International Journal of Cyber-Security and Digital Forensics (IJCSDF) 2(3): 27-34
The Society of Digital Information and Wireless Communications, 2013 (ISSN: 2305-0012)

The Improved 32nd-Order Differential Attack on 8 Rounds of MISTY2 without FL Functions

Yasutaka Igarashi[1], Toshinobu Kaneko[2], Yutaka Eguchi[1], Takahiro Murai[1], Ryutaro Sueyoshi[1], Yosuke Hashiguchi[1], Seiji Fukushima[1], and Tomohiro Hachino[1]
[1] Kagoshima University
1-21-40 Korimoto, Kagoshima, 890-0065 Japan
{igarashi, fukushima, hachino}@eee.kagoshima-u.ac.jp
[2] Tokyo University of Science
2641 Yamazaki, Noda, Chiba, 278-8510 Japan
kaneko@ee.noda.tus.ac.jp

ABSTRACT

We study the 32nd-order differential attack on 8 rounds of MISTY2 without FL functions. MISTY2 is a 64-bit block cipher with a 128-bit secret key proposed by Matsui of Mitsubishi Electric Corp. in 1996. We found the new 32nd-order differential characteristic of MISTY2 without FL functions, which makes the 32nd-order differential of the upper 23 bits out of a 32-bit input to the 8th-round FO function be zero. Using the characteristics, we show that 8 rounds of MISTY2 without FL functions can be simply attacked with 2^{35} blocks of chosen plain text and $2^{81.4}$ times of FO operation. Moreover we reduce the number of times of FO operation required for this attack by using a modulo 2 occurrence distribution, which is derived by a partial sum technique proposed by Ferguson et al. We apply this distribution to the intermediate data of encryption function, and show that the number of times of FO operation can be reduced to $2^{57.4}$. This work is the first 8-round attack on MISTY2 as far as we know, while previously known higher-order differential attacks on MISTY2 are 5-round attack and 7-round attack.

KEYWORDS

Cryptanalysis, higher-order differential attack, block cipher, MISTY2, partial sum technique, modulo 2 occurrence distribution

1 INTRODUCTION

Recently various cryptographic systems and subsystems have been proposed [1], [2], [3], [4], [5]. Among them, MISTY2 is a 64-bit block cipher with a 128-bit secret key designed by Matsui of Mitsubishi Electric Corp. in 1996 [6]. The designer recommends using 12 rounds of FO function and 14 sets of FL function. Because FL is a linear function as far as a secret key is fixed, FL does not determine provable security against a differential attack and a linear attack. On the other hand FO determines the security because it is a nonlinear function. Therefore we analyze the security against a higher-order differential attack [7] on MISTY2 without FLs.

Table 1 shows data complexity and computational complexity of a higher-order differential attack on MISTY2 without FLs. Sugita reported a 5-round attack with the 7th-order differential, 2^7 blocks of chosen plain text, and 2^{39} times of FO operation [8]. We previously reported a 7-round attack with 7th-order

differential, 2^{11} blocks of chosen plain text, and 2^{83} times of FO operation [9]. This time we found the new characteristics of MISTY2 without FLs, which is that the 32nd-order differential of the upper 23 bits out of the 32-bit input to FO_8 function becomes zero. We show a simple 8-round attack with the 32nd-order differential we found, 2^{35} blocks of chosen plain text, and $2^{81.4}$ times of FO operation. This is the first 8-round attack on MISTY2. Moreover we reduce the number of times of FO operation required for the attack by exploiting a modulo 2 occurrence distribution (MOD), which is derived by a partial sum technique proposed by Ferguson et al. We apply this distribution to the intermediate data of encryption function, and show that the number of times of FO operation can be reduced to $2^{57.4}$.

Table 1. Complexity of a higher-order differential attack on MISTY2 without FLs. "*" denotes this article.

rd.	order	data	# of times	ref.
5	7	2^7	2^{39}	[8]
7	7	2^{11}	2^{83}	[9]
8	32	2^{35}	$2^{81.4}$	*
8	32	2^{35}	$2^{57.4}$	*

2 EIGHT ROUNDS OF MISTY2 WITHOUT FLS

In this section we show the 8-rounds data-mixing part of MISTY2 without FLs, and show its components FO_i and FI_{ij} (i = 1, 2, …, 7, 8. j = 1, 2, 3). Next we study the equivalent modification of FI_{ij} for reducing computational complexity of an attack.

