_1, respectively. Then, the objective function to search truncated differential trails is

$$\min \left[\begin{array}{l} \sum_{i=0}^{n_0-1} \omega_0 \cdot \tilde{u}_i + \sum_{i=0}^{n_1-1} \omega_1 \cdot \tilde{l}_i + \\ \sum_{j=1}^{r_m} \sum_{i=0}^{n_s-1} \left(\omega_{DDT} \cdot isDDT_i^j + \omega_{FBCT} \cdot isFBCT_i^j + \omega_{DDT2} \cdot isDDT2_i^j \right) \end{array} \right].$$

After obtaining the truncated differential trails, instantiate the differential trails over E_0 and E_1, and compute experimentally the probability r for E_m. The rest steps are the same as Hadipour et al.'s model.

Rationale of Our Model. First, our model is also an easy-to-use tool and has a low time cost as the model of [HNE22]. Except for the different modeling for E_m, the other parts are the same as Hadipour et al.'s model. For E_0 and E_1, we encode independently the propagation of truncated differentials and just count the number of active S-boxes in the upper and lower differential trails. The changes in the modeling and objective function have a limited impact on the time cost. Second, due to the refined modeling for E_m, better truncated boomerang trails may be found by our model. It has been shown by the example of two boomerang distinguishers on 14-round WARP in Section 3.2 that the number of common active S-boxes in E_m is not the only factor affecting the connection probability r. For the middle part E_m with the same number of rounds, Distinguisher I has 3 common active S-boxes and 7 constrained S-boxes, while Distinguisher II found by our model has 4 common active S-boxes and 1 constrained S-box. For the constrained S-boxes, their output differences affect the input differences of the other active S-boxes, which are treated as the intermediate variables in the theoretical computation formula of r. Then the corresponding DDTs will be involved and may lead to a lower probability. So, there are more DDTs involved in the formula of r for Distinguisher I when computing the probabilities of specific boomerang characteristics. In other words, when the number of constrained S-boxes increases, their impact on r exceeds the impact of common active S-boxes. To refine the modeling for E_m, we classify the S-boxes in E_m into full-active S-boxes, constrained S-boxes and free S-boxes, and encode them by new binary variables according to the associated tables of DDT, DDT^2, FBCT and its variants in the computation of r. These variables are considered into the objective function of MILP, so our model may return better truncated boomerang trails for the target cipher.

5 Applications

5.1 Application to WARP

Applying our model to WARP, we find new boomerang distinguishers for 14 to 23 rounds, and the full specification is listed in Table 11 in Appendix A. Our program codes are written in Python, executed with Gurobi as the solver, and the CPU is Intel(R)Core(TM)i5-1135@2.40GHz. The running times range from 10 to 205 seconds for 14 to 23 rounds. In our model, the weights $w_0, w_1, w_{DDT}, w_{FBCT}$ and w_{DDT2} are listed in Table 6, which follow that $w_{DDT} < w_{FBCT} < w_{DDT2}$ and $w_{DDT2} < w_0, w_{DDT2} < w_1$, according to the impact of these tables on the probability. We note that these weights can be adjusted manually. We first set $w_0 = w_1 = 6, w_{DDT} = 1, w_{FBCT} = 1.5$ and $w_{DDT2} = 3$ by experiments, referring to the public codes of [HNE22]. When there are several optimal solutions for the same value of objective function, our model randomly outputs one of them. We adjust the values of w_{DDT2} to 2, 3 and 4, instantiate the output truncated trails and select the

one with the highest probability. The connection probability r for the middle part E_m is estimated by experiments with data far greater than r^{-1}. Compared with all boomerang distinguishers on round-reduced WARP given in [HNE22], our distinguishers have the higher probabilities, which are presented in Table 7. Particularly, we improve the probability of boomerang distinguisher on 15-round WARP by a factor of $2^{5.78}$. The probability of our 23-round distinguisher is $2^{-111.36}$, which is the best result presented so far for boomerang distinguishers on WARP. It demonstrates the advantage of our model.

Table 6: Weights of variables in our model for WARP.

Rounds	w_0	w_1	w_{DDT}	w_{FBCT}	w_{DDT2}
14, 15	6	6	1	1.5	2
16, 17, ..., 21	6	6	1	1.5	3
22, 23	6	6	1	1.5	4

Table 7: Comparison with all boomerang distinguishers on WARP in [HNE22].

Rounds	r_0	r_m	r_1	p	q	r	Probability	Reference
14	2	10	2	2^{-4}	2^{-4}	$2^{-3.11}$	$2^{-19.11}$	Ours
	2	10	2	2^{-4}	2^{-4}	$2^{-4.58}$	$2^{-20.58}$	[HNE22]
15	2	10	3	2^{-4}	2^{-4}	$2^{-8.80}$	$2^{-22.80}$	Ours
	2	10	3	2^{-4}	2^{-8}	$2^{-4.58}$	$2^{-28.58}$	[HNE22]
16	3	10	3	2^{-8}	2^{-4}	$2^{-8.80}$	$2^{-30.80}$	Ours
	3	10	3	2^{-8}	2^{-4}	$2^{-10.50}$	$2^{-34.50}$	[HNE22]
20	5	10	5	2^{-14}	2^{-14}	$2^{-14.81}$	$2^{-70.81}$	Ours
	5	10	5	2^{-14}	2^{-14}	$2^{-19.96}$	$2^{-75.96}$	[HNE22]
21	5	10	6	2^{-10}	2^{-18}	$2^{-23.36}$	$2^{-79.36}$	Ours
	5	10	6	2^{-10}	2^{-19}	$2^{-26.55}$	$2^{-84.55}$	[HNE22]
22	6	10	6	2^{-16}	2^{-18}	$2^{-23.36}$	$2^{-91.36}$	Ours
	6	10	6	2^{-16}	2^{-19}	$2^{-26.55}$	$2^{-96.55}$	[HNE22]
23	7	10	6	2^{-26}	2^{-18}	$2^{-23.36}$	$2^{-111.36}$	Ours
	6	10	7	2^{-24}	2^{-20}	$2^{-27.59}$	$2^{-115.59}$	[HNE22]

5.2 On the Applicability of Our Model

Our model is more suitable for the generalized Feistel ciphers with many branches. For SPN ciphers and the Feistel ciphers with few branches, the impact of semi-active S-boxes in E_m on the whole boomerang probability is limited. It is the reason why we choose WARP of 32 branches as a target, which has shown a good effect. We also apply our model to two other ciphers TWINE and LBlock-s, which are the generalized Feistel structure with 16 branches. We find better 14-round and 15-round boomerang distinguishers for TWINE and LBlock-s than the previous distinguishers in [HNE22] and [LMR22]. A comparison is shown in Table 8, and the specification is presented in Table 12 and Table 13 in Appendix B. Note that by our model we find the same boomerang distinguishers for 13,16 rounds of TWINE and LBlock-s as those in [HNE22].

