Improved the Automated Evaluation Algorithm Against Differential Attacks and Its Application to WARP

Jiali Shi, Guoqiang Liu and Chao Li

August 29, 2022

Improved the Automated Evaluation Algorithm against Differential Attacks 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, China
2. Hunan Enginnering Research Center of Commercial Cryptography Theory and Technology Innovation, Changsha, China
3. State Key Laboratory of Information Security, Institute of Information Engineering, Beijing, China
{jiali00@126.com,liuguoqiang87@hotmail.com,lichao_nudt@sina.com}

Abstract. This paper presents a heuristic approach to find the key recovery-friendly distinguishers for block ciphers, which aims to attack more rounds with a lower complexity. Firstly, we construct an SAT model to search for a set of distinguishers with the minimum number of active input-output words (and optimal probability). Subsequently, based on the discovered distinguishers, we select the advantageous distinguisher with fewer key bits involved in the key recovery phase. Finally, the guess-and-check for the key recovery attack is performed using the manual approach to compute the attack parameters accurately. By applying our new technique to WARP proposed in SAC 2020, we identify some 19-round and 20-round advantageous differentials. Simultaneously, the high-probability chain of Sbox leads to a stronger clustering effect of the differential trails for WARP, so we effectively improve the probability of the advantageous distinguisher. Also, we perform the first 25-round differential attacks by extending a 19-round and a 20-round distinguisher, respectively. The results cover 2 more rounds than the previous known differential attacks.

Keywords: Differential Attack, SAT/SMT Model, Clustering Effect, WARP

1 Introduction

The differential attack was introduced by Biham and Shamir [5]. The goal of a differential attack is to attack more rounds with a lower complexity. There are generally two phases to achieve this goal, i.e., constructing a distinguisher and then launching a key recovery attack upon the distinguisher. Nowadays, most automated models focus on searching for differential trials with optimal probabilities, such as branch and bound method [12], CP [18], MILP [13, 19], SAT [11, 16], SMT [8]. However, a good differential attack is affected by many

Table 1: Summary of cryptanalytic results on WARP

Approach	Rounds	Data	Time	Memory	Ref
Differential Attack	21(2+16+3)	2^{113}	2^{113}	2^{72}	[9]
	23(2+18+3)	$2^{106.62}$	$2^{106.68}$	$2^{106.62}$	[21]
	25(3+19+3)	$2^{117.92}$	$2^{114.27}$	$2^{117.92}$	Sect. 4.2
	25(2+20+3)	$2^{123.71}$	$2^{120.06}$	$2^{123.71}$	Sect. 4.3
Rectangle Attack	24(1+21+2)	$2^{126.06}$	$2^{122.49}$	$2^{127.06}$	[21]
	26(1+22+3)	$2^{120.6}$	$2^{115.9}$	$2^{120.6}$	[10]
Integral Attack	32(1+22+9)	2^{127}	2^{127}	2^{108}	[7]

factors. In addition to the probability of the distinguisher, the cryptanalyst also needs to consider the input and output differences of the differential, the number of rounds extended by the distinguisher, and the number of key bits involved in the key recovery phase, etc. These factors influence and constrain each other, so how to trade off these factors is the key to executing better attacks.

To search for a advantageous distinguisher, Zong et al. [23] studied the key-recovery-attack friendly differentials and performed the first 27-round differential attack on GIFT-128 [3]. At SAC 2021, to facilitate a 20-round attack on GIFT-64, Sun et al. [17] identified the advantageous distinguisher by exhaustively checking all the 13-round differential trails with probabilities no less than 2^{-64}. Motivated by this observation, we improve the automated evaluation algorithm against differential attacks on block ciphers.

WARP was proposed by Banik et al. at SAC 2020 [1]. It is a lightweight block cipher with a 128-bit block and key. The design goal of WARP is the small-footprint circuit in the field of 128-bit block ciphers. Its structure is a variant of the 32-branch Type-II generalized Feistel network (GFN). The designers [1] evaluated the resistance of WARP to the differential, linear, integral, impossible differential, and meet-in-the-middle attacks. For differential attack, they provided the minimum number of active Sboxes for the first 19 rounds differential trails. Then, a 23-round differential attack and 24-round rectangle attack for WARP were proposed by Teh and Biryukov in [21]. This work has not yet determined the minimum active Sbox and optimal probability for the 20-round differential trail.

Our Contributions. Compared with Zong's method [23], our method is more efficient and general. Zong et al. first found the initial set containing the input-output differences of such distinguisher: extend more rounds and fewer keys involved in the key recovery phase, then identified the valid distinguisher from the initial set. Using this method, they performed a 27-round differential attack on GIFT-128, one more round than the existing results. However, this method requires a lot of computational resources, and the probability of the distinguisher constructed from the input-output differences in the initial set may not be optimal. Therefore, we first search for the distinguisher with the optimal probability and then deduce the advantageous distinguisher that involves fewer key bits in the key recovery phase. In more detail, we achieve the following:

- We provide a two-step strategy to search for the differentials which have advantages in the distinguishing phase and key recovery phase. Firstly, we construct an SAT model to enumerate all the input-output differential patterns of the differential trail with optimal probability and the minimum number of active words (bytes/nibbles/bits) for its input-output differences. Secondly, for each input-output differential pattern of these distinguishers, we utilize the SMT model to describe the differential propagation in the extended rounds and count the number of key bits involved in the key recovery phase. These experimental results guide us to identify advantageous distinguishers which lead to attack more rounds with a lower complexity.
- We apply this new technique to search for the advantageous distinguishers of WARP. With the observations in WARP, we provide some tips such as reducing constraints and reducing the search space to accelerate the search of differential trails. By using this model, we find the 19-round and 20-round advantageous distinguishers efficiently. Interestingly, the discovered 19-round advantageous distinguisher involves fewer master keys in the key recovery phase than the 18-round and 20-round distinguishers.
- We propose the first 25-round key recovery attacks on WARP. We notice that the strong clustering effect of the trails benefits from the high probability chain of the S-box. Hence, we improve the probability of a 19-round distinguisher from 2^{-132} to $2^{-116.92}$ by enumerating 34566 trails with probabilities 2^{-132} sharing the same input-output differences. Then, based on this distinguisher, we launch a 25-round key recovery attack by extending 3 rounds forward and 3 rounds backward. Similarly, we find that a 20-round advantageous distinguisher can also be used to perform an effective 25-round key recovery attack. The results cover 2 more rounds than the existing differential attacks, as are summarized in Table 1.

Outline. The structure of WARP, and its observations are introduced in Sect. 2. We provide details about the SAT and SMT models for searching the advantageous distinguishers in Sect. 3. In Sect. 4, we find some advantageous distinguishers and perform the 25 rounds key recovery attacks on round-reduced WARP. Sect. 5 concludes the paper.

2 Preliminaries

2.1 Specification of WARP

WARP was proposed by Banik et al. at SAC 2020 [1]. WARP is a 128-bit block cipher with a 128-bit key, and its round number is 41. The structure of WARP is a variant of Type-II GFN. The round function is shown in Fig. 6 of Appendix A. In each round function, three operations are performed in sequence, i.e. 4-bit Sbox, nibble XOR, and shuffle operation. The shuffle operation π in the 41st round is omitted.

- Sbox. To implement a lightweight threshold circuit, WARP applies the Sbox of Midori [2], as is illustrated in Table 7 of Appendix A.

Table 2: The notations

$X^r = X_0^r\|\|X_1^r\|\|\ldots\|\|X_{31}^r$	The 32-nibble input state of the $(r+1)$th round.
$Y^r = Y_0^r\|\|Y_1^r\|\|\ldots\|\|Y_{31}^r$	The 32-nibble input state of the shuffle operation in the $(r+1)$th round.
Mk	The 128-bit master key.
K^r	The 64-bit subkey in the $(r+1)$th round.
ΔX	The difference in the state X.

– Nibble XOR. The output nibble of the Sbox is xored to the subkey and the internal state.

– Shuffle operation. The shuffle operation π is worked on 32 nibbles, mapping the ith nibble to the $\pi(i)$th nibble. It is listed in Table 8 of Appendix A.

Key Schedule. The 128-bit master key Mk is divided into two 64-bit keys $Mk = Mk^0||Mk^1$. Let Mk_j^i denote one nibble, where $i \in \{0, 1\}$, $j \in \{0, 15\}$. i.e. $Mk^0 = Mk_0^0||Mk_1^0||\ldots||Mk_{15}^0$, $Mk^1 = Mk_0^1||Mk_1^1||\ldots||Mk_{15}^1$. The $(r+1)$th subkey is given as $K^r = Mk^{r \bmod 2}$, where $0 \leq r \leq 40$.

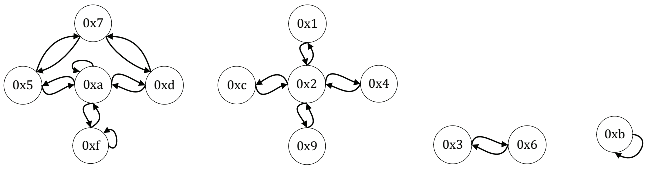
In addition, there are two nibbles of constants that are xored to the 1st and 3rd nibbles of the state. Since the differential attack does not consider the effect of constants, the introduction of constants is ignored. The details of WARP can be found in [1]. The notations are summarized in Table 2.

