A Fast Calculus for the Linearizing Attack and Its Application to an Attack on KASUMI

Nobuyuki Sugio[1], Shunichi Nambu[2], and Toshinobu Kaneko[2]

[1] NTT DoCoMo, 3-5 Hikari-no-oka, Yokosuka, Kanagawa 239-8536, Japan
sugio@nttdocomo.co.jp
[2] Tokyo University of Science, 2641, Yamazaki, Noda, Chiba 278-8510, Japan
j7303655@ed.noda.tus.ac.jp, kaneko@ee.noda.tus.ac.jp

Abstract. This paper describes a linearizing attack with fast calculus for higher order differential attack. The linearizing attack, proposed by Shimoyama et al. [13], [15], linearizes the attack equation and determines the key by Gaussian elimination. The cost of calculating the coefficient matrix is dominant overhead in this attack. We improve the algorithm used to calculate the coefficient matrix by applying a bit-slice type implementation [3]. We apply this method to five-round KASUMI and show that it need $2^{27.5}$ chosen plaintexts and 2^{34} KASUMI encryptions.

1 Introduction

Higher order differential attack is a well-known attack against block ciphers. It exploits the properties of the higher order differentials of functions and derives an attack equation to determine the key. Jakobsen et al. applied it to $\mathcal{KN}$ cipher [8]. They used exhaustive search to solve the attack equation. Shimoyama et al. proposed an effective method of solving the attack equation [15] and Moriai et al. generalized it for the attack on CAST cipher [13]. Their method, which we call *linearizing attack* in this paper, linearizes the attack equation and solves the key by using Gaussian elimination. Hatano et al. proposed an optimization for linearizing attack [6] that is based on linear dependency between unknowns in the attack equation; it decreases the number of independent variables.

In the linearizing attack, the major computational cost is estimated to be the cost of calculating the coefficient matrix of unknown variables.[1] In this paper, we propose a fast calculus for an enhanced linearizing attack. We apply the bit-slice type implementation proposed by Biham [3] to the attack equation to calculate the coefficient matrix. We analyze elements of the coefficient matrix of unknown variables and calculate it using the T function proposed in this paper instead of a round function.

We apply the fast calculus to attack the 64-bit block cipher KASUMI. KASUMI [1] is based on the known block cipher MISTY1 [11] and is optimized for

[1] If the size of coefficient matrix is small, this computational cost ignores the complexity of solving the system of equations [13].

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 163–172, 2006.

Table 1. Comparison to previous attacks on KASUMI

Cipher	Rounds	Complexity		Comments
		Data	Time	
KASUMI	4*1	$2^{10.5}$	$2^{22.11}$	Higher Order Differential Attack [18]
	5	$2^{39.4}$	2^{117}	Higher Order Differential Attack [19]
	5	$2^{22.1}$	$2^{60.7}$	Higher Order Differential Attack [16]
	5	$2^{27.5}$	$2^{39.9}$	Higher Order Differential Attack [20]
	5	$2^{27.5}$	2^{34}	**This paper**

*1-this attack is on a version of the cipher without FL functions.
Time complexity is measured in encryption units.

implementation in hardware. It is used in the confidentiality and integrity algorithm of 3GPP mobile communications. Table 1 lists the known attacks on KASUMI. Our method needs $2^{27.5}$ chosen plaintexts and 2^{34} KASUMI encryptions.

2 Preliminaries

2.1 Higher Order Differential [10]

Let $F(\cdot)$ be an encryption function as follows.

$$Y = F(X; K) \tag{1}$$

where $X \in \mathrm{GF}(2)^n$, $Y \in \mathrm{GF}(2)^m$, and $K \in \mathrm{GF}(2)^s$. X, K, and Y denote a plaintext, a key and a ciphertext, respectively. Let $\{A_1, \cdots, A_i\}$ be a set of linearly independent vectors in $\mathrm{GF}(2)^n$ and $V^{(i)}$ be the sub-space spanned by these vectors. The i-th order differential is defined as follows.

$$\Delta^{(i)}_{V^{(i)}} F(X; K) = \bigoplus_{A \in V^{(i)}} F(X \oplus A; K) \tag{2}$$

In the following, $\Delta^{(i)}$ denotes $\Delta^{(i)}_{V^{(i)}}$, when it is clearly understood.