Table 8: Boomerang distinguishers for 14 and 15 rounds of TWINE and LBlock-s.

Ciphers	Rounds	Ours	[HNE22]	[LMR22]
TWINE	14	$2^{-39.11}$	$2^{-42.25}$	
	15	$2^{-47.19}$	$2^{-51.03}$	$2^{-47.7}$
Lblock-s	14	$2^{-36.6}$	$2^{-38.47}$	
	15	$2^{-44.82}$	$2^{-46.49}$	

6 Rectangle Attacks on Round-Reduced WARP

6.1 Properties of Local Structures and Precomputed Table

Before giving the rectangle attacks on WARP, we present some properties of two local structures shown in Figure 8 and Figure 9, which are helpful for constructing and filtering data in our rectangle attacks. We also present a kind of precomputed tables related to the local structure in Figure 9, which will be used in the key recovery stage.

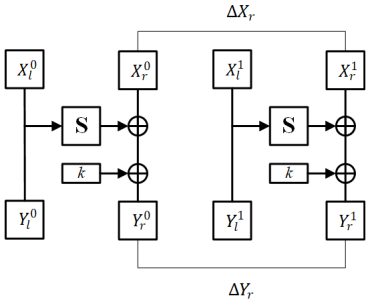

Figure 8: A pair of states through a local structure.

Property 1. In the structure of Figure 8, if X_l^0, X_l^1 and X_r^0 are known, and the difference $\Delta Y_r = Y_r^0 \oplus Y_r^1$ is known, then the value of X_r^1 can be computed directly.

Property 2. In the structure of Figure 8, let $\Delta X_r = X_r^0 \oplus X_r^1$, $v_0 = S(Y_l^0) \oplus Y_r^0$ and $v_1 = S(Y_l^1) \oplus Y_r^1$, then $v_0 \oplus v_1 = \Delta X_r$.

In Figure 8, since

$$k = S(X_l^0) \oplus X_r^0 \oplus Y_r^0 = S(X_l^1) \oplus X_r^1 \oplus Y_r^1,$$

and $X_l^0 = Y_l^0$, $X_l^1 = Y_l^1$, we have

$$\begin{aligned} X_r^1 &= S(X_l^0) \oplus S(X_l^1) \oplus X_r^0 \oplus \Delta Y_r, \\ v_0 \oplus v_1 &= S(Y_l^0) \oplus Y_r^0 \oplus S(Y_l^1) \oplus Y_r^1 = \Delta X_r. \end{aligned}$$

In our rectangle attack, Property 1 will be used to construct the plaintext pair (P_1, P_2) satisfying the desired difference, and Property 2 will be used to construct the desired ciphertext quartets. When the difference ΔX_r is known for the distinguisher, we can compute the values (v_1, v_2) for (C_1, C_2) and (v_3, v_4) for (C_3, C_4) according to Property 2, and then find the quartets (C_1, C_2, C_3, C_4) by the collision of $(v_1, v_2) \oplus (v_3, v_4) = (\Delta X_r, \Delta X_r)$.

Property 3. In the structure of Figure 9, if the differences $\Delta X_r = X_r^0 \oplus X_r^1$, $\Delta Z_r = Z_r^0 \oplus Z_r^1$, $\Delta Z_l' = Z_l^{0\prime} \oplus Z_l^{1\prime}$ are known, and $\Delta Z_l = Z_l^0 \oplus Z_l^1 = 0$, then the differences $\Delta_{in} = X_l^0 \oplus X_l^1$ and $\Delta_{out} = S(X_l^0) \oplus S(X_l^1)$ can be computed directly.

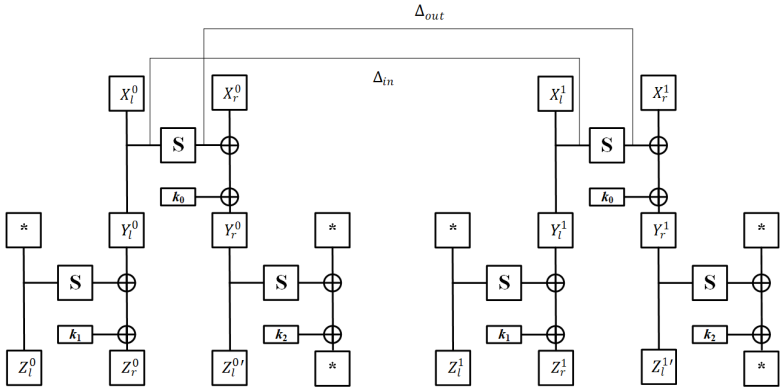

Figure 9: A pair of states through another local structure.

In Figure 9, it can be seen that

$$\Delta Y_r = Y_r^0 \oplus Y_r^1 = Z_l^{0\prime} \oplus Z_l^{1\prime} = \Delta Z_l', \Delta_{out} = \Delta X_r \oplus \Delta Y_r = \Delta X_r \oplus \Delta Z_l'.$$

Since $\Delta Z_l = Z_l^0 \oplus Z_l^1 = 0$, we have that

$$\Delta_{in} = Y_l^0 \oplus Y_l^1 = Z_r^0 \oplus Z_r^1 = \Delta Z_r.$$

In the rectangle attack, for the ciphertext quartets (C_1, C_2, C_3, C_4), by Property 3 we can compute the input difference Δ_{in} and output difference Δ_{out} of S-box for (C_1, C_3) and (C_2, C_4). For S-box of `WARP`, a random pair of input and output differences is a possible differential with probability about $2^{-1.32}$. It will provide about 2.64 bits filter for the ciphertext quartets by looking up the DDT of S-box.

Construct Precomputed Table. In Figure 9, if we know $Z_l^0, Z_r^0, Z_l^{0\prime}, Z_l^1, Z_r^1, Z_l^{1\prime}$ and the key k_1, then the difference $\Delta X_r = X_r^0 \oplus X_r^1$ can be computed by

$$\Delta X_r = S(S(Z_l^0) \oplus Z_r^0 \oplus k_1) \oplus S(S(Z_l^1) \oplus Z_r^1 \oplus k_1) \oplus Z_l^{0\prime} \oplus Z_l^{1\prime}.$$

In the key recovery stage of a rectangle attack, for a quartet (C_1, C_2, C_3, C_4) we need to determine the candidates of k_1 when knowing the values at $Z_l^0, Z_r^0, Z_r^{0\prime}$ for C_1, C_2, the values at $Z_l^1, Z_r^1, Z_r^{1\prime}$ for C_3, C_4, and the value $\Delta X_r = \alpha$. To reduce the time complexity of key recovery, we can precompute a table T_α of size 2^{48} as follows. Initialize T_α with all elements $\mathbb{NULL}$.