2.2 The Observations and Property of WARP

Since WARP is based on Type-II GFN, we make clear what GFN means. Given an even number $2q$ of blocks $(X_0^r, \ldots, X_{2i-1}^r)$, a set of cryptographic keyed function denoted as F_i^r, a permutation π which iterated R rounds over $2q$ elements, the $(r+1)$th round of the $2i$-branch Type-II GFN is defined as

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

where $0 \leq r < R$, $0 \leq i < q$. For such a structure, the cryptographic keyed function F_i^r usually consists of the Sbox and the XOR of a subkey K_i^r. For instance, TWINE [20], WARP. Therefore, the differential propagation of Sboxes is crucial to differential attacks. There are some observations and properties of WARP that can be used to improve search models or optimize key recovery attacks.



(a) 0xa (b) 0x2 (c) 0x3 (d) 0xb

Fig. 1: The high-probability chains centered at 0xa, 0x2, 0x3, 0xb of the Sbox

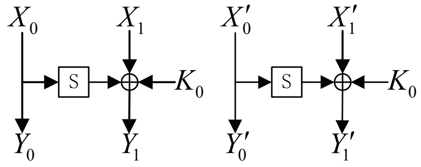


Fig. 2: A pair nibbles of the Feistel-subround

Observation 1 *([22]) The Sbox has a high-probability chain*

$$(\Delta S_0, \Delta S_1, \ldots, \Delta S_L)$$

if $\Delta S_i \xrightarrow{Sbox} \Delta S_{i+1}$ *is a high-probability transition for all* $i \in \{0, 1, \ldots, L\}$*, where* $0 < L$*. For the Sbox of* `WARP`*, there are several high-probability chains which have the iterative property illustrated in Fig. 1. For instance, in Fig. 1(a), the high-probability chain centered at* 0xa *of the Sbox is as follows.*

$$\left.\begin{array}{c} \text{0x5} \overset{Sbox}{\rightleftharpoons} \text{0xa} \\ \text{0xf} \overset{Sbox}{\rightleftharpoons} \text{0xf} \end{array}\right\} \overset{Sbox}{\rightleftharpoons} \text{0xa} \overset{Sbox}{\rightleftharpoons} \text{0xd} \overset{Sbox}{\rightleftharpoons} \text{0x7} \overset{Sbox}{\rightleftharpoons} \text{0x5}. \quad (1)$$

Property 1. ([21]) As illustrated in Fig. 2, the Feistel-subround performs on two nibbles, and the XOR 4-bit key is executed after the Sbox. Since the key K_0 has no influence on the differences ΔX_1 and ΔY_1, it allows partial encryption or decryption based on known differences $\Delta X_0||\Delta X_1$, ΔY_1 and checks if the given pairs $(X_0||X_1, X'_0||X'_1)$ are valid without guessing the key K_0. The same goes for the decryption direction. This property can be used to filter wrong pairs in the key recovery phase.

Observation 2 *When encrypting, we can detect whether the input pair is valid without guessing the key according to Property 1. A similar situation exists in the decryption direction. Therefore,*

- *for the 25-round key recovery attack based on a 19-round distinguisher, without guessing the subkey, the following 14 nibbles in the 1st and 25th rounds can be used to directly filter the wrong pairs.*

$$\Delta Y^0_{1,3,13,15,19,29} = \Delta X^{24}_{1,9,11,25,27} = \text{0x0}, \Delta X^{24}_{23,31} = \text{0xa}.$$

- *Similarly, for the 25-round differential attack by extending a 20 rounds distinguisher, there are the following 16 nibbles in the 1st and 25th rounds to directly check whether the pairs are right.*

$$\Delta Y^0_{21} = \text{0xa}, \Delta Y^0_{3,13,15,17,19,29,31} = \text{0x0},$$
$$\Delta X^{24}_{7} = \text{0x5}, \Delta X^{24}_{15} = \text{0xa}, \Delta X^{24}_{9,11,15,17,25,27} = \text{0x0}.$$

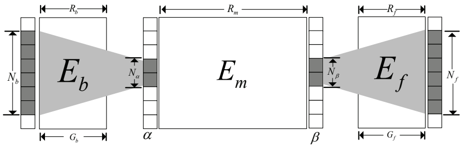


Fig. 3: Differential attack on block ciphers

3 Improved the Automated Evaluation Algorithm against Differential Attacks

To search for an advantageous distinguisher, Zong et al. [23] first found the initial set δ_{ini} containing the input-output differences of distinguishers with fewer involved key bits in the key recovery phase. Then, they deduced the advantageous differential from the initial set δ_{ini}. However, this method requires a lot of computational resources. In addition, for some ciphers, especially the Feistel-structured ciphers, the probability of a differential constructed from the input-output differences in δ_{ini} may not be optimal. To overcome this obstacle, we are motivated to improve the automated evaluation algorithm against differential attacks on block ciphers. We first introduce the basic strategy of differential attacks and then provide the SAT and SMT models used in the attack.

3.1 The Strategy Towards Advantageous Distinguishers

For the differential attack shown in Fig. 1, an R-round cipher E is decomposed into three consecutive keyed permutations $E = E_f \circ E_m \circ E_b$. The block/key size is n/k-bit. Generally, the analyst uses an R_m-round distinguisher dominated by a differential trail with optimal probability P_{opt} and launch the key recovery attack by extending R_b-round forward and R_f-round backward.

However, there are multiple factors that affect differential attacks, such as the number R_m of rounds and the probability of the distinguisher (α, β), the input and output differences of the distinguisher (the minimum number of active bits for the differences α, β is denoted as N_α, N_β, respectively), the number of key bits to be guessed in the key recovery phase (there are G_b/G_f key bits involved in the E_b/E_f part), and the number of active bits for the differences of the plaintexts and ciphertexts (similarly, the number of active bits for their differences are represented as N_b, N_f). These factors influence and restrict each other, e.g., the distinguisher with a minimum number of active bits for differences α, β can filter wrong pairs more effectively during data collection, so such a distinguisher has more advantages in the key recovery phase. Therefore, we need to trade off these factors and explore a longer attack with a lower complexity.

To execute longer attacks, we focus on such advantageous distinguishers (α, β): **(a)**. the trail with long rounds $(R_m)_{max}$ and optimal probability P_{opt}. **(b)**. the minimum number $(N_\alpha+N_\beta)_{min}$ of active words (bytes/nibbles/bits) for its input and output differences α, β. **(c)**. more rounds $(R_b + R_f)_{max}$ are extended by the distinguisher. **(d)**. fewer key bits are involved in the extended rounds (the minimum total number of the involved key bits is denoted as $(G_b + G_f)_{min}$). Currently, most works focus on constructing differential trails with P_{opt}. Few works have been devoted to deducing advantageous distinguishers for performing better key recovery attacks. In FSE 2022, Zong et al. searched for the key recovery-friendly distinguishers with high probability, **(c)** and **(d)**. Inspired by this work, we improve the automatic search algorithm against differential attack, and provide the SAT and SMT models to search for the advantageous distinguishers with **(a)**, **(b)**, **(c)** and **(d)**, so as to launch better key recovery attacks. Specifically, we adopt the following two-step strategy to achieve this goal.

- **Step 1.** We first utilize the SAT model to search for differential trail with $(R_m)_{max}$, P_{opt} and $(N_\alpha + N_\beta)_{min}$. All input-output differential patterns of such differential (α, β) are denoted as $(act_\alpha,\ act_\beta)$.
- **Step 2.** The SMT model is used to describe the differential propagation in the extended rounds and count the number of involved key bits in the key recovery phase. From each pattern $(act_\alpha,\ act_\beta)$ obtained in the **Step 1**, we apply the SMT model to determine the distinguisher with $(R_b + R_f)_{max}$ rounds and $(G_b + G_f)_{min}$ involved key bits.

In this way, we can find the advantageous distinguishers with **(a)**, **(b)**, **(c)** and **(d)**. Also, for the selected distinguisher, we employ the strong clustering effect of differential trails to improve the probability of the distinguisher, thereby achieving a better key recovery attack. The process of the key recovery attack summarized in [14] is as follows.

- Data collection. We construct $2^t = 2 \cdot 2^{-N_b} \frac{N_e}{P_{opt}}$ structures and each structure includes N_b bits traversed, where N_e is the expected number of right pairs.
- $(n - N_f)$ bits inactive differences of the ciphertexts are used to filter wrong pairs. There are $2^{t+2N_b-1-(n-N_f)}$ pairs remaining.
- Set $2^{G_b+G_f}$ empty counters for counting correct partial subkey values. Continue guessing to filter the remaining pairs to determine the correct candidate key bits. The time complexity of this process is abbreviated as σ.