In this paper, we use the following properties of the higher order differential.

Property 1. If the degree of $F(X; K)$ with respect to X equals N, then

$$deg_X\{F(X; K)\} = N \Rightarrow \begin{cases} \Delta^{(N+1)} F(X; K) = 0 \\ \Delta^{(N)} F(X; K) = const \end{cases} \tag{3}$$

Property 2. The higher order differential has linear property on Exclusive-OR sum.

$$\Delta^{(N)}\{F(X_1; K_1) \oplus F(X_2; K_2)\} = \Delta^{(N)} F(X_1; K_1) \oplus \Delta^{(N)} F(X_2; K_2) \tag{4}$$

2.2 Attack of a Block Cipher

Consider an R-round block cipher. Let $H_{R-1}(X) \in \mathrm{GF}(2)^m$ be a part of the $(R-1)$-th round output and $C(X) \in \mathrm{GF}(2)^m$ be the ciphertext for the plaintext $X \in \mathrm{GF}(2)^n$. $H_{R-1}(X)$ is described as follows.

$$H_{R-1}(X) = F_{R-1}(X; K_1, \cdots, K_{R-1}) \tag{5}$$

Let K_i be an i-th round key and $F_i(\cdot)$ be a function of $\mathrm{GF}(2)^n \times \mathrm{GF}(2)^{s \times i} \rightarrow \mathrm{GF}(2)^m$.

If the degree of $F_{R-1}(\cdot)$ with respect to X is $N-1$, we have

$$\Delta^{(N)} H_{R-1}(X) = 0 \tag{6}$$

Let $\tilde{F}(\cdot)$ be a function that outputs $H_{R-1}(X)$ from the ciphertext $C(X) \in \mathrm{GF}(2)^m$.

$$H_{R-1}(X) = \tilde{F}(C(X); K_R) \tag{7}$$

where $K_R \in \mathrm{GF}(2)^s$ denotes the round key to decode $H_{R-1}(X)$ from $C(X)$. From Eq. (6), (7) and Property 1, the following equation holds.

$$0 = \Delta^{(N)} \tilde{F}(C(X); K_R) \tag{8}$$

In the following, we refer to Eq. (8) as the attack equation.

2.3 Linearizing Attack

Shimoyama et al. proposed an effective method of solving attack Eq. (8) [13], [15]. This method, called linearizing attack in this paper, linearizes the attack equation by treating every higher order variable like $k_i k_j$ with new independent variables like k_{ij}. In the following, we use the term *linearized attack equation* to refer to an attack equation that is regarded as a linear equation.

Let L be the number of unknowns in the linearized attack equation of Eq. (8). Since the attack Eq. (8) is derived by using an m-bit sub-block, we can rewrite it as follows.

$$\mathbf{Ak} = \mathbf{b}\ ,\ \mathbf{k} = {}^t(k_1, k_2, \ldots, k_1k_2, \ldots, k_1k_2k_3, \cdots) \tag{9}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{k}$ are the $m \times L$ coefficient matrix, the m-dimensional vector, and the L-dimensional vector over GF(2), respectively. $\mathbf{k}$ denotes linearized unknowns that are expressed as monomials of the R-th round key K_R.

We can obtain m linearized attack equations from one N-th order differential because Eq. (8) is an m-bit equation. Therefore, we need $\lfloor L/m \rfloor$ sets of the N-th order differential to determine a unique solution.

Since one set of N-th order differential requires 2^N chosen plaintexts, the number of plaintexts, M, needed to determine the key is estimated as

$$M = 2^N \times \left\lfloor \frac{L}{m} \right\rfloor \tag{10}$$

If we use the same technique shown in [13], [15], Eq. (9) requires $2^N \times (L+1)$ times $\tilde{F}(\cdot)$ calculations. Since we have to prepare $\lfloor L/m \rfloor$ sets of N-th order differentials to determine $\mathbf{k}$, the computational cost is estimated as

$$T = 2^N \times (L+1) \times \left\lfloor \frac{L}{m} \right\rfloor \tag{11}$$

3 Fast Calculus for the Linearizing Attack

Each element of the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ in Eq. (9) can be expressed as a Boolean expression of ciphertext $C(X) = (c_1, c_2, \ldots, c_m)$ like $c_1 + \cdots + c_1c_2 + \cdots + c_1c_2c_3 + \cdots$. Let $\mathbf{a}_j$ $(j = 1, 2, \cdots, L+1)$ be a m-dimensional column vector of $\mathbf{A}$ and $\mathbf{b}$. $\mathbf{a}_j$ is calculated by using N-th order differentials, and is defined as follows.