2.1 Data-Mixing Part of MISTY2 without FLs

Fig. 1 shows 8-round data-mixing part of MISTY2 without FLs. Both an input plain text P and an output cipher text C are 64 bits. P_L and P_R are the upper and the lower 32 bits of P, respectively. C_L and C_R are the upper and the lower 32 bits of C, respectively. $P = P_L \| P_R$, $C = C_L \| C_R$ where the symbol "||" denotes concatenation of two data. MISTY2 consists of XOR (⊕) and FO_i. We call the component with 64-bit input/output (I/O) including one FO_i and the following XOR "round."

Fig. 2 shows FO_i (i = 1, 2, …, 8). Both an input data and an output data are 32 bits. It consists of XOR and FI_{ij} (i = 1, 2, …, 8. j = 1, 2, 3). KO_{ij} (i = 1, 2, …, 8. j = 1, 2, 3, 4) represents a 16-bit extended key.

Fig. 3 shows FI_{ij} (i = 1, 2, …, 8. j = 1, 2, 3). Both an input and an output are 16 bits. It consists of XOR and two kinds of S-boxes S_9 and S_7. S_9 represents an S-box with 9-bit I/O, and S_7 represents an S-box with 7-bit I/O. KI_{ij1} and KI_{ij2} represent a 7-bit and a 9-bit extended key, respectively.

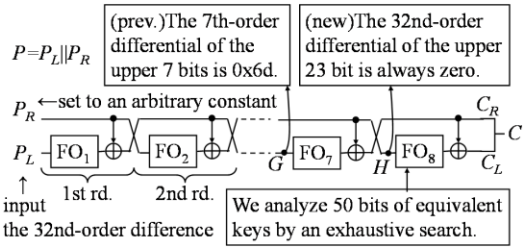

Figure 1. Eight-round data-mixing part of MISTY2 without FLs.

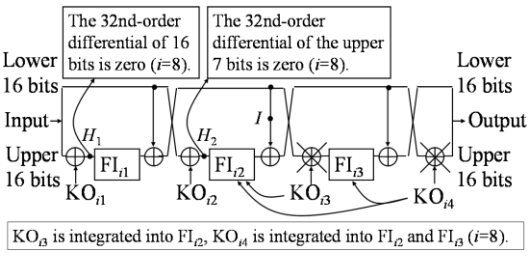

Figure 2. FO_i (i=1, 2, …, 8).

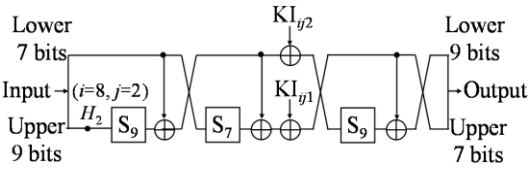

Figure 3. FI_{ij} (i=1, 2, …, 8. j=1, 2, 3).

2.2 Equivalent Modification

We study the equivalent modification of FI_{8j} to reduce the total number of bits of the extended keys attacker has to analyze, which reduces the computational complexity of an attack.

Since XOR operators of the extended key can be freely moved unless they jump over S-boxes in Fig. 2 and 3, we move them in order to integrate as many extended as possible. We call such a modified FI_{8j} an equivalent FI_{8j} shown in Fig. 4. KI'_{8jk} (j = 2, 3. k = 1, 2, 3) is called an equivalent key that is an integration of some extended keys given by

$$KI'_{821} = KI_{822}, \quad (1)$$

$$KI'_{822} = (KI_{822})^{R7} \oplus KI_{821} \oplus (KO_{83} \oplus KO_{84})^{L7}, \quad (2)$$

$$KI'_{823} = \left(00 \,\|\, (KO_{83} \oplus KO_{84})^{L7}\right) \oplus (KO_{83} \oplus KO_{84})^{R9}, \quad (3)$$

$$KI'_{831} = KI_{832}, \quad (4)$$

$$KI'_{832} = (KI_{832})^{R7} \oplus KI_{831} \oplus (KO_{84})^{L7}, \quad (5)$$

$$KI'_{833} = \left(00 \,\|\, (KO_{84})^{L7}\right) \oplus (KO_{84})^{R9}. \quad (6)$$

$(x)^{Li}$ denotes the upper i bits of data x. $(x)^{Ri}$ denotes the lower i bits of data x. For example, the first term of the right side of (6) is 9-bit data, whose upper 2 bits are 00 and the lower 7 bits are the upper 7 bits of KO_{84}. Because KO_{83} and KO_{84} in Fig. 2 are integrated into KI'_{822}, KI'_{823}, KI'_{832}, and KI'_{833}, they are removed from Fig. 2.