If we get these parameters: $P_{opt}, N_b, N_f, G_b, G_f$, we can calculate that the data complexity is 2^{t+N_b}, and the time complexity is $2^t \cdot 2^{2N_b-1-(n-N_f)} \cdot \sigma \approx 2 \cdot \frac{N_e}{P_{opt}} \cdot 2^{N_b+N_f-n} \cdot \sigma$. The data/time complexity is constrained by the block/key size.

$$\begin{cases} 2^{t+N_b} < 2^n, \\ 2 \cdot \frac{N_e}{P_{opt}} \cdot 2^{N_b+N_f-n} \cdot \sigma < 2^k. \end{cases} \tag{2}$$

Since the round function of WARP performs on nibbles, for the above parameters, we mainly focus on the nibble-oriented number, such as $\frac{N_\alpha}{4}$, $\frac{N_\beta}{4}$, $\frac{N_b}{4}$, $\frac{N_f}{4}$, $\frac{G_b}{4}$, and $\frac{G_f}{4}$ nibbles.

Remarks. The Differential attack is affected by multiple factors, and the difficulty is how to trade-off these parameters to construct a longer attack. In this paper, we select an advantageous distinguisher considering more parameters, but the obtained distinguisher is not necessarily globally optimal. But the new technique is general and easy to implement. The distinguishers used in related research works have similar characteristics. For instance, some distinguishers [14] used to improve the boomerang attack on SKINNY [4] have fewer active nibbles for their input-output differences than the distinguishers utilized in [6].

3.2 SAT Model for Searching Advantageous Differentials

In this section, we provide the SAT model for searching all input-output differential patterns of the trails with more rounds $(R_m)_{max}$, optimal probability P_{opt} and $(\frac{N_\alpha+N_\beta}{4})_{min}$ nibbles. Specifically, the automation model is as follows.

- Step 1. The SAT model $\mathcal{M}_1$ is used to search for the differential trail with $(R_m)_{max}$ and P_{opt}. We construct this model to describe the differential propagation of the round function and set an objective function of the optimal probability P_{opt}. Besides, according to the property of WARP, we accelerate the search of the differential trail with P_{opt} by reducing the codomain of each nibble variable and reducing the number of constraints of the Sbox. After that, the solver returns a differential trial with the input and output differences α_0, β_0. We take the total number of active nibbles of the differences α_0, β_0 as the initial value $(\frac{N_{\alpha_0}+N_{\beta_0}}{4})_{ini}$.
- Step 2. The SAT model $\mathcal{M}_2$ is applied to search for the trails with $(R_m)_{max}$, P_{opt} and $(\frac{N_\alpha+N_\beta}{4})_{min}$. Based on the model $\mathcal{M}_1$ and the value $(\frac{N_{\alpha_0}+N_{\beta_0}}{4})_{ini}$, we add the constraints of the active nibbles of the differences α, β of the distinguisher and the objective function of minimizing the total number of active nibbles $(\frac{N_\alpha+N_\beta}{4})_{min}$. Then, we call the Cryptominisat5[1] solver until it returns a trail with P_{opt} and the minimum value $(\frac{N_\alpha+N_\beta}{4})_{min}$. The input-output differential pattern of the differences α, β is written as $act_{\alpha,i}$, $act_{\beta,i}$, where $act_{\alpha,i}, act_{\beta,i} \in \{0,1\}$, $0 \leq i < 32$. That is, if the ith nibble of the input differences α is active, then $act_{\alpha,i} = 1$, otherwise $act_{\alpha,i} = 0$. The activeness of the output differences β is also expressed in the same way.
- Step 3. We utilize the model $\mathcal{M}_2$ to enumerate all the input-output differential patterns of the distinguisher (α, β) with $(R_m)_{max}$, P_{opt} and $(\frac{N_\alpha+N_\beta}{4})_{min}$. For every discovered pattern, we add a blocking clause that contains only $act_{\alpha,0}, \ldots, act_{\alpha,31}$, $act_{\beta,0}, \ldots, act_{\beta,31}$ to the model, and solve it again until the model has no solution, which means we have found all the solutions.

For the constraints of basic operations such as XOR and branching, please refer to [16] for more details. Thereafter, we introduce the constraints and objective functions used in the model.

Constraints describing the codomain of a nibble. Let a nibble difference be $\Delta x_3||\Delta x_2||\Delta x_1||\Delta x_0$. Due to the iterative property of the high-probability chain of the Sbox, we can construct the differential trails with optimal probability by using the chain centered on 0xa given in Equation (1).

[1] https://github.com/msoos/cryptominisat

At this time, the codomain of the input and output differences for the Sbox is $\delta_N = \{0\mathrm{x}0, 0\mathrm{x}5, 0\mathrm{x}7, 0\mathrm{xa}, 0\mathrm{xd}, 0\mathrm{xf}\}$. In order to reduce the search space, we constrain the codomain of a nibble of all state variables in the round function to be δ_N. The following constraints represented as CN_1 are used to describe the codomain of a nibble.

$$\begin{cases} \Delta x_2 \vee \neg\Delta x_0 & = 1, \\ \neg\Delta x_2 \vee \Delta x_0 & = 1, \\ \Delta x_3 \vee \neg\Delta x_1 \vee \Delta x_0 & = 1, \\ \neg\Delta x_3 \vee \Delta x_1 \vee \Delta x_0 & = 1. \end{cases} \tag{3}$$

To compare the effectiveness of the model under different constraints, we also provide another constraint denoted as CN_2 that reduces the search space. From Observation 1, it can be seen that there is no hight-probability (2^{-2}) transition for the differential with the input or output differences 0x8. Therefore, to reduce the search space, we can limit the codomain of a nibble to not include 0x8, that is, $\Delta x_3||\Delta x_2||\Delta x_1||\Delta x_0 \neq 0\mathrm{x}8$. The corresponding constraint CN_2 is as follows.

$$\neg\Delta x_0 \vee \Delta x_1 \vee \Delta x_2 \vee \Delta x_3 = 1. \tag{4}$$

Constraints on the differential propagation with high-probability chain centered on 0xa **of the Sbox**. The high-probability chain centered on 0xa has iterative property, so we can construct the trails with optimal probability. Therefore, we describe these valid differential propagation in Equation (1) with CNF constraints. The input and output differences of the Sbox are represented as $\Delta X = \Delta x_3||\Delta x_2||\Delta x_1||\Delta x_0$ and $\Delta Y = \Delta y_3||\Delta y_2||\Delta y_1||\Delta y_0$ respectively. The differential probability written as DP belongs to $\{0, 2^{-2}, 1\}$. An additional variable $2 \cdot p_0$ describes the weight of the differential (Weight is the negative value of the binary logarithm of the differential probability).

$$p_0 = \begin{cases} 1, & \text{if } DP(\Delta X, \Delta Y) = 2^{-2}, \\ 0, & \text{if } DP(\Delta X, \Delta Y) = 1. \end{cases}$$

The differential propagation with probability dp_0 is denoted as $a_3||a_2||a_1||a_0 \rightarrow b_3||b_2||b_1||b_0$, then enumerate valid combinations of $(9 \cdot m)$-bit vectors

$$a_3^{(i)}||a_2^{(i)}||a_1^{(i)}||a_0^{(i)}||b_3^{(i)}||b_2^{(i)}||b_1^{(i)}||b_0^{(i)}||dp_0^{(i)}.$$

The original differential model of the Sbox consists of the following m clauses.

$$\bigvee_{j=0}^{3}(\Delta x_j \oplus a_j^{(i)}) \vee \bigvee_{j=0}^{3}(\Delta y_j \oplus b_j^{(i)}) \vee (p_0 \oplus dp_0^{(i)}) = 1, 0 \leq i \leq m-1.$$

Thereafter, the 9-bit Boolean function is defined as

$$f(\Delta X||\Delta Y||p_0) = \begin{cases} 1, & \text{if } \Delta X \rightarrow \Delta Y \text{ in Equation (1)}, \\ 0, & \text{otherwise}. \end{cases}$$

After that, we use Logic Friday[2] to simplify the CNF expressions. As a result, for all valid differentials with probability 2^{-2}, the constraints CS_{vd4} are made up of 27 clauses with 9 variables. Similarly, the high-probability chain centered on 0xa is described by the constraints $CS_{vd4,a}$ which include 16 clauses with 9 variables $(\Delta x_3, \ldots, \Delta x_0, \Delta y_3, \ldots, \Delta y_0, p_0)$.

Furthermore, to analyze the efficiency of searching for models under different constraints, we also provide different constraints for describing the differential propagation of Sboxes. Generally, for searching the trail with optimal probability, the constraints CS_{vd} used to describe all valid differentials of the Sbox are composed of 57 clauses with 11 variables $(\Delta x_3, \ldots, \Delta x_0, \Delta y_3, \ldots, \Delta y_0, p_2, p_1, p_0)$, where p_2, p_1, p_0 are used to represent the weight of the differential.

