SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2019, Vol. 62 032111:1–032111:15
https://doi.org/10.1007/s11432-018-9658-0

Automatic search method for multiple differentials and its application on MANTIS

Shiyao CHEN[1], Ru LIU[2], Tingting CUI[3] & Meiqin WANG[1*]

[1]*Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan* 250100*, China;*
[2]*Huawei Digital Technologies (Suzhou) Co.,Ltd., Suzhou* 215123*, China;*
[3]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou* 310018*, China*

Received 4 September 2018/Accepted 12 November 2018/Published online 29 January 2019

Abstract Multiple differential cryptanalysis is one of the extensions of classic differential cryptanalysis. In this paper, we present a generic automatic search method for clustering multiple differentials on a target block cipher. Our search method has two steps. Firstly, the sets of input and output differences will be determined. With these sets, we get different multiple differentials. Then for each one of these multiple differentials, we enumerate and record all satisfied differential trails, which leads to a more accurate evaluation of the multiple differentials distinguisher. Among these different multiple differentials distinguishers, we can choose the best one for key recovery attack. We demonstrate our search method by applying it on the part of differentials of the block cipher MANTIS. As a result, we find a new 10-round multiple differentials distinguisher with probability $2^{-55.98}$ and an 11-round multiple differentials distinguisher with probability $2^{-63.71}$, which is the longest distinguisher for MANTIS so far as we know. This new 10-round distinguisher can lead to a better signal-to-noise ratio, so we derive an improved key recovery attack on MANTIS-6 with the complexity of about $2^{51.79}$ chosen-plaintext queries, $2^{51.91}$ encryptions and data-time product $2^{103.70}$, which is better than the previous best one with data-time product $2^{110.61}$. Aiming at exploring the gap between the performance of multiple differential attack and the security margin on MANTIS, we also use the 11-round distinguisher to derive a key recovery attack on MANTIS-7 with the complexity of about $2^{61.86}$ chosen-plaintext queries, $2^{102.92}$ encryptions and data-time product $2^{164.78}$. It does not threat the security of full version MANTIS (MANTIS-7) since the security bound of data-time product claimed by the designers is 2^{126}.

Keywords multiple differential cryptanalysis, structure attack, automatic search, MANTIS, MILP

Citation Chen S Y, Liu R, Cui T T, et al. Automatic search method for multiple differentials and its application on MANTIS. Sci China Inf Sci, 2019, 62(3): 032111, https://doi.org/10.1007/s11432-018-9658-0

1 Introduction

Differential cryptanalysis [1] is a classic cryptanalytic method, based on which many cryptanalytic techniques have been developed. One of them is the multiple differential cryptanalysis, proposed by Blondeau et al. [2] at FSE 2011. Its main improvement is using many differentials to reduce the data complexity. Later, at FSE 2013, Wang et al. [3] proposed structure attack, which is a subclass of multiple differential attack in general. The structure attack uses multiple input differences and a single output difference. Wang et al. found that multiple output differences may not be better than the single output difference according to the complexities of data and time on some block ciphers such as PRESENT [4].

Traditionally, there is no explicit rule to direct how to cluster multiple differentials so as to mount the final distinguisher that is used in key recovery attack. In order to identify a distinguisher with

* Corresponding author (email: mqwang@sdu.edu.cn)

Table 1 Summary of attacks on MANTIS

Target version	Data (D)	Time (T)	$D \times T$	Source
MANTIS-5	2^{28}	2^{38}	2^{66}	Ref. [5]
MANTIS-6	$2^{55.09}$	$2^{55.52}$	$2^{110.61}$	Ref. [6]
MANTIS-6	$2^{51.79}$	$2^{51.91}$	$2^{103.70}$	Section 4
MANTIS-7	$2^{61.86}$	$2^{102.92}$	$2^{164.78}$	Section 5

high probability, we need to cluster differential characteristics with high probability as many as possible. Based on this underlying rule, the clustered differential characteristics with high probability may be constrained with some specific properties, such as the active S-box number and branch number of linear layer. These properties do let us find many good differential characteristics efficiently, but it also let us ignore some differential characteristics with high probability, which may lead to the better multiple differentials distinguisher for key recovery attack.

We can observe the impact of these restrictions on clustering multiple differentials by comparing two previous differential attacks [5, 6] on MANTIS, which is a tweakable block cipher proposed by Beierle et al. [7] at CRYPTO 2016. Encouraged to attack the aggressive versions by the designers, a practical attack on MANTIS-5 was given by Dobraunig et al. [5] with 2^{28} chosen-plaintexts and about 2^{38} encryptions, which used a 9-round multiple differentials distinguisher. This 9-round distinguisher was found by hand and simple to be evaluated. Dobraunig et al. started with a differential characteristic and just found the similar characteristics that could be clustered. Later, Eichlseder et al. [6] derived a key recovery attack on MANTIS-6 with $2^{55.09}$ chosen-plaintexts and $2^{55.52}$ encryptions, which used a 10-round multiple differentials distinguisher. The data-time product is $2^{110.61}$ below the designers' bound 2^{126}. Different from the previous attack, Eichlseder et al. generalized the clustering approach from the attack on MANTIS-5. They started from a truncated differential characteristic and considered all compatible differential characteristics, which relaxed some restrictions on the clustered differentials. By clustering more characteristics, they got a longer distinguisher to attack more rounds.

Concentrating on clustering multiple differentials, Eichlseder et al. [6] proposed a general method for finding and evaluating the multiple differentials on target block cipher, but it still has some restrictions on these differentials during the clustering process. Taking MANTIS as an example, it will forbid the branch number > 4. Hence, it may loose some differentials with high probability during the clustering process. So, we want to find a way to relax these restrictions during the differentials clustering process, even just a part of the differentials. However, there are some problems when relaxing the restrictions on clustering multiple differentials: how to search out different kinds of differential characteristics with high probability, how to select the differentials to be used in the final distinguisher and how to evaluate the final probability.

Our contributions. To deal with the problems above, we firstly propose a generic automatic search method for clustering multiple differentials. With this new search method, we can get different sets of input and output differences as many as possible and we classify these sets according to their active patterns of input and output differences. Then under each set of input and output differences, we enumerate all satisfied differential characteristics. At the same time, the detailed distribution of these satisfied differential trails is recorded, hence leading to a more accurate and easier evaluation of the multiple differentials distinguisher. Finally, we can choose the best distinguisher for key recovery attack. We also give an improved enumerating algorithm as a part of our new automatic search method.

Secondly, based on the inner structure in [6], we apply our automatic search method to find multiple differentials on some rounds of MANTIS. We find a new 10-round multiple differentials distinguisher and give an improved key recovery attack on MANTIS-6. Then we find an 11-round multiple differentials distinguisher as well, which is the longest distinguisher for MANTIS so far as we know, and we derive a key recovery attack to explore the security margin of full version MANTIS against multiple differential attack. All results are shown in Table 1.

Outline. This paper is organized as follows. In Section 2, we give a brief description of MANTIS and introduce several lemmas related to MILP technique. Then some notations used in this paper are

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

Figure 1 SubCells (S).

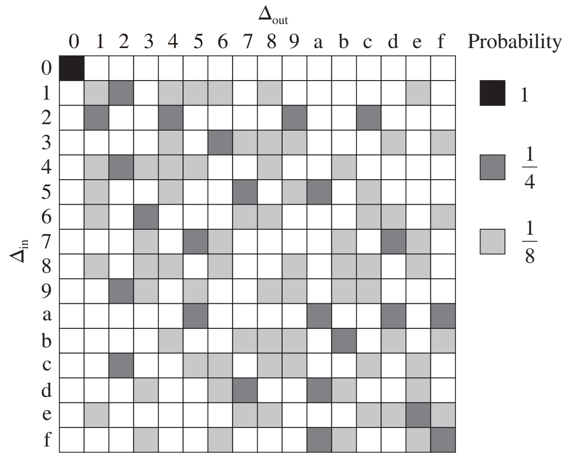


Figure 2 DDT of MANTIS.

given. In Section 3, the new automatic search method for clustering multiple differentials is presented and applied on MANTIS. Based on the two multiple differentials distinguishers we find, we give an improved key recovery attack on MANTIS-6 in Section 4 and explore the security margin of full version MANTIS in Section 5. In Section 6, we conclude the paper.