$$\mathbf{a}_j = \Delta^{(N)}\mathbf{A}_j\mathbf{c}\ ,\ \mathbf{c} = {}^t(c_1, \cdots, c_1c_2, \cdots, c_1c_2c_3, \cdots) \tag{12}$$

where $\mathbf{A}_j$ is an $m \times D$ constant matrix determined from Eq. (8) and $\mathbf{c}$ is a D-dimensional vector. The elements of $\mathbf{c}$ are ciphertext monomials which include higher order degrees. We can rewrite Eq. (12) as follows.

$$\mathbf{a}_j = \mathbf{A}_j\Delta^{(N)}\mathbf{c} \tag{13}$$

$\mathbf{c}$ is determined from ciphertexts. Since we calculate $\Delta^{(N)}\mathbf{c}$ for each set of N-th order differential, we are able to determine $\mathbf{a}_j$ by calculating Eq. (13) without using the $\tilde{F}(\cdot)$ function. Therefore, we can determine coefficient matrix $\mathbf{A}$ and vector $\mathbf{b}$ from Eq. (13).

Consider the derivation of $\mathbf{c}$ by using T function to calculate ciphertexts. We take T to be a D-bit output function that outputs elements of $\mathbf{c}$ and implement T by using the bit-slice method [3]. Since S-boxes are generally implemented as tables in an encryption function, we embed T as a table in the same way. If we implement it on a 32-bit processor, we need $\lfloor D/32 \rfloor$ table look-ups to retrieve D-bit elements. In this paper, we consider that the computational costs of table S-box and T function look-ups as being the same.

In the following, we introduce an algorithm for key derivation and estimate the necessary number of chosen plaintexts and the computational cost.

Algorithm for key derivation
Step 0: Prepare $\lfloor L/m \rfloor$ sets of N-th order differentials.
Step 1: Calculate $\Delta^{(N)}\mathbf{c}$ using one set of N-th order differential and repeat the calculation for $\lfloor L/m \rfloor$ sets.
Step 2: Calculate $\mathbf{a}_j$ $(j = 1, 2, \cdots, L+1)$ from Eq. (13).
Step 3: Determine the key by solving Eq. (9) with a method such as Gaussian elimination.

The necessary number of chosen plaintexts M' for key derivation is the same as Eq. (10).

$$M' = 2^N \times \left\lfloor \frac{L}{m} \right\rfloor \tag{14}$$

We estimate the computational cost for each step of the algorithm for key derivation as follows.

Step 1: It needs $2^N \times \lfloor D/32 \rfloor$ table look-ups to calculate $\Delta^{(N)}\mathbf{c}$ for each N-th order differential. Thus Step 1 has computational cost of T'_{Step1} as follows.

$$T'_{Step1} = 2^N \times \left\lfloor \frac{D}{32} \right\rfloor \times \left\lfloor \frac{L}{m} \right\rfloor \tag{15}$$

Step 2: In calculating Eq. (13), we calculate inner products of m sets of row vectors of $\mathbf{A}_j$ and $\Delta^{(N)}\mathbf{c}$. This needs $2\times\lfloor D/32 \rfloor \times m \times (L+1)$ table look-ups. Since we prepare $\lfloor L/m \rfloor$ sets of N-th order differentials, the necessary computational cost of Step 2 is estimated to be

$$T'_{Step2} = 2 \times \left\lfloor \frac{D}{32} \right\rfloor \times m \times (L+1) \times \left\lfloor \frac{L}{m} \right\rfloor \approx 2 \times \left\lfloor \frac{D}{32} \right\rfloor \times L^2 \tag{16}$$

Step 3: Solving Eq. (9) with a method such as Gaussian elimination is generally estimated to cost about L^3. In this paper, since we evaluate computational cost assuming the use of a 32-bit processor, Step 3 costs $T1_{Step3}$ as follows.

$$T'_{Step3} = \left\lfloor \frac{L^3}{32} \right\rfloor \tag{17}$$

Therefore the necessary computational cost, T', of this algorithm is evaluated as follows.

