The Improved 96th-Order Differential Attack on 11 Rounds of the Block Cipher CLEFIA

Yasutaka Igarashi, Seiji Fukushima, and Tomohiro Hachino
Kagoshima University, Kagoshima, Japan
Email: {igarashi, fukushima, hachino}@eee.kagoshima-u.ac.jp

Toshinobu Kaneko
Tokyo University of Science, Chiba, Japan
Email: kaneko@ee.noda.tus.ac.jp

***Abstract*—CLEFIA is a 128-bit block cipher proposed by Shirai of SONY et al. in 2007. Its key size is 128, 192, or 256 bits. The number of the round of data processing part depends on a key size, viz. it is 18, 22, or 26 rounds for 128, 192, or 256 bits of a key size, respectively. Such a characteristic of CLEFIA have been known that the 96th-order differential of 64 bits out of 128 bits of the 8th-round's output is zero. With this characteristic, we reported the 96th-order differential attack on 11 rounds of CLEFIA that requires $2^{98.3}$ blocks of plain text and 2^{159} times of data encryption. In this paper, we reduce this number of the times of the encryption, (viz. computational complexity) by applying a partial sum technique proposed by Ferguson et al. With the technique, we sequentially derive a modulo 2 occurrence distribution of intermediate data of cryptanalysis. We also reduce the complexity by introducing a nested structure of iterative computations to the attack algorithm. As a result we reduce the complexity to $2^{106.6}$, which is $1/2^{52.4}$ of the conventional complexity.**

***Index Terms*—cryptanalysis, higher-order differential attack, block cipher, CLEFIA**

I. Introduction

CLEFIA is a 128-bit block cipher proposed by Shirai of SONY et al. in 2007 [1], [2]. A data processing part of CLEFIA consists of some rounds of 4-branch type-2 generalized Feistel structure, where two similar nonlinear functions are placed in parallel. CLEFIA supports 128, 192, and 256 bits of secret keys. The number of round of data processing part depends on a key size, viz. it is 18, 22, or 26 rounds for 128, 192, or 256 bits of secret key, respectively. CLEFIA has been applied and evaluated for cryptographic techniques towards the revision of the e-Government Recommended Ciphers List in FY 2013 in Japan [3].

It has been known that the synchronous 8th-order differential of 24 bits out of 128 bits becomes zero at the 6th-round output of the data processing part [4]. It has been also known that the 96th-order differential of 64 bits out of 128 bits becomes zero at the 8th-round output [5]. Exploiting this property we reported the 96th-order differential attack [6] on 11 rounds of CLEFIA with $2^{98.3}$ blocks of plain text and 2^{159} times of data processing [7]. In the article we reduce the number of times of data processing for the 96th-order differential attack by using a modulo 2 occurrence distribution (MOD) of the intermediate data of data processing part. MOD is derived by using a partial sum technique proposed by Ferguson *et al*. [8]. Moreover we reduce the number of the times by applying a nested structure of iterative computation to the attack algorithm and optimizing the order of the nest.

As a result we show that the 96th-order differential attack can be performed with $2^{99.8}$ blocks of plain text and $2^{106.6}$ times of data processing, which is $1/2^{52.4}$ of computational complexity compared to the conventional attack. Note that the best known higher-order differential attack on CLEFA is the 105th-order differential attack on 14 rounds of CLEFIA with $2^{108.5}$ blocks of plain text and $2^{223.0}$ times of data processing so far [9].

II. Data Processing Part of CLEFIA

In this section we describe the data processing part of CLEFIA. We omit the detail specification of CLEFIA [2] being not required for the article.

Fig. 1 shows an 11-round data processing part of CLEFIA where input plain text and output cipher text are 128 bits represented by X_i and $C_i^{(11)}$ (i = 0, 1, 2, 3), respectively. A bit size of X_i and $C_i^{(11)}$ is 32. $C_i^{(j)}$ (j = 1, 2, …, 11) represents 32-bit output of the jth round given by

$$C_i^{(j)} = (C_{i0}^{(j)}, C_{i1}^{(j)}, C_{i2}^{(j)}, C_{i3}^{(j)})^T \quad (1)$$

where $C^{(j)}_{ik}$ (k= 0, 1, 2, 3) is an 8-bit data. The superscript T represents transposition of a vector or a matrix. The symbol $\oplus$ represents an XOR operation. WK_i (i = 0, 1) is a 32-bit whitening key, and RK_ℓ (i = 0, 1, 2, …, 21) is a 32-bit round key used in F_i (i = 0, 1). F_i is a bijective nonlinear function with 32-bit input/output (I/O) adopting SP structure. 4-branch 32-bit data X_i is mixed by XOR and substituted by a nonlinear function. Such a structure is called 4-branch generalized Feistel.