The total number of bits of extended keys is 64 as shown in the right side of (1)--(6); KI_{822}, KI_{823}, KI_{832}, KI_{833}, KO_{83}, KO_{84} before the equivalent modification. On the other hand, the total number of bits of equivalent keys is 50 as shown in the left side of (1)--(6) after the modification. It is reduced by 14 bits. Note that the characteristics of FI_{8j} do not change by the equivalent modification.

For the rest of this article we study a higher-order differential attack by focusing on the equivalent FI_{8j} and the equivalent keys.

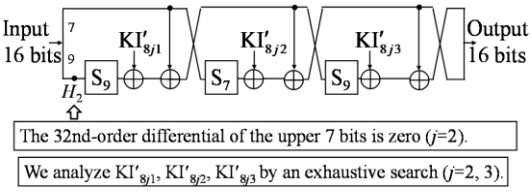

Figure 4. Equivalent FI_{8j} (j=2, 3).

3 HIGHER-ORDER DIFFERENTIAL

In this section, we describe the definition of higher-order differential and some of its properties [7] related to this article, and we describe an attack equation using these properties. Next we show the new 32nd-order differential characteristics of MISTY2 we found, and describe the attack equation using the found characteristics

3.1 Definition, Property, and Attack Equation

Fig. 5 shows a block diagram of an encryption process. E_1 and E_2 represent components of an encryption process. $K_1 \in GF(2)^p$ and $K_2 \in GF(2)^q$ represent p bits and q bits of the extended keys used in E_1 and E_2, respectively. $P = (p_1, p_2, \ldots, p_n)$ and $\Delta P \in GF(2)^n$ represent n bits of

input plain text and input difference, respectively. $H \in \mathrm{GF}(2)^m$ represents m bits of the output of E_1. $C(P \oplus \Delta P) \in \mathrm{GF}(2)^\ell$ represents ℓ bits of the output cipher text corresponding to $P \oplus \Delta P$. We assume $V^{(i)}$ as an ith-order subspace of $\mathrm{GF}(2)^n$ consisting of i sets of linear independent vector in $\mathrm{GF}(2)^n$ ($i \le n$), and call it an input differential. The ith-order differential of $E_1(P; K_1)$ with respect to $V^{(i)}$ is defined by $\Delta^{(i)}E_1(P; K_1)$ as follows

$$\Delta^{(i)}E_1(P; K_1) \equiv \sum_{\Delta P \in V^{(i)}}^{\oplus} E_1(P \oplus \Delta P; K_1). \quad (7)$$

$\Sigma^{\oplus}$ denotes a summation in XOR. If the algebraic degree of $E_1(P; K_1)$ with respect to P is N ($\le n$), the (N+1)th-order differential of $E_1(P; K_1)$ becomes zero regardless of P and K_1 as follows

$$\Delta^{(N+1)}E_1(P; K_1) = 0. \quad (8)$$

Moreover, if Boolean polynomial of $E_1(P; K_1)$ does not include the jth-order term of p_t ($1 \le t \le n$), jth-order differential of $E_1(P; K_1)$ with respect to $V^{(j)}$, which corresponds to the jth-order term of p_t, becomes zero regardless of P and K_1 as follows

$$\Delta^{(j)}E_1(P; K_1) = 0. \quad (9)$$

Since $E_1(P; K_1) = E_2^{-1}(C(P); K_2)$, which is the inverse function of E_2, (8) and (9) can be rewritten as

$$\Delta^{(N+1)}E_2^{-1}(C(P); K_2) \equiv \sum_{\Delta P \in V^{(N+1)}}^{\oplus} E_2^{-1}(C(P \oplus \Delta P); K_2) = 0 \quad (10)$$

and

$$\Delta^{(j)}E_2^{-1}(C(P); K_2) = 0. \quad (11)$$

(10) or (11) is always correct if K_2 is correct, while they are stochastically correct if K_2 is incorrect. This is why attacker can estimate K_2 and can check the correctness of K_2 by (10) or (11). The incorrect K_2 can be eliminated by solving some sets of (10) or (11) whose plain texts P are different from each other. Actually, attacker has to solve at least $\lceil q/m \rceil$ different sets of (10) or (11). Such attack using (10) or (11) is called a higher-order differential attack, and (10) and (11) are called attack equations.

