Chinese Journal of Electronics
vol. 33, no. 1, pp. 80–89, January 2024
https://doi.org/10.23919/cje.2022.00.132

RESEARCH ARTICLE

Constructing the Impossible Differential of Type-II GFN with Boolean Function and Its Application to WARP

Jiali SHI[1], Guoqiang LIU[1,2,3], and Chao LI[1,2,3]

1. *College of Liberal Arts and Sciences, National University of Defense Technology, Changsha 410000, China*
2. *Hunan Engineering Research Center of Commercial Cryptography Theory and Technology Innovation, Changsha 410000, China*
3. *State Key Laboratory of Information Security, Institute of Information Engineering, Beijing 100000, China*

Corresponding author: Guoqiang LIU, Email: liuguoqiang87@hotmail.com
Manuscript Received May 16, 2022; Accepted August 22, 2022

Abstract — Type-II generalized Feistel network (GFN) has attracted a lot of attention for its simplicity and high parallelism. Impossible differential attack is one of the powerful cryptanalytic approaches for word-oriented block ciphers such as Feistel-like ciphers. We deduce the impossible differential of Type-II GFN by analyzing the Boolean function in the middle round. The main idea is to investigate the expression with the variable representing the plaintext (ciphertext) difference words for the internal state words. By adopting the miss-in-the-middle approach, we can construct the impossible differential of Type-II GFN. As an illustration, we apply this approach to WARP, a lightweight 128-bit block cipher with a 128-bit key which was presented by Banik *et al.* at SAC 2020. The structure of WARP is a 32-branch Type-II GFN. Therefore, we find two 21-round truncated impossible differentials and implement a 32-round key recovery attack on WARP. For the 32-round key recovery attack on WARP, some observations are used to mount an effective attack. Taking the advantage of the early abort technique, the data, time, and memory complexities are $2^{125.69}$ chosen plaintexts, $2^{126.68}$ 32-round encryptions, and 2^{100}-bit, repectively. To the best of our knowledge, this is the best attack on WARP in the single-key scenario.

Keywords — WARP, Feistel cipher, Impossible differential attack.
Citation — Jiali SHI, Guoqiang LIU, Chao LI, "Constructing the Impossible Differential of Type-II GFN with Boolean Function and Its Application to WARP," *Chinese Journal of Electronics*, vol. 33, no. 1, pp. 80–89, 2024. doi: 10.23919/cje.2022.00.132.

I. Introduction

The impossible differential attack is one of the most effective cryptanalytic approaches used to evaluate the security of block ciphers. This method was introduced by Biham *et al.* [1] and Knudsen [2] independently. Its core idea is to use (truncated) differentials with zero probability to eliminate the wrong keys. Truncated differential focus on whether there are differences on some bytes and do not care about the values of the differences. The most important part of this attack is to find the longest impossible differential. Generally, such differentials are constructed by the miss-in-the-middle method [3], [4]. Given a differential $\alpha \to \beta$, it propagates α and β in forwards and backwards by r_0 and r_1 rounds, respectively. Then, we check whether there is a contradiction in the middle round. If there is a contradiction, it is a $(r_0 + r_1)$-round impossible differential. Based on the method, some automated tools called U-method [5], UID-method [6] and WW-method [7] have been exploited for searching impossible differentials. These methods focus on the impossible differential independent with the Sboxes. In addition, automated models such as MILP (mixed integer linear programming), CP, SMT and SAT are also often used for searching impossible differentials [8]–[11].

The Boolean function plays an important role in the security of block cipher. Generally, the output words in each round of block cipher can be regarded as Boolean functions over plaintext and key words. From the per-

Associate Editor: Prof. Jianfeng MA, Guizhou University.

spective of distinguishing attacks, the cryptanalysts try to construct an impossible differential which covers as many rounds as possible. At EUROCRYPT 2016, Sun *et al.* proved that the upper bound on the length of impossible differentials of substitution-permutation network (SPN) structure by the primitive index of the linear layers [12] without considering the details inside Sboxes. Inspired by this method, Zhang *et al.* evaluated the resistance of SPN block cipher against impossible differential attack by counting the occurrences of the plaintext (ciphertext) words appearing in the expression of the internal state words [13]. For SKINNY, they found all of the 11-round impossible differentials. Thus, we try to construct the impossible differential for Type-II-based ciphers [14] by exploring the algebraic expression in the middle round.

Type-II GFN is a popular design for block ciphers. There are many Type-II-based ciphers, e.g., TWINE [15] and WARP [16]. WARP is a lightweight block cipher, with a 128-bit block and a 128-bit key. This cipher submitted by Banik *et al.* at SAC 2020, suits the small-footprint circuit which can be used as a direct replacement for AES-128. Designers of WARP provided the security evaluations, they found a 20-round integral distinguisher by utilizing the MILP model proposed in [17] and a 21-round impossible differential. In terms of the meet-in-the-middle attack, they considered that it is difficult to amount 32-round WARP. For differential and linear attacks, they used the MILP model [18] to obtain the lower bound on the number of active Sboxes for up to 19-round. Afterward, Teh *et al.* [19] offered a 23-round differential attack. They also implemented 24-round rectangle attack on WARP with memory/data/time complexity $2^{127.06}/2^{126.06}/2^{122.49}$ by using the Feistel boomerang connectivity table (FBCT) proposed in [20].

Our contributions In this paper, we evaluate the security of WARP against impossible differential attacks. The output words in the middle round are represented as Boolean functions over plaintext and key words. These algebraic formulas provide precious information related to impossible differentials. Therefore, we deduce the impossible differentials by investigating the algebraic representation in the middle round. Further, we launch a key recovery attack on the reduced round WARP based on the impossible differential. More precisely,

- For Type-II GFN, we construct impossible differentials by utilizing the Boolean functions of the internal state words. Unlike the method provided in [13], we not only focus on the number of occurrences of the variable in the Boolean function, but also analyze the effect of the plaintext (ciphertext) difference words on the differential pattern of the internal state words.
- Apply this method on WARP. We deduce two 21-round truncated impossible differentials by analyzing the algebraic formulas of the internal state words. To verify the impossible differential, we search the contradictions in each round by using the SMT model proposed in [11].
- We proposed a 32-round key recovery attack on WARP. Based on a 21-round impossible differential, we launch a 32-round key recovery attack by pre-appending and appending five rounds and six rounds, respectively. In the process of the key recovery attack, we exploit the key schedule and remove the redundancy in the subkeys involved in the extended rounds. We also construct a hash table and adopt the early abort technique [21] to improve the attack. As a result, the data, time, and memory complexities are about $2^{125.69}$ chosen plaintexts, $2^{126.68}$ 32-round encryptions, and 2^{100} bits. This decreases the security margin of WARP to 22%. The cryptanalytic results for WARP in the single-key scenario are summarized in Table 1.