The objective function for searching the trails with optimal probability P_{opt}. For the r-round trail, the probability variable of the jth Sbox in the ith round is $p_0^{(i,j)}$ by using the constraints $CS_{vd4,a}$, where $0 \le i < r$, $0 \le j < 16$. The prospective value of differential weight is w_{DT} $(P_{opt} = 2^{-w_{DT}})$, then the objective function is

$$\sum_{i=0}^{r-1} \sum_{j=0}^{15} 2 \cdot p_0^{(i,j)} \le w_{DT}. \tag{5}$$

Constraints on the activeness of each nibble for the input-output differences α, β of a distinguisher. The 4-bit difference is denoted as Δx_3, Δx_2, Δx_1, Δx_0, an extra binary variable t is required to represent whether the nibble is active or not. If $\Delta x_3 || \Delta x_2 || \Delta x_1 || \Delta x_0 \neq$ 0x0, the nibble is active, that is, $t = 1$, otherwise, $t = 0$. The corresponding clauses are as follows.

$$\begin{cases} \neg \Delta x_0 \vee t & = 1, \\ \neg \Delta x_1 \vee t & = 1, \\ \neg \Delta x_2 \vee t & = 1, \\ \neg \Delta x_3 \vee t & = 1, \\ \Delta x_3 \vee \Delta x_2 \vee \Delta x_1 \vee \Delta x_0 \vee \neg t & = 1. \end{cases}$$

The objective function for minimizing the number of the active nibbles $(\frac{N_\alpha + N_\beta}{4})_{min}$ of the input-output differences α, β. For a differential (α, β), let the i-nibble of the input/output differences α / β be $act_{\alpha,i}/\ act_{\beta,i}$, where $0 \le i < 32$. We aim at minimizing the number of active nibbles for the differences α, β. The prospective value of $\frac{N_\alpha + N_\beta}{4}$ is $(\frac{N_\alpha + N_\beta}{4})_{min}$. Accordingly, the objective functions are as follows.

$$\sum_{i=0}^{31} (act_{\alpha,i} + act_{\beta,i}) \le (\frac{N_\alpha + N_\beta}{4})_{min}. \tag{6}$$

In some cases, to execute a better differential attack, it may be necessary to constrain the minimum number of active nibbles of the input and output differences

[2] https://web.archive.org/web/20131022021257/http:/www.sontrak.com/

of the distinguisher, respectively. Hence, assuming that the prospective value of $\frac{N_\alpha}{4}$ and $\frac{N_\beta}{4}$ is $(\frac{N_\alpha}{4})_{min}$ and $(\frac{N_\beta}{4})_{min}$, the objective function is

$$\sum_{i=0}^{31} act_{\alpha,i} \leq (\frac{N_\alpha}{4})_{min}, \sum_{i=0}^{31} act_{\beta,i} \leq (\frac{N_\beta}{4})_{min}. \tag{7}$$

The essential form of Equation (5), (6) and (7) is $\sum_{i=0}^{m-1} x_i \leq t$. We apply the sequential encoding approach [15] to convert this function into CNF formulas.

Finally, to illustrate the search efficiency of the SAT model, we compare the time consumed by searching of the 18-round trails with optimal probability 2^{-122} under different constraints. Our experiments deploy a server with Intel(R) Xeon(R) E5-2680 CPU*2 with 2.50GHZ, 256GB RAM. The search time for the SAT model with CS_{vd} is 12605 seconds. On this basis, after only replacing the constraints CS_{vd} by CS_{vd4}, the search time is 4513 seconds, Furthermore, in order to reduce the search space, the SAT model with CS_{vd4} and CN_2 takes 913 seconds. Lastly, the SAT model with $CS_{vd4,a}$ and CN_1 returns a valid solution after 558 seconds. As a result, we verified that the optimal probability for 20 rounds trail is 2^{-140}, which was not confirmed in [21].

3.3 SMT Model Oriented to Key Recovery

In this section, for WARP, we construct an SMT model to describe differential propagation in the extended rounds and count the number of the master key nibbles involved in the key recovery phase. Based on the distinguisher (α, β), we extend R_b rounds forward from input differences α and R_f rounds backward from output differences β. Then, we mark the subkeys that need to be guessed in the extended rounds, and count the total number $\frac{G_b+G_f}{4}$ of the involved master key nibbles by considering the key schedule of WARP.

There are two types of differences in the extended rounds: constant difference, and unknown difference. For each nibble variable, we introduce an extra variable x_{con} to describe whether the nibble difference ΔX is known. The constant difference and the unknown difference can be expressed as follows.

- Constant difference. Let a nibble difference ΔX be a constant δ, where $\delta \in \{0x0, 0x1, \ldots, 0xf\}$. It is written as $\Delta X = \delta, x_{con} = 0$.
- Unknown difference. If a nibble difference ΔX is unknown, it is denoted as $\Delta X = 0xf, x_{con} = 1$.

Constraints on the differential propagation in the E_b and E_f parts. Since the differential propagation in the E_b and E_f parts is similar, we only take Feistel-subround shown in Fig. 4 as an example to introduce the differential propagation in the E_b part. Given $\Delta Y_0, y_{con,0}$ and $\Delta Y_1, y_{con,1}$, $\Delta X_0, x_{con,0}$ and $\Delta X_1, x_{con,1}$ are derived. The details are as follows.

- For the left branch, the expressions describing the differential propagation are written as $\Delta X_0 = \Delta Y_0, x_{con,0} = y_{con,0}$.

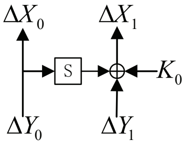


Fig. 4: The differential propagation of the Feistel-subround

- For the right branch, the input difference $\mathit{\Delta} X_1$ will be affected by the output difference $\mathit{\Delta} Y_0$. The relations can be described with the following equations.

$$\begin{cases} \mathit{\Delta} Y_0 & = 0\text{x}0 \Rightarrow \mathit{\Delta} X_1 = \mathit{\Delta} Y_1, x_{con,1} = y_{con,1}, \\ \mathit{\Delta} Y_0 & \neq 0\text{x}0 \Rightarrow \mathit{\Delta} X_1 = 0\text{xf}, x_{con,1} = 1. \end{cases}$$

Constraints describing the subkeys involved in the E_b and E_f parts. In [21], they proposed a 23-round key recovery attack by extending 2 rounds forward and 3 rounds backward the 18-round distinguisher. Therefore, we choose $R_b, R_f \in \{2, 3\}$ to deduce a better key recovery attack for a given distinguisher. Next, for `WARP`, we take $R_b = 3$ as an example to illustrate how to mark the subkey nibbles involved in the extended rounds.

- In the 3rd round, there is no need to guess the subkeys according to Property 1, then $K_i^2 = 0$, where $0 \leq i < 16$.
- The following subkey nibbles need to be guessed in the 2nd round are determined based on the nibbles that need to calculate the values in the 3rd round. $(0 \leq i < 16)$

$$K_i^1 = \begin{cases} 1, & \text{if } \mathit{\Delta} Y_{2i+1}^1 \neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

- The following subkey nibbles involved in the 1st round are determined based on the nibble values to be calculated in the 2nd and 3rd rounds. $(0 \leq i < 16$, $j = \lfloor \frac{\pi^{-1}(2i)}{2} \rfloor)$

$$K_j^0 = \begin{cases} 1, & \text{if } \mathit{\Delta} X_{2i+1}^1 \neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Constraints on counting the total number $\frac{G_b+G_f}{4}$ of the master key nibbles involved in the E_b and E_f parts. We obtain the corresponding relations between the subkeys and the master keys according to the key schedule of `WARP`. The 128-bit master key is divided into 32 nibbles $Mk = Mk_0^0||\ldots||Mk_{15}^0||Mk_0^1||\ldots||Mk_{15}^1$. If all subkeys K_i^r corresponding to the master key nibble $Mk_i^{r \bmod 2}$ do not need to be guessed, then $Mk_i^{r \bmod 2} = 0$, otherwise, $Mk_i^{r \bmod 2} = 1$, where $r \in [0, R_b) \cup [R_b + R_m, R_b + R_m + R_f)$, $0 \leq i < 16$.

Table 3: The experimental results for differential trails of WARP

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\#AS$	0	1	2	3	4	6	8	11	14	17	22	28	34	40	47	52	57	61	66	70
P_{opt}	1	2^{-2}	2^{-4}	2^{-6}	2^{-8}	2^{-12}	2^{-16}	2^{-22}	2^{-28}	2^{-34}	2^{-44}	2^{-56}	2^{-68}	2^{-80}	2^{-94}	2^{-104}	2^{-114}	2^{-122}	2^{-132}	2^{-140}
$(\frac{N_\alpha+N_\beta}{4})_{ran}$	2	3	4	5	6	8	10	13	16	17	22	28	24	26	29	21	14	18	21	19
$(\frac{N_\alpha+N_\beta}{4})_{min}$	2	3	4	5	6	8	10	11	14	17	22	22	24	23	29	21	14	18	15	19

Let $t = r \mod 2$, where $t \in \{0,1\}$. The expression is specified as follows.

$$Mk_i^t = \begin{cases} 0, \text{if } \sum_{r=0}^{R_b-1} K_i^r + \sum_{r=R_b+R_m}^{R_b+R_m+R_f-1} K_i^r = 0, \\ 1, \text{otherwise.} \end{cases}$$

The total number of the master key nibbles involved in the extended rounds is

$$\sum_{i=0}^{15} (Mk_i^0 + Mk_i^1) = \frac{G_b + G_f}{4}.$$

4 Differential Attack on WARP

In this section, we propose the 25-round key recovery attacks on WARP by extending a 19-round and a 20-round distinguisher, respectively. To find advantageous distinguishers, we obtain all input-output differential patterns of the trail with $(R_m)_{max}$, P_{opt} and $(\frac{N_\alpha+N_\beta}{4})_{min}$ by utilizing the SAT model. For each input-output differential pattern, we choose the input-output differences (α, β) belonging to the pattern to count the number of master keys involved in the extended rounds, and try to identify the input-output differential patterns of the distinguishers with fewer involved master keys. Eventually, we find a 19-round, a 20-round advantageous distinguisher which can be used to launch the 25-round key recovery attacks.