2 Preliminary

2.1 The brief description of MANTIS

MANTIS is a tweakable block cipher proposed by Beierle et al. [7] at CRYPTO 2016. It has several versions MANTIS-r according to the number of total rounds $2r$. The message block M, tweak T, and secret key K of each version are 64, 64, and 128 bits, respectively. What's more, every 64-bit internal state can be regarded as a 4×4 matrix with sixteen 4-bit nibbles N_i ($0 \leqslant i \leqslant 15$), where (N_0, N_1, N_2, N_3) is the first row. The key schedule of MANTIS is simple, the master key is divided into two 64-bit k_0 and k_1. Then k_0 and $k_0' = (k_0 \ggg 1) + (k_0 \gg 63)$ are used as whitening keys, k_1 is used as subkey in each round function.

Similar to PRINCE [8], MANTIS-r is composed by r forward rounds R_i ($i = 1, \dots, r$) and r backward rounds R_i^{-1} ($i = r+1, \dots, 2r$), and an unkeyed inner layer $S \circ M \circ S$ between the r-th and $(r+1)$-th round. The round function R_i and R_i^{-1} are shown in (1), which are made up of five components and described as follows. For more details about MANTIS, we refer to [7].

$$\begin{aligned} R_i &= M \circ P \circ A_i \circ C_i \circ S, \\ R_i^{-1} &= S \circ C_i \circ A_i \circ P^{-1} \circ M. \end{aligned} \tag{1}$$

• SubCells (S). The involutive S-box S given in Figure 1 is applied to each nibble of the state. The differential distribution table (DDT) is shown in Figure 2.

• AddTweakey$_i$ (A) and AddConstant$_i$ (C). Add the round constant C_i, round tweakey state $h^i(T)\oplus k_1$ (for R_i) or $h^i(T) \oplus k_1 + \alpha$ (for R_i^{-1}), where α is the constant. The tweakey update function h simply permutes the order of nibbles as specified in Figure 3.

• PermuteCells (P). All nibbles of the state are permuted by P, specified in Figure 4.

• MixColumns (M). Multiply each column of the state with an involutive near-MDS matrix M, specified in Figure 5.

2.2 Sun et al.'s MILP-based automatic search model

Here, we briefly recall the automatic differential search method based on MILP (mixed integer linear

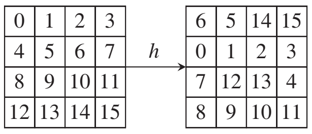


Figure 3 Update function h.

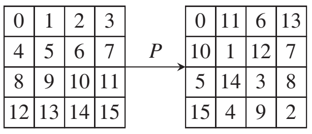


Figure 4 PermuteCells (P).

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure 5 MixColumns (M).

programming) by Sun et al. in [9, 10].

Constraints of subsets in $\{0,1\}^n \subseteq \mathbb{R}^n$. For a subset $X \subseteq \mathbb{R}^n$, the convex hull conv(X) is defined as the smallest convex set in $\mathbb{R}^n$ containing X. In Lemma 1, we describe the H-representation.

Lemma 1. The set of all solutions of the following system of (in)equalities

$$\begin{cases} \lambda_{0,0}x_0 + \cdots + \lambda_{0,n-1}x_{n-1} + \lambda_{0,n} \geqslant 0, \\ \qquad\qquad \vdots \\ \lambda_{i,0}x_0 + \cdots + \lambda_{i,n-1}x_{n-1} + \lambda_{i,n} \geqslant 0, \\ \qquad\qquad \vdots \end{cases} \tag{2}$$

is the convex, where $\lambda_{i,j}$ is fixed as a real number. For any subset $X \subseteq \mathbb{R}^n$ with finite discrete points, there exists a system $H_{\text{conv}}(X)$ of linear inequalities of the form like (2) such that $\text{Sol}(H_{\text{conv}}(X)) = \text{conv}(X)$, and we call $H_{\text{conv}}(X)$ the H-representation of conv(X).

Constraints of differential model. According to Lemma 1, all propagation pattern of difference through every component such as S-box, XOR and Branch operations can be described by a set of linear inequalities, as long as input and output differences are regarded as variables. By combining the inequalities of each component of a target cipher, we can describe the differential propagation behaviors precisely. As a result, each solution is a differential characteristic. By setting the objective function such as the total number of active S-boxes, the model can be solved by any MILP optimizer like Gurobi [11].

Constraint of removing a solution. Assume x is a solution of the MILP model $\tilde{M}$, if all solutions in model $\tilde{M}$ except x satisfy the constraint $l^{(x)}$. Then, we can add $l^{(x)}$ to model $\tilde{M}$ to remove the solution x according to Lemma 2, which is used in [9] and firstly proposed by Balas et al. [12].

Lemma 2. For a given 0-1 vector $\sigma = (\sigma_0, \ldots, \sigma_{n-1}) \in \{0,1\}^n \subseteq \mathbb{R}^n$. All 0-1 vectors except σ satisfy the constraint $\sum_{i=0}^{n-1}[\sigma_i + (-1)^{\sigma_i}x_i] \geqslant 1$.

2.3 Notations

- $|\mathbb{S}|$: the size of the set $\mathbb{S}$.
- χ^L: all input differences on plaintext.
- $\langle\chi^L\rangle$: the linear span generated by χ^L.
- l: the size of the best key candidates list.
- β: the filtering probability on ciphertext.
- δ: the difference of one nibble in MANTIS.
- Δ: the difference of one state in MANTIS.
- Δ_{in}: the input difference of a r-round differential characteristic.
- Δ_{out}: the output difference of a r-round differential characteristic.

- δ_i: the i-th nibble of the state in MANTIS.
- $\delta_{i\oplus j\oplus k}$: the XORed value of the i, j, k-th nibbles of the state in MANTIS.
- $\{0,1,2,3,4,5,6,7,8,9,\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d},\mathrm{e},\mathrm{f}\}$: the state differences representation (arranged by nibble's index $N_0, N_1, N_2, N_3, \ldots, N_{15}$) in MANTIS.

3 New method to search multiple differentials and its application on MANTIS

In this section, we propose a new automatic search method for clustering multiple differentials, consisting of two steps. The first step is to get the sets of multiple input and output differences for multiple differentials. Then in the second step, with the sets of multiple input and output differences obtained in the first step, we construct corresponding multiple differentials search model and enumerate out all satisfied differential characteristics from this model. During this step, we need to record all gained differential characteristics to help us cluster and precisely evaluate the final distinguisher that is used for key recovery attack.

3.1 Determing the sets of input and output differences

For multiple differential cryptanalysis, it is not easy to determine which sets of input and output differences can lead to the final best multiple differentials for key recovery attack. In other words, it is difficult to cluster the best multiple differentials distinguisher for the target cipher. However, in general, the best multiple differentials include at least one best differential characteristic. Inspired by this fact, we propose a new method to search multiple differentials. Before introducing our search algorithm, we give the definition of Pattern as follows.

Definition 1. Assume Δ is a difference state, which can be represented on nibble-level as $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{n-1}\}$, then $\bar{\Delta} = \{\bar{\delta_0}, \bar{\delta_1}, \ldots, \bar{\delta}_{n-1}\}$ is called the Pattern of Δ, where

$$\bar{\delta_i} = \begin{cases} 0, & \text{if } \delta_i = 0, \\ 1, & \text{otherwise.} \end{cases}$$

With the definition of Pattern, we illustrate the basic idea of our new search method. Firstly, we search out enough differential characteristics with best probability, then store the pair of Patterns of input and output differences for each characteristic. Then, due to the little contribution of the differential with low probability on multiple differentials, we only focus on the differentials with high probability under each pair of Patterns. As a result, their corresponding sets of input and output differences only involve part of difference values on active nibbles. Thus, under each pair of Patterns, we search out sufficient differentials with high probability to get the specific sets of input and output differences, which lead to a multiple differentials distinguisher. The whole detailed process is shown in Algorithm 1.