Table 1 Summary of cryptanalytic results on WARP

Approach	Rounds	Memory	Data	Time	Ref.
Differential	23	$2^{106.62}$	$2^{106.62}$	$2^{106.68}$	[19]
Rectangle	24	$2^{127.06}$	$2^{126.06}$	$2^{122.49}$	[19]
Impossible differential	32	2^{100}	$2^{125.69}$	$2^{126.68}$	Sect. IV

Outline Section II provides the notions and the descriptions of Type-II GFN and WARP. In Section III, we introduce how to construct the impossible differentials with the Boolean function of the internal state words for the Type-II GFN, and apply this generic method to WARP. Based on a 21-round impossible differential, we amount a 32-round key recovery attack on WARP in Section IV. And finally, Section V concludes this work.

II. Preliminaries

1. Notations

- Mk: the 128-bit master key.
- X_j^{r-1}: the jth nibble input in the rth round.
- Y_j^{r-1}: the jth nibble input of the π operation in the rth round.
- K_i^{r-1}: the ith nibble subkey in the rth round.
- ΔX_j^{r-1}: the jth nibble input difference in the rth round.
- ΔY_j^{r-1}: the jth nibble input difference of the π operation in the rth round.
- X^{r-1}: $X_0^{r-1}||X_1^{r-1}||\cdots||X_{31}^{r-1}$.

2. Type-II GFN

Zheng *et al.* introduced the Type-II Feistel which is a generalization of the original Feistel construction [22]. The Type-II Feistel apply the cyclically right-shifted to achieve Shannon's concepts of diffusion. Later, Nyberg [14] proposed the following Type-II GFN depicted in Figure 1.

Definition 1 The n-bit plaintext is devided into $q = 2p$ subblocks denoted as $X_0, X_1, \ldots, X_{2p-1}$. The size of subblock X_i is m-bit. The ciphertext of the Type-II GFN is obtained by iterating R rounds of the plaintext. Let F_j^r be the cryptographic keyed functions, where

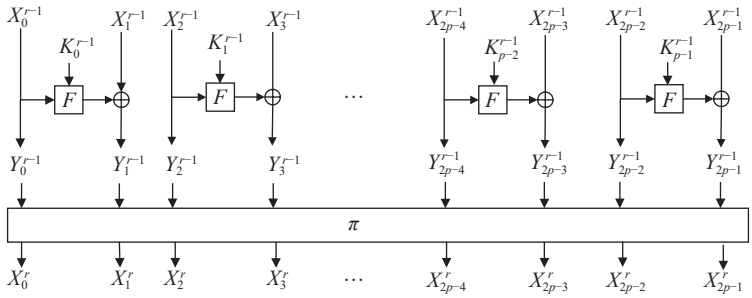

Figure 1 One round of a Type-II GFN with $2p$ subblocks.

$0 < r \leq R$, $0 \leq j < p$, and π be a permutation over $2p$ subblocks. The round function in the rth round is

$$(X_0^r, X_1^r, \ldots, X_{2p-1}^r) \leftarrow \pi(X_0^{r-1}, F_0^{r-1}(X_0^{r-1}) \oplus X_1^{r-1}, \ldots, X_{2p-2}^{r-1}, F_{p-1}^{r-1}(X_{2p-2}^{r-1}) \oplus X_{2p-1}^{r-1})$$

Let Y_i^{r-1} be the input of the permutation π in the rth round, i.e., $Y_1^{r-1} = F_0^{r-1}(X_0^{r-1}) \oplus X_1^{r-1}$. In other words, we say that X_0^{r-1} or X_1^{r-1} diffuses to Y_1^{r-1}. As in [23], the definition of diffusion round is as follows.

Definition 2 The permutation over $2p$ branches is denoted as π. X_i^0 fully diffusion means that X_i^0 diffuses to all $j \in \{0, 1, \ldots, 2p-1\}$, X_j^r after r rounds. π permutation to reach full diffusion means full diffusion for all $i \in \{0, 1, \ldots, 2p-1\}$, X_i^0 after r rounds. The diffusion round is defined as the minimum number of round that satisfies this property for the block X_i^0.

Note that the diffusion rounds for encryption and decryption is not necessarily equal. When designing the cipher, the permutation and its inverse can reach full diffusion as quickly as possible. The diffusion round of the permutation π is as follows.

Definition 3 Let $DR_i(\pi)$ be the minimum number of rounds r which X_i^0 reach fully diffusion after r rounds. The diffusion round of π is written as

$$DR_m = \max_{0 \leq i \leq 2p-1} \{DR_i(\pi), DR_i(\pi^{-1})\}$$

3. Description of WARP

`WARP` is a 128-bit lightweight block cipher with 128-bit key [16]. It adopts a variant of Type-II GFN with 32 nibbles. The round function of `WARP` consists of a 4-bit Sbox, 4-bit XOR operation, and a permutation over 32 nibbles. There are 41 rounds. The round function can be described as follows.

$$Y_{2i+1}^{r-1} = S(X_{2i}^{r-1}) \oplus X_{2i+1}^{r-1} \oplus K_i^{r-1}$$
$$Y_{2i}^{r-1} = X_{2i}^{r-1}$$

Then, the input nibbles of π in the rth round are denoted as Y_j^{r-1}, and the corresponding output nibbles are written as $X_{\pi(j)}^r = Y_j^{r-1}$ where $1 \leq r \leq 41$, $0 \leq i \leq 15$, and $0 \leq j \leq 31$. There are four following operations in the round function.

1) Sbox: Apply the 4-bit Sbox of `MIDORI` [24] which is described in Table 2.

Table 2 The 4-bit Sbox

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

2) Add subkey: The XOR subkey operation is executed after the Sbox. This operation XOR the subkey and the internal state.

3) Permutation: The 32-branch permutation π is given in Table 3.

Table 3 The permutation over 32 nibbles

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

4) Add round constants: XOR the round constants and the internal state in the 1st and 3rd branches, respectively.

The permutation π is omitted in the last round. In addition, we do not introduce the round constants, which do not affect the feasibility of impossible differential attacks. For more details, please refer to [16]. Next, we will briefly describe the key schedule.

Key schedule The 128-bit master key is represented as two 64-bit keys $Mk = Mk^0||Mk^1$, i.e., $Mk^0 = k_0^0||k_1^0||\cdots||k_{15}^0$, $Mk^1 = k_0^1||k_1^1||\cdots||k_{15}^1$. Then, the subkeys in the rth round is $Mk^{(r-1) \bmod 2}$.

III. Constructing the Impossible Differentials for Type-II GFN

In this section, we construct the impossible differential by analyzing the Boolean function of the internal state words in the middle round. First, we review how to construct the impossible differential by using the miss-in-the-middle approach. Second, we provide a generic approach for constructing the impossible differentials of the Type-II GFN by investigating the internal state expres-

sion in the middle round. Finally, we take WARP as an example to deduce the impossible differential by using the new method and verify these impossible differentials with the SMT model [11].

1. Reviewing the miss-in-the-middle method

We focus on the truncated difference and ignore the details inside Sboxes. Usually, the differential pattern of the internal state is the linear combination of the following four differential patterns.