Manuscript received April 28, 2013; revised July 17, 2013

doi: 10.12720/ijoee.1.3.147-151

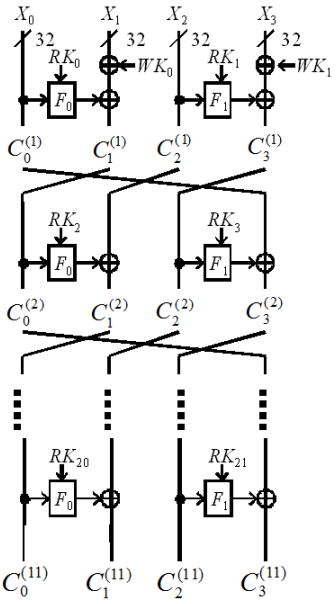

Figure 1. 11-round data processing part of CLEFIA.

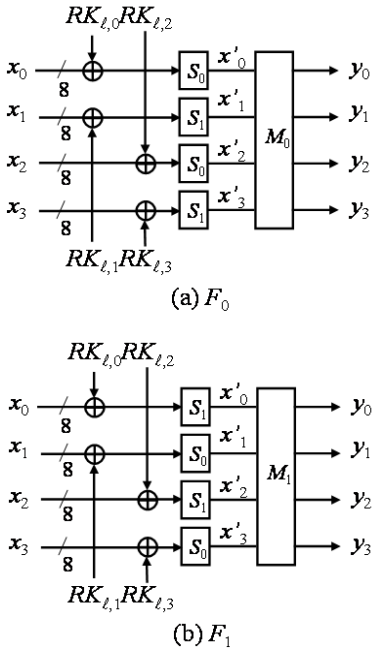

Figure 2. (a) F0 and (b) F_1.

Fig. 2 shows (a) F_0 and (b) F_1 where its input and output data are represented by an 8-bit x_i and an 8-bit y_i (i = 0, 1, 2, 3), respectively. $RK_{\ell,i}$ (i = 0, 1, 2, 3) is an 8-bit key satisfying the following equation: $RK_\ell = (RK_{\ell,0}, RK_{\ell,1}, RK_{\ell,2}, RK_{\ell,3})^T$. S_0 and S_1 are bijective nonlinear functions with 8-bit I/O. Their output is represented by x'_i. M_0 and M_1 are 4×4 nonsingular matrices given by

$$\begin{aligned}&(y_0 \quad y_1 \quad y_2 \quad y_3)^T \\ &= M_i(x'_0 \quad x'_1 \quad x'_2 \quad x'_3)^T \qquad (i = 0, 1),\end{aligned} \tag{2}$$

$$M_0 = \begin{pmatrix} M_{00} \\ M_{01} \\ M_{02} \\ M_{03} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 & 6 \\ 2 & 1 & 6 & 4 \\ 4 & 6 & 1 & 2 \\ 6 & 4 & 2 & 1 \end{pmatrix} \tag{3}$$

$$M_1 = \begin{pmatrix} M_{10} \\ M_{11} \\ M_{12} \\ M_{13} \end{pmatrix} = \begin{pmatrix} 1 & 8 & 2 & a \\ 8 & 1 & a & 2 \\ 2 & a & 1 & 8 \\ a & 2 & 8 & 1 \end{pmatrix} \tag{4}$$

where "a" is a hexadecimal number. The multiplications of a vector and a matrix are performed in GF(2^8) defined by the primitive polynomial $z^8 + z^4 + z^3 + z^2 + 1$.

III. The 96th-Order Differential Characteristics of CLEFIA and Its Attack Equation

In this section we show the 96th-order differential characteristics of CLEFIA and its attack equation.