In Algorithm 1, N_1 different best differentials are searched out to deduce enough expected Patterns $(\bar{\Delta}_{\mathrm{in}}, \bar{\Delta}_{\mathrm{out}})$ at first, and we store such Patterns into the set $\mathrm{Act}_{\Delta_{\mathrm{in}},\Delta_{\mathrm{out}}}$. Then for each pair of Patterns in $\mathrm{Act}_{\Delta_{\mathrm{in}},\Delta_{\mathrm{out}}}$, we search out N_2 different best differentials and store their actual input and output differences $(\Delta_{\mathrm{in}}, \Delta_{\mathrm{out}})$ into the corresponding set $\mathrm{Dist}_{(\bar{\Delta}_{\mathrm{in}},\bar{\Delta}_{\mathrm{out}})}$. As a result, $\mathrm{Dist}_{(\bar{\Delta}_{\mathrm{in}},\bar{\Delta}_{\mathrm{out}})}$ leads to a multiple differentials distinguisher for the target cipher. Actually, we can obtain $|\mathrm{Act}_{\Delta_{\mathrm{in}},\Delta_{\mathrm{out}}}|$ multiple differentials and choose the most suitable one to derive the key recovery attack.

In order to explain the relationship among Δ_{in}, $\bar{\Delta}_{\mathrm{in}}$ and $\mathrm{Dist}_{\bar{\Delta}_{\mathrm{in}}}$ more clearly, we present an example as follows. If an input difference of MANTIS is

$$\Delta_{\mathrm{in}} = \{\mathrm{a},0,0,0,\mathrm{f},0,0,0,0,0,0,0,0,0,0,0\},$$

then its Pattern is

$$\bar{\Delta}_{\mathrm{in}} = \{1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\},$$

Algorithm 1 New method to cluster multiple differentials

```
Input: $\mathcal{M}$: $r$-round MILP differential model of the target cipher.
    $N_1$: the number of sample differential characteristics to find enough Patterns.
    $N_2$: the number of sample differential characteristics under each pair of Patterns.
Output: Dist $= \{\mathrm{Dist}_{(\alpha,\gamma)} | (\alpha,\gamma) \in \mathrm{Act}_{\Delta_{\rm in},\Delta_{\rm out}}\}$: $|\mathrm{Act}_{\Delta_{\rm in},\Delta_{\rm out}}|$ multiple differentials.
 1: $\mathcal{M}_1 = \mathcal{M}$;
 2: sol_count = 0;
 3: $\mathrm{Act}_{\Delta_{\rm in},\Delta_{\rm out}}$ := None;
 4: while sol_count $< N_1$ do
 5:     curr_sol = Solve the model $\mathcal{M}_1$ to get the best differential characteristic;
 6:     Extract $(\Delta_{\rm in}, \Delta_{\rm out})$ from curr_sol and get $(\bar{\Delta}_{\rm in}, \bar{\Delta}_{\rm out})$;
 7:     $\mathrm{Act}_{\Delta_{\rm in},\Delta_{\rm out}} = \mathrm{Act}_{\Delta_{\rm in},\Delta_{\rm out}} \cup (\bar{\Delta}_{\rm in}, \bar{\Delta}_{\rm out})$;
 8:     Add $l^{(\text{curr\_sol})}$ to update model $\mathcal{M}_1$;   /* Remove current solution from the model */
 9:     sol_count++;
10: end while
11: Dist := None;
12: for each pair of Patterns $(\bar{\Delta}_{\rm in}, \bar{\Delta}_{\rm out}) \in \mathrm{Act}_{\Delta_{\rm in},\Delta_{\rm out}}$ do
13:     $\mathcal{M}_2 = \mathcal{M}$;
14:     sol_count = 0;
15:     $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$ := None;
16:     Add constrains corresponding with $(\bar{\Delta}_{\rm in}, \bar{\Delta}_{\rm out})$ to model $\mathcal{M}_2$;
17:     while sol_count $< N_2$ do
18:         curr_sol = Solve the model $\mathcal{M}_2$ to get the best differential characteristic;
19:         Extract $(\Delta_{\rm in}, \Delta_{\rm out})$ from curr_sol;
20:         $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})} = \mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})} \cup (\Delta_{\rm in}, \Delta_{\rm out})$;
21:         Add $l^{(\text{curr\_sol})}$ to update model $\mathcal{M}_2$;
22:         sol_count++;
23:     end while
24:     Dist = Dist $\cup\, \mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$;
25: end for
26: return Dist.
```

which means that only the 0-th and 4-th nibbles are active (non-zero difference). What's more, assume we find two differentials under this input Pattern, whose input differences are $\Delta_{\rm in}^0$ and $\Delta_{\rm in}^1$ as follows:

$$\Delta_{\rm in}^0 = \{\mathrm{a}, 0, 0, 0, \mathrm{f}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\},$$

$$\Delta_{\rm in}^1 = \{\mathrm{f}, 0, 0, 0, \mathrm{a}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}.$$

Now, the input differences set of such multiple differentials can be denoted as

$$\mathrm{Dist}_{\bar{\Delta}_{\rm in}} = \{I_1, 0, 0, 0, I_2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\},$$

where $I_1, I_2 = \{\mathrm{a}, \mathrm{f}\}$.

Note that there are two parameters in Algorithm 1, which are N_1 and N_2. In general, the more differential trails we sample, the more number of Patterns and sets of input and output differences we may cover. But, the large values for N_1 and N_2 will lead to long excution time. To strike a balance, values for N_1 and N_2 can be heuristically explored. During our searching process, we can get enough Patterns and sets of input and output differences from just small part of best differential characteristics. In addition, with these sets returned by Algorithm 1, we still can refine and reduce the set $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$ according to the specific property of the target cipher (like linear layer and numbers of keys to be guessed in key recovery phase), which can save the searching time for the second step.

3.2 Enumerating process for searching multiple differentials

With each set of input and output differences $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$ returned by Algorithm 1, we still need an enumerating algorithm to search out all satisfied differential characteristics to evaluate the probability. Sun et al. [9] proposed an enumerating algorithm to search differential (with fixed input and output difference), truncate differential and boomerang/rectangle distinguishers, while it does not cover multiple differentials. So before starting the enumerating process, we need to use the constraints of subset in Subsection 2.2 to describe the set $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$. Then combined with r-round differential model, we get the corresponding multiple differentials search model.

However, during our enumerating process, we find that there are some invalid[1)] solutions when solving the model. To overcome this problem and record the detailed infomations of all satisfied characteristics, we give our improved multiple differentials enumerating algorithm in Algorithm 2. Here, for an invalid solution during the enumerating process, we do not discard it directly. We extract all variables from the invalid solution, and we set the approximate binary values for such continuous values to get an approximate solution. Then, we check this approximate solution, if it is an invalid characteristic, we discard it. If it is a valid one, we still need to check whether it is repeated with the obtained characteristics before. So, we use a hash table H to check these repeated characteristics. With checking and recording on the run, we discard the invalid solution and repeated solution characteristic promptly, which gains more proficiency and reduces the unnecessary impacts (no need to add the removing constraints) on the multiple differentials search model.