4.1 The Differential Friendly to Key Recovery Attacks

As far as we know, solvers like STP[3], Gurobi[4]. can be regarded as a black box that returns a valid solution according to the constructed model. Usually, there is only one objective function P_{opt} for the SAT model $\mathcal{M}_1$ given in Sect. 3.2 when

[3] https://github.com/stp/stp

[4] http://www.gurobi.com

Table 4: The input-output differential patterns of the differential trails with P_{opt} and $(\frac{N_\alpha+N_\beta}{4})_{min}$ of `WARP`

Round	Input pattern	Output pattern
18	$act^{18}_{\alpha_0}$ = 0b00000000111100010000110011010000 $act^{18}_{\alpha_1}$ = 0b00001100110100000000000000011110001	$act^{18}_{\beta_0}$ = 0b00100000010000001100000011000011000 $act^{18}_{\beta_1}$ = 0b00000110000011000000100000010000011
19	$act^{19}_{\alpha_0}$ = 0b11000000000001000000001000000110000 $act^{19}_{\alpha_1}$ = 0b000100000001100001100000000000010000	$act^{19}_{\beta_0}$ = 0b10010000000010010101100000101000000 $act^{19}_{\beta_1}$ = 0b10110000101000000100100000000010010
20	$act^{20}_{\alpha_0}$ = 0b00001100110100000000000000011110001 $act^{20}_{\alpha_1}$ = 0b00000000111100010000110011010000	$act^{20}_{\beta_0}$ = 0b10010000000010010101100000101000000 $act^{20}_{\beta_1}$ = 0b10110000101000000100100000000010010

Table 5: The total number of master keys involved in the extended rounds based on the differential of `WARP`

Differential Pattern	R_b(rounds)	R_f(rounds)	$(\frac{G_b+G_f}{4})$ nibbles
$act^{18}_\alpha, act^{18}_\beta$	2	3	18
$act^{19}_\alpha, act^{19}_\beta$	2	3	16
	3	2	15
	3	3	20
$act^{20}_\alpha, act^{20}_\beta$	2	3	17

searching for the trails. The solver returns a trail with P_{opt}. However, the number of active nibbles for their input and output differences is random. Then, we use the model $\mathcal{M}_2$ which add another objective function $(\frac{N_\alpha+N_\beta}{4})_{min}$ (Equation (6)) to minimize the total number of active nibbles for the input-output differences (α, β), and get a trail with P_{opt} and $(\frac{N_\alpha+N_\beta}{4})_{min}$.

With the help of the SAT model, we obtain the experimental results for the first 20 rounds shown in Table 3. $\#AS$ denotes the minimum number of active Sboxes for the trail. $(\frac{N_\alpha+N_\beta}{4})_{min}/(\frac{N_\alpha+N_\beta}{4})_{ran}$ represents the minimum/random number of active input-output nibbles of the trail with optimal probability P_{opt} by using the model $\mathcal{M}_2/\mathcal{M}_1$. For instance, the number $(\frac{N_\alpha+N_\beta}{4})_{ran}$ of a 19-round trail with P_{opt} returned by the model $\mathcal{M}_1$ is 21-nibble, and the minimum number $(\frac{N_\alpha+N_\beta}{4})_{min}$ searched by the model $\mathcal{M}_2$ is 15 nibbles. The latter filters wrong pairs more effectively during the data collection.

In Table 4, we get two input-output differential patterns of the 18/19/20-round trial by utilizing the model $\mathcal{M}_2$. As in [21], Teh et al. applied a 18-round distinguisher whose input-output differences belong to the pattern $(act^{18}_{\alpha_0}, act^{18}_{\beta_0})$ to perform a 23-round key recovery attack. Referring to this attack, we choose $R_b, R_f \in \{2, 3\}$. Then, for each input-output pattern shown in Table 4, we apply the SMT model to deduce the minimum number $(\frac{G_b+G_f}{4})_{min}$ of master keys involved in the extended rounds. The results show that the two input-output patterns of 18/19/20 rounds distinguishers involve the same number of master keys by extending the same rounds. As illustrated in Table 5, it can be seen that the 19-round distinguisher involves fewer master keys than the 18-round distinguisher in the key recovery phase. For instance, by extending $R_b = 2$, $R_f = 3$ rounds, the number $\frac{G_b+G_f}{4}$ of a 18-round, 19-round and 20-

round distinguisher is 18, 16 and 17 nibbles. When $R_b = R_f = 3$, the number of master keys involved in the 19-round distinguisher is 20-nibble. When $R_b = 2$, $R_f = 3$, the number for the 20-round distinguisher is 17-nibble. However, the optimal probability of the 19/20-round trail is $2^{-132}/2^{-140}$. So if we can improve the probability of the 19-round or the 20-round distinguisher, then it is possible to perform an efficient key recovery attack based on this distinguisher.

Motivated by the above observation and analysis, we apply the SAT model to search the trails sharing the same input and output differences. We utilize the clustering effect of the trails to improve the probability of the distinguishers. The results are listed in Table 9 of Appendix B. the symbol "‡" means that we have find some trails with optimal probability sharing the same input and output differences, but not all of them. When searching for differential trails with the same input and output differences, we focus on the trials with optimal probability. So compared to the result in [21], the clustering effect of the trails starts from the 8th round instead of the 10th round, and we obtain a higher probability of the distinguisher by exploiting the clustering effect of fewer trails. E.g. for the following 19-round differential $(\alpha^{19}, \beta^{19})$ given in [21],

$$\alpha^{19} = \mathrm{0x00005a00aa07000000000000a55a0005},$$
$$\beta^{19} = \mathrm{0xa0007005a00a5a0000a00a0005000050}.$$

they improved the probability from 2^{-132} to $2^{-118.07}$ by utilizing the clustering effect of 594111 trails. While we find another 19-round differential $(\alpha_0^{19}, \beta_0^{19})$ belonging to $(act_{\alpha_0}^{19}, act_{\beta_0}^{19})$, and use 34566 differential trails with 2^{-132} to improve the probability to $2^{-116.92}$.

$$\alpha_0^{19} = \mathrm{0xaa000000000a0000000a0000005a0000},$$
$$\beta_0^{19} = \mathrm{0x500a0000000a0050a05a000050a00000}.$$

The strong clustering effect of `WARP` benefits from the high-probability chain of the Sbox, which also makes it more effective to improve the probability of the distinguisher. In addition, it can be observed that the following 20-round differential $(\alpha_1^{20}, \beta_1^{20})$ given by Teh et al [21]. belongs to the pattern $(act_{\alpha_1}^{20}, act_{\beta_1}^{20})$, and its probability is improved from 2^{-140} to $2^{-122.71}$ using 545054 trails.

$$\alpha_1^{20} = \mathrm{0x00000000faa5000f00007500aa050000},$$
$$\beta_1^{20} = \mathrm{0xa05f0000a0500000a0050000000a00a0},$$

After obtaining the parameter information of P_{opt}, N_b, N_f and G_b, G_f, and substituting into Equation (2) in Sect. 3.1, we can estimate the complexity of the differential attack. Lastly, we execute an effective 25 rounds key recovery attack based on the 19-round, 20-round distinguisher, respectively.