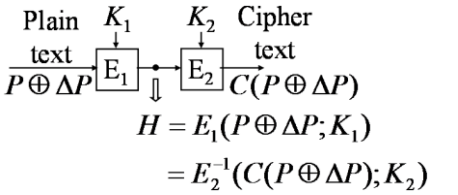

Figure 5. Block diagram of an encryption process.

3.2 The 32nd-Order Differential Characteristics and Attack Equation of MISTY2 without FLs

In this subsection, we describe the higher-order differential characteristics of MISTY2 without FLs and the attack equation exploiting the characteristics.

We show the two higher-order differential characteristics in Fig. 1. One has been known previously [9], and the other is found in this time. Previously, the 7th-order differential is put into the lower 7 bits of plain text P, while the remaining 57 bits are arbitrary constants. Namely, all 2^7 kinds of data from 0x00 to 0x7f are put into the lower 7 bits where the symbol "0x" represents that its following value is hexadecimal. In this case it has been known that the 7th-order differential of the lower 7 bits out of 32 bits becomes 0x6d at the point G in Fig. 1. This time, we put the 32nd-order differential into the upper 32 bits of plain text, while the lower 32 bits are arbitrary constants. Then we

experimentally found the characteristic, which is that the 32nd-order differential of the upper 23 bits out of 32 bits becomes zero at the point H in Fig. 1. By exploiting the characteristics we can derive the following equation corresponding to (10) or (11), and then can analyze the equivalent key $\mathrm{KI'}_{8jk}$ used in the equivalent FO_8.

$$\sum_{\Delta P \in V^{(32)}}^{\oplus} \left\{ \mathrm{FO}_8^{-1}(C_L \oplus C_R; \mathrm{KI'}_{8jk}) \right\}^{\mathrm{L23}} = 0, \quad (j, k = 1, 2, 3) \tag{12}$$

$$(C_L \| C_R) = C(P \oplus \Delta P) \tag{13}$$

where FO_8^{-1} represents an inverse function of FO_8. Attacker has to analyze total 75 bits of equivalent keys used in FI_{8j} and compute the 32nd-order differential at the points H_1 and H_2 in Fig. 2 to solve (12) because of $H = (H_1 \| H_2)$. On the other hand, attacker only has to analyze 50 bits of equivalent keys used in FI_{82} and FI_{83} and compute the 32nd-order differential at the point H_2 if we focus on the lower 7 bits of (12), which is given by

$$\sum_{\Delta P \in V^{(32)}}^{\oplus} \left\{ \mathrm{FI}_{82}^{-1}(I \oplus x_3; \mathrm{KI'}_{82k}) \right\}^{\mathrm{L7}} = 0 \tag{14}$$

where

$$I = \mathrm{FI}_{83}^{-1}(x_3 \oplus (C_L \oplus C_R)^{R16}; \mathrm{KI'}_{83k}), \quad x_3 = (C_L \oplus C_R)^{L16}. \tag{15}$$

FI_{82}^{-1} and FI_{83}^{-1} represent inverse functions of FI_{82} and FI_{83}, respectively. The variable I in (15) corresponds to data I in Fig. 2. We intend to solve (14) and (15) in the next section.

4 COST ESTIMATION OF THE ATTACK

In this section we estimate the number of blocks of chosen plain text and the number of times of FO operation required to solve (14) by an exhaustive search.