$$T' = T'_{Step1} + T'_{Step2} + T'_{Step3} \tag{18}$$

4 Higher Order Differential Attack on KASUMI

4.1 KASUMI

KASUMI is a Feistel type block cipher with 64-bit data block and 128-bit secret key. It is based on MISTY1 [11] which has provable security against linear and differential cryptanalysis [4], [12]. In 2000, the 3rd Generation Partnership Project (3GPP)[2] selected KASUMI as the mandatory cipher in Wideband Code Division Multiple Access (W-CDMA). It is used in the confidentiality and integrity algorithm of 3GPP mobile communications. Fig. 1 outlines its block diagrams with equivalent FO and FI functions; we call it KASUMI hereafter.

4.2 Previous Results

Tanaka et al. proposed the first attack on 5-round KASUMI with a 32-nd order differential by using a bijective round function feature [19]. Sugio et al. searched for an effective chosen plaintext by computer simulations and reduced

[2] 3GPP is a consortium that standardize the 3rd Generation Mobile System.

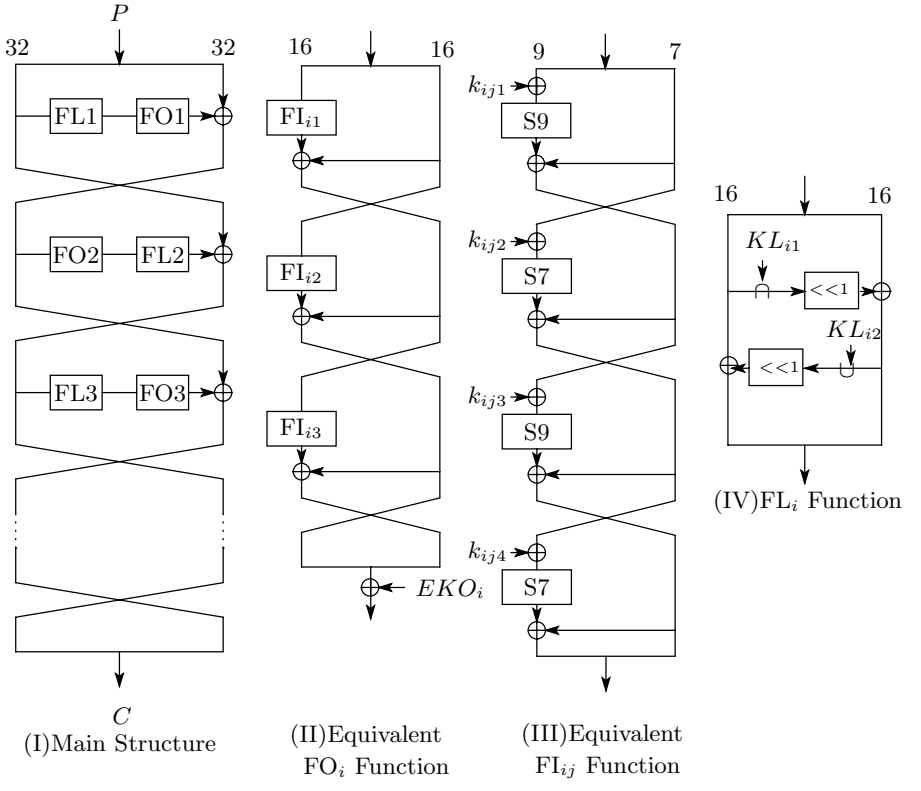

Fig. 1. KASUMI

the necessary number of plaintexts and computational cost by using a 16-th order differential [16]. In the following, we introduce the outline of [16].

Let $H_i = (h_{i4}, h_{i3}, h_{i2}, h_{i1})$ where $h_{i4}, h_{i2} \in \mathrm{GF}(2)^7$ and $h_{i3}, h_{i1} \in \mathrm{GF}(2)^9$ are the right half of the i-th round output. With KASUMI, plaintext X is divided into eight sub-blocks as follows.

$$X = (X_7, X_6, \ldots, X_0) \quad X_i \in \begin{cases} \mathrm{GF}(2)^9 \ (i = \text{odd}) \\ \mathrm{GF}(2)^7 \ (i = \text{even}) \end{cases} \tag{19}$$