1. For each of 2^{24} values of $(Z_l^0, Z_r^0, Z_r^{0\prime}, Z_l^1, Z_r^1, Z_r^{1\prime})$, guess 2^4 values of k_1 and compute ΔX_r by the formula above. If $\Delta X_r = \alpha$, then store $(Z_l^0, Z_r^0, Z_r^{0\prime}, Z_l^1, Z_r^1, Z_r^{1\prime})$ in a table L indexed by the corresponding value of k_1. For each index in L, there are 2^{20} 24-bit terms on average.

2. For each index k_1 in L, combine any two terms from the 2^{20} terms to get 2^{40} ordered term pairs, and store k_1 in T_α indexed by the 48-bit values of the ordered term pairs.

On average, there are $2^{40+4-48} = 2^{-4}$ values in each index of T_α. That is, a quartet (C_1, C_2, C_3, C_4) can suggest candidates of k_1 with probability 2^{-4}. Using the precomputed table T_α, we can filter the quartet or obtain the candidate of k_1 by one table lookup in the key recovery stage. The time and memory complexities for constructing the precomputed table T_α are not more than 2^{48}.

6.2 Improved Rectangle Attack on 26-Round WARP

Based on the 21-round boomerang distinguisher of probability $2^{-79.36}$ found by our model, we mount an improved rectangle attack on 26-round WARP by extending two rounds before and three rounds at the end of the distinguisher, which is illustrated in Figure 10. Compared with the 26-round rectangle attack in [LMR22], the time, data and memory complexities are reduced by $2^{4.4}$, $2^{14.42}$ and $2^{14.42}$, respectively. The probability of the 21-round rectangle distinguisher transformed from the 21-round boomerang distinguisher is $2^{-128-79.36} = 2^{-207.36}$, and the input and output differences are

$$\Delta_1 = \texttt{0x0000000000000f00000000faa5000000},$$
$$\nabla_3 = \texttt{0x0000a0a00000000df00a00700500a0a0}.$$

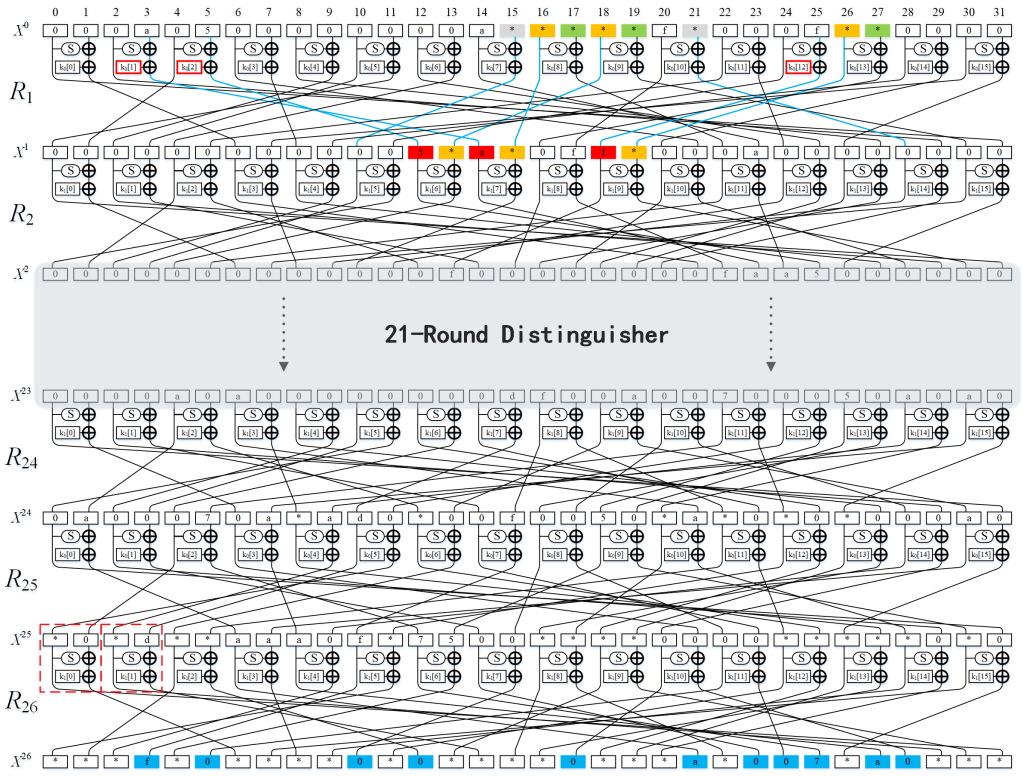

Figure 10: Rectangle attack on 26-round WARP.

Precomputation. Construct two precomputed tables T_0 and $T_{\texttt{a}}$ of size 2^{48} according to the method introduced in Section 6.1, where $\alpha = 0$ and 0xa. The time and memory complexities are not more than $2 \times 2^{48} = 2^{49}$.

The attack process is as follows.

1. Choose 2^t plaintext structures each containing 2^{24} plaintexts, where the 3 orange nibbles and 3 green nibbles of X^0 in Figure 10 traverse all possible values and the other 26 nibbles take constants randomly.

2. For each plaintext structure Ω, construct the related plaintext structure Ω' such that plaintext pairs (P_1, P_2), $P_1 \in \Omega$, $P_2 \in \Omega'$, satisfy the desired 26-nibble difference in X^0, of which the differences at 24 white nibbles are known and the differences

at 2 gray nibbles ΔX_{15}^0 and ΔX_{21}^0 can be computed as follows. Since the value of X_{14}^0 for P_1 is known constant, and the differences $\Delta X_{14}^0 = \texttt{0xa}$, $\Delta X_{10}^1 = 0$, we have $\Delta X_{15}^0 = S(X_{14}^0) \oplus S(X_{14}^0 \oplus \texttt{0xa})$. ΔX_{21}^0 can be computed in the same way.

3. Initialize a list of 2^{48} counters, which correspond to 12 nibbles of key $k_0[1]$, $k_0[2]$, $k_0[12]$, $k_1[1]$, $k_1[2]$, $k_1[3]$, $k_1[5]$, $k_1[6]$, $k_1[8]$, $k_1[9]$, $k_1[12]$ and $k_1[13]$.

4. Guess $k_0[1]$, $k_0[2]$ and $k_0[12]$, and construct the desired plaintext pairs (P_1, P_2), $P_1 \in \Omega$, $P_2 \in \Omega'$ as follows. For each plaintext structure Ω, partly encrypt it to obtain the values at the 3 red nibbles in X^1. Then, the differences at the 3 orange nibbles in X^1 can be computed based on Property 1. Furthermore, for each $P_1 \in \Omega$, we can compute the values at the 3 green nibbles in X^0 for P_2 based on Property 1. There are 2^{t+24} plaintext pairs (P_1, P_2) obtained for each guessed key.