- Z: the difference is zero;
- N_{fix}: the difference is non-zero and fixed;
- N_{noz}: the difference can take any value except zero;
- U: the difference can be any value.

By applying the miss-in-the-middle approach, there are 3 types of contradictions for constructing impossible differentials.

- Type-1 contradiction. There is contradiction between the zero difference and the non-zero difference, that is, $Z \neq N_{\text{noz}}$ or $Z \neq N_{\text{fix}}$.
- Type-2 contradiction. There is contradiction between the two different fixed values. For example, if $N_{\text{fix},0} = \alpha$, $N_{\text{fix},1} = \beta$, then there is a contradiction in $N_{\text{fix},0} \neq N_{\text{fix},1}$, where $\alpha \neq \beta$;
- Type-3 contradiction. Disjointness of the codomain of two differential patterns can also lead to contradictions.

In the middle round, if there is always at least one type of contradiction for the differential of the internal state, we call this differential an impossible differential. Note that regardless of whether the cipher is based on the SPN structure or the Feistel structure, the number of rounds of the impossible differentials is closely related to the number of diffusion round, and so is the GFN. Some works are devoted to the study of the impossible differentials for GFN, for example, to evaluate the security of GFN, Suzaki *et al.* [24] determined the number of rounds of the impossible differential from DR_m. They applied the U-method to search for the impossible differential, and they pointed out that the number of rounds for the impossible differentials is one of $(2DR_m-2)$, $(2DR_m-1)$ and $(2DR_m+1)$. Therefore, we can construct the impossible differential of GFN by analyzing the Boolean functions in the (DR_m+1), DR_m or (DR_m-1) round.

2. A generic approach for constructing the impossible differentials

For Type-II GFN, the output nibbles in the rth round can be described by tweakable Boolean functions, which contain both secret variables (key words) and public variables (plaintext words). Assume the subkey K_j^r is used in the keyed function F_j^r. Since $K_j^r \oplus K_j^r = 0$, thus, the key words do not influence the value of the input difference and can be ignored. In other words, the keyed Sbox has the same difference distribution table as the unkeyed Sbox. Hence, the output subblocks in the rth round are represented as Boolean functions which only contained the public variables (the plaintext difference words). Generally, the non-linear operation in the keyed function F_i^r is called Sbox represented as S. Let ΔX denote the difference of the variable X. The output differences of the subblocks in the rth round is written as

$$(\Delta X_0^r, \Delta X_1^r, \ldots, \Delta X_{2p-1}^r) \leftarrow \pi(\Delta X_0^{r-1}, S(\Delta X_0^{r-1}) \oplus \Delta X_1^{r-1}, \ldots, \Delta X_{2p-2}^{r-1}, S(\Delta X_{2p-2}^{r-1}) \oplus \Delta X_{2p-1}^{r-1})$$

Similarly, we obtain the Boolean function of the output difference nibble in the rth round with the variables ΔX_j^0 representing the input difference nibbles, where $0 \leq j < 2p$. To describe the effect of the variables ΔX_j^0 on the output difference, we write the Boolean function f of the ith output nibble ΔX_i^r as follows.

$$\begin{aligned} \Delta X_i^r &= f(\Delta X_0^0, \Delta X_1^0, \ldots, \Delta X_{2p-1}^0) \\ &\equiv G_{\text{CN}}(\Delta X_I^0) \oplus \Delta X_t^0 \end{aligned} \tag{1}$$

where $0 \leq i < 2p$, $G_{\text{CN}}(\Delta X_I^0)$ represents a composite function of nonlinear operations with the variables indexed by the subset $I \subseteq \{0, 1, \ldots, 2p-1\}$. ΔX_t^0 indicates that this variable has a linear relationship with the output function, and $t \in \{0, 1, \ldots, 2p-1\}$.

Example 1 To demonstrate these notions, let

$$f(\Delta X_0^0, \Delta X_1^0, \Delta X_2^0, \Delta X_3^0) = S(S(S(\Delta X_0^0) \oplus \Delta X_1^0) \oplus \Delta X_2^0) \oplus S(\Delta X_2^0) \oplus \Delta X_3^0$$

be a polynomial in four variables. Clearly, $I = \{0, 1, 2\}$ is an index subset, and

$$\begin{aligned} G_{CN} &= S(S(S(\Delta X_0^0) \oplus \Delta X_1^0) \oplus \Delta X_2^0) \oplus S(\Delta X_2^0) \\ \Delta X_t^0 &= \Delta X_3^0 \end{aligned}$$

It can be seen that the difference of ΔX_3^0 and the output difference $f(\Delta X_0^0, \Delta X_1^0, \Delta X_2^0, \Delta X_3^0)$ have a linear relationship. Therefore, if the difference of ΔX_I^0 is fixed to 0, the differential pattern of $f(\Delta X_0^0, \Delta X_1^0, \Delta X_2^0, \Delta X_3^0)$ is consistent with the differential pattern of ΔX_3^0.

3. Applications to WARP

Considering WARP as an example, we analyze how to construct the impossible differentials from the Boolean function point of view. Usually, we focus on the impossible differential with one active input-output difference nibble. Therefore, by observing the Boolean function in equation (1), we deduce the corresponding output differential patterns in the rth round.

- f_Z: There is no variable representing the active difference in the Boolean function.
- $f_{N_{\text{fix}}}$: Only the variable ΔX_t^0 in the Boolean function denotes fixed differences N_{fix}, and the differences of all other variables ΔX_I^0 are fixed to 0.
- $f_{N_{\text{noz}}}$: The variable with active difference passes through the composite nonlinear function G_{CN} and the variable representing active difference appears only once

in G_{CN}. For instance, in Example 1, such variable could be ΔX_0^0 or ΔX_1^0, but not ΔX_2^0.

• f_U: The variable representing active difference appears at least twice in the Boolean function, e.g., in Example 1, if the difference of the variable ΔX_2^0 is active, it can make the value of the output difference unknown.

Since the number of full diffusion rounds of WARP is 10, that is, $DR_m = 10$. In addition, with the help of the MILP model given in [23], the designer found a 21-round impossible differential [16] by considering the details inside the Sbox. Therefore, we first investigate the output function in the 10th round from the encryption and decryption.

1) For the encryption direction, we get the Boolean function denoted as ΔeX_j^{10} of the output difference nibble in the 10th round according to the variable ΔX_i^0 representing the plaintext differences, where $0 \le i < 31$, $0 \le j < 32$. By choosing different input differences, we find that only the following two equations can construct a fixed differential pattern $f_{N_{\mathrm{fix}}}$.

$$\Delta eX_7^{10} = G_{\mathrm{CN}}(eI_0) \oplus \Delta X_3^0$$
$$\Delta eX_{23}^{10} = G_{\mathrm{CN}}(eI_1) \oplus \Delta X_{19}^0$$

where $eI_0 = \{0, 1, 2, 4, \ldots, 31\}$ and $eI_1 = \{0, 1, \ldots, 18, 20, \ldots, 31\}$.