The following plaintext, obtained by computer simulations, is the effective chosen plaintext that enables us to reduce the necessary number of chosen plaintexts and computational cost.

$$X \in (\mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{X}_2, \mathcal{X}_1, \mathcal{C}, \mathcal{C}) \quad \mathcal{X}_1, \mathcal{X}_2 : \text{variable}, \ \mathcal{C} : \text{fixed sub-block} \tag{20}$$

Using the above chosen plaintext, we have a constant value that denotes $\Delta^{(16)} h_{33} = 0 \in \mathrm{GF}(2)^9$. Accordingly, we derive the attack equation as follows.

$$\Delta^{(16)} \{\mathrm{FO}_5^9(\mathrm{FL5}(C_L; KL_5); KO_5) \oplus C_R^9\} = 0, \tag{21}$$

where C_L and C_R denote the left and right 32-bit ciphertext, respectively, and $\mathrm{FO}_5^9(\cdot)$ and C_R^9 denote the 9-bits corresponding to h_{33}. Eq. (21) has 82-bit equivalent keys[3]. Sugio et al. estimated the key by combining exhaustive search with the linearizing attack [16]. It needs 2^{22} chosen plaintexts and 2^{63} (FO+FL) function operations.

Nambu et al. analyzed unknown variables $L = 26,693$ in linearized attack equations using the computer software REDUCE. They estimated 82 equivalent key bits by the linearizing attack. It needs $2^{27.5}$ chosen plaintexts and $2^{42.2}$ (FO+FL) function operations [20].

4.3 Application of Fast Calculus to an Attack on KASUMI

In the following, we will demonstrate an application of fast calculus to an attack on KASUMI. We linearize Eq. (21) and express it as follows.

$$\mathbf{Ak} = \mathbf{b} \tag{22}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{k}$ are the $9 \times 26,693$ coefficient matrix, the 9-dimensional vector, and the 26,693-dimensional vector, respectively. If we determine the coefficient matrix $\mathbf{A}$ and the vector $\mathbf{b}$ by calculating Eq. (13), we need to analyze the constant matrixes $\mathbf{A}_j$ and the vector $\mathbf{c}$. Therefore, we analyzed $\mathbf{A}_j$ and $\mathbf{c}$ by expanding the Boolean expressions of Eq. (21) with the computer software REDUCE. We show the number of elements of $\mathbf{c}$ in Table 2.

Table 2. Analysis of the number of elements of $\mathbf{c}$

bit position	# of elements of $\mathbf{c}$
16-th bit	6537
17-th bit	6686
18-th bit	6237
19-th bit	6433
20-th bit	6419
21-th bit	6713
22-th bit	6569
23-th bit	6493
24-th bit	6854
all	9109

'*all*' denotes the number of all elements of $\mathbf{c}$ in 9-bits.

As a result, we determined $\mathbf{A}_j$ $(j = 1, 2, \ldots, 26,694)$ as the 9×9109 matrixes and $\mathbf{c}$ as the 9109-dimensional vector. In the following, we estimate the number of chosen plaintexts needed and the computational cost for the fast calculus.

[3] $KL_5 = (KL_{51}, KL_{52})$ 32 bits and $KO_5 = (k_{511}, k_{512}, k_{513}, k_{521}, k_{522}, k_{523})$ 50 bits.

Estimation of Complexity
Since unknown variables $L = 26,693$ exist in the linearized system of equations, we need $\lfloor 26,693/9 \rfloor$ sets of 16-th order differentials to determine the key. Therefore, the necessary number of chosen plaintexts is estimated as follows.

$$M' = 2^{16} \times \left\lfloor \frac{26,693}{9} \right\rfloor \approx 2^{27.5} \tag{23}$$

We can estimate the computational cost, T', by calculating Eq. (15),$\cdots$,(18).