5. For each ciphertext pair (C_1, C_2) of 2^{t+24} plaintext pairs (P_1, P_2), compute the values of (v_{i1}, v_{i2}), $i = 1, 2$, based on Property 2 for 2 local structures marked by red squares in the final round. Insert (C_1, C_2) into a hash table H indexed by 26 nibbles, which consists of the values of (C_1, C_2) at the 11 blue nibbles and the 2 values of (v_{i1}, v_{i2}). Construct the ciphertext quartets (C_1, C_2, C_3, C_4) by looking up the table H, such that for (C_1, C_3) and (C_2, C_4) the differences at above 13 nibbles are equal to the known values. There are about $Q = 2^{2t+47-26\times4} = 2^{2t-57}$ quartets obtained.

6. For each of Q quartets, derive the candidates of 9-nibble key $k_1[13]$, $k_1[1]$, $k_1[12]$, $k_1[6]$, $k_1[8]$, $k_1[2]$, $k_1[5]$, $k_1[9]$ and $k_1[3]$ by looking up the precomputed tables successively, which are shown in Table 9. Each table provides a filter of 2^{-4} for quartets. For example, lookup the table $T_{\texttt{a}}$ to find the candidates of $k_1[13]$, according to the 12-nibble values of (C_1, C_3, C_2, C_4) at $X_{19}^{26}, X_{16}^{26}, X_{29}^{26}$. If there is a candidate for $k_1[13]$, then continue to look up the next table in the same way, otherwise, go to the next quartet. If there is a candidate for the 9-nibble key together with the guessed $k_0[1]$, $k_0[2]$ and $k_0[12]$, then add 1 to the corresponding counter.

7. Select the top 2^{48-b} hits in the counters to be the candidates of the 12-nibble key. Exhaustively search the remaining 20 nibbles of the key and verify them.

Table 9: Precomputation tables for key recovery attack on 26-round WARP.

Key	Positions related with index	Precomputed table	Complexity
$k_1[13]$	$X_{19}^{26}, X_{16}^{26}, X_{29}^{26}$	$T_{\texttt{a}}$	Q
$k_1[1]$	$X_{29}^{26}, X_{14}^{26}, X_{31}^{26}$	T_0	$2^{-4}Q$
$k_1[12]$	$X_{11}^{26}, X_{18}^{26}, X_{1}^{26}$	T_0	$2^{-8}Q$
$k_1[6]$	$X_{25}^{26}, X_{4}^{26}, X_{7}^{26}$	T_0	$2^{-12}Q$
$k_1[8]$	$X_{15}^{26}, X_{22}^{26}, X_{9}^{26}$	$T_{\texttt{a}}$	$2^{-16}Q$
$k_1[2]$	$X_{1}^{26}, X_{12}^{26}, X_{11}^{26}$	T_0	$2^{-20}Q$
$k_1[5]$	$X_{3}^{26}, X_{0}^{26}, X_{13}^{26}$	T_0	$2^{-24}Q$
$k_1[9]$	$X_{13}^{26}, X_{30}^{26}, X_{15}^{26}$	T_0	$2^{-28}Q$
$k_1[3]$	$X_{21}^{26}, X_{8}^{26}, X_{19}^{26}$	T_0	$2^{-32}Q$

Complexity Analysis. The data complexity is $2 \times 2^t \times 2^{24} = 2^{t+25}$ for 2^t plaintext structures and the related plaintext structures. For each guessed key, there are 2^{t+24} plaintext pairs (P_1, P_2) satisfying the input difference of the 21-round distinguisher. Then, there are about 2^{2t+47} quartets satisfying the input differences of the rectangle distinguisher. Note that the probability of the rectangle distinguisher is $2^{-207.36}$. For the expected number of right quartets $s = 2^{2t+47-207.36} = 4$, we need $t = 81.18$. So the data complexity is $2^{t+25} = 2^{106.18}$. The time complexity of Step 4 is about $2^{12} \times (2^t + 2^{t+24}) \times 3 \times 2 \approx 2^{t+37.58}$

S-box computations, which is equivalent to $2^{t+37.58}/(26 \times 16) \approx 2^{t+28.71}$ encryptions of 26-round WARP. The time complexity of Step 5 is about $2^{12} \times 2^{t+25} \times 2$ S-box computations and $2^{t+24+12}$ memory access, which is equivalent to $2^{t+38.32}/(26 \times 16) \approx 2^{t+29.45}$ encryptions. The time complexity of Step 6 is about $2^{12} \times Q = 2^{2t-45}$ table lookups, which is equivalent to $2^{2t-53.87}$ encryptions. The complexity of Step 7 is 2^{128-b} encryptions. So the overall time complexity is $2^{t+25} + 2^{t+28.71} + 2^{t+29.45} + 2^{2t-53.87} + 2^{128-b}$. Taking $b = 24$ for the success probability of 84%, the overall time complexity is $2^{111.5}$. The memory complexity is $2^{106.18}$, which is bounded by the hash table H.

6.3 Rectangle Attack on 27-Round WARP

Employing the probabilistic extension technique for the rectangle attacks proposed by Song et al. in [SYC+24], based on the same 21-round boomerang distinguisher, we mount a rectangle attack on 27-round WARP by extending two rounds before and four rounds at the end of the distinguisher, which is illustrated in Figure 11.