Algorithm 2 Improved multiple differentials enumerating algorithm

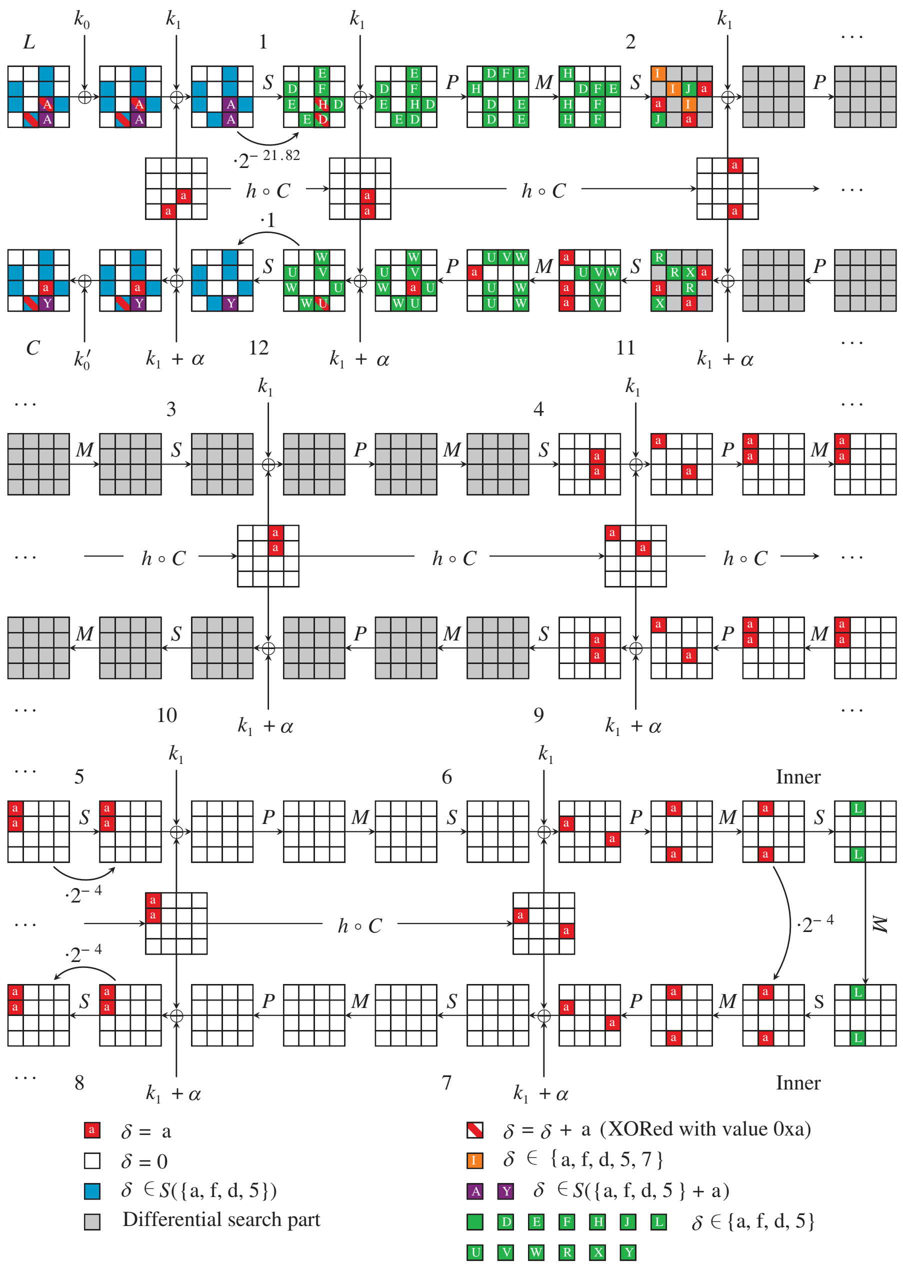
```
Input: $\mathcal{M}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$: multiple differentials model of the target cipher under $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$.
Output: $O_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$: all characteristics for multiple differentials under $\mathrm{Dist}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$.
 1: $H :=$ None;
 2: while True do
 3:     Solve the model $\mathcal{M}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$;
 4:     if model $\mathcal{M}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$ is infeasible then
 5:         break;
 6:     end if
 7:     curr_sol = extract a characteristic from the solution returned by solver (get the approximate solution when the
        solution has continuous values);
 8:     Check the status of curr_sol;
 9:     if the status of curr_sol is an invalid characteristic then
10:         Continue;
11:     end if
12:     if Hash(curr_sol) $\in H$ then
13:         Continue;
14:     else
15:         $H = H \cup$ Hash(curr_sol);
16:         Record curr_sol to $O_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$;
17:     end if
18:     Add $l^{(\text{curr\_sol})}$ to update model $\mathcal{M}_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$;
19: end while
20: return $O_{(\bar{\Delta}_{\rm in},\bar{\Delta}_{\rm out})}$.
```

3.3 Mount 10-round multiple differentials distinguisher on MANTIS

Both of the previous attacks on MANTIS use the same inner structure in their distinguishers, because in [6] the authors said, "most of the resulting solutions display the same inner structure as the existing 5-round characteristic". So we still use this good inner structure as a part of our distinguisher. Except this inner part, there may be many other differentials with the high probability in other parts of the distinguisher. Clustering such other differentials with high probability, not only will it provide more choices for our key recovery attack, but also may enhance the probability of distinguisher and even to mount a longer one.

Now, we use our new proposed automatic search method to mount a new 10-round multiple differentials distinguisher, shown in Figure 6 and consisting of three parts. The first part is the good inner structure mentioned above. The second part is from round 2 to 4. The third part is from round 9 to 11.

1) Some solutions have continuous values rather than exactly binary values. We find that this is caused by Gurobi optimizer due to the limitations of finite-precision arithmetic.



L
C
k_0
k_0'
k_1
$k_1 + \alpha$
$h \circ C$
S P M
$\cdot 2^{-21.82}$
$\cdot 1$
$\cdot 2^{-4}$
Inner
$\delta = \mathrm{a}$
$\delta = 0$
$\delta \in S(\{\mathrm{a, f, d, 5}\})$
Differential search part
$\delta = \delta + \mathrm{a}$ (XORed with value 0xa)
$\delta \in \{\mathrm{a, f, d, 5, 7}\}$
$\delta \in S(\{\mathrm{a, f, d, 5}\} + \mathrm{a})$
$\delta \in \{\mathrm{a, f, d, 5}\}$

Figure 6 (Color online) Multiple differential attack on MANTIS-6.

The first part (rounds 4 to 9). It starts from the the state after S in round 4 with the difference state

$$\{0, 0, 0, 0, 0, 0, \mathrm{a}, 0, 0, 0, \mathrm{a}, 0, 0, 0, 0, 0\},$$

and ends at the state before S in round 9 with

$$\{0, 0, 0, 0, 0, 0, \mathrm{a}, 0, 0, 0, \mathrm{a}, 0, 0, 0, 0, 0\}.$$

As we take this inner structure, we also need take its all tweak difference (owing to the linear tweak schedule in MANTIS), which is shown in Figure 6.

The second part (rounds 2 to 4). For this 3-round multiple differentials (includes three layers of Soperation), it ends at the state after S in round 4 with

$$\Delta_{\text{out}} = \{0, 0, 0, 0, 0, 0, \mathrm{a}, 0, 0, 0, \mathrm{a}, 0, 0, 0, 0, 0\}.$$

Table 2 The distribution of active nibbles' differences of the state after S in round 2 of 1664 trails with probability 2^{-28} under $\mathrm{Dist}'_{\alpha_0}$

Differences	a	f	d	5	7
N_0	1408	160	16	16	64
N_5*	1408	160	16	16	64
N_6	1160	328	88	88	0
N_7	1664	0	0	0	0
N_8	1664	0	0	0	0
N_{10}*	1408	160	16	16	64
N_{12}	1160	328	88	88	0
N_{14}	1664	0	0	0	0

* The 5-th and 10-th nibbles are equaling and can be used in key recovery phase.

So, we have just one output difference here.

Step 1. To get the sets of multiple input differences for this 3-round multiple differentials, we need to set N_1 and N_2 for the sampling according to Algorithm 1. After some trials, we let $N_1 = 200$ and $N_2 = 200$, which are enough to get the detailed results. After sampling N_1 best characteristics (the best probability of this 3-round differential characteristic is 2^{-28}) in just several minutes[2)], we get only one pair of Patterns of input and output differences $\mathrm{Act}_{\Delta_{\mathrm{in}},\Delta_{\mathrm{out}}} = \{(\alpha_0, \gamma_0)\}$,

$$\alpha_0 = \{1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0\},$$

$$\gamma_0 = \{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0\}.$$

With (α_0, γ_0), after sampling N_2 differential characteristics, we get the set of input differences:

$$\mathrm{Dist}_{\alpha_0} = \{G_1, 0, 0, 0, 0, G_2, G_3, G_4, G_1, 0, G_3, 0, G_1, 0, G_3, 0\},$$

where $G_1, G_3, G_4 = \{\mathrm{a}, \mathrm{f}, \mathrm{d}, 5\}$ and $G_2 = \{\mathrm{a}, \mathrm{f}, \mathrm{d}, 5, 7\}$.