$$T'_{Step1} = 2^{16} \times \left\lfloor \frac{9109}{32} \right\rfloor \times \left\lfloor \frac{26,693}{9} \right\rfloor \approx 2^{35.69} \tag{24}$$

$$T'_{Step2} \approx 2 \times \left\lfloor \frac{9109}{32} \right\rfloor \times 26,693^2 \approx 2^{38.56} \tag{25}$$

$$T'_{Step3} = \left\lfloor \frac{26,693^3}{32} \right\rfloor \approx 2^{39.11} \tag{26}$$

$$T' = T'_{Step1} + T'_{Step2} + T'_{Step3} = 2^{39.94} \tag{27}$$

We compare Eq. (27) to the previous results. In Eq. (27), we estimate the computational cost as the number of table look-ups of the T function and matrix calculations. According to Fig. 1, each FI function has two S9-boxes and two S7-boxes and so each FO function has (S9 $\times$ 2 + S7 $\times$ 2) $\times$ 3 = 12 S-boxes. Therefore, we regard the computational cost of Eq. (27) as $2^{39.94}/12 \approx 2^{36.4}$ (FO+FL) function operations and this is equivalent to $2^{36.4}/5 \approx 2^{34}$ KASUMI encryptions.[4] We summarize the results of this fast calculus in Table 1.

5 Conclusion

In this paper we applied higher order differential attack to five-round KASUMI. We proposed a linearizing attack with fast calculus that can reduce the complexity incurred in calculating the coefficient matrix **A** and the vector **b**. Our attack requires $2^{27.5}$ chosen plaintexts and 2^{34} encryptions.

In the linearizing attack, we solve the system of linearized equations by using Gaussian elimination. If the number of unknown variables L is large, we can't ignore the computational cost of Gaussian elimination. Therefore, if we decrease the number of unknown variables, we can diminish the total computational cost, T'. We outlined a technique that eliminates unknown variables for the fast calculus in the appendix. We will be able to reduce the computational cost for the key derivation by using this elimination technique.

[4] We discuss here an attack on a 5-round variant.

References

1. 3GPP TS 35202. "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification", http://www.3gpp.org/tb/other/algorithms.htm
2. S. Babbage and L. Frisch. "On MISTY1 higher order differential cryptanalysis", proceedings of 3rd International Conference on Information Security and Cryptology, Lecture Notes in Computer Science 2015, pp. 22-36, Springer-Verlag, 2000.
3. E. Biham. "A Fast New DES Implementation in Software", proceedings of Fast Software Encryption 4th International Workshop, Lecture Notes in Computer Science 1267, pp. 260-272, Springer-Verlag, 1997.
4. E. Biham, A.Shamir. "Differential Cryptanalysis of DES-Like Cryptosystems", Journal of Cryptology, 4(1), pp.3-72, 1991.
5. Y. Hatano, H. Sekine, T. Kaneko. "Higher Order Differential Attack of Camellia (II)", proceedings of Selected Areas in Cryptography 2002 9th Annual International Workshop, pp. 129-146, Lecture Notes in Computer Science 2595 Springer-Verlag 2003.
6. Y. Hatano, H.Tanaka, T.Kaneko. "Optimization for the algebraic method and its application to an attack of MISTY1", IEIEC TRANS. FUNDAMENTALS, Vol.E87-A, No.1, pp.18-27, January, 2004.
7. T. Iwata, K. Kurosawa. "Probabilistic Higher Order Differential Attack and Higher Order Bent Functions", proceedings of Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Lecture Notes in Computer Science 1716, pp. 62-74, Springer-Verlag 1999.
8. T. Jakobsen and L. R. Knudsen. "The Interpolation Attack on Block Cipher", proceedings of Fast Software Encryption 4th International Workshop, Lecture Notes in Computer Science 1267, pp. 28-40, Springer-Verlag, 1997.
9. L. R. Knudsen and D. Wagner. "Integral Cryptanalysis", proceedings of Fast Software Encryption 9th International Workshop, Lecture Notes in Computer Science 2365, pp. 112-127, Springer-Verlag, 2002.
10. X. Lai. "Higher Order Derivatives and Differential Cryptanalysis", proceedings of Communications and Cryptography, pp.227-233, Kluwer Academic Publishers, 1994.
11. M. Matsui. "New Block Encryption Algorithm MISTY", proceedings of Fast Software Encryption 4th International Workshop, Lecture Notes in Computer Science 1267, pp. 54-67, Springer-Verlag, 1997.
12. M. Matsui. "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology, proceedings of EUROCRYPT'93, Lecture Notes in Computer Science 765, pp. 386-397, Springer-Verlag, 1994.
13. S. Moriai, T. Shimoyama, and T. Kaneko. "Higher Order Differential Attack of a CAST Cipher", proceedings of Fast Software Encryption 5th International Workshop, Lecture Notes in Computer Science 1372, pp. 17-31, Springer-Verlag, 1998.
14. S. Moriai, T. Shimoyama, T. Kaneko. "Higher Order Differential Attack Using Chosen Higher Order Differences", proceedings of Selected Areas in Cryptography 1998, pp. 106-117, Lecture Notes in Computer Science 1556 Springer-Verlag 1999.
15. T. Shimoyama, S. Moriai, T. Kaneko, and S. Tsujii. "Improving Higher Order Differential Attack and Its Application to Nyberg-Knudesen's Designed Block Cipher", IEIEC Trans. Fundamentals, Vol.E82-A, No.9, pp. 1971-1980, September, 1999.