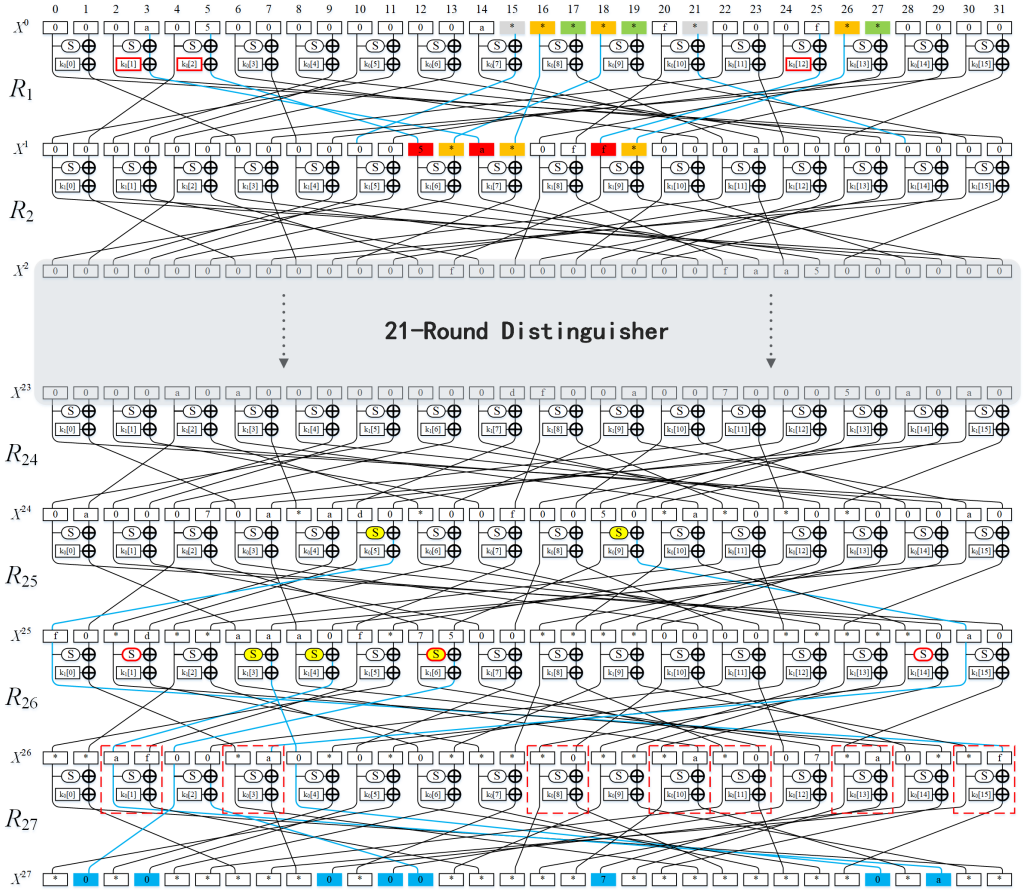

Figure 11: Rectangle attack on 27-round WARP.

Probabilistic Extension. In Figure 11, the output differences of the five yellow-colored S-boxes are set to fixed values each with probability 2^{-2}. That is, the differentials 0xd⟶0xf and 0x5⟶0xa for the two S-boxes in the 25th round, and the differentials 0xa⟶0xa, 0xa⟶0xa and 0x7⟶0x5 for the three S-boxes in the 26th round. The probability of the

extension part for two sides is $2^{-2\times5\times2} = 2^{-20}$, so the total probability of the rectangle distinguisher is $2^{-128-79.36-20} = 2^{-227.36}$.

Precomputation. Construct three precomputed tables T_0, $T_{\mathtt{a}}$ and $T_{\mathtt{d}}$ of size 2^{48} according to the method introduced in Section 6.1, where $\alpha = 0, \mathtt{0xa}$ and $\mathtt{0xd}$. The time and memory complexities are not more than $3 \times 2^{48} \approx 2^{49.58}$.

Choosing 2^t plaintext structures, guessing $k_0[1]$,$k_0[2]$,$k_0[12]$ and constructing plaintext pairs are the same as the Steps of 1, 2, 4 in the 26-round attack. For each guessed key, we get 2^{t+24} desired plaintext pairs. The rest process is as follows:

1. Initialize a list of 2^{36} counters, which correspond to 9 nibbles of key $k_0[1]$, $k_0[2]$, $k_0[3]$, $k_0[4]$, $k_0[10]$, $k_0[12]$, $k_0[13]$, $k_0[14]$ and $k_0[15]$.

2. For each ciphertext pair (C_1, C_2) of 2^{t+24} plaintext pairs (P_1, P_2), compute the values of (v_{i1}, v_{i2}), $i = 1, 2, ..., 7$, based on Property 2 for 7 local structures marked by red squares in the final round. Insert (C_1, C_2) into a hash table H indexed by 30 nibbles, which consists of the values of (C_1, C_2) at 8 blue nibbles and the 7 values of (v_{i1}, v_{i2}). Construct the ciphertext quartets (C_1, C_2, C_3, C_4) by looking up the table H, such that for (C_1, C_3) and (C_2, C_4) the differences at above 15 nibbles are equal to the known values. There are about $2^{2t+47-30\times4} = 2^{2t-73}$ quartets obtained.

3. For each of 2^{2t-73} quartets (C_1, C_2, C_3, C_4), based on Property 3 we can compute the input and output differences of the 3 red-marked S-boxes in the 26th round for (C_1, C_3) and (C_2, C_4), respectively. Note that for any given pair of input and output differences, it is a possible differential for S-box of WARP with probability about $2^{-1.32}$. That is, there exists 1.32-bit filter for each pair of input and output differences. In total, there is a filter of $2^{-1.32\times3\times2} = 2^{-7.92}$ for the quartets. Then, the expected number of quartets left is $Q = 2^{2t-73-7.92} = 2^{2t-80.92}$.

4. For each of Q quartets, derive the candidates of 6-nibble key $k_0[15]$, $k_0[14]$, $k_0[10]$, $k_0[13]$, $k_0[4]$ and $k_0[3]$ by looking up the precomputed tables successively, which are shown in Table 10.

5. Select the top 2^{36-b} hits in the counters to be the candidates of the 9-nibble key. Exhaustively search the remaining 23 nibbles of the key and verify them.

Table 10: Precomputation tables for key recovery attack on 27-round WARP.

Key	Positions related with index	Precomputed table	Complexity
$k_0[15]$	$X_7^{27}, X_{26}^{27}, X_{21}^{27}$	T_0	Q
$k_0[14]$	$X_9^{27}, X_{20}^{27}, X_{23}^{27}$	$T_{\mathtt{d}}$	$2^{-4}Q$
$k_0[10]$	$X_{17}^{27}, X_{28}^{27}, X_{27}^{27}$	$T_{\mathtt{a}}$	$2^{-8}Q$
$k_0[13]$	$X_{19}^{27}, X_{16}^{27}, X_{29}^{27}$	T_0	$2^{-12}Q$
$k_0[4]$	$X_{27}^{27}, X_2^{27}, X_{17}^{27}$	T_0	$2^{-16}Q$
$k_0[3]$	$X_{21}^{27}, X_8^{27}, X_{19}^{27}$	T_0	$2^{-20}Q$

Complexity Analysis. There are about 2^{2t+47} quartets satisfying the input differences of the rectangle distinguisher. Note that the probability of the rectangle distinguisher is $2^{-227.36}$. For the expected number of right quartets $s = 2^{2t+47-227.36} = 4$, we need $t = 91.18$. So the data complexity is $2^{t+25} = 2^{116.18}$. The time complexity of constructing plaintext pairs is about $2^{12} \times (2^t + 2^{t+24}) \times 3 \times 2 \approx 2^{t+37.58}$ S-box computations, which is equivalent to $2^{t+37.58}/(27 \times 16) \approx 2^{t+28.83}$ encryptions of 27-round WARP. The time complexity of Step 2 is about $2^{12} \times 2^{t+25} \times 7$ S-box computations and $2^{t+24+12}$ memory access, which is equivalent to $2^{t+39.91}/(27\times16) \approx 2^{t+31.16}$ encryptions. The time complexity of Step 3 is about 2^{2t-73} XORs and lookups DDT of S-box, which is equivalent to $2^{2t-80.75}$

encryptions. The time complexity of Step 4 is about $2^{12} \times Q = 2^{2t-68.92}$ table lookups, which is equivalent to $2^{2t-77.67}$ encryptions. The complexity of Step 5 is 2^{128-b} encryptions. So the overall time complexity is $2^{t+25} + 2^{t+28.83} + 2^{t+31.16} + 2^{2t-80.75} + 2^{2t-77.67} + 2^{128-b}$. Taking $b = 12$ for the success probability of 84.11%, the overall time complexity is $2^{122.63}$. The memory complexity is $2^{116.18}$, which is bounded by the hash table H.