Step 2. With Dist_{α_0} obtained in Step 1, we divide it into two sets as follows:

$$\mathrm{Dist}'_{\alpha_0} = \{G_1, 0, 0, 0, 0, G_5, G_3, G_4, G_1, 0, G_3, 0, G_1, 0, G_3, 0\},$$

$$\mathrm{Dist}''_{\alpha_0} = \{G_1, 0, 0, 0, 0, 7, G_3, G_4, G_1, 0, G_3, 0, G_1, 0, G_3, 0\},$$

where $G_1, G_3, G_4, G_5 = \{\mathrm{a}, \mathrm{f}, \mathrm{d}, 5\}$. We build corresponding multiple differentials search model for each set of input differences and use the improved enumerating algorithm to search out all satisfied differential characteristics.

Under these two sets of input differences, we get 2192 differential characteristics in total (this enumerating process costs about 5 hours). All these characteristics have only two kinds of probabilities, 2^{-28} and 2^{-32}. For $\mathrm{Dist}'_{\alpha_0}$, we get 1664 characteristics with probability 2^{-28} (256 input differences) and 336 characteristics with probability 2^{-32} (36 input differences). For $\mathrm{Dist}''_{\alpha_0}$, we get 32 characteristics with probability 2^{-28} (4 input differences) and 160 characteristics with probability 2^{-32} (16 input differences). We summarize the part of search results in Table 2, which decides the second part of the final distinguisher used for key recovery attack.

The third part (rounds 9 to 11). For this 3-round multiple differentials (includes 3 layers of S operation), it starts at the state before S in round 9 with

$$\Delta_{\mathrm{in}} = \{0, 0, 0, 0, 0, 0, \mathrm{a}, 0, 0, 0, \mathrm{a}, 0, 0, 0, 0, 0\}.$$

So, we have just one input difference here. It can noted that this part is the same as the inverse of the second part (rounds 4 to 2). We can reuse the search results in the second part, from which we just give the final results in Table 3.

2) By default, the programs are running on a PC using one thread with Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz and 4 GB RAM.

Table 3 Number of trails for each output state[a)]

State after S in round 12	Number of differential trails
$\{\mathrm{a},0,0,0,0,\mathrm{a},\mathrm{a},\mathrm{a},\mathrm{a},0,\mathrm{a},0,\mathrm{a},0,\mathrm{a},0\}$	32
$\{\mathrm{a},0,0,0,0,\mathrm{a},G_6,\mathrm{a},\mathrm{a},0,G_6,0,\mathrm{a},0,G_6,0\}$	52
$\{G_7,0,0,0,0,\mathrm{a},G_6,\mathrm{a},G_7,0,G_6,0,G_7,0,G_6,0\}$	116
$\{\mathrm{a},0,0,0,0,\mathrm{a},G_6,G_8,\mathrm{a},0,G_6,0,\mathrm{a},0,G_6,0\}$	208
$\{\mathrm{a},0,0,0,0,G_9,G_6,G_8,\mathrm{a},0,G_6,0,\mathrm{a},0,G_6,0\}$	640

a) $G_6,G_7,G_8,G_9=\{\mathrm{a},\mathrm{f},\mathrm{d},5\}$

The final 10-round multiple differentials. Now, we use the three parts above to combine the final 10-round multiple differentials. However, not all trails contribute to the probability of the distinguisher, the collection of data and the key recovery phase, so we need to carefully select the multiple input or output differences according to detailed search results of all characteristics above.

For the second part, we just use 1664 differential characteristics with probablity 2^{-28} under the set $\mathrm{Dist}'_{\alpha_0}$. It starts with

$$\{G_1,0,0,0,0,G_5,G_3,G_4,G_1,0,G_3,0,G_1,0,G_3,0\},$$

where $G_1,G_3,G_4,G_5=\{\mathrm{a},\mathrm{f},\mathrm{d},5\}$. It ends with

$$\{0,0,0,0,0,0,\mathrm{a},0,0,0,\mathrm{a},0,0,0,0,0\}.$$

So, the probability of the second part is $\frac{2^{-28}\cdot 1664}{256}\approx 2^{-25.30}$. Then, for the third part, we start with

$$\{0,0,0,0,0,0,\mathrm{a},0,0,0,\mathrm{a},0,0,0,0,0\}.$$

Considering data and time complexities of the whole attack, we use the output differences set with

$$\{\mathrm{a},0,0,0,0,G_9,G_6,G_8,\mathrm{a},0,G_6,0,\mathrm{a},0,G_6,0\},$$

where $G_6,G_8,G_9=\{\mathrm{a},\mathrm{f},\mathrm{d},5\}$. The probability of the third part is $2^{-28}\cdot 640=2^{-18.68}$. Finally, we combine these three parts to gain a new 10-round multiple differentials distinguisher with probability $p_*=2^{-25.30}\cdot 2^{-12}\cdot 2^{-18.68}=2^{-55.98}>2^{-58}$ (here the random probability is 2^{-58}, because there are 2^6 output differences).

3.4 Mount 11-round multiple differentials on MANTIS

Similarly, we also find an 11-round multiple differentials with the same inner part. Owing to the space limitations, we just give the input and output differences of our distinguisher, shown in Figure 7. It starts at the state before S in round 2 with

$$\{0,0,\mathrm{a},0,F,0,\mathrm{a},0,F,0,0,K,0,\mathrm{a},0,0\},$$

where $F=\{\mathrm{a},\mathrm{f}\}$ and K means being included in key recovery (so the probability is calculated in key recovery phase, not included in distinguisher). It ends at the state after S in round 12 with

$$\{\mathrm{a},0,0,0,0,\mathrm{a},\mathrm{a},K,\mathrm{a},0,\mathrm{a},0,\mathrm{a},0,\mathrm{a},0\}.$$

Then we get an 11-round multiple differentials distinguisher with probability $p_*=2^{-30.71}\cdot 2^{-12}\cdot 2^{-21}=2^{-63.71}>2^{-64}$ (random probability).

4 Improved key recovery attack on MANTIS-6

In this section, we give an improved key recovery attack on MANTIS-6 based on the 10-round multiple differentials distinguisher with probability $2^{-55.98}$ in Subsection 3.3. The data and time complexities are $2^{51.79}$ chosen-plaintexts and $2^{51.91}$ encryptions respectively, so the data-time product is $2^{103.70}$. The whole attack involves data collection phase and key recovery phase, whose details are given in this section.

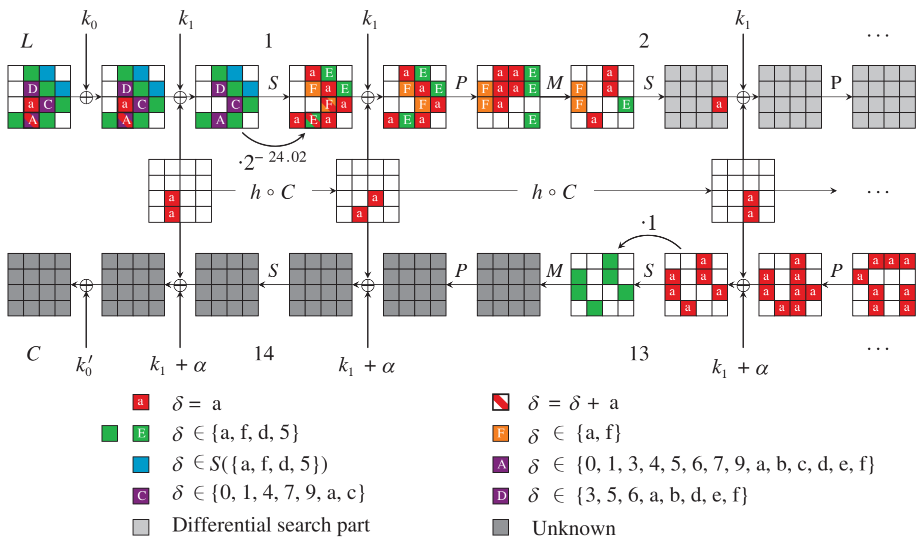


Figure 7 (Color online) Multiple differential attack on MANTIS-7.