16. N. Sugio, H. Tanaka, and T. Kaneko. "A Study on Higher Order Differential Attack of KASUMI", proceedings of International Symposium on Information Theory and its Applications 2002, pp. 755-758, 2002.
17. H. Tanaka, K. Hisamatsu, T. Kaneko. "Strength of MISTY1 without FL Function for Higher Order Differential Attack", proceedings of Applied Algebra, Algebraic Algorithms and Error-Correcting Codes 13th International Symposium, Lecture Notes in Computer Science 1719, pp. 221-230, Springer-Verlag 1999.
18. H. TanakaCC. IshiiCT. Kaneko. "On the Strength of KASUMI without FL Functions against Higher Order Differential Attack", proceedings of 3rd International Conference on Information Security and Cryptology, Lecture Notes in Computer Science 2015, pp. 14-21, Springer-Verlag, 2000.
19. H. TanakaCC. IshiiCT. Kaneko. "On the strength of block cipher KASUMI and MISTY", proceedings of Symposium on Cryptography and Information Security 2001 (in Japanese), pp. 647-652, 2001.
20. S. Nambu, T. Kaneko. "A Study on Higher Order Differential Attack of KASUMI (Ⅲ)", proceedings of The 27th Symposium on Information Theory and Its Applications 2004 (in Japanese), pp.45-48, 2004.

A A Technique for Eliminating Unknown Variables for the Fast Calculus

We will outline an elimination technique of unknown variables for the fast calculus by using lots of the linearized attack equation.

We linearize the attack equation in the same way as Eq. (9) and divide unknown variables L into L' and L''. Let $\mathbf{a}_j$ $(j = 1, 2, \ldots, L+1)$ be a m-dimensional column vector of $\mathbf{A}$ and $\mathbf{b}$, and let the elements of $\mathbf{a}_j$ $(j = 1, 2, \ldots, L')$ be D' ciphertext monomials which include higher order degrees and elements of $\mathbf{a}_j$ $(j = L'+1, \ldots, L+1)$ be the same as those of $\mathbf{c}$. Therefore, we can rewrite Eq. (13) as

$$\mathbf{a}_j = \begin{cases} \mathbf{A}_j \Delta^{(N)} \mathbf{c}'_i & (i = 1, 2, \ldots\ ,\ j = 1 \sim L') \\ \mathbf{A}_j \Delta^{(N)} \mathbf{c}_i & (i = 1, 2, \ldots\ ,\ j = L'+1 \sim L+1), \end{cases} \tag{28}$$

where $\mathbf{c}'_i$ is a D'-dimensional vector that is composed of a part of the elements of D-dimensional vector $\mathbf{c}_i$. Therefore, if we prepare $Q(> D')$ sets of N-th order differentials and calculate each $\Delta^{(N)}\mathbf{c}'_i$ $(i = 1, 2, \ldots, Q)$, we can determine $\Delta^{(N)}\mathbf{c}'_i = \mathbf{0}$ $(i = D'+1, \ldots, Q)$ by using linear dependency of $\Delta^{(N)}\mathbf{c}'_i$. In Eq. (28), if $\Delta^{(N)}\mathbf{c}'_i$ equals $\mathbf{0}$, we can determine $\mathbf{a}_j = \mathbf{0}$. Since $\mathbf{a}_j$ $(j = 1, 2, \ldots, L')$ that correspond to unknown variables L' equals $\mathbf{0}$, it is not necessary to estimate L'.

If many unknown variables L exist in the linearized attack equation, we will be able to reduce the computational cost for the key derivation by using this elimination technique.