7 Conclusions

In this paper, we revisit the model of searching boomerang distinguishers for Feistel structure proposed by Hadipour et al., and show the model is not optimal when applied to the generalized Feistel structure. It is shown that a boomerang distinguisher with more common active S-boxes may have a higher probability, which is caused by the semi-active S-boxes in E_m only active in either the upper or the lower differential trail. The semi-active S-boxes are not considered by the objective function in Hadipour et al.'s Model. We classify the active S-boxes in E_m in more detail according to the associated tables of DDT, DDT^2, FBCT and its variants in the computation of boomerang probability. Then, we propose an improved MILP model to search boomerang distinguishers for generalized Feistel structure. Applying this model to WARP, we find better boomerang distinguishers for all rounds than that found by Hadipour et al.'s model. For 15-round WARP, the probability of boomerang distinguisher is improved by a factor of $2^{5.78}$. We present a boomerang distinguisher on 23-round WARP with probability $2^{-111.36}$, resulting in the best result presented on WARP so far. We also apply our model to TWINE and Lblock-s, and find better 14-round and 15-round boomerang distinguishers of these two ciphers. Based on our 21-round boomerang distinguisher for WARP, we improved the 26-round rectangle attack on WARP with the time complexity of $2^{111.5}$ reduced by $2^{4.4}$. Exploiting the properties of two local structures and the probabilistic extension technique, we give the first rectangle attack on 27-round WARP, which improves the best previous rectangle attack on this cipher by one round.

References

[BBI+20] Subhadeep Banik, Zhenzhen Bao, Takanori Isobe, Hiroyasu Kubo, Fukang Liu, Kazuhiko Minematsu, Kosei Sakamoto, Nao Shibata, and Maki Shigeri. WARP : Revisiting GFN for lightweight 128-bit block cipher. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 535–564. Springer, 2020.

[BDK01] Eli Biham, Orr Dunkelman, and Nathan Keller. The rectangle attack - rectangling the Serpent. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, volume 2045 of *Lecture Notes in Computer Science*, pages 340–357. Springer, 2001.

[BHL+20] 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.

[BK09] 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.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.*, 4(1):3–72, 1991.

[CHP+18] 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.

[DDV20] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. Catching the fastest boomerangs application to SKINNY. *IACR Trans. Symmetric Cryptol.*, 2020(4):104–129, 2020.

[DKS14] 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.

[HBS21] Hosein Hadipour, Nasour Bagheri, and Ling Song. Improved rectangle attacks on SKINNY and CRAFT. *IACR Trans. Symmetric Cryptol.*, 2021(2):140–198, 2021.

[HDE24] Hosein Hadipour, Patrick Derbez, and Maria Eichlseder. Revisiting differential-linear attacks via a boomerang perspective with application to aes, ascon, clefia, skinny, present, knot, twine, warp, lblock, simeck, and SERPENT. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 38–72. Springer, 2024.

[HE22] Hosein Hadipour and Maria Eichlseder. Integral cryptanalysis of WARP based on monomial prediction. *IACR Trans. Symmetric Cryptol.*, 2022(2):92–112, 2022.

[HNE22] Hosein Hadipour, Marcel Nageler, and Maria Eichlseder. Throwing boomerangs into feistel structures application to CLEFIA, WARP, LBlock, LBlock-s and TWINE. *IACR Trans. Symmetric Cryptol.*, 2022(3):271–302, 2022.

[KKS00] John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified boomerang attacks against reduced-round MARS and Serpent. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 75–93. Springer, 2000.

[LMR22] Virginie Lallemand, Marine Minier, and Loïc Rouquette. Automatic search of rectangle attacks on feistel ciphers: Application to WARP. *IACR Trans. Symmetric Cryptol.*, 2022(2):113–140, 2022.

[Mur11] Sean Murphy. The return of the cryptographic boomerang. *IEEE Trans. Inf. Theory*, 57(4):2517–2521, 2011.

[Sel08] Ali Aydin Selçuk. On probability of success in linear and differential cryptanalysis. *J. Cryptol.*, 21(1):131–147, 2008.

[SLLM24] Jiali Shi, Guoqiang Liu, Chao Li, and Jianfeng Ma. Constructing the impossible differential of type-ii gfn with boolean function and its application to warp. *Chinese Journal of Electronics*, 33(1):80–89, 2024.

[SMMK12] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight block cipher for multiple platforms. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer, 2012.

[SWW22] Ling Sun, Wei Wang, and Meiqin Wang. Key-recovery attacks on CRAFT and WARP. In Benjamin Smith and Huapeng Wu, editors, *Selected Areas in Cryptography - 29th International Conference, SAC 2022, Windsor, ON, Canada, August 24-26, 2022, Revised Selected Papers*, volume 13742 of *Lecture Notes in Computer Science*, pages 77–95. Springer, 2022.

[SYC+24] Ling Song, Qianqian Yang, Yincen Chen, Lei Hu, and Jian Weng. Probabilistic extensions: A one-step framework for finding rectangle attacks and beyond. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 339–367. Springer, 2024.

[TB22] Je Sen Teh and Alex Biryukov. Differential cryptanalysis of WARP. *J. Inf. Secur. Appl.*, 70:103316, 2022.

[Wag99] 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.

[WP19] 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.

[WZ11] Wenling Wu and Lei Zhang. Lblock: A lightweight block cipher. In Javier López and Gene Tsudik, editors, *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, volume 6715 of *Lecture Notes in Computer Science*, pages 327–344, 2011.

A Boomerang Distinguishers for WARP

In Figure 12 and Figure 13, the red cells and lines indicate the propagation of active cells in the upper trail, and the blue cells and lines indicate the propagation of active cells in the lower trail. The purple cells and lines indicate they are common active in the upper and lower trails, and the S-boxes marked with green circles indicate they are the constrained S-boxes. The capital letters in cells are variables of the differences in the upper trail, and the lowercase letters in cells are variables of the differences in the lower trail.