Because (14) is 7 sets of Boolean equation, it is satisfied with probability 2^{-7} even if an estimated key is false. There are 2^{50} candidates of the key since its total bit size is 50. Therefore attacker needs to solve 8 sets of (14) with different P in order to identify the true key where the probability that a false key survives is 2^{-6} (= $(2^{-7})^8 \times 2^{50}$). Because attacker has to compute the 32nd-order differential to prepare one set of (14), the number of blocks of chosen plain text to prepare 8 different sets of (14) is given by D as follows:

$$D = 8 \times 2^{32} = 2^{35}. \tag{16}$$

Next we study the number of times of FO_i operation required to solve 8 different sets of (14). If attacker solves the first set of (14) for all 2^{50} candidates of the key, the number of candidates is reduced to 2^{43} (= $2^{50} \times 2^{-7}$). And then he solves the second set of (14) for the remaining 2^{43} candidates. Its number is reduced to 2^{36} (= $2^{43} \times 2^{-7}$). By solving 8 different set of (14), the last remaining key will be the true key. (14) includes two functions (FI_{82}^{-1} and FI_{83}^{-1}). It is natural to assume that the computational complexity of FI_{82}^{-1} and FI_{83}^{-1} is equal to that of FI_{ij}. FO_i includes three sets of FI_{ij}. Therefore the total number of times of $\Sigma^{\oplus}$ operation in (14) times 2/3 corresponds to the total number of times of FO_i operation (T) required for this attack as follows:

$$T = 2^{50} \times 2^{32} \times \sum_{i=0}^{7} 2^{-7i} \times \frac{2}{3} \approx 2^{81.4}. \tag{17}$$

5 COMPLEXITY REDUCTION BY EXPLOITING A MOLULO 2 OCCURRENCE DISTRIBUTION

In this section, we show that the computational complexity of (14) can be reduced by exploiting a modulo 2 occurrence distribution (MOD) for intermediate data of (14), which is sequentially derived by using a partial sum technique proposed by Ferguson et al. [10]. The advantage of using MOD is as follows. Even number of times of XOR operations of a certain variable x is zero, while odd number of times of them is x. Therefore even number of XOR operations of x becomes unnecessary, and odd number of them can be substituted with x by using MOD.

Next we show the algorithm to reduce the complexity of (14). Fig. 6 shows the output part of FO_8. x_1, x_4, x_7, x_{3L}, x_{8L}, x_{9R}, x_{10}, x_{11}, and x_{14} denote 9 bits of intermediate data. x_2, x_5, x_{3R}, x_{8R}, x_{9L}, and x_{12} denote 7 bits of intermediate data. x_3 and x_8 denote 16 bits of intermediate data where $x_8 = x_{8L} \| x_{8R} = x_{9L} \| x_{9R}$. By using the intermediate data x_i in Fig. 6, (14) can be rewritten as

$$\sum_{\Delta P \in V^{(32)}}^{\oplus} \left\{ \mathrm{FI}_{82}^{-1}(I \oplus x_3; \mathrm{KI}'_{82k}) \right\}^{\mathrm{L7}} = \sum_{\Delta P \in V^{(32)}}^{\oplus} \left\{ S_9^{-1}(x_{14} \oplus \mathrm{KI}'_{821}) \right\}^{\mathrm{L7}} = 0, \quad (18)$$

$$x_{14} = x_{11} \oplus S_7^{-1}(x_{12} \oplus \mathrm{KI}'_{822}), \quad (19)$$

$$x_{12} = (x_{11} \oplus x_{9L})^{R7}, \quad (20)$$

$$x_{11} = S_9^{-1}(x_{10} \oplus \mathrm{KI}'_{823}), \quad (21)$$

$$x_{10} = x_{9L} \oplus x_{9R}, \quad (22)$$

$$x_{9L} \| x_{9R} = x_{8L} \| x_{8R}, \quad (23)$$

$$x_{9L} = (x_{8L})^{L7}, \quad x_{9R} = (x_{8L})^{R2} \| x_{8R}, \quad (24)$$

$$x_{8L} = x_{3L} \oplus S_9^{-1}(x_7 \oplus \mathrm{KI}'_{831}), \quad (25)$$

$$x_{8R} = x_{3R} \oplus S_7^{-1}(x_5 \oplus \mathrm{KI}'_{832}), \quad (26)$$

$$x_7 = x_4 \oplus S_7^{-1}(x_5 \oplus \mathrm{KI}'_{832}), \quad (27)$$

$$x_5 = (x_4 \oplus x_2)^{R7}, \quad (28)$$

$$x_4 = S_9^{-1}(x_1 \oplus \mathrm{KI}'_{833}), \quad (29)$$

$$x_{3L} \| x_{3R} = x_3, \quad (30)$$

$$x_1 = x_2 \oplus x_0, \quad (31)$$

$$x_2 \| x_0 = x_3 \oplus v, \quad (32)$$

$$x_3 \| v = C_L \oplus C_R. \quad (33)$$

Our attack algorithm to operate (13), (33) to (18) in the inverse order is as follows.