First we focus on an input plain text (see Fig. 1). We set X_0 or X_2 to take arbitrary constant, and input the 96th-order differential to the three remaining X_i. Namely, all the 2^{96} kinds of data from 0 to 2^{96} -1 are put into the three X_i. At this time the 96th-order differentials of $C_{10}^{(9)}$ and $S_0(C_{00}^{(9)} \oplus RK_{16,0})$ becomes the same value [7] (see Fig. 3 that shows the equivalently modified data processing part from the 9th round to the 11th round). M_0^{-1} is the inverse matrix of M_0 that is given by $M_0^{-1} = M_0$. There is the product of the matrices M_0^{-1} and M_1 at the right side of the 11th round that is given by

$$M_0^{-1} \cdot M_1 = \begin{pmatrix} 25 & 2e & 22 & 28 \\ 2e & 25 & 28 & 22 \\ 22 & 28 & 25 & 2e \\ 28 & 22 & 2e & 25 \end{pmatrix} \tag{5}$$

By using the 96th-order differential characteristics the following attack equations can be derived:

$$\sum_{\Delta X \in V^{(96)}} {}^{\oplus} S_0(C_{00}^{(9)} \oplus RK_{16,0}) = \sum_{\Delta X \in V^{(96)}} {}^{\oplus} C_{10}^{(9)} \tag{6}$$

$$C_{10}^{(9)} = C_{00}^{(10)} = W_{20}^{(11)} \oplus 28\, S_0(C_{23}^{(11)} \oplus RK_{21,3}) \tag{7}$$

$$W_{20}^{(11)} = W_{10}^{(11)} \oplus 22\, S_1(C_{22}^{(11)} \oplus RK_{21,2}) \tag{8}$$

$$W_{10}^{(11)} = W_{00}^{(11)} \oplus 2e\, S_0(C_{21}^{(11)} \oplus RK_{21,1}) \tag{9}$$

$$W_{00}^{(11)} = C'^{(11)}_{30} \oplus 25\, S_1(C_{20}^{(11)} \oplus RK_{21,0}) \tag{10}$$

$$C'^{(11)}_{30} = C_{30}^{(11)} \oplus 2C_{31}^{(11)} \oplus 4C_{32}^{(11)} \oplus 6C_{33}^{(11)} \tag{11}$$

$$C_{00}^{(9)} = W_{20}^{(10)} \oplus a\, S_0(C_{23}^{(10)} \oplus RK_{19,3}) \tag{12}$$

$$W_{20}^{(10)} = W_{10}^{(10)} \oplus 2\, S_1(C_{22}^{(10)} \oplus RK_{19,2}) \tag{13}$$

$$W_{10}^{(10)} = W_{00}^{(10)} \oplus 8\, S_0(C_{21}^{(10)} \oplus RK_{19,1}) \tag{14}$$

$$W_{00}^{(10)} = C_{30}^{(10)} \oplus S_1(C_{20}^{(10)} \oplus RK_{19,0}), \quad C_{30}^{(10)} = C_{20}^{(11)} \tag{15}$$

$$C_2^{(10)} = U_2^{(11)} \oplus (6,4,2,1)^T S_1(C_{03}^{(11)} \oplus RK_{20,3}) \quad (16)$$

$$U_2^{(11)} = U_1^{(11)} \oplus (4,6,1,2)^T S_0(C_{02}^{(11)} \oplus RK_{20,2}) \quad (17)$$

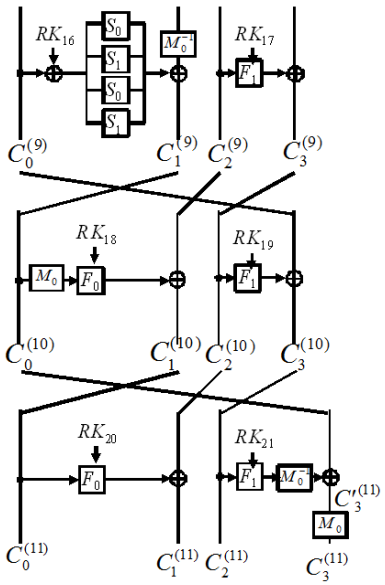

Figure 3. Equivalently-modified data processing part from the 9th round to the 11th round.