4.1 Data collection

As is shown in Figure 6, the plaintext covers $|\chi^L| = 13^6 \cdot 14^2 \approx 2^{29.82}$ input differences, and the size of its linear span[3] is $|\langle\chi^L\rangle| = 16^8 = 2^{32}$. To generate our data efficiently, we describe the structure as follows: with a fixed plaintext $L \in \chi^L$ and a fixed tweak T, we get a plaintext pair (L, T) and $(L, T \oplus \mathrm{a})$ under the tweaks $(T, T \oplus \mathrm{a})$. Then, the set $\bigcup_L \{L \oplus \Delta | \Delta \in \langle\chi^L\rangle\}$ is called a structure. So, with $|\langle\chi^L\rangle| = 2^{32}$ chosen-plaintext pair queries, we obtained $|\chi^L| \cdot |\langle\chi^L\rangle| = 2^{61.82}$ compatible plaintext pairs.

The probability of a plaintext pair to follow the input of distinguisher in Figure 6 is $2^{-21.82}$. For example, the difference of nibble N_2 of the state before S in round 1 are in the set $S(\{\mathrm{a,f,d,5}\})$. From DDT of MANTIS, we know $S(\{\mathrm{a,f,d,5}\}) = \{1,3,4,5,6,7,9,\mathrm{a,b,c,d,e,f}\}$, which has 13 elements in total. Thus, the transition probability of the set $S(\{\mathrm{a,f,d,5}\})$ to the set $\{\mathrm{a,f,d,5}\}$ is $\frac{1}{13} \cdot 4 = 2^{-1.7}$. Meanwhile, we also need to consider the probability for three equaling set E, which is $2^{-2} \cdot 2^{-2} = 2^{-4}$. Similarly, we can calculate the transition probabilities for other nibbles. Finally, we can get the probability $2^{-21.82}$.

To include a right pair following the whole 12-round differentials, we need to generate $2^{55.98+21.82} = 2^{77.80}$ pairs, that means we need $N_{\mathrm{st}} = 2^{77.80}/2^{61.82} = 2^{15.98}$ structures illustrated above. The data complexity for one right pair is $N_Q = 2 \cdot N_{\mathrm{st}} \cdot |\langle\chi^L\rangle| = 2^{48.98}$ chosen-plaintext queries.

For the filter probability β on ciphertext, there are two parts: inactive nibbles and active nibbles. To satisfy the 8 inactive nibbles on ciphertext, the probability is $2^{-4 \cdot 8} = 2^{-32}$, while the probability to satisfy 5 nibbles with difference $S(\{\mathrm{a,f,d,5}\})$, 1 nibbles with difference $S(\{\mathrm{a,f,d,5}\}) + \mathrm{a}$, 1 nibble with difference $S(\{\mathrm{a,f,d,5}\} + \mathrm{a})$ and 1 nibbles with difference a, the probability is $(\frac{13}{16})^6 \cdot \frac{14}{16} \cdot \frac{1}{16} \approx 2^{-5.99}$. As a result, $\beta = 2^{-37.99}$ and $N_F = N_{\mathrm{st}} \cdot |\chi^L| \cdot |\langle\chi^L\rangle| \cdot \beta = 2^{39.81}$ filtered pairs containing one right pair will be used in key recovery phase. Since we expect $N_{\mathrm{right}} = 7$ right pairs in the following key recovery phase, thus we need $N_{\mathrm{right}} \cdot N_Q = 2^{51.79}$ chosen-plaintexts to generate $N_{\mathrm{right}} \cdot N_F = 2^{42.62}$ filtered pairs.

4.2 Key recovery

Before mounting a key recovery attack, we generalize an important key relation of MANTIS used in [6] as Lemma 3 and give the proof in Appendix A.

Lemma 3. In MANTIS, for any $0 \leqslant i, j, k < 15$ (i, j, k are different from each other), assume knowing the i-th, j-th, k-th nibbles of $(k_0 + k_1)$ and the i-th, j-th, k-th nibbles of $(k_0' + k_1)$, 3 out of 4 bits in $\delta_{i \oplus j \oplus k}$ of k_1 can be directly deduced by guessing another bit.

Recovery 62 bits from the filtered pairs. We use $N_{\mathrm{right}} \cdot N_F = 2^{42.62}$ filtered pairs to recovery the right 62 bits key informations according to the conditions in Table 4 as follows.

3) Here, we can regard all bits in active nibbles as a binary vector.

Table 4 Conditions and their filter probabilities in key recovery on MANTIS-6

Rounds	Condition	Nibbles	Difference	Probability	Key bits
1	(C1)	$\delta_{2,8,13}$	$\{\mathrm{a,f,d,5}\}$, equal*	$2^{-9.1}$	12
	(C2)	$\delta_{4,11,14}$	$\{\mathrm{a,f,d,5}\}$, equal[+a]**	$2^{-9.21}$	12
	(C3)	δ_{10}	$\{\mathrm{a,f,d,5}\}+\mathrm{a}$	$2^{-1.81}$	4
	(C4)	δ_{6}	$\{\mathrm{a,f,d,5}\}$	$2^{-1.7}$	4
12	(C5)	$\delta_{2,8,13}$	$\{\mathrm{a,f,d,5}\}$	$2^{-9.1}$	12
	(C6)	$\delta_{4,11,14}$	$\{\mathrm{a,f,d,5}\}$, equal[+a]	$2^{-9.21}$	12
	(C7)	δ_{6}	$\{\mathrm{a,f,d,5}\}$	$2^{-1.7}$	4
2,11	(C8)	$\delta_{2\oplus8\oplus13}$	$\{\mathrm{a}\}$	2^{-4}	1
	(C9)	$\delta_{4\oplus11\oplus14}$	$\{\mathrm{a,f,d,5}\}$	2^{-2}	1

* The differences of corresponding nibbles are equal.
** The differences or the difference XORed with value a of corresponding nibbles are equal.

Table 5 Conditions for key recovery and filter probabilities on MANTIS-6 (for 7 right pairs)

Round	Condition	Nibbles	Difference	Probability
2	(V1)	δ_{14}	$\{\mathrm{a}\}$	2^{-2}
2	(V2)	$\delta_{5,10}$	$\{\mathrm{a,f,d,5,7}\}$, equal	$2^{-3.30}$
11	(V3)	δ_{10}	$\{\mathrm{a,f,d,5}\}$	2^{-1}
11	(V4)	δ_{14}	$\{\mathrm{a}\}$	2^{-2}

(1) Guess 12 bits in $\delta_{2,8,13}$ of k_0+k_1 in round 1 and compute forward to check the condition (C1). $2^{42.62}\cdot 2^{-9.1}=2^{33.52}$ pairs will be left.

(2) Guess 12 bits in $\delta_{2,8,13}$ of $k_0'+k_1$ in round 12 and compute forward to check the condition (C5). $2^{33.52}\cdot 2^{-9.1}=2^{24.42}$ pairs will be left.

(3) Guess 1 bit in $\delta_{2\oplus8\oplus13}$ of k_1, compute forward and check the condition (C8). $2^{24.42}\cdot 2^{-4}=2^{20.42}$ pairs will be left.

(4) Guess 12 bits in $\delta_{4,11,14}$ of k_0+k_1 in round 1, compute forward and check the condition (C2). $2^{20.42}\cdot 2^{-9.21}=2^{11.21}$ pairs will be left.