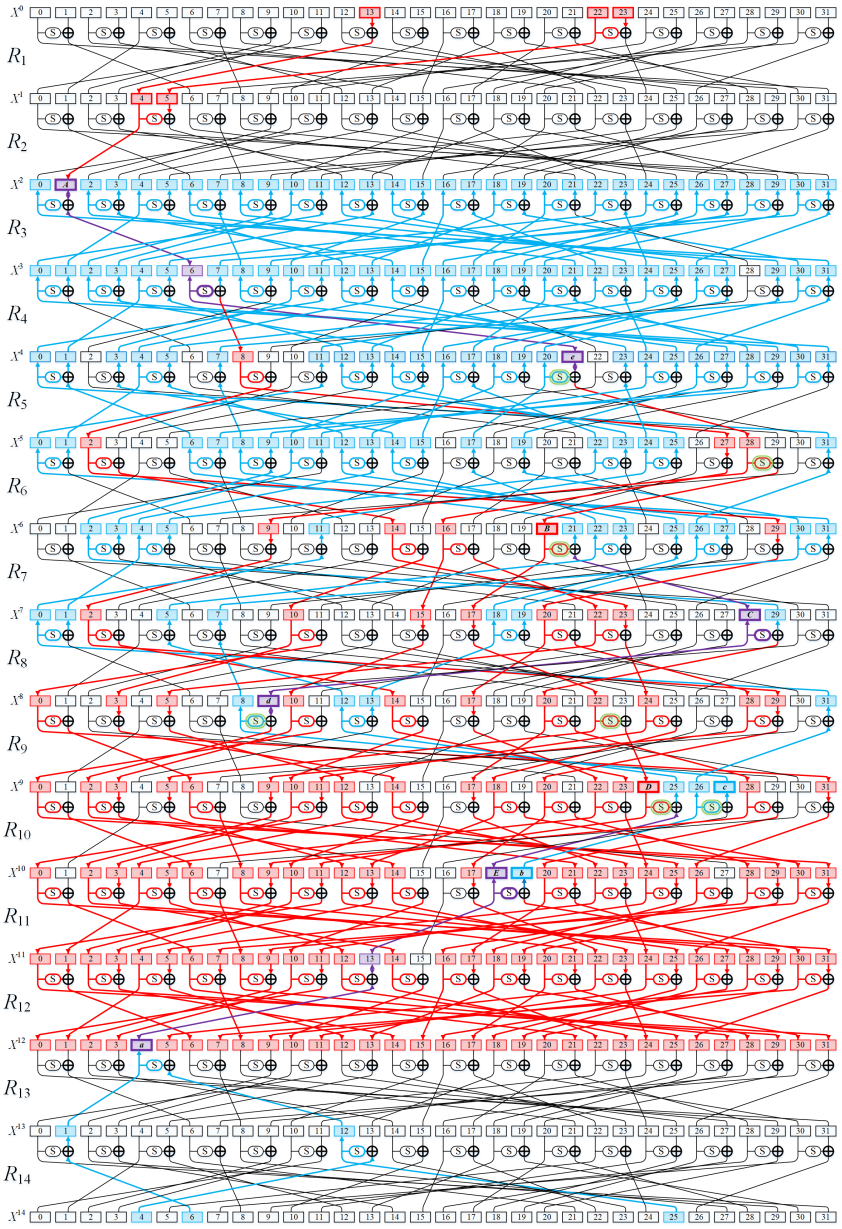

Figure 12: Boomerang distinguisher I on 14-round WARP [HNE22].

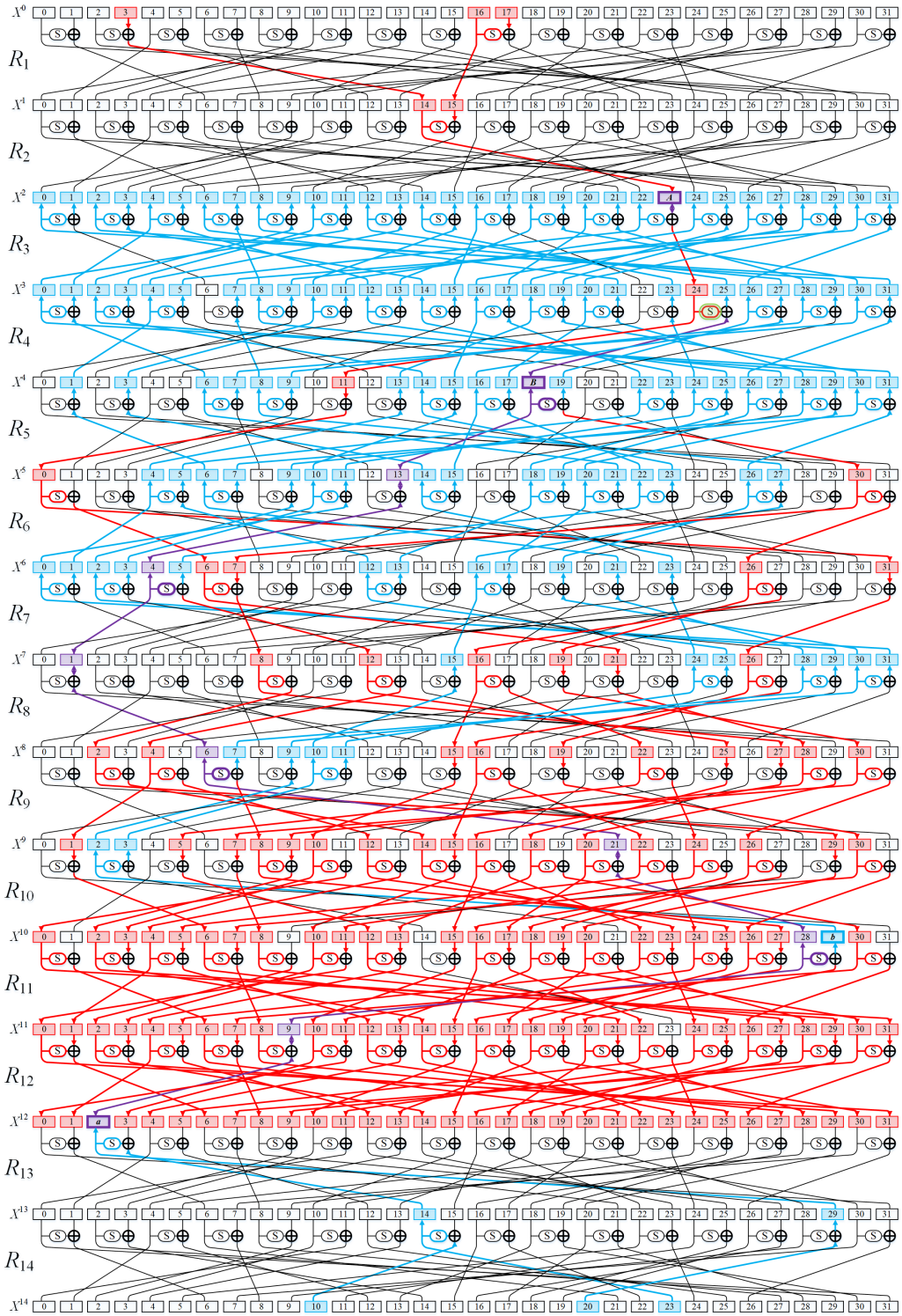

Figure 13: Boomerang distinguisher II on 14-round WARP.