2) Similarly, for the decryption direction, we deduce the Boolean function denoted as ΔdX_i^{10} of output differences in the 10th round with the variables ΔX_j^{21} representing the output differences in the 21th round. We find that the output Boolean functions ΔdX_7^{10}, ΔdX_{23}^{10} also contradict the corresponding Boolean functions ΔeX_7^{10}, ΔeX_{23}^{10} in the encryption direction, that are

$$\Delta dX_7^{10} = G_{\mathrm{CN}}(dI_0) \oplus \Delta X_0^{21}$$
$$\Delta dX_{23}^{10} = G_{\mathrm{CN}}(dI_1) \oplus \Delta X_{16}^{21}$$

where $dI_0 = dI_1 = \{0, 1, \ldots, 31\}$.

The specific expression of ΔeX_7^{10} is in Appendix A. For instance, let $\Delta X_3^0 = \Delta X_0^{21} = \delta$, where $\delta \in \{1, 2, \ldots, 15\}$, then

$$\Delta eX_7^{10} = \delta$$
$$\Delta dX_7^{10} = S(S(S(S(\delta)))) \oplus \delta$$

As a result, we get 2 following 21-round truncated impossible differentials for WARP.

$$\Omega_0 = (0\text{x}000\delta000000000000000000000000000 \nrightarrow 0\text{x}\delta000000000000000000000000000000)$$
$$\Omega_1 = (0\text{x}000000000000000000\delta000000000000 \nrightarrow 0\text{x}000000000000000\delta000000000000000)$$

The 21-round impossible differential given in [16] is Ω_1 with $\delta = 0\text{x}8$.

Verify the contradiction in the impossible differential We manually verify the impossible differential, and the specific contradiction marked in red of the impossible differential Ω_0 is shown in Table 4, where "$*$" denotes the non-zero difference, and "?" represents the unknown difference. In addition, we also apply the SMT model given in [11] to check the contradictions in each round. For the impossible differential Ω_0, Table 5 shows the one nibble contradiction positions of each intermediate round from the 1st round to the 20th round. For instance, when the connecting round is the 8th round, there is a contradiction in the 7th or the 19th nibble.

IV. Key Recovery Attack on the 32-Round WARP

In this section, with the 21-round impossible differential Ω_0, we launch a 32-round impossible differential attack on WARP by extending five rounds and six rounds forward and backward the impossible differential, respectively. To lower the complexity of this attack, we deploy several techniques, such as exploiting the properties and observations of WARP, constructing a hash table and applying the early abort method.

1. Properties and observations on WARP

We further investigate the structure of WARP and the key schedule. Some properties and observations can be used to improve the key recovery attack, where Property 1 was used in [20]. Then we elaborate on them.

Property 1 For the two-branch Feistel-subround shown in the Figure 2 of WARP, the XOR key operation is executed after the Sbox. Therefore, part of the encryption and decryption operations can be performed without guessing the subkey. Specifically,

• For the encryption direction, given an input pair $(X_0^0||X_1^0, X_0^{0\prime}||X_1^{0\prime})$ which satisfies the input difference $\Delta X_0^0||\Delta X_1^0$. The fixed difference ΔY_1^0 can be used to check whether the given input pair is valid without guessing the subkey K_0^0.

• Analogously, for the decryption direction, the fixed difference ΔX_1^0 can be utilized to check whether the given pair $(Y_0^0||Y_1^0, Y_0^{0\prime}||Y_1^{0\prime})$ is valid or not without guessing the subkey K_0^0.

Observation 1 According to Property 1, in the key recovery phase in Section IV, there are some nibbles with fixed differences in the 1st and the 32nd round can be directly used to filter the corresponding plaintext-ciphertext pairs without guessing the subkey. Such 44-bit differences are as follows.

$$\Delta Y_{7,9,17,23,27}^0 = 0\text{x}0$$
$$\Delta X_{3,11,21,25,27,29}^{31} = 0\text{x}0$$

Observation 2 From the 1st to the 5th round and the 27th to the 32nd round, we can derive the relations among the subkeys according to the key schedule of WARP. The 128-bit master key is represented as two 64-bit keys $Mk = Mk^0||Mk^1$, i.e., $Mk^0 = k_0^0||k_1^0||\cdots||k_{15}^0$, $Mk^1 = k_0^1||k_1^1||\cdots||k_{15}^1$. Thus, the relations between the subkeys and the master key can be expressed as follows.

Table 4 The 21-round impossible differential Ω_0

	Round	Difference
Encryption	ΔeX^0	0x000δ 0000 0000 0000 0000 0000 0000 0000
	ΔeX^1	0x0000 0000 0000 00δ0 0000 0000 0000 0000
	ΔeX^2	0x0000 0000 00 $*$ 0 0000 0000 000δ 0000 0000
	ΔeX^3	0x $*$ 00 $*$ 0000 0000 0000 0000 0000 δ000 0000
	ΔeX^4	0x0000 00 $*$ 0 0δ00 00 $*$ 0 00 $*$ 0 0000 0000 000$*$
	ΔeX^5	0x0000 0000 $*$ 0 $*$ 0 0 $*$ 00 0000 0 $*$ 0 $*$ 0 00 $*$ 0 00 $*$ 0
	ΔeX^6	0x $*$ 0 $*$ $*$ $*$ 0 $*$ $*$ 0000 0000 $*$ 00 $*$ 0000 $*$ 0 $*$ $*$ $*$ 00δ
	ΔeX^7	0x0 $*$ 00 00 $*$ 0 ? $*$ 0 $*$ $*$0?$*$?0 $*$ $*$ $*$ $*$ $*$ 0 00δ0 0 $*$ $*$
	ΔeX^8	0x $*$ 0?0 $*$ $*$ $*$ $*$ $*$0?0 0 $*$ 0? $*$ $*$0δ $*$ $*$?? $*$ $*$?? ?0?0
	ΔeX^9	0x? $*$ $*$? $*$?$*$? ???$*$?0?$*$?$*$?? ?$*$?0 ?0?$*$??$\delta*$
	ΔeX^{10}	0x?$*$?? ???δ ???? ???? ???? ?$*$?? ???? ?$*$??
Decryption	ΔdX^{10}	0x$*$??? ?? $*$? ???? ???? ???? ???? ???? ????
	ΔdX^{11}	0x??$*$? ???? δ??? ??$*$? ???? $*$ $*$?? ???? $*$? $*$ $*$
	ΔdX^{12}	0x$*$??? $*$? $*$ $*$? $*$ $*$? $*$??? 0$*$?? $*$? $*$ $*$??0δ 0$*$??
	ΔdX^{13}	0x0 $*$ $*$ $*$ $*$ $*$ $*$? 00$*$? $*$?00 δ $*$ 00 $*$ $*$ $*$? 0 $*$ $*$? $*$? $*$ $*$
	ΔdX^{14}	0x $*$ $*$0 $*$ 0 $*$ $*$ $*$ $*$ $*$00 000δ $*$ $*$ $*$ $*$ $*$ $*$00 $*$ $*$00 0 $*$ 00
	ΔdX^{15}	0x0000 0000 00δ $*$ $*$ $*$ $*$ $*$ 0 $*$ 00 $*$ $*$00 000 $*$ 000$*$
	ΔdX^{16}	0x000δ 0000 0000 0000 $*$ $*$00 00 $*$ $*$ 0 $*$ $*$0 0000
	ΔdX^{17}	0x0000 0 $*$ 00 0000 00δ $*$ 00 $*$ $*$ 0000 0000 0000
	ΔdX^{18}	0x0000 0000 0000 $*$ $*$00 0000 000δ 0000 0000
	ΔdX^{19}	0x0000 0000 0000 0000 0000 0000 δ $*$ 00 0000
	ΔdX^{20}	0x0000 0000 000δ 0000 0000 0000 0000 0000
	ΔdX^{21}	0xδ000 0000 0000 0000 0000 0000 0000 0000