4.2 The 25-round Key Recovery Attack on `WARP` Based on the 19-round Distinguisher

In this section, based on the 19-round distinguisher $(\alpha_0^{19}, \beta_0^{19})$ with probability $2^{-116.92}$, we perform the 25-round key recovery attack by extending 3 rounds at

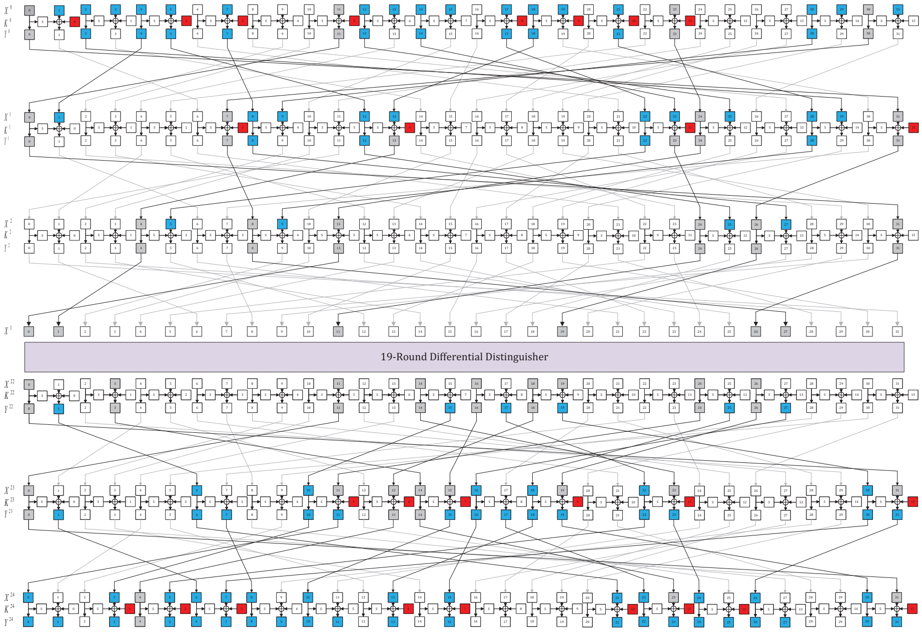


Fig. 5: The 25-round key recovery attack on WARP based on the 19-round differential. The nibble with zero difference is marked in white, the nibble with a nonzero difference is marked in grey, the nibble of unknown difference is marked in blue, and the subkeys involved in the extended rounds are marked in red.

the top and 3 rounds in the bottom. The 25-round key recovery attack is shown in Fig. 5. The attack procedures are as follows.

Data collection. For WARP, there is no whitening key at the input, we can construct structures at the position of X^0. By traversing $N_f = 17 \times 4$ bits values of $X^0_{1,\ldots,5,7,12,\ldots,15,17,18,19,21,28,29,31}$, and the remaining 60 bits are fixed differences, each structure includes 2^{135} pairs. Construct 2^t structures, we obtain 2^{t+135} pairs.

Key Recovery.

Step 1. By utilizing inactive bits in the output differences ΔX^{25}, we filter the wrong pairs. For each pair, we obtain the corresponding pair of ciphertexts $(X^{25}, X^{25'})$ by querying the oracle. There are 44 inactive bits for ΔX^{25}. In addition, according to Observation 2, 14 nibbles allow an immediate check of whether the given pair is valid. Then, the number of valid pairs would be reduced to $2^{t+135-44-56} = 2^{t+35}$. Let $2^m = 2^{t+35}$.

Step 2. We guess the value of a nibble master key Mk^0_3 and check the 8-bit difference $\Delta Y^1_9 = \text{0xa}$, $\Delta X^{23}_{31} = \text{0x5}$. The remaining 2^{m-8} pairs will participate in the following process. This guess-and-check procedure is repeated for all the 18 parts until all the 80-bit master keys are traversed. The time complexity

Table 6: Detailed computation of complexity for the 25-round key recovery attack based on the 19-round distinguisher

step	GMK	Condition on the difference	$\#\{Remaining\ pairs\}$	Time complexity
2.1	Mk_3^0	$\Delta Y_9^1 = $ 0xa,$\Delta X_{31}^{23} = $ 0x0	$2^m \cdot 2^{-8}$	$2 \cdot 2^m \cdot 2^4 \cdot 4$
2.2	Mk_2^0	$\Delta Y_{13}^1 = $ 0xa,$\Delta X_{23}^{23} = $ 0x5	$2^{m-8} \cdot 2^{-8}$	$2 \cdot 2^{m-8} \cdot 2^4 \cdot 2^4 \cdot 4$
2.3	Mk_{11}^0	$\Delta Y_{25}^1 = $ 0x0,$\Delta X_{15}^{23} = $ 0xa	$2^{m-16} \cdot 2^{-8}$	$2 \cdot 2^{m-16} \cdot 2^8 \cdot 2^4 \cdot 4$
2.4	Mk_{10}^0	$\Delta Y_{29}^1 = $ 0x0,$\Delta X_7^{23} = $ 0x0	$2^{m-24} \cdot 2^{-8}$	$2 \cdot 2^{m-24} \cdot 2^{12} \cdot 2^4 \cdot 4$
2.5	Mk_8^0	$\Delta Y_{23}^1 = $ 0x0	$2^{m-32} \cdot 2^{-4}$	$2 \cdot 2^{m-32} \cdot 2^{16} \cdot 2^4 \cdot 2$
2.6	Mk_{11}^1	$\Delta Y_{25}^2 = $ 0xa,$\Delta X_{15}^{22} = $ 0x0	$2^{m-36} \cdot 2^{-8}$	$2 \cdot 2^{m-36} \cdot 2^{20} \cdot 2^4 \cdot 8$
2.7	Mk_5^0	$\Delta Y_1^1 = $ 0x0	$2^{m-44} \cdot 2^{-4}$	$2 \cdot 2^{m-44} \cdot 2^{24} \cdot 2^4 \cdot 2$
2.8	Mk_{15}^0	$\Delta X_1^{23} = $ 0x0	$2^{m-48} \cdot 2^{-4}$	$2 \cdot 2^{m-48} \cdot 2^{28} \cdot 2^4 \cdot 2$
2.9	Mk_1^0	$\Delta X_{11}^{23} = $ 0x5	$2^{m-52} \cdot 2^{-4}$	$2 \cdot 2^{m-52} \cdot 2^{32} \cdot 2^4 \cdot 2$
2.10	Mk_7^0	$\Delta X_{17}^{23} = $ 0x0	$2^{m-56} \cdot 2^{-4}$	$2 \cdot 2^{m-56} \cdot 2^{36} \cdot 2^4 \cdot 2$
2.11	Mk_6^0	$\Delta X_{19}^{23} = $ 0xa	$2^{m-60} \cdot 2^{-4}$	$2 \cdot 2^{m-60} \cdot 2^{40} \cdot 2^4 \cdot 2$
2.12	Mk_6^1	$\Delta Y_5^2 = $ 0x0	$2^{m-64} \cdot 2^{-4}$	$2 \cdot 2^{m-64} \cdot 2^{44} \cdot 2^4 \cdot 4$
2.13	Mk_{15}^1	$\Delta X_1^{22} = $ 0x0	$2^{m-68} \cdot 2^{-4}$	$2 \cdot 2^{m-68} \cdot 2^{48} \cdot 2^4 \cdot 4$
2.14	Mk_7^1	$\Delta X_{17}^{22} = $ 0x0	$2^{m-72} \cdot 2^{-4}$	$2 \cdot 2^{m-72} \cdot 2^{52} \cdot 2^4 \cdot 4$
2.15	Mk_{12}^0	$\Delta X_{19}^{22} = $ 0xa	$2^{m-76} \cdot 2^{-4}$	$2 \cdot 2^{m-76} \cdot 2^{56} \cdot 2^4 \cdot 4$
2.16	Mk_5^1	$\Delta X_{25}^{22} = $ 0x0	$2^{m-80} \cdot 2^{-4}$	$2 \cdot 2^{m-80} \cdot 2^{60} \cdot 2^4 \cdot 4$
2.17	Mk_9^1	$\Delta X_{27}^{22} = $ 0x0	$2^{m-84} \cdot 2^{-4}$	$2 \cdot 2^{m-84} \cdot 2^{64} \cdot 2^4 \cdot 4$
2.18	Mk_9^0	$\Delta Y_{27}^2 = $ 0x0	$2^{m-88} \cdot 2^{-4}$	$2 \cdot 2^{m-88} \cdot 2^{68} \cdot 2^4 \cdot 4$
2.19	Mk_3^1,Mk_0^0	$\Delta Y_9^2 = $ 0x0	$2^{m-92} \cdot 2^{-4}$	$2 \cdot 2^{m-92} \cdot 2^{72} \cdot 2^4 \cdot 4$
Total				$2^{m+7.10}$

and the number of remaining pairs in each step are detailed in Table 6. Where "GMK" means that the nibble master key needs to be guessed.

Complexity Analysis. We set $N_e = 1$ pairs remaining for the right key guess, then construct $2^t = 2 \cdot 2^{-68} \cdot \frac{N_e}{p} \approx 2^{49.92}$ structures. The data complexity is $2^{t+68} = 2^{117.92}$. There are about $2^{m-96} = 2^{-11.08}$ pairs remaining for the wrong key guess. The memory complexity is $2^{49.92+68} + 2^{80} \cdot \frac{80}{128} = 2^{117.92}$ 128-bit blocks, and the time complexity is about $2^{t+68} \cdot \frac{2}{25} = 2^{t+68} \cdot 2^{-3.65} \approx 2^{114.27}$ 25-round encryptions.

4.3 The 25-round Key Recovery Attack on WARP Based on the 20-round Distinguisher

We launch a 25-round key recovery attack by extending the 20-round distinguisher $(\alpha_1^{20}, \beta_1^{20})$ with probability $2^{-122.71}$ given in Sect. 4.1. The whole attack details are demonstrated in Fig. 7 of Appendix B.