Attack Algorithm

1. Make MOD of total 32-bit data x_1, x_2, x_{3L}, and x_{3R} (referred to as MOD1) from 2^{32} blocks of cipher text $C(P \oplus \Delta P)$ via (13), (33), (32), (31), and (30). The maximum and the average number of the elements of MOD1 are 2^{32} and 2^{31}, respectively.
2. Guess KI'_{833}, and make MOD of total 32-bit data x_4, x_5, x_{3L}, and x_{3R} (MOD2) from MOD1 via (29) and (28) with at most 2^{32} times and the average 2^{31} times of S_9^{-1} operation. The maximum and the average number of the elements of MOD2 are 2^{32} and 2^{31}, respectively.
3. Guess KI'_{832}, and make MOD of total 25-bit data x_7, x_{8R}, and x_{3L} (MOD3) from MOD2 via (27) and (26) with at most 2^{32} times and the average 2^{31} times of S_7^{-1} operation. The maximum and the average number of the elements of MOD3 are 2^{25} and 2^{24}, respectively.
4. Guess KI'_{831}, and make MOD of total 16-bit data x_{10} and x_{9L} (MOD4) from MOD3 via (25), (24), (23), and (22) with at most 2^{25} times and the average 2^{24} times of S_9^{-1} operation. The maximum and the average

number of the elements of MOD4 are 2^{16} and 2^{15}, respectively.

5. Guess KI'_{823}, and make MOD of total 16-bit data x_{11} and x_{12} (MOD5) from MOD4 via (21) and (20) with at most 2^{16} times and the average 2^{15} times of S_9^{-1} operation. The maximum and the average number of the elements of MOD5 are 2^{16} and 2^{15}, respectively.
6. Guess KI'_{822}, and make MOD of 9-bit data x_{14} (MOD6) from MOD5 via (19) with at most 2^{16} times and the average 2^{15} times of S_7^{-1} operation. The maximum and the average number of the elements of MOD6 are 2^9 and 2^8, respectively.
7. Guess KI'_{821}, and compute (18) from MOD6 with at most 2^9 times and the average 2^8 times of S_9^{-1} operation, and confirm the authenticity of the guessed six keys KI'_{833}, KI'_{832}, KI'_{831}, KI'_{823}, KI'_{822}, KI'_{821} by (18).

We execute Step 1 one time. Steps 2, 3, …, 7 are executed by using a nested structure of loop iterations. Step 2 is the outermost loop, and step 7 is the innermost loop. We apply this algorithm for all the candidate keys, and confirm its authenticity by (18) as described in the previous section. The maximum number (T_{max}) and the average number (T_{av}) of times of S_i^{-1} operation for our attack algorithm are given by

$$T_{\max} = \sum_{i=0}^{7} T_2 \approx 2^{60.6}, \quad (34)$$

$$T_2 = 2^9(2^{32} + T_3),\ T_3 = 2^7(2^{32} + T_4), \quad (35)$$

$$T_4 = 2^9(2^{25} + T_5),\ T_5 = 2^9(2^{16} + T_6), \quad (36)$$

$$T_6 = 2^7(2^{16} + T_7),\ T_7 = 2^9 \times 2^9 \times 2^{-7i}, \quad (37)$$

and

$$T_{av} = \sum_{i=0}^{7} T_2' \approx 2^{59.6}, \quad (38)$$

$$T_2' = 2^9(2^{31} + T_3'),\ T_3' = 2^7(2^{31} + T_4'), \quad (39)$$

$$T_4' = 2^9(2^{24} + T_5'),\ T_5' = 2^9(2^{15} + T_6'), \quad (40)$$

$$T_6' = 2^7(2^{15} + T_7'),\ T_7' = 2^9 \times 2^8 \times 2^{-7i}. \quad (41)$$

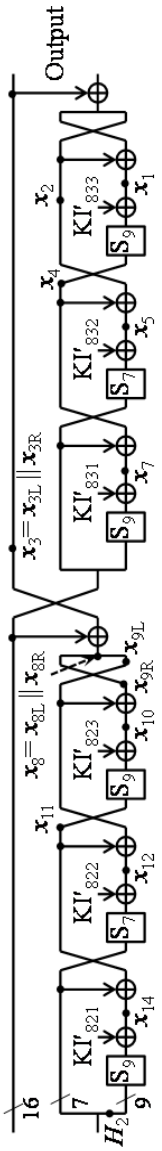

Figure 6. Output part of FO_8.