Table 5 The one nibble contradiction position in each intermediate round for the 21-Round impossible differential Ω_0

Intermediate round	1-nibble contradiction position
1	0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
2	0, 2, 3, 4, 5, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
3	4, 5, 8, 9, 10, 12, 14, 15, 16, 17, 18, 22, 23, 24, 28, 29, 30, 31
4	2, 3, 10, 11, 12, 13, 20, 22, 24, 25, 26, 27
5	0, 4, 5, 14, 16, 17, 18, 19
6	12, 22, 23, 30, 31
7	24, 25, 26
8	18, 19
9	30
10	7
11	8, 21
12	17, 27, 28
13	9, 15, 16, 19, 22
14	2, 5, 10, 13, 15, 23, 27, 30
15	1, 3, 4, 5, 7, 10, 12, 16, 19, 23, 24, 25, 29
16	1, 3, 5, 6, 8, 9, 11, 12, 13, 14, 15, 18, 20, 21, 24, 25, 29, 30, 31
17	0, 1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 17, 18, 20, 21, 23, 25, 26, 27, 28, 29, 31
18	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31
19	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
20	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31

$$K_i^{r-1} = \begin{cases} k_i^0, & \text{if } (r-1) \mod 2 = 0 \\ k_i^1, & \text{otherwise} \end{cases}$$

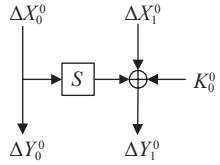

Figure 2 The differential propagation of the Feistel-subround.

2. The 32-round key recovery attack on WARP

For the key recovery attack shown in Figure 3, X_j^{r-1} denotes the jth input nibble in the rth round. The ith input nibble of π_0 operation in the rth round is written as Y_j^{r-1}. Where $0 \le j \le 31$ and $1 \le r \le 41$. In the following, we will elaborate on the attack.

Step 1 We construct 2^n structures. In each structure, according to Observation 1, we select a set of 2^{52} plaintexts which has 76-bit fixed values. These plaintexts can form about 2^{103} plaintext pairs. These plaintext pairs satisfy the following differential pattern.

$$\Delta X^0 = 0\text{x}000 * 00 * *\delta * 000000 * *0000 * *0 * * * 0 * 00$$

Where $*$ represents unknown difference and δ de-

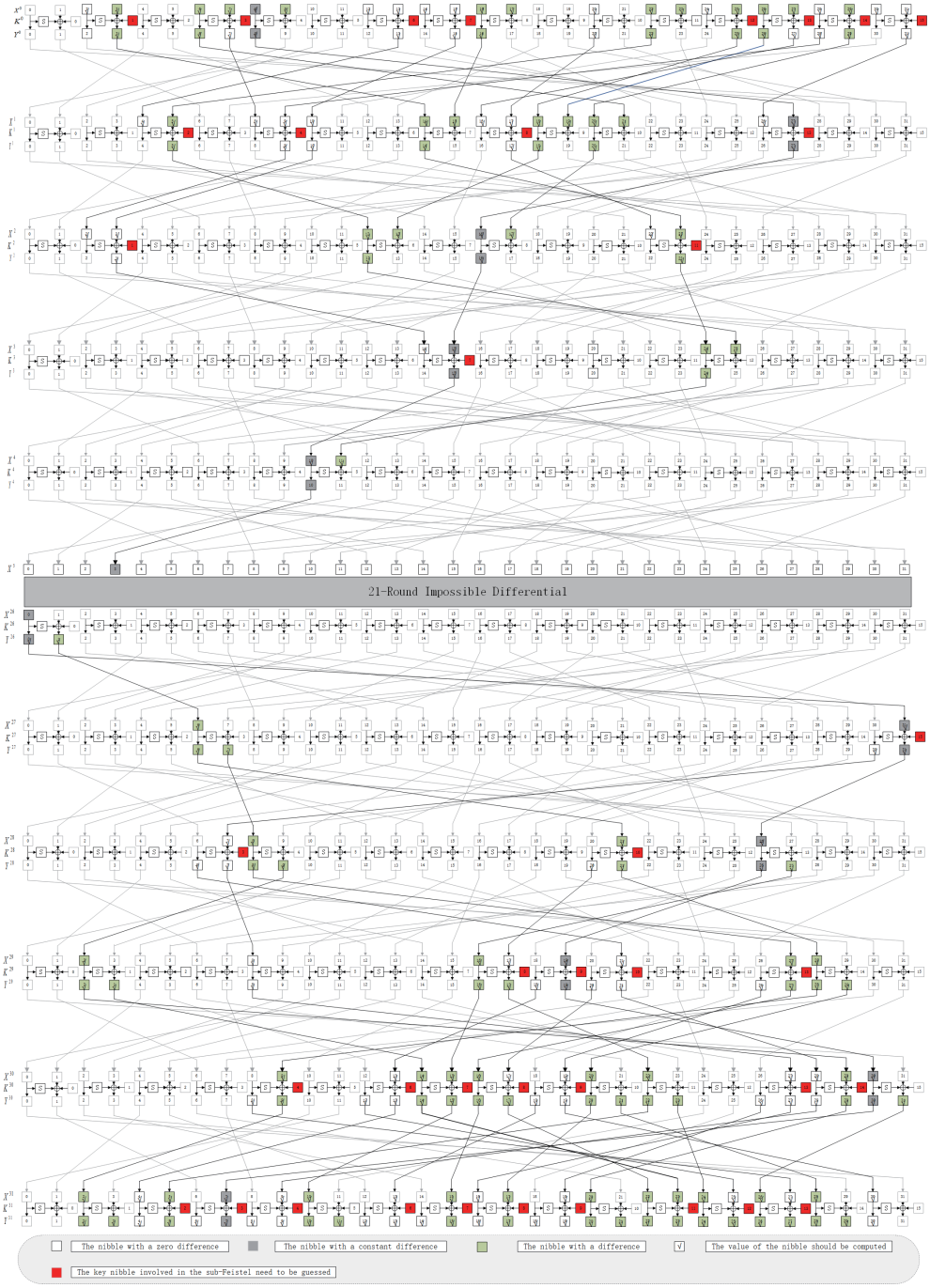

Figure 3 Type-II generalized Feistel network.