In the key recovery attack, we prepare 2^t structures and each structure includes 14 nibbles traversed. Then, we obtain 2^{t+111} pairs. For each pair, there are 11-nibbles fixed differences for the ciphertexts, and 16-nibble fixed differences in the 1st and 25th rounds (Observation 2). Thus, the remaining 2^{t+13} pairs will participate in the following processes. Let $2^m = 2^{t+13}$. For each pair, we repeat the guess-and-check procedure for the 17-nibble master key involved in the extended rounds. The time complexity and the number of remaining pairs in each step are detailed in Table 10 of Appendix B.

Complexity Analysis. For the right key guess, there are $N_e = 1$ pairs remaining. We collect $2^t = 2 \cdot 2^{-14 \cdot 4} \frac{N_e}{p} \approx 2^{67.71}$ structures. The data complexity is $2^{t+56} = 2^{123.71}$. There are about $2^{m-80} = 2^{-0.71}$ pairs remaining for the wrong key guess. The memory complexity is $2^{t+56} + 2^{68} \cdot \frac{68}{128} = 2^{123.71}$ 128-bit blocks, and the time complexity is about $2^{67.71+56} \cdot \frac{2}{25} \approx 2^{120.06}$ 25-round encryptions.

5 Conclusions

We provide an algorithm to search for the distinguishers that have advantages in both the distinguishing phase and the key recovery phase. This new technique is widely applicable and easy to implement for block ciphers. Taking WARP as an illustration, we propose the SAT model to search for the differential trail with optimal probability and a minimum number of active nibbles for its input-output differences. Subsequently, for each input and output differential pattern of these discovered distinguishers, we construct the SMT model to describe the differential propagation and amount the number of master key bits involved in the extended rounds. Later, we obtain some 19-round and 20-round advantageous distinguishers. Furthermore, we improve the probability of the distinguisher by utilizing the clustering effect of the differential trail. At last, we launch the 25-round key recovery attacks based on a 19-round and a 20-round distinguisher. The results cover 2 more rounds than the previous differential attack.

Acknowledgements. The authors would like to thank the reviews for their valuable comments. This work is supported by the National Natural Science Foundation of China (No. 61702537, No. 62172427).

References

1. Banik, S., Bao, Z., Isobe, T., Kubo, H., Liu, F., Minematsu, K., Sakamoto, K., Shibata, N., Shigeri, M.: WARP : Revisiting GFN for lightweight 128-bit block cipher. In: Dunkelman, O., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12804, pp. 535–564. Springer (2020), https://doi.org/10.1007/978-3-030-81652-0_21
2. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I-I. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015), https://doi.org/10.1007/978-3-662-48800-3_17
3. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 321–345. Springer (2017), https://doi.org/10.1007/978-3-319-66787-4_16

4. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5, https://doi.org/10.1007/978-3-662-53008-5_5
5. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer (1990), https://doi.org/10.1007/3-540-38424-3_1
6. Hadipour, H., Bagheri, N., Song, L.: Improved rectangle attacks on SKINNY and CRAFT. IACR Trans. Symmetric Cryptol. **2021**(2), 140–198 (2021). https://doi.org/10.46586/tosc.v2021.i2.140-198, https://doi.org/10.46586/tosc.v2021.i2.140-198
7. Hadipour, H., Eichlseder, M.: Integral cryptanalysis of WARP based on monomial prediction. IACR Trans. Symmetric Cryptol. **2022**(2), 92–112 (2022). https://doi.org/10.46586/tosc.v2022.i2.92-112, https://doi.org/10.46586/tosc.v2022.i2.92-112
8. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 161–185. Springer (2015), https://doi.org/10.1007/978-3-662-47989-6_8
9. Kumar, M., Yadav, T.: MILP based differential attack on round reduced WARP. In: Batina, L., Picek, S., Mondal, M. (eds.) Security, Privacy, and Applied Cryptography Engineering - 11th International Conference, SPACE 2021, Kolkata, India, December 10-13, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13162, pp. 42–59. Springer (2021). https://doi.org/10.1007/978-3-030-95085-9_3, https://doi.org/10.1007/978-3-030-95085-9_3
10. Lallemand, V., Minier, M., Rouquette, L.: Automatic search of rectangle attacks on feistel ciphers: Application to WARP. IACR Trans. Symmetric Cryptol. **2022**(2), 113–140 (2022). https://doi.org/10.46586/tosc.v2022.i2.113-140, https://doi.org/10.46586/tosc.v2022.i2.113-140
11. Liu, Y., Wang, Q., Rijmen, V.: Automatic search of linear trails in ARX with applications to SPECK and chaskey. In: Manulis, M., Sadeghi, A., Schneider, S.A. (eds.) Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9696, pp. 485–499. Springer (2016), https://doi.org/10.1007/978-3-319-39555-5_26
12. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993), https://doi.org/10.1007/3-540-48285-7_33
13. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C., Yung, M., Lin, D. (eds.) Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers. Lecture

Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2011), https://doi.org/10.1007/978-3-642-34704-7_5

14. Qin, L., Dong, X., Wang, X., Jia, K., Liu, Y.: Automated search oriented to key recovery on ciphers with linear key schedule applications to boomerangs in SKINNY and forkskinny. IACR Trans. Symmetric Cryptol. **2021**(2), 249–291 (2021), https://doi.org/10.46586/tosc.v2021.i2.249-291
15. Sinz, C.: Towards an optimal CNF encoding of boolean cardinality constraints. In: van Beek, P. (ed.) Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3709, pp. 827–831. Springer (2005), https://doi.org/10.1007/11564751_73
16. Sun, L., Wang, W., Wang, M.: Accelerating the search of differential and linear characteristics with the SAT method. IACR Trans. Symmetric Cryptol. **2021**(1), 269–315 (2021), https://doi.org/10.46586/tosc.v2021.i1.269-315
17. Sun, L., Wang, W., Wang, M.: Improved attacks on GIFT-64. In: AlTawy, R., Hülsing, A. (eds.) Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13203, pp. 246–265. Springer (2021), https://doi.org/10.1007/978-3-030-99277-4_12
18. Sun, S., Gérault, D., Lafourcade, P., Yang, Q., Todo, Y., Qiao, K., Hu, L.: Analysis of aes, skinny, and others with constraint programming. IACR Trans. Symmetric Cryptol. **2017**(1), 281–306 (2017), https://doi.org/10.13154/tosc.v2017.i1.281-306
19. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer (2014), https://doi.org/10.1007/978-3-662-45611-8_9
20. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: Twine: A lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7707, pp. 339–354. Springer (2012), https://doi.org/10.1007/978-3-642-35999-6_22
21. Teh, J.S., Biryukov, A.: Differential cryptanalysis of WARP. IACR Cryptol. ePrint Arch. p. 1641 (2021), https://eprint.iacr.org/2021/1641
22. Todo, Y., Sasaki, Y.: Designing s-boxes providing stronger security against differential cryptanalysis for ciphers using byte-wise XOR. In: AlTawy, R., Hülsing, A. (eds.) Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13203, pp. 179–199. Springer (2021), https://doi.org/10.1007/978-3-030-99277-4_9
23. Zong, R., Dong, X., Chen, H., Luo, Y., Wang, S., Li, Z.: Towards key-recovery-attack friendly distinguishers: Application to GIFT-128. IACR Trans. Symmetric Cryptol. **2021**(1), 156–184 (2021), https://doi.org/10.46586/tosc.v2021.i1.156-184