(5) Guess 12 bits in $\delta_{4,11,14}$ of $k_0'+k_1$ in round 12, compute forward and check the condition (C6). $2^{11.21}\cdot 2^{-9.21}=2^{2.00}$ pairs will be left.

(6) Guess 1 bit in $\delta_{4\oplus11\oplus14}$ of k_1, compute forward and check the condition (C9). $2^{2.00}\cdot 2^{-2}=1$ pair will be left.

(7) Guess 4 bits in δ_6 of k_0+k_1 in round 1, compute forward and check the condition (C3). $1\cdot 2^{-1.81}=2^{-1.81}$ pair will be left.

(8) Guess 4 bits in δ_{10} of $k_0'+k_1$ in round 1, compute forward and check the condition (C4). $2^{-1.81}\cdot 2^{-1.7}=2^{-3.51}$ pair will be left.

(9) Guess 4 bits in δ_{10} of $k_0'+k_1$ in round 12, compute forward and check the condition (C7). $2^{-3.51}\cdot 2^{-1.7}=2^{-5.21}$ pair will be left.

After this process, the right key suggests at least 7 pairs, the wrong key suggests $2^{-5.21}$ pair on average. The signal-to-noise ratio is $S_N=\frac{7}{2^{-5.21}}=2^{8.02}$. The key suggested with the most times is expected to be the right key. The 62 bits key material is found. We expect that only the 7 right pairs remain. So, the major time complexity of this phase is $2^{59.49}$ S-box operations, which can be reduced to $2^{59.49}\cdot\frac{1}{12}\cdot\frac{1}{16}=2^{51.91}$ encryptions.

Recovery 26 bits from the right pairs. We can use the 7 right pairs and 62 bits right key to recover another 26 bits key information according to the conditions in Table 5 as follows. In round 2, the nibbles N_5, N_{10} of the state after S have the same difference with probability $2^{-3.30}$, the nibble N_{14} of the state after S has the difference $\{\mathrm{a}\}$ with probability 2^{-2}. Similarly, the filter probability for round 11 is 2^{-3}. All involved keys are illustrated in Figure 8. With already known 62 bits, another 26 bits key need to be guessed. Condition (V1 & V2 & V3 & V4) holds with the probability of $2^{-8.30\cdot7}=2^{-58.10}$ for 7 right pairs. We expect that only the right 26-bit subkey is remained. Then, the left $128-62-26=40$ bits key material can be exhausted. So, the time complexity of this phase is about 2^{40} encryptions.

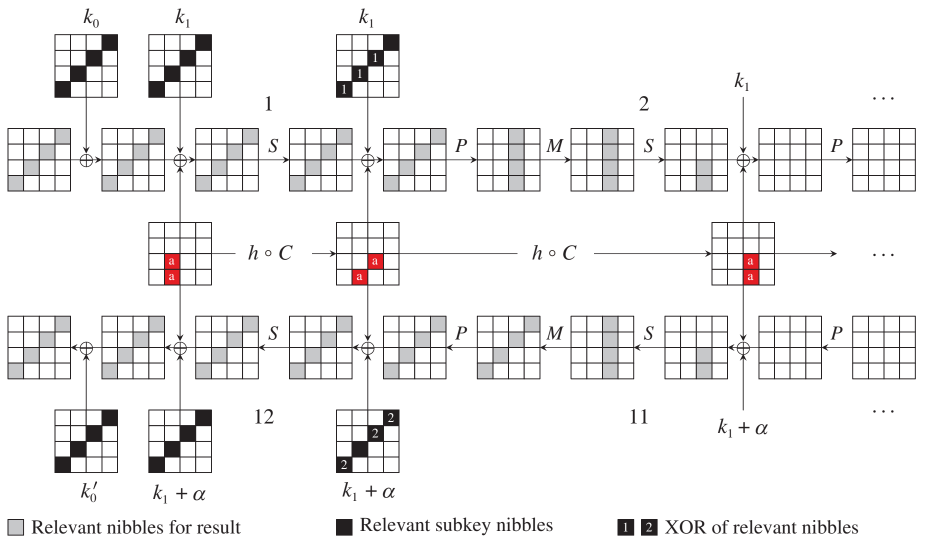


Figure 8 (Color online) Key cells involved in condition (V1 & V2 & V3 & V4).

4.3 Complexity and success probability

In the whole attack on MANTIS-6, the data complexity is $2^{51.79}$ chosen-plaintext queries, the time complexity is about $2^{51.91}$ 12-round encryptions, so the data-time product is $2^{103.70}$. We calculate the success probability by the following formula in [13]:

$$P_S = \Phi\left(\frac{\sqrt{\mu S_N} - \Phi^{-1}(1-2^{-a})}{\sqrt{S_N+1}}\right), \tag{3}$$

where Φ is the normal distribution, μ is the number of right pair, S_N is the signal-to-noise ratio and a is the advantage key bits. As we leave only one best candidate, that means $l = 2^0, a = 62 - 0 = 62$. So, the success probability is $P_S = 98.1\%$.

5 Key recovery attack on MANTIS-7

In order to explore the gap between the multiple differential attack and security margin on full version MANTIS claimed by the designers, we mount a key recovery attack on MANTIS-7 without considering the data-time product restriction.

Data collection. We give the data collection phase as follows.

Step 1. Put $2^{60.855}$ ciphertexts under tweak T and $T+\alpha$ (totally $2^{61.855}$ ciphertexts) into a 2^{24} size hash table T with size of 2^{24} (indexed by 24 bits, the 5 inactive nibbles and 1 nibble with difference {a}) on plaintext.

Step 2. In each entry of the table T, it has $2^{60.855-24} = 2^{36.855}$ pairs on average. So we can combine $2^{36.855\cdot 2} = 2^{73.71}$ pairs in each entry.

Step 3. We get $2^{73.71+24} = 2^{97.71}$ plaintext pairs. After filtered by the active bits on plaintext, it remains $N_F = 2^{97.71} \cdot \frac{2^{27.02}}{2^{35}} = 2^{89.73}$ filtered pairs.

Step 1 needs $2^{61.855}$ chosen-ciphertext queries. Step 2 needs $2^{97.71}$ memory access. Step 3 needs $2^{97.71}$ memory access. The filtered N_F pairs including one right pair will be used in key recovery phase.

Key recovery. We list all filter conditions in Table 6. For each pair in $N_F = 2^{89.73}$ filter pairs, we do the following steps:

(1) Guess 2^3 differences of $\delta_{4,13}$ before S in round 2. Then we can squeeze out one 44-bit (corresponding to 11 nibbles $\delta_{1,2,5,6,7,9,10,11,12,13,14}$) key information of $k_0 + k_1$ on average.

(2) Guess 2^{12} differences of $\delta_{2,4,6,8,11,13}$ after S in round 13. Then we can squeeze out one 64-bit (corresponding to all 16 nibbles) key information of $k_0' + k_1$ on average.

Table 6 Conditions for key recovery and filter probability on MANTIS-7

Round	Condition	Nibbles	Difference	Probability	Key bits
1	(C1)	δ_3	{a}	1	4
2	(C2)	$\delta_{2\oplus7\oplus13}$	{a}	2^{-2}	1
13	(C3)	$\delta_{2\oplus7\oplus13}$	{a}	2^{-2}	1
	(C4)	$\delta_{1\oplus11\oplus14}$	{a}	2^{-2}	1
	(C5)	$\delta_{3\oplus6\oplus9}$	{a}	2^{-2}	1
	(C6)	$\delta_{3\oplus9\oplus12}$	{a}	2^{-2}	1
	(C7)	$\delta_{0\oplus10\oplus15}$	{a}	2^{-2}	4
	(C8)	$\delta_{0\oplus5\oplus10}$	{a}	2^{-2}	4
12	(C9)	$\delta_{2\oplus8\oplus13}$	{a}	2^{-2}	4

(3) Guess 1 bit in $\delta_{2\oplus7\oplus13}$ of k_1. If the pair satisfies (C2) and (C3) in Table 6 (with probability $2^{-2-2} = 2^{-4}$), then we remain it, otherwise discard it. $2^{1-4} = 2^{-3}$ keys will be left.