notes nonzero difference. Thus, we collect about 2^{n+103} plaintext pairs. For each plaintext, we can get the corresponding ciphertext by encrypting the plaintext. The ciphertext pairs are required to conform to the following differential pattern $\Delta X^{32} = \Delta Y^{31}$.

$$\Delta X^{32}\text{=0x00}**0*0\delta00**000*0*00**********00$$

There are 60 inactive bits in ΔX^{32}. In addition, according to Observation 1, we also can perform a 44-bit filter without guessing the subkey nibbles, and remain about $2^{n+103-60-44} = 2^{n-1} = 2^m$ plaintext pairs. In this step, we simplify the selection of proper pairs by constructing a hash table HT_p.

Step 2 To reduce the time complexity in the key recovery phase, we take the property of the key schedule into account and utilize the early abort method. The involved key bits, the number of remaining pairs, and the time complexity in Step 2 are detailed in Table 6 (Note that the unit of the time complexity is one Sbox operation). The detailed procedure is as follows.

- Step 2.1: Guess the 4-bit subkey K_7^{31}. For each remaining pair, we guess the value K_7^{31} and compute the pair of $(X_{15}^{31}, X_{15}^{31})$. Since $Y_{16}^{30} = X_{15}^{31}$ and $Y_{17}^{30} = X_{22}^{31}$, we can check whether the 4-bit difference ΔX_{17}^{30} hold or not according to the Property 1. Then there are about 2^{m-4} plaintext pairs remaining. The time complexity is $2 \cdot 2^m \cdot 2^4 \cdot 2$. Similarly, for these subkey K_8^{31}, K_2^{31}, K_3^{31}, K_1^0, K_{12}^0, K_{14}^0 and K_6^0, we guess each 4-bit subkey and check the validity of the condition $\Delta X_{21}^{30} = 0$, $\Delta X_{23}^{30} = 0$, $\Delta X_{31}^{30} = 0$, $\Delta Y_{15}^{1} = 0$, $\Delta Y_{19}^{1} = 0$, $\Delta Y_{21}^{1} = 0$ and $\Delta Y_{13}^{2} = 0$, respectively. At the end of this step, the expected number of the remaining pairs is about 2^{m-32}.
- Step 2.2: Guess the 12-bit subkey K_4^{31}, K_{14}^{31} and K_{11}^{31}. The 4-bit subkey K_{14}^{31} and K_{14}^0 guessed in Step 2.1 share the same master key k_{14}^0 according to the Observation 2, so we only need to guess the 8-bit subkey K_4^{31} and K_{11}^{31}. For each pair, we compute the value of X_9^{31} by guessing the 4-bit subkey K_4^{31}. Then, as the values of $Y_{28}^{30} = X_{29}^{31}$ and $Y_{29}^{30} = X_{20}^{31}$ are known, we use the value of the master key k_{14}^0 to calculate the value of X_{29}^{31}. We obtain the value of X_{23}^{29} with the enumeration of the 4-bit value K_{11}^{31}. From the relationship $Y_3^{29} = X_{14}^{30} = Y_{14}^{30} = X_{23}^{31}$, $Y_3^{29} = X_{29}^{30}$ between these values, and Property 1, we verify whether the condition $\Delta X_3^{29} = 0$. The expected number of remaining pairs is about 2^{m-36}, and the time complexity is $2 \cdot 2^{m-32} \cdot 2^{32} \cdot 2^8 \cdot 3$. Analogously, we enumerate the values of these subkey K_7^{30}, K_{15}^0 and K_{13}^1, K_{11}^2, K_4^{30}, K_6^{31} and K_9^{30}, K_{12}^{31} and K_9^{29}, K_{10}^{28}, K_3^0 separately and verify that the differences $\Delta X_3^{29} = 0$, $\Delta X_{17}^{29} = 0$, $\Delta Y_{17}^{2} = 0$, $\Delta Y_{25}^{3} = 0$, $\Delta X_{29}^{29} = 0$, $\Delta X_{9}^{28} = 0$, $\Delta X_{27}^{28} = 0$, $\Delta X_{7}^{27} = 0$ and $\Delta y_{11}^{4} = 0$ are satisfied. Therefore, we expect about 2^{m-68} remaining pairs.
- Step 2.3: Guess the 28-bit subkey K_8^{30}, K_{10}^{29}, K_{13}^{30}, K_{15}^{27}, K_9^{31} and K_3^{28}. With the Observation 2 of the key schedule, the 8-bit subkey K_9^{31}, K_3^{28} and K_9^1, K_3^0 guessed share the same 8-bit master key as k_9^1, k_3^0, so this step only needs to guess the 20-bit subkey. For each of 2^{m-68}