$$U_1^{(11)} = U_0^{(11)} \oplus (2,1,6,4)^T S_1(C_{01}^{(11)} \oplus RK_{20,1}) \quad (18)$$

$$U_0^{(11)} = C_1^{(11)} \oplus (1,2,4,6)^T S_0(C_{00}^{(11)} \oplus RK_{20,0}) \quad (19)$$

where

$$U_i^{(j)} = (U_{i0}^{(j)}, U_{i1}^{(j)}, U_{i2}^{(j)}, U_{i3}^{(j)})^T \quad (20)$$

$$W_i^{(j)} = (W_{i0}^{(j)}, W_{i1}^{(j)}, W_{i2}^{(j)}, W_{i3}^{(j)})^T \quad (21)$$

$$C^{(j)}(X \oplus \Delta X) = \begin{pmatrix} C_0^{(j)} \\ C_1^{(j)} \\ C_2^{(j)} \\ C_3^{(j)} \end{pmatrix} \quad (22)$$

$C^{(j)}(X \oplus \Delta X)$ is the jth-round 128-bit output corresponding to the input plain text $(X \oplus \Delta X)$ where ΔX is an input difference. $U_i^{(j)}$ and $W_i^{(j)}$ are 32-bit intermediate data of the equivalent circuit shown in Fig. 4 and Fig. 5 where the hexadecimal number in a box represents multiplication. Because (6) is 8 sets of Boolean equation, it holds with probability 2^{-8} even if a guessed key is false. On the other hand it holds with probability 1 if the guessed key is correct. Therefore attacker can recover the total 104 bits of the round keys, $RK_{16,0}$, RK_{19}, RK_{20}, and RK_{21} in (6)-(22) by analyzing (6)-(22) for sufficient number of times.

IV. Attack Algorithm and Cost Estimation

In this section we show the attack algorithm to recover the total 104 bits of round keys, and estimate the cost of the attack. We reduce the computational complexity of the attack by exploiting MOD for intermediate data of (6)-(22), which is sequentially derived by using a partial sum technique proposed by Ferguson *et al.* [8]. The advantages of using MOD are as follows. Even number of XOR operations of a certain variable x is zero, while

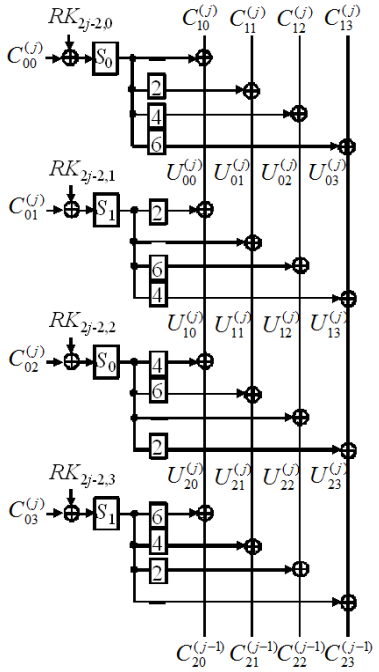

Figure 4. Equivalent circuit whose inputs are $C_0^{(j)}$, $C_1^{(j)}$, and RK_{2j-2}. The output is $C_2^{(j-1)}$.

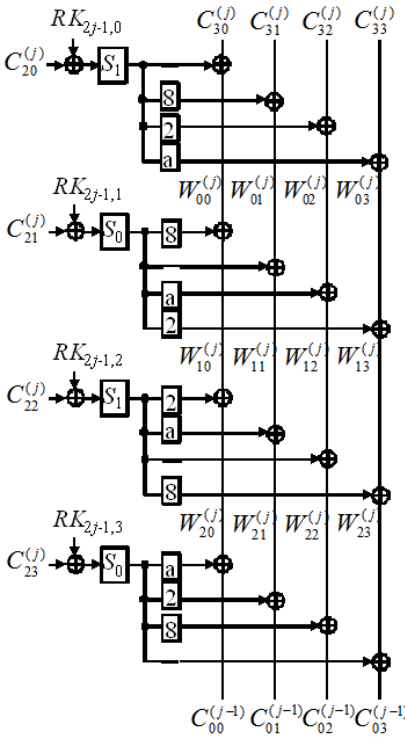

Figure 5. Equivalent circuit whose inputs are $C_2^{(j)}$, $C_3^{(j)}$, and RK_{2j-1}. The output is $C_0^{(j-1)}$.