(4) Guess 1 bit in $\delta_{1\oplus11\oplus14}$ of k_1. If the pair satisfies (C4), then we remain it, otherwise discard it. $2^{-3+1-2} = 2^{-4}$ keys will be left.

(5) Guess 4 bit in δ_3 of $k_0 + k_1$. $2^{-4+4} = 2^0$ keys will be left.

(6) Guess 1 bit in $\delta_{3\oplus6\oplus9}$ of k_1. If the pair satisfies (C5), then we remain it, otherwise discard it. $2^{0+1-2} = 2^{-1}$ keys will be left.

(7) Guess 1 bit in $\delta_{3\oplus9\oplus12}$ of k_1. If the pair satisfies (C6), then we remain it, otherwise discard it. $2^{-1+1-2} = 2^{-2}$ keys will be left.

(8) Guess 4 bits in $\delta_{0\oplus10\oplus15}$ of k_1. If the pair satisfies (C7), then we remain it, otherwise discard it. $2^{-2+4-2} = 1$ keys will be left.

(9) Guess 4 bits in $\delta_{0\oplus5\oplus10}$ of k_1. If the pair satisfies (C8), then we remain it, otherwise discard it. $2^{0+4-2} = 2^2$ keys will be left.

(10) Guess 4 bits in $\delta_{2\oplus8\oplus13}$ of k_1. If the pair satisfies (C9), then we remain it, otherwise discard it. $2^{2+4-2} = 2^4$ keys will be left.

After this phase, $2^{89.73+15+4} = 2^{108.73}$ candidate keys are suggested in total. While the squeezed and guessed bits include $44 + 64 + 20 = 128$ (the 20 bits are guessed in step (3) to (10)) bits independent key informations. The time complexity of this phase is $2^{110.73}$ S-box operations, which can be reduced to $2^{110.73} \cdot \frac{1}{14 \cdot 16} = 2^{102.92}$ encryptions, the signal-to-noise ratio $S_N = \frac{1}{2^{108.73}/2^{128}} = 2^{19.27}$.

Complexity and success probability. With one right pair, the data complexity is about $2^{61.86}$ chosen-ciphertexts. The major time complexity of the whole attack is about $2^{102.92}$ encryptions. As we leave 2^{96} best candidate keys (time complexity of this brute searching part can be ignored), that means $l = 2^{96}$, $a = 128 - 96 = 32$. So the success probability is $P_S = 83.94\%$.

6 Conclusion

In this paper, we propose a generic automatic search method for clustering multiple differentials. Compared to traditional searching process, our method removes restrictions on searching multiple differentials as much as possible. We apply this new search method on MANTIS and find a new 10-round distinguisher to derive an improved key recovery attack, which improves the data-time product of the previous best differential attack on MANTIS from $2^{110.61}$ down to $2^{103.70}$. We also find an 11-round distinguisher for MANTIS, which is the longest one until now. Based on this distinguisher, we explore the gap between multiple differential attack and the security margin on MANTIS. The result shows that full version MANTIS is still secure enough against multiple differential attack.

Acknowledgements This work was supported by National Cryptography Development Fund (Grant No. MMJJ201701-02), National Natural Science Foundation of China (Grant No. 61572293), Major Scientific and Technological Innovation Projects of Shandong Province (Grant No. 2017CXGC0704), and Fundamental Research Fund of Shandong Academy of Sciences (Grant No. 2018:12-16).

References

1 Biham E, Shamir A. Differential Cryptanalysis of the Data Encryption Standard. Berlin: Springer, 1993

2 Blondeau C, Gérard B. Multiple differential cryptanalysis: theory and practice. In: Proceedings of International Workshop on Fast Software Encryption, 2011. 35–54

3 Wang M Q, Sun Y, Tischhauser E, et al. A model for structure attacks, with applications to PRESENT and Serpent. In: Proceedings of International Workshop on Fast Software Encryption, 2012. 49–68

4 Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems, 2007. 450–466

5 Dobraunig C, Eichlseder M, Kales D, et al. Practical key-recovery attack on MANTIS5. In: Proceedings of International Workshop on Fast Software Encryption, 2016. 248–260

6 Eichlseder M, Kales D. Clustering related-tweak characteristics: application to MANTIS-6. In: Proceedings of International Workshop on Fast Software Encryption, 2018. 111–132

7 Beierle C, Jean J, Kölbl S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Proceedings of Annual International Cryptology Conference, 2016. 123–153

8 Borghoff J, Canteaut A, Güneysu T, et al. PRINCE – a low-latency block cipher for pervasive computing applications. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2012. 208–225

9 Sun S W, Hu L, Wang M Q, et al. Automatic enumeration of (related-key) differential and linear characteristics with predefined properties and its applications. 2014. https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2014/747&version=20140926:084100&file=747.pdf

10 Sun S W, Hu L, Wang P, et al. Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2014. 158–178

11 Gurobi Optimization. Gurobi optimizer reference manual. 2018. http://www.gurobi.com

12 Balas E, Jeroslow R. Canonical cuts on the unit hypercube. SIAM J Appl Math, 1972, 23: 61–69

13 Selçuk A A. On probability of success in linear and differential cryptanalysis. J Cryptol, 2008, 21: 131–147

Appendix A Proof of Lemma 3

Proof. For the sake of simplicity, we use $K_{0,1}[a,b]$ to denote

$$k_0[(a)\text{mod}64] \oplus k_1[(b)\text{mod}64],$$

$K_1[a,b,c]$ to denote

$$k_1[(a)\text{mod}64] \oplus k_1[(b)\text{mod}64] \oplus k_1[(c)\text{mod}64].$$

If we know the i-th, j-th, k-th ($i,j,k \in \{0,1,\ldots,14\}$) nibbles of k_0+k_1 and $k_0'+k_1$, which are the following 24 bits:

$$K_{0,1}[4i,4i],\ K_{0,1}[4i+1,4i+1],\ K_{0,1}[4i+2,4i+2],\ K_{0,1}[4i+3,4i+3], \tag{A1}$$

$$K_{0,1}[4j,4j],\ K_{0,1}[4j+1,4j+1],\ K_{0,1}[4j+2,4j+2],\ K_{0,1}[4j+3,4j+3], \tag{A2}$$

$$K_{0,1}[4k,4k],\ K_{0,1}[4k+1,4k+1],\ K_{0,1}[4k+2,4k+2],\ K_{0,1}[4k+3,4k+3], \tag{A3}$$

$$K_{0,1}[4i-1,4i],\ K_{0,1}[4i,4i+1],\ K_{0,1}[4i+1,4i+2],\ K_{0,1}[4i+2,4i+3], \tag{A4}$$

$$K_{0,1}[4j-1,4j],\ K_{0,1}[4j,4j+1],\ K_{0,1}[4j+1,4j+2],\ K_{0,1}[4j+2,4j+3], \tag{A5}$$

$$K_{0,1}[4k-1,4k],\ K_{0,1}[4k,4k+1],\ K_{0,1}[4k+1,4k+2],\ K_{0,1}[4k+2,4k+3], \tag{A6}$$

then we can combine first 3 bits in (A1) and last 3 bits in (A4) to get 3 bits

$$K_1[4i,4i+1],\ K_1[4i+1,4i+2],\ K_1[4i+2,4i+3].$$

Similarly, we can get

$$K_1[4j,4j+1],\ K_1[4j+1,4j+2],\ K_1[4j+2,4j+3],$$
$$K_1[4k,4k+1],\ K_1[4k+1,4k+2],\ K_1[4k+2,4k+3].$$

Once we get one bit $K_1[4i,4j,4k]$, as we already know

$$K_1[4i,4i+1] \oplus K_1[4j,4j+1] \oplus K_1[4k,4k+1],$$

then we get another three bits

$$K_1[4i+1,4j+1,4k+1],\ K_1[4i+2,4j+2,4k+2],\ K_1[4i+3,4j+3,4k+3].$$