Table 11: Specification of boomerang distinguishers for 14 to 23 rounds of WARP.

14 rounds						
$Pr = 2^{-19.11}$	$r_0 = 2$	$r_m = 10$	$r_1 = 2$	$p = 2^{-4}$	$q = 2^{-4}$	$r = 2^{-3.11}$
ΔX^0				ΔX^2		
0x000a000000000000ff00000000000000				0x0000000000000000000000a00000000		
∇X^{12}				∇X^{14}		
0x00a0000000000000000000000000000				0x00000000007000000000a00500000000		
15 rounds						
$Pr = 2^{-22.80}$	$r_0 = 2$	$r_m = 10$	$r_1 = 3$	$p = 2^{-4}$	$q = 2^{-4}$	$r = 2^{-8.80}$
ΔX^0				ΔX^2		
0xff00000000000000000a000000000000				0x0000000a0000000000000000000000		
∇X^{12}				∇X^{15}		
0x00000000000a0000000000000000000				0x00000000a000000000000500000a00000		
16 rounds						
$Pr = 2^{-30.80}$	$r_0 = 3$	$r_m = 10$	$r_1 = 3$	$p = 2^{-8}$	$q = 2^{-4}$	$r = 2^{-8.80}$
ΔX^0				ΔX^3		
0x0000000000af00000000a50000050000				0x0000000000000000000000a00000000		
∇X^{13}				∇X^{16}		
0x00000000000000000000000000000a0000				0x0000050000a0000000000000a0000000		
17 rounds						
$Pr = 2^{-38.80}$	$r_0 = 3$	$r_m = 10$	$r_1 = 4$	$p = 2^{-8}$	$q = 2^{-8}$	$r = 2^{-8.80}$
ΔX^0				ΔX^3		
0x0000000000af00000000a50000050000				0x0000000000000000000000a00000000		
∇X^{13}				∇X^{17}		
0x00000000000000000000000000000a0000				0xf00a0000000a500000a0000000000000		
18 rounds						
$Pr = 2^{-46.81}$	$r_0 = 4$	$r_m = 10$	$r_1 = 4$	$p = 2^{-8}$	$q = 2^{-8}$	$r = 2^{-14.81}$
ΔX^0				ΔX^4		
0x00000000000005000000007daf000000				0x0000000000000000000000000a00000		
∇X^{14}				∇X^{18}		
0x00000000000000000000000000000a00				0x00500a00000000000a000000f000a000		
19 rounds						
$Pr = 2^{-58.81}$	$r_0 = 5$	$r_m = 10$	$r_1 = 4$	$p = 2^{-14}$	$q = 2^{-8}$	$r = 2^{-14.81}$
ΔX^0				ΔX^4		
0x0000000000005aa50ffa000a00000000				0x0000000000000000000000000a00000		
∇X^{14}				∇X^{19}		
0x00000000000000000000000000000a00				0x00500a00000000000a000000f000a000		
20 rounds						
$Pr = 2^{-70.81}$	$r_0 = 5$	$r_m = 10$	$r_1 = 5$	$p = 2^{-14}$	$q = 2^{-14}$	$r = 2^{-14.81}$
ΔX^0				ΔX^5		
0x0000000000005aa50ffa000a00000000				0x0000000000000000000000000a00000		
∇X^{15}				∇X^{20}		
0x00000000000000000000000000000a00				0x000000000a0aa0a00050d0a000000500		
21 rounds						
$Pr = 2^{-79.36}$	$r_0 = 5$	$r_m = 10$	$r_1 = 6$	$p = 2^{-10}$	$q = 2^{-18}$	$r = 2^{-23.36}$
ΔX^0				ΔX^5		
0x0000000000000f00000000faa5000000				0x0000000000000000a00a000000000000		
∇X^{15}				∇X^{21}		
0x00000000000000000000000a00000000				0x0000a0a00000000df00a00700500a0a0		

22 rounds						
$Pr = 2^{-91.36}$	$r_0 = 6$	$r_m = 10$	$r_1 = 6$	$p = 2^{-16}$	$q = 2^{-18}$	$r = 2^{-23.36}$
ΔX^0				ΔX^6		
0x0000000000005aa50ffa000a00000000				0x0000000000000000a00a000000000000		
∇X^{16}				∇X^{22}		
0x00000000000000000000000a00000000				0x0000a0a00000000df00a00700500a0a0		
23 rounds						
$Pr = 2^{-111.36}$	$r_0 = 7$	$r_m = 10$	$r_1 = 6$	$p = 2^{-26}$	$q = 2^{-18}$	$r = 2^{-23.36}$
ΔX^0				ΔX^7		
0x000d0a00000000af7daa750005750000				0x0000000000000000a00a000000000000		
∇X^{17}				∇X^{23}		
0x00000000000000000000000a00000000				0x0000a0a00000000df00a00700500a0a0		

B Boomerang Distinguishers for TWINE and LBlock-s

Table 12: Specification of boomerang distinguishers for 14 and 15 rounds of TWINE.

14 rounds						
$Pr = 2^{-39.11}$	$r_0 = 4$	$r_m = 7$	$r_1 = 3$	$p = 2^{-8}$	$q = 2^{-4}$	$r = 2^{-15.11}$
ΔX^0	0x079800a700000000			ΔX^4	0x00000000a0000000	
∇X^{11}	0x00000d0000000000			∇X^{14}	0x00e00000010000d0	
15 rounds						
$Pr = 2^{-47.19}$	$r_0 = 4$	$r_m = 8$	$r_1 = 3$	$p = 2^{-8}$	$q = 2^{-8}$	$r = 2^{-15.19}$
ΔX^0	0x0065002a00050000			ΔX^4	0x0000000000006000	
∇X^{12}	0x0000000000000020			∇X^{15}	0x02a0a00000070090	

Table 13: Specification of boomerang distinguishers for 14 and 15 rounds of LBlock-s.

14 rounds						
$Pr = 2^{-36.6}$	$r_0 = 4$	$r_m = 7$	$r_1 = 3$	$p = 2^{-8}$	$q = 2^{-4}$	$r = 2^{-12.6}$
ΔX^0	0x0004040000040440			ΔX^4	0x0400000000000000	
∇X^{11}	0x0000000000000040			∇X^{14}	0x0040000400000040	
15 rounds						
$Pr = 2^{-44.82}$	$r_0 = 4$	$r_m = 7$	$r_1 = 4$	$p = 2^{-8}$	$q = 2^{-8}$	$r = 2^{-12.82}$
ΔX^0	0x0004040000040440			ΔX^4	0x0400000000000000	
∇X^{11}	0x0000000004000000			∇X^{15}	0x0004404000004400	