Odd number 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. These advantages result in complexity reduction.

Next we show the attack algorithm exploiting MOD as follows.

A. Attack Algorithm

0) Make MOD of the total 72-bit data, $C_0^{(11)}$, $C_1^{(11)}$, and $C_{20}^{(11)}$ (= $C_{30}^{(10)}$) (name as MOD1) and MOD of the total 40-bit data, $C_2^{(11)}$ and $C'_{30}{}^{(11)}$ (name as MOD2) from the 2^{96} blocks of cipher text $C^{(11)}(X \oplus \Delta X)$ where $\Delta X \in V^{(96)}$. The number of elements of MOD1 is at most 2^{72} and the average is 2^{71}. The number of elements of MOD2 is at most 2^{40} and the average is 2^{39}.

1) Guess 8-bit $RK_{20,0}$, and make MOD of the total 64-bit data, $C_{01}^{(11)}$, $C_{02}^{(11)}$, $C_{03}^{(11)}$, $U_0^{(11)}$, and $C_{30}^{(10)}$ (MOD3) from MOD1 through the equivalent circuit shown in Fig. 4 with at most 2^{72} times and the average 2^{71} times of S_0 operation and the following multiplications, which corresponds to (19). The number of elements of MOD3 is at most 2^{64} and the average is 2^{63}.

2) Guess 8-bit $RK_{20,1}$, and make MOD of the total 56-bit data, $C_{02}^{(11)}$, $C_{03}^{(11)}$, $U_1^{(11)}$, and $C_{30}^{(10)}$ (MOD4) from MOD3 through the equivalent circuit shown in Fig. 4 with at most 2^{64} times and the average 2^{63} times of S_1 operation and the following multiplications, which corresponds to (18). The number of elements of MOD4 is at most 2^{56} and the average is 2^{55}.

3) Guess 8-bit $RK_{20,2}$, and make MOD of the total 48-bit data, $C_{03}^{(11)}$, $U_2^{(11)}$, and $C_{30}^{(10)}$ (MOD5) from MOD4 through the equivalent circuit shown in Fig. 4 with at most 2^{56} times and the average 2^{55} times of S_0 operation and the following multiplications, which corresponds to (17). The number of elements of MOD5 is at most 2^{48} and the average is 2^{47}.

4) Guess 8-bit $RK_{20,3}$, and make MOD of the total 40-bit data, $C_2^{(10)}$ and $C_{30}^{(10)}$ (MOD6) from MOD5 through the equivalent circuit shown in Fig. 4 with at most 2^{48} times and the average 2^{47} times of S_1 operation and the following multiplications, which corresponds to (16). The number of elements of MOD6 is at most 2^{40} and the average is 2^{39}.

5) Guess 8-bit $RK_{19,0}$, and make MOD of the total 32-bit data, $C_{21}^{(10)}$, $C_{22}^{(10)}$, $C_{23}^{(10)}$, and $W_{00}^{(10)}$ (MOD7) from MOD6 through the equivalent circuit shown in Fig. 5 with at most 2^{40} times and the average 2^{39} times of S_1 operation corresponding to (15). The number of elements of MOD7 is at most 2^{32} and the average is 2^{31}.

6) Guess 8-bit $RK_{21,0}$, and make MOD of the total 32-bit data, $C_{21}^{(11)}$, $C_{22}^{(11)}$, $C_{23}^{(11)}$, and $W_{00}^{(11)}$ (MOD8) from MOD2 through the equivalent circuit shown in Fig. 5 with at most 2^{40} times and the average 2^{39} times of S_1 operation and the following multiplication, which corresponds to (10). The number of elements of MOD8 is at most 2^{32} and the average is 2^{31}.

7) Guess 8-bit $RK_{19,1}$, and make MOD of the total 24-bit data, $C_{22}^{(10)}$, $C_{23}^{(10)}$, and $W_{10}^{(10)}$ (MOD9) from MOD7 through the equivalent circuit shown in Fig. 5 with at most 2^{32} times and the average 2^{31} times of S_0 operation and the following multiplication, which corresponds to (14). The number of elements of MOD9 is at most 2^{24} and the average is 2^{23}.