Appendix A: The Round Function, Sbox and Shuffle Operation of WARP

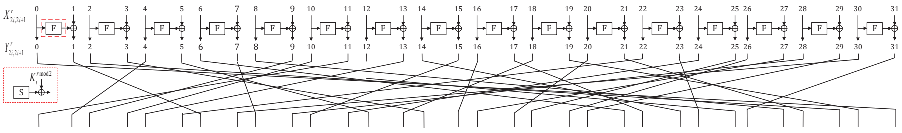


Fig. 6: The round function of WARP

Table 7: The 4-bit Sbox of WARP

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

Table 8: The shuffle operation on 32 nibbles of WARP

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

Appendix B: The Experimental Results on WARP

Table 9: The experimental results for the differentials of WARP

Round	Input difference	Output difference	P_{opt}	$\#Trail$	DP	Ref
8	0x0000000000aa00000000da00000a0000	0x0a0000000000a000f00a0a0000a00000	2^{-22}	4	2^{-20}	Sect. 4.1
9	0x0000da00000a00000000000000a50000	0xa00a00a007005050000050500000000a	2^{-28}	6	$2^{-26.68}$	Sect. 4.1
10	0x00000000210004000000002421000004	0x20040020010020200000202000000004	2^{-34}	4	2^{-32}	Sect. 4.1
	0x00000000ee000e00000000eeee00000e	0xe00e00e00e00e0e00000e0e00000000e	2^{-34}	7	$2^{-33.19}$	[21]
11	0x02120002000042001200000200001292	0x00002020000000004200100200400202 0	2^{-44}	4	2^{-42}	Sect. 4.1
	0x24000001000021210921000100002100	0x40020010020010100000101000000002	2^{-44}	7	$2^{-43.19}$	[21]
12	0x7500a5da0a0000000005da00000a0a00	0x00a000000fa000a0fa00000d0000aaf0	2^{-56}	24	$2^{-52.68}$	Sect. 4.1
	0x04240004000021002400000400002121	0x01000404212020010020212020200020	2^{-56}	5	$2^{55.42}$	[21]
13	0x00024200000202004200929202000000	0x00000004202001002024000401200021	2^{-68}	2240	$2^{-62.15}$	Sect. 4.1
	0x12001212020000000002420000020200	0x20210004042000210000000120200200	2^{-68}	1600	$2^{-62.37}$	[21]
14	0xaa00575a0a000000000ada0000070a00	0x005000a0a0005000a00a500a50000a0a	2^{-80}	1824	$2^{-70.38}$	Sect. 4.1
	0x000c240000010100210029210100000 0	0x2024200c20000c042024002020010000	2^{-80}	21528	$2^{-72.14}$	[21]
15	0xaa00aa7505000000000a5700000a0500	0x005050005000a0aa0a0057505a5a5a50	2^{-94}	632	$2^{-84.70}$	Sect. 4.1
	0x0005750000050a00a500a5a50a000000	0x0aa0a7505a555aa00070000a500a005a	2^{-94}	497248	$2^{-85.54}$	[21]
16	0x0000aa00aa0a0000000000005a5a000a	0x500a55000aa05a000000a000a00000a0	2^{-104}	13581‡	$2^{-90.27}$	Sect. 4.1
	0x0000a500a50a000000000000aaa5000a	0xa005a50005a057000000a000a00000a0	2^{-104}	800152	$2^{-90.52}$	[21]
17	0xff000000000a0000000a000000aa0000	0x000005a0000d700000700000a00000a5	2^{-114}	13280‡	$2^{-100.30}$	Sect. 4.1
	0x00070000005a000057000000000a0000	0x00a00000a00000aa000005500005a000	2^{-114}	734494	$2^{-95.66}$	[21]
18	0x0000000057aa000a0000a500fa0a0000	0x00a00000a00000a500000aa00005a000	2^{-122}	626723	$2^{-104.62}$	[21]
19	0xaa000000000a0000000a0000005a0000	0x500a0000000a0050a05a000050a00000	2^{-132}	34566‡	$2^{-116.92}$	Sect. 4.1
	0x00005a00aa07000000000000a55a0005	0xa0007005a00a5a0000a00a0005000050	2^{-132}	594111	$2^{-118.07}$	[21]
20	0x00000000faa5000f00007500aa050000	0xa05f0000a0500000a0050000000a00a0	2^{-140}	545054	$2^{-122.71}$	[21]

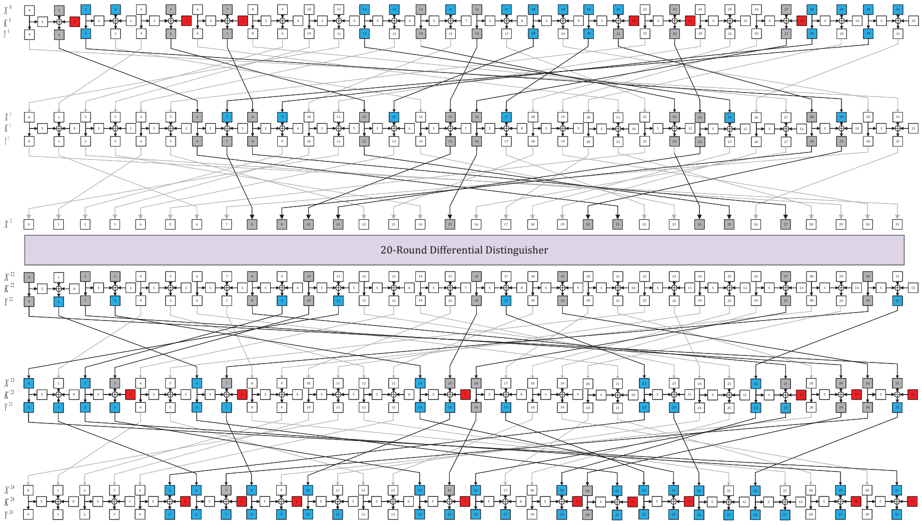


Fig. 7: The 25-round key recovery attack on `WARP` based on the 20-round differential. The nibble of zero difference is marked in white, the nibble with a nonzero difference is marked in grey, the nibble of unknown difference is marked in blue, and the subkeys involved in the extended rounds are marked in red.

Table 10: The Detailed computation of complexity for the 25-round key recovery attack based on the 20-round distinguisher

step	GMK	Condition on the difference	$\#\{Remaining\ pairs\}$	Time complexity
1	Mk_3^0	$\Delta Y_9^1 = 0\text{x}0, \Delta X_{31}^{23} = 0\text{xa}$	$2^m \cdot 2^{-8}$	$2 \cdot 2^m \cdot 2^4 \cdot 4$
2	Mk_2^0	$\Delta Y_{13}^1 = 0\text{x}0, \Delta X_{23}^{23} = 0\text{x}0$	$2^{m-8} \cdot 2^{-8}$	$2 \cdot 2^{m-8} \cdot 2^4 \cdot 2^4 \cdot 4$
3	Mk_{10}^0	$\Delta Y_{29}^1 = 0\text{x}7, \Delta X_6^{23} = 0\text{xa}$	$2^{m-16} \cdot 2^{-8}$	$2 \cdot 2^{m-16} \cdot 2^8 \cdot 2^4 \cdot 4$
4	Mk_{11}^0	$\Delta Y_{25}^1 = 0\text{x}0, \Delta X_{15}^{23} = 0\text{xa}$	$2^{m-24} \cdot 2^{-8}$	$2 \cdot 2^{m-24} \cdot 2^{12} \cdot 2^4 \cdot 2$
5	Mk_0^0	$\Delta Y_7^1 = 0\text{xf}$	$2^{m-32} \cdot 2^{-4}$	$2 \cdot 2^{m-32} \cdot 2^{16} \cdot 2^4 \cdot 2$
6	Mk_{13}^0	$\Delta Y_{17}^1 = 0\text{x}0$	$2^{m-36} \cdot 2^{-4}$	$2 \cdot 2^{m-36} \cdot 2^{20} \cdot 2^4 \cdot 2$
7	Mk_{15}^0	$\Delta X_1^{23} = 0\text{xf}$	$2^{m-40} \cdot 2^{-4}$	$2 \cdot 2^{m-40} \cdot 2^{24} \cdot 2^4 \cdot 2$
8	Mk_{14}^0	$\Delta X_3^{23} = 0\text{xf}$	$2^{m-44} \cdot 2^{-4}$	$2 \cdot 2^{m-44} \cdot 2^{28} \cdot 2^4 \cdot 2$
9	MK_7^0	$\Delta X_{17}^{23} = 0\text{x}0,\ \Delta Y_9^1 = 0\text{x}0$	$2^{m-48} \cdot 2^{-4}$	$2 \cdot 2^{m-48} \cdot 2^{32} \cdot 2^4 \cdot 2$
10	Mk_9^0	$\Delta X_{27}^{23} = 0\text{xa}$	$2^{m-52} \cdot 2^{-4}$	$2 \cdot 2^{m-52} \cdot 2^{36} \cdot 2^4 \cdot 2$
11	Mk_{15}^1	$\Delta X_1^{22} = 0\text{x}0$	$2^{m-56} \cdot 2^{-4}$	$2 \cdot 2^{m-56} \cdot 2^{40} \cdot 2^4 \cdot 4$
12	Mk_{13}^1	$\Delta X_9^{22} = 0\text{x}0$	$2^{m-60} \cdot 2^{-4}$	$2 \cdot 2^{m-60} \cdot 2^{44} \cdot 2^4 \cdot 4$
13	Mk_1^1	$\Delta X_{11}^{22} = 0\text{x}0$	$2^{m-64} \cdot 2^{-4}$	$2 \cdot 2^{m-64} \cdot 2^{48} \cdot 2^4 \cdot 4$
14	Mk_7^1	$\Delta X_{17}^{22} = 0\text{x}0$	$2^{m-68} \cdot 2^{-4}$	$2 \cdot 2^{m-68} \cdot 2^{52} \cdot 2^4 \cdot 4$
15	Mk_3^1	$\Delta X_{31}^{22} = 0\text{x}0$	$2^{m-72} \cdot 2^{-4}$	$2 \cdot 2^{m-72} \cdot 2^{56} \cdot 2^4 \cdot 4$
16	Mk_{14}^1,Mk_4^0	$\Delta X_3^{22} = 0\text{xf}$	$2^{m-76} \cdot 2^{-4}$	$2 \cdot 2^{m-76} \cdot 2^{60} \cdot 2^8 \cdot 4$
Total				$2^{m+7.09}$