Table 6 Detailed computation of complexity in the key recovery phase

Step	Guessed subkey	Condition	#Remaining pairs	Time complexity
Step 2.1	K_7^{31}	$\Delta x_{17}^{30} = 0$	$2^m \cdot 2^{-4}$	$2 \cdot 2^m \cdot 2^4 \cdot 2$
	K_8^{31}	$\Delta x_{21}^{30} = 0$	$2^{m-4} \cdot 2^{-4}$	$2 \cdot 2^{m-4} \cdot 2^4 \cdot 2^4 \cdot 2$
	K_2^{31}	$\Delta x_{23}^{30} = 0$	$2^{m-8} \cdot 2^{-4}$	$2 \cdot 2^{m-8} \cdot 2^8 \cdot 2^4 \cdot 2$
	K_3^{31}	$\Delta x_{31}^{30} = 0$	$2^{m-12} \cdot 2^{-4}$	$2 \cdot 2^{m-12} \cdot 2^{12} \cdot 2^4 \cdot 2$
	K_1^0	$\Delta y_{15}^{1} = 0$	$2^{m-16} \cdot 2^{-4}$	$2 \cdot 2^{m-16} \cdot 2^{16} \cdot 2^4 \cdot 2$
	K_{12}^0	$\Delta y_{19}^{1} = 0$	$2^{m-20} \cdot 2^{-4}$	$2 \cdot 2^{m-20} \cdot 2^{20} \cdot 2^4 \cdot 2$
	K_{14}^0	$\Delta y_{21}^{1} = 0$	$2^{m-24} \cdot 2^{-4}$	$2 \cdot 2^{m-24} \cdot 2^{24} \cdot 2^4 \cdot 2$
	K_6^0	$\Delta y_{13}^{2} = 0$	$2^{m-28} \cdot 2^{-4}$	$2 \cdot 2^{m-28} \cdot 2^{28} \cdot 2^4 \cdot 2$
Step 2.2	K_4^{31},K_{11}^{31}	$\Delta x_3^{29} = 0$	$2^{m-32} \cdot 2^{-4}$	$2 \cdot 2^{m-32} \cdot 2^{32} \cdot 2^8 \cdot 3$
	K_7^{30}	$\Delta x_{17}^{29} = 0$	$2^{m-36} \cdot 2^{-4}$	$2 \cdot 2^{m-36} \cdot 2^{40} \cdot 2^4 \cdot 4$
	K_{15}^0,k_{13}^1	$\Delta y_{17}^{2} = 0$	$2^{m-40} \cdot 2^{-4}$	$2 \cdot 2^{m-40} \cdot 2^{44} \cdot 2^8 \cdot 2$
	K_{11}^2	$\Delta y_{25}^{3} = 0$	$2^{m-44} \cdot 2^{-4}$	$2 \cdot 2^{m-44} \cdot 2^{52} \cdot 2^4 \cdot 3$
	K_4^{30}	$\Delta x_{29}^{29} = 0$	$2^{m-48} \cdot 2^{-4}$	$2 \cdot 2^{m-48} \cdot 2^{56} \cdot 2^4 \cdot 3$
	K_6^{31}, k_9^{30}	$\Delta x_9^{28} = 0$	$2^{m-52} \cdot 2^{-4}$	$2 \cdot 2^{m-52} \cdot 2^{60} \cdot 2^8 \cdot 4$
	K_{12}^{31}, k_9^{29}	$\Delta x_{27}^{28} = 0$	$2^{m-56} \cdot 2^{-4}$	$2 \cdot 2^{m-56} \cdot 2^{68} \cdot 2^8 \cdot 4$
	K_{10}^{28}	$\Delta x_7^{27} = 0$	$2^{m-60} \cdot 2^{-4}$	$2 \cdot 2^{m-60} \cdot 2^{76} \cdot 2^4 \cdot 3$
	K_3^0	$\Delta y_{11}^{4} = 0$	$2^{m-64} \cdot 2^{-4}$	$2 \cdot 2^{m-64} \cdot 2^{80} \cdot 2^4 \cdot 5$
Step 2.3	K_8^{30}, K_{10}^{29},K_{13}^{30},K_{15}^{27}	$\Delta x_1^{26} = 0$	ϵ	$2 \cdot 2^{m-68} \cdot 2^{84} \cdot 2^{20} \cdot 7$

remaining pairs, check whether the input nibble difference satisfies $\Delta X_1^{26} = 0$ of the 21-round impossible differential. Hence, the probability for a wrong joint subkey surviving is about 2^{-4}. Thus these wrong values are eliminated from the table HT_p.

Step 3 In the above two steps, we already guess the value of the 100-bit subkey. And with the Observation 2, the corresponding 100-bit master key are $k^0_{1,3,4,6,7,8,9,10,11,12,13,14,15}$ and $k^1_{2,3,4,6,7,8,9,10,11,12,13,15}$. We also guess the remaining 28-bit master key, then check whether this key is correct by about 2^{28} encryptions. If this guessed key is correct, end this attack. Otherwise, return to Step 2 to continue guessing until the only correct key is recovered.

Complexity of the attack In the key recovery phase, we use ϵ to denote the expected number of the wrong master keys remaining after Step 2.3, where

$$\epsilon = 2^{100} \cdot (1-2^{-4})^{2^{m-68}} = 2^{100} \cdot (1-2^{-4})^{2^{n-69}}$$

The time complexity of Step 3 is

$$2 \cdot 2^{100} \cdot (1 + (1-2^{-4}) + \cdots + (1-2^{-4})^{2^{n-69}}) + \epsilon \cdot 2^{28}$$

As for the whole key recovery attack, the data and memory complexities are 2^{n+52} chosen plaintexts and $(2^{n-1} \cdot 4 \cdot 128 + 2^{100})$-bit. The time complexity is about 2^{n+52} 32-round encryptions which are dominated by collecting proper pairs in Step 1. Let $n = 73.69$, with about $\epsilon \cdot 2^{28}$ encryptions, the correct master key will be recovered. Therefore, the data, time and memory complexities are about $2^{125.69}$ chosen plaintexts, $2^{126.68}$ 32-round encryptions and 2^{100}-bit.

V. Conclusion

In this paper, from the Boolean function point of view, we analyze the truncated impossible differential of Type-II GFN. An application of this method to WARP gives two 21-round truncated impossible differentials. Then, with the aid of the SMT model, we recognize the contradiction in the connection round. Furthermore, we launch the key recovery attack on 32-round WARP based on the 21-round impossible differential. To decrease the complexity, we adopt some techniques such as the early abort method and the properties of WARP. Therefore, with a 21-round impossible differential, we amount the 32-round key recovery attack on WARP. This is the best result on WARP in the single-key setting.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61702537 and 62172427).

Appendix A

After 10 rounds of encryption, the output nibble ΔeX_7^{10} can be represented by the following Boolean function with the variables denoting the input difference nibbles ΔX_i^0, where $0 \le i < 32$.

$$\begin{aligned}\Delta eX_7^{10} =& S(S(S(S(S(S(S(\Delta X_{22}^0) + \Delta X_{23}^0) + \Delta X_{12}^0) + S(\Delta X_{30}^0) + \Delta X_{31}^0) + S(S(\Delta X_{14}^0) + \Delta X_{15}^0) + \Delta X_{24}^0)\\
&+ S(S(S(\Delta X_{10}^0) + \Delta X_{11}^0) + \Delta X_4^0) + S(\Delta X_{18}^0) + \Delta X_{19}^0) + S(S(S(S(\Delta X_{20}^0) + \Delta X_{21}^0) + \Delta X_2^0) + S(\Delta X_0^0)\\
&+ \Delta X_1^0) + S(S(\Delta X_6^0) + \Delta X_7^0) + \Delta X_{28}^0) + S(S(S(S(S(\Delta X_{10}^0) + \Delta X_{11}^0) + \Delta X_4^0) + S(\Delta X_{18}^0) + \Delta X_{19}^0)\\
&+ S(S(\Delta X_{28}^0) + \Delta X_{29}^0) + \Delta X_6^0) + S(S(S(\Delta X_8^0) + \Delta X_9^0) + \Delta X_{10}^0) + S(\Delta X_{26}^0) + \Delta X_{27}^0)\\
&+ S(S(S(S(S(\Delta X_{12}^0) + \Delta X_{13}^0) + \Delta X_{22}^0) + S(\Delta X_{24}^0) + \Delta X_{25}^0) + S(S(\Delta X_{26}^0) + \Delta X_{27}^0) + \Delta X_{20}^0)\\
&+ S(S(S(\Delta X_{16}^0) + \Delta X_{17}^0) + \Delta X_{14}^0) + S(\Delta X_4^0) + \Delta X_5^0) + S(S(S(S(\Delta X_{18}^0) + \Delta X_{19}^0) + \Delta X_0^0) + S(\Delta X_6^0)\\
&+ \Delta X_7^0) + S(S(\Delta X_{20}^0) + \Delta X_{21}^0) + \Delta X_2^0) + S(S(S(S(S(S(\Delta X_{30}^0) + \Delta X_{31}^0) + \Delta X_8^0) + S(\Delta X_{28}^0) + \Delta X_{29}^0)\\
&+ S(S(\Delta X_8^0) + \Delta X_9^0) + \Delta X_{10}^0) + S(S(S(\Delta X_4^0) + \Delta X_5^0) + \Delta X_{18}^0) + S(\Delta X_{16}^0) + \Delta X_{17}^0) + S(S(S(S(\Delta X_{22}^0)\\
&+ \Delta X_{23}^0) + \Delta X_{12}^0) + S(\Delta X_{30}^0) + \Delta X_{31}^0) + \Delta X_{15}^0) + \Delta X_{24}^0) + S(S(S(S(S(\Delta X_6^0) + S(S(\Delta X_{14}^0) + \Delta X_7^0)\Delta\\
&+ X_{28}^0) + S(\Delta X_{14}^0) + \Delta X_{15}^0) + S(S(\Delta X_{30}^0) + \Delta X_{31}^0) + \Delta X_8^0) + S(S(S(\Delta X_{26}^0) + \Delta X_{27}^0) + \Delta X_{20}^0)\\
&+ S(\Delta X_2^0) + \Delta X_3^0\end{aligned}$$