8) Guess 8-bit $RK_{21,1}$, and make MOD of the total 24-bit data, $C_{22}^{(11)}$, $C_{23}^{(11)}$, and $W_{10}^{(11)}$ (MOD10) from MOD8 through the equivalent circuit shown in Fig. 5 with at most 2^{32} times and the average 2^{31} times of S_0 operation and the following multiplication, which corresponds to (9). The number of elements of MOD10 is at most 2^{24} and the average is 2^{23}.

9) Guess 8-bit $RK_{19,2}$, and make MOD of the total 16-bit data, $C_{23}^{(10)}$ and $W_{20}^{(10)}$ (MOD11) from MOD9 through the equivalent circuit shown in Fig. 5 with at most 2^{24} times and the average 2^{23} times of S_1 operation and the following multiplication, which corresponds to (13). The number of elements of MOD11 is at most 2^{16} and the average is 2^{15}.

10) Guess 8-bit $RK_{21,2}$, and make MOD of the total 16-bit data, $C_{23}^{(11)}$ and $W_{20}^{(11)}$ (MOD12) from MOD10 through the equivalent circuit shown in Fig. 5 with at most 2^{24} times and the average 2^{23} times of S_1 operation and the following multiplication, which corresponds to (8). The number of elements of MOD12 is at most 2^{16} and the average is 2^{15}.

11) Guess 8-bit $RK_{19,3}$, and make MOD of the total 8-bit data $C_{00}^{(9)}$ (MOD13) from MOD11 through the equivalent circuit shown in Fig. 5 with at most 2^{16} times and the average 2^{15} times of S_0 operation and the following multiplication, which corresponds to (12). The number of elements of MOD13 is at most 2^8 and the average is 2^7.

12) Guess 8-bit $RK_{21,3}$, and compute the right-hand side of (6) via (7) from MOD12 through the equivalent circuit shown in Fig. 5 with at most 2^{16} times and the average 2^{15} times of S_0 operation and the following multiplication.

13) Guess 8-bit $RK_{16,0}$, and compute the left-hand side of (6) from MOD13 with at most 2^8 times and the average 2^7 times of S_0 operation. The guessed keys at the steps 1-13 are the candidate of correct key if (6) holds, otherwise they are false keys.

Attacker guesses all the 2^{104} kinds of the round key at the steps 1-13. Attacker executes step 0 one time. Steps 1-13 are executed by using a nested structure of loop iterations.

Step 1 is an outermost loop, and step 13 is an innermost loop.

First, attacker confirms the authenticity of the total 2^{104} sets of the guessed key by computing (6). And then, the number of candidate keys is reduced to 2^{96} (= $2^{104} \times 2^{-8}$). Second, attacker confirms the authenticity of 2^{96} sets of the guessed keys by computing (6) again where the plain text X is different from the one at the first time. And then, the number of candidate keys is reduced to 2^{88}. Attacker repeats this confirmation 14 times, and then the average number of candidate keys is reduced to 2^{-8} (= $2^{104} \times (2^{-8})^{14}$) where the last key shall be a correct key. Because attacker has to compute the 96ht-order differential to prepare one set of (6), the number of chosen plain texts to prepare 14 different sets of (6) is given by D as follows:

$$D = 14 \times 2^{96} \approx 2^{99.8} \quad (23)$$

The maximum number (T_{max}) of data processing of 11-round CLEFIA, which include 88 sets of S_i (i = 0, 1), for the attack algorithm are given by

$$T_{max} = \sum_{i=0}^{13} 2^{-8i} T_1 / 88 \approx 2^{106.6} \quad (24)$$

$$T_1 = 2^8(2^{72} + T_2),\ T_2 = 2^8(2^{64} + T_3),\ T_3 = 2^8(2^{56} + T_4) \quad (25)$$

$$T_4 = 2^8(2^{48} + T_5),\ T_5 = 2^8(2^{40} + T_6),\ T_6 = 2^8(2^{40} + T_7) \quad (26)$$

$$T_7 = 2^8(2^{32} + T_8),\quad T_8 = 2^8(2^{32} + T_9) \quad (27)$$

$$T_9 = 2^8(2^{24} + T_{10}),\quad T_{10} = 2^8(2^{24} + T_{11}) \quad (28)$$

$$T_{11} = 2^8(2^{16} + T_{12}),\ T_{12} = 2^8(2^{16} + T_{13}),\ T_{13} = 2^8 \cdot 2^8. \quad (29)$$

T_i (i = 1, 2, …, 13) is the number of S_i (i = 0, 1) operations for the nested loops from step i to step 13.