It is natural to assume that the computational complexity of S_i^{-1} is equivalent to that of S_i. T_i and T'$_i$ are correspond to the computational complexity of the nested structure of

steps i to 7 (i = 2, 3, …, 7). Because FO_i (i = 1, 2, …, 8) includes 9 sets of S_i (i = 7, 9), the maximum number (T'_{max}) and the average number (T'_{av}) of times of FO_i operation are given by

$$T'_{max} = \frac{T_{max}}{9} \approx 2^{57.4}, \quad T'_{av} = \frac{T_{av}}{9} \approx 2^{56.4} \quad (42)$$

In the previous section, attacker requires 2^{35} blocks of chosen plain text and $2^{81.4}$ times of FO_i operation. This attack algorithm has reduced the number of times of FO_i operation to up to $1/2^{25}$. Note that the number of blocks of chosen plain text required for this attack is 2^{35}.

6 CONCLUSIONS

We have study the 32nd-order differential attack on 8 rounds of MISTY2 without FL functions. We found the new 32nd-order differential characteristic of MISTY2 without FL functions. Using the characteristics and MOD, we showed that 8 rounds of MISTY2 without FL functions can be attacked with 2^{35} blocks of chosen plain text and $2^{57.4}$ times of FO operation.

REFERENCES

1. Dey, S.: An Image Encryption Method: SD-Advanced Image Encryption Standard: SD-AIES. In: International Journal of Cyber-Security and Digital Forensics, vol. 1, no. 2, pp. 82--88, The Society of Digital Information and Wireless Communications, (2013).
2. Dey, S.: Amalgamation of Cyclic Bit Operation in SD-EI Image Encryption Method: An Advanced Version of SD-EI Method: SD-EI VER-2, In: International Journal of Cyber-Security and Digital Forensics, vol. 1, no. 3, pp. 221--225, The Society of Digital Information and Wireless Communications, (2013).
3. Sulaiman, S., Muda, Z., Juremi, J., Mahmod, R., Yasin, S. Md.: A New Shiftcolumn Transformation: An Enhancement of Rijndael Key Scheduling, In: International Journal of Cyber-Security and Digital Forensics, vol. 1, no. 3, pp. 160--166, The Society of Digital Information and Wireless Communications, (2013).
4. Juremi, J., Mahmod, R., Sulaiman, S., Ramli, J.: Enhancing Advanced Encryption Standard S-box Generation Based on Round Key, In: International Journal of Cyber-Security and Digital Forensics, vol. 1, no. 3, pp. 183--188, The Society of Digital Information and Wireless Communications, (2013).
5. Abusukhon, A., Talib, M., Ottoum, I.: Secure Network Communication Based on Text-to-image Encryption, In: International Journal of Cyber-Security and Digital Forensics, vol. 1, no. 4, pp. 263--271, The Society of Digital Information and Wireless Communications, (2013).
6. Matsui, M.: New Block Encryption Algorithm MISTY. In: Lecture Notes in Computer Science, vol. 1267, pp. 54--68, Springer, (1997).
7. Lai, X.: Higher Order Derivatives and Differential Cryptanalysis. Communications and Cryptography, vol. 276, pp. 227--233, Springer, (1994).
8. Sugita, M.: Higher Order Differential Attack of Block Ciphers MISTY1, 2. In: IEICE Tech. Report, vol. 98, no. 48, pp. 31--40, (1998).
9. Hatano, Y., Tanaka, H., Kaneko, T.: Higher Order Differential Attack of MISTY2 without FL Functions. In: Proc. The International Conference on Fundamentals of Electronics, Communications and Computer Sciences, Sect. 17, pp. 6--10, (2002).
10. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Lecture Notes in Computer Science, vol. 1978, pp. 136--141, Springer, (2001).