References

[1] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, Prague, Czech Republic, pp. 12–23, 1999.

[2] S. Lucks, "On the security of the 128-bit block cipher DEAL," in *Proceedings of the 6th International Workshop on Fast Software Encryption*, Rome, Italy, pp. 60–70, 1999.

[3] E. Biham, A. Biryukov, and A. Shamir, "Miss in the middle attacks on IDEA and Khufu," in *Proceedings of the 6th International Workshop on Fast Software Encryption*, Rome, Italy, pp. 124–138, 1999.

[4] A. Biryukov, "Miss-in-the-middle attack," in *Encyclopedia of Cryptography and Security*, 2nd ed., H. C. A. van Tilborg and S. Jajodia, Eds. Springer, New York, NY, USA, pp. 786, 2011.

[5] J. Kim, S. Hong, J. Sung, *et al.*, "Impossible differential cryptanalysis for block cipher structures," in *Proceedings of the 4th International Conference on Progress in Cryptology*, New Delhi, India, pp. 82–96, 2003.

[6] Y. Y. Luo, X. J. Lai, Z. M. Wu, *et al.*, "A unified method for finding impossible differentials of block cipher structures," *Information Sciences*, vol. 263, pp. 211–220, 2014.

[7] S. B. Wu and M. S. Wang, "Automatic search of truncated impossible differentials for word-oriented block ciphers," in *Proceedings of the 12th International Conference on Progress in Cryptology*, Kolkata, India, pp. 283–302, 2012.

[8] T. T. Cui, S. Y. Chen, K. T. Jia, *et al.*, "New automatic search tool for impossible differentials and zero-correlation linear approximations," Available at: http://eprint.iacr.org/

2016/689, 2016.

[9] Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects," in *Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Paris, France, pp. 185–215, 2017.

[10] L. Sun, D. Gerault, W. Wang, *et al.*, "On the usage of deterministic (related-key) truncated differentials and multidimensional linear approximations for SPN ciphers," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 3, pp. 262–287, 2020.

[11] X. C. Hu, Y. Q. Li, L. Jiao, *et al.*, "Mind the propagation of states," in *Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security*, Daejeon, South Korea, pp. 415–445, 2020.

[12] B. Sun, M. C. Liu, J. Guo, *et al.*, "Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis," in *Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vienna, Austria, pp. 196–213, 2016.

[13] W. Y. Zhang, M. C. Cao, J. Guo, *et al.*, "Improved security evaluation of SPN block ciphers and its applications in the single-key attack on SKINNY," *IACR Transactions on Symmetric Cryptology*, vol. 2019, no. 4, pp. 171–191, 2020.

[14] K. Nyberg, "Generalized Feistel networks," in *Proceedings of the International Conference on the Theory and Applications of Crypotology and Information Security*, Kyongju, Korea, pp. 91–104, 1996.

[15] T. Suzaki, K. Minematsu, S. Morioka, *et al.*, "TWINE: A lightweight block cipher for multiple platforms," in *Proceedings of the 19th International Conference on Selected Areas in Cryptography*, Windsor, Canada, pp. 339–354, 2013.

[16] S. Banik, Z. Z. Bao, T. Isobe, *et al.*, "WARP: revisiting GFN for lightweight 128-bit block cipher," in *Proceedings of the 27th International Conference on Selected Areas in Cryptography*, Halifax, NS, Canada, pp. 535–564, 2021.

[17] Z. J. Xiang, W. T. Zhang, Z. Z. Bao, *et al.*, "Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers," in *Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security*, Hanoi, Vietnam, pp. 648–678, 2016.

[18] N. Mouha, Q. J. Wang, D. W. Gu, *et al.*, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Proceedings of the 7th International Conference on Information Security and Cryptology*, Beijing, China, pp. 57–76, 2012.

[19] J. S. Teh and A. Biryukov, "Differential cryptanalysis of WARP," Available at: https://eprint.iacr.org/2021/1641, 2021.

[20] H. Boukerrou, P. Huynh, V. Lallemand, *et al.*, "On the Feistel counterpart of the boomerang connectivity table," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 1, pp. 331–362, 2020.

[21] J. Q. Lu, J. Kim, N. Keller, *et al.*, "Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1," in *Proceedings of the Cryptographers' Track at the RSA Conference on Topics in Cryptology*, San Francisco, CA, USA, pp. 370–386, 2008.

[22] Y. L. Zheng, T. Matsumoto, and H. Imai, "On the construction of block ciphers provably secure and not relying on any unproved hypotheses," in *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, pp. 461–480, 1989.

[23] S. Banik, A. Bogdanov, T. Isobe, *et al* ., "Midori: A block cipher for low energy," in *Proceedings of* the 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, pp. 411–436, 2015.

[24] T. Suzaki and K. Minematsu, "Improving the generalized Feistel," in *Proceedings of* the 17th International Workshop on Fast Software Encryption, Seoul, Korea, pp. 19–39, 2010.

Jiali SHI was born in 1992. She is a Ph.D. candidate of National University of Defense Technology. Her research interests include design and analysis of the block ciphers.
(Email: jiali00@126.com)

Guoqiang LIU was born in 1986. He received the Ph.D. degree in Information Engineering University. His research interests include design and cryptanalysis of block ciphers.
(Email: liuguoqiang87@hotmail.com)

Chao LI was born in 1966. He is a Ph.D. Researcher and Ph.D. Supervisor in National University of Defense Technology. His research interests include coding theory and symmetric-key cryptography.
(Email: lichao_nudt@sina.com)