V. Conclusioins

We have studied the 96th-order differential attack on 11 rounds of CLEFIA. We reduced the number of times of data processing for the attack by applying a partial sum technique proposed by Ferguson et al. With the technique, we sequentially derived MOD of intermediate data of the data processing part. Moreover we reduced the number of the times by applying a nested structure of iterative computation to the attack algorithm and optimizing the order of the nest. As a result we showed that the 96th-order differential attack can be performed with $2^{99.8}$ blocks of plain text and $2^{106.6}$ times of data processing, which is $1/2^{52.4}$ of computational complexity compared to the conventional attack. The future work is to apply our attack algorithm to [4].

References

[1] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA (Extended abstract)," in *Lecture Notes in Computer Science*, Springer-Verlag, vol. 4593, pp. 181-195, 2007.

[2] CLEFIA The 128-bit Blockcipher. [Online]. Available: http://www.sony.net/Products/cryptography/clefia/

[3] CRYPTREC topics. [Oline]. Available: http://www.cryptrec.go.jp/english/topics/cryptrec_20101001_callforattack.html

[4] N. Shibayama and T. Kaneko, "A peculiar higher order differential of CLEFIA," in *Proc. International Symposium on Information Theory and its Applications*, 2012, pp. 526-530.

[5] Y. Tsunoo, E. Tsujihara, H. Kubo, M. Shigeri, and T. Kawabata, "Saturation characteristics of generalized Feistel structure," *IEICE Trans. Fundamentals*, vol. J93-A, no. 4, pp. 269-276. 2010.

[6] X. Lai, "Higher order derivatives and differential cryptanalysis," in *Communications and Cryptography*, Springer US, vol. 276, pp. 227-233, 1994.

[7] N. Shibayama, Y. Igarashi, T. Kaneko, and S. Hangai, "Security evaluation of CLEFIA against saturation cryptanalysis," in *Proc. Symposium on Cryptography and Information Security*, no. 2B1-4, 2011.

[8] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved cryptanalysis of Rijndael," in *Lecture Notes in Computer Science*, Springer, vol. 1978, 2001, pp. 136-141.

[9] N. Shibayama and T. Kaneko, "New higher order differential property of CLEFIA," in *Proc Symposium on Cryptography and Information Security*, no. 2B4-2, 2013.

Yasutaka Igarashi received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Saitama University, Japan, in 2000, 2002, and 2005. He is currently an assistant professor of Kagoshima University. His research is involved with optical CDMA and the cryptanalysis of symmetric-key cryptography. Dr. Igarashi is a member of IEICE and RISP.

Toshinobu Kaneko received the B.E., M.E., and Ph.D. degrees in Electrical Engineering from the University of Tokyo, in 1971, 1973, and 1976, respectively. He is currently a Professor of Tokyo University of Science. He has been engaged in coding theory and information security. Prof. Kaneko is a member of CRYPTREC and served as a chairman of Symmetric-Key Cryptography subcommittee in 2001--2003. Prof. Kaneko is a member of IEICE, IEEJ, IPSJ, and IEEE.

Seiji Fukushima received the B.S., M.S., and Ph.D. degrees in electrical engineering from Kyushu University in 1984, 1986, and 1993, respectively. He is currently a Professor at Kagoshima University. His research interests include photonics/radio hybrid communication systems and their related devices. Prof. Fukushima is a member of IEICE, IEEE/Photonic Society, Japan Society of Applied Physics, Japanese Liquid Crystal Society, and Optical Society of America.

Tomohiro Hachino received the B.S., M.S., and Dr. Eng. degrees in electrical engineering from Kyushu Institute of Technology in 1991, 1993, and 1996, respectively. He is currently an Associate Professor at Kagoshima University. His research interests include nonlinear control and identification, signal processing, and evolutionary computation. Dr. Hachino is a member of IEEJ, SICE, and ISCIE.