The 128-bit Blockcipher CLEFIA

Self Evaluations Report

Version 1.0

Sony Corporation

January 29, 2010

Revision History

Jan 29, 2010 version 1.0

Contents

Chapter 1

Introduction

This document describes self-evaluation results of the 128-bit blockcipher CLEFIA.

Cryptographic technologies are advancing: new techniques on attack, design and implementation are extensively studied. In these years cryptographic functions are implemented in wider field of applications, and there are growing needs for low-cost implementation of cryptographic technologies with high level of security and high performance.

The design philosophy of CLEFIA is to achieve both of high security and high performance on many platforms in software and hardware using the state-of-the-art techniques for design and cryptanalysis.

Design Philosophy

Security There are many known cryptanalytic techniques for blockciphers. It is essential to show a quantitative evaluation on the security against general cryptanalyses such as differential cryptanalysis [10] and linear cryptanalysis [43] to have confidence in security. CLEFIA adopts the Diffusion Switching Mechanism (DSM) [63–65], which is a novel technique to enhance the immunity against differential cryptanalysis and linear cryptanalysis by using plural different diffusion matrices. We aimed to show quantitative evaluations on the security against these attacks. Moreover, CLEFIA is designed considering extensively immunity against all other known cryptanalysis as far as we know.

Furthermore, CLEFIA is designed based on the state-of-the-art cryptanalytic techniques presented after the blockciphers in the current e-Government recommended ciphers list were designed, because cryptanalytic techniques for blockciphers are evolved day by day [9, 14, 15, 18]. In particular, recent researches of related-key attacks make remarkable progress. These attacks are serious threats for blockciphers with a simple key scheduling part such as AES. The key scheduling part of CLEFIA is designed to show a quantitative

evaluation on the security against differential attacks, and to resist against related-key attacks.

Performance As cryptographic primitives are implemented in many applications and in various environments, they are expected to be implemented on a wide range of platforms. From this point of view, AES has exellent properties. Therefore, CLEFIA is aimed to be designed to have advantages over AES. Since AES was designed, several 128-bit blockciphers have been designed. However there are few blockciphers that achieve higher performances than those of AES. Our design goal was to achieve both of speed and cost for implementations keeping high level of security by using state-of-the-art techniques for design and cryptanalysis. As a result, software performance of CLEFIA is comparable to AES, and hardware efficiency of CLEFIA provides remarkable advantages over AES.

Advantages over Existing Blockciphers Since CLEFIA was accepted and presented at Fast Software Encryption Workshop (FSE 2007) in 2007 [66], many results on cryptanalysis of CLEFIA have been published [73,78,79,83, 86], however, there is no known security concern on the full-round CLEFIA so far. On the other hand, it is revealed that AES with 192-bit and 256-bit keys do not have expected security, although it is under a special related-key attack scenario [14, 15]. CLEFIA is designed to resist related-key attacks, which is of advantageous to increase confidence in its security.

In software, CLEFIA with 128-bit keys achieves about 12.9 cycles/byte, 1.48 Gbps on a 2.4 GHz AMD Athlon 64. This result shows that software performace of CLEFIA is classified into the fastest group of blockciphers in the current e-Government recommended ciphers list.

In hardware, CLEFIA with 128-bit keys can be implemented with less than 5K gates by using a 0.09 μm CMOS ASIC library. This is in the smallest class among the blockciphers in the current e-Government recommended ciphers list. For speed optimized implementations, the performance of CLEIFA achieves 1.6 Gbps with about 6 Kgates and 3 Gbps with about 12 Kgates. From these results, CLEFIA is unique in hardware efficiency, which is defined by throughput per gate.

On the Use in e-Government Systems To compile the e-Government recommended ciphers list (draft), CRYPTREC performed security evaluations in order to select cryptographic techniques that satisfy the level of security sufficient for the e-Government system. CRYPTREC required that symmetric-key cryptographic techniques should satisfy either of the following conditions [27].

- Even with the best attacking technique available to date, computational cost of 2^{128} or more (i.e. exhaustive search for a secret key) is

required to break selected symmetric-key cryptographic techniques. It is necessary for the techniques to be shown that they are secure against typical attacking techniques such as differential and linear cryptanalysis.

- Widely used symmetric-key cryptographic techniques which have been evaluated in details and have no security problems in a realistic system, are selected. In this case, computational cost of 2^{100} or more is required to break them.

As shown in Chapter 3, security of CLEFIA satisfy the first item above, therefore, it is considered that CLEFIA satisfies the level of security sufficient for the e-Government system.

Regarding implementation performance, as shown in Chapter 4, CLEFIA achieves high performance both in software and hardware. Therefore, CLEFIA is suitable for all applications in Japan e-Government systems that require high implementation performance.

Furthermore, CLEFIA has advantages in compact hardware implementations. So it is recommended to use CLEFIA in products and systems with constrained environments.

Chapter 2

Design Rationale

This chapter describes design rationale of CLEFIA [66, 71].

CLEFIA is designed to realize good balance on three fundamental directions which are considered as important for practical ciphers: (1) security, (2) speed, and (3) cost for implementations. To achieve these goal, several kinds of design technologies are contributed. Summary of special features of CLEFIA in design aspect is listed as follows.

1. The first blockcipher employing the Diffusion Switching Mechanism (DSM) to enhance the immunity against the differential attack and the linear attack [10, 43]
2. Compact F-functions realized by employing a 4-branch generalized Feistel structure
3. Enhanced immunity against a certain class of attacks by using a two S-boxes system
4. Using only lightweight components for efficient implementations of software and hardware
5. Enabling shared implementation of the data processing part and the key scheduling part
6. A new key scheduling algorithm realizing strong immunity against related-key attacks

The details of these features are explained in the following sections.

2.1 Data Processing Part

In this section, design rationale for the data processing part of CLEFIA are described.

2.1.1 Fundamental Structure

CLEFIA employs a generalized Feistel structure which is an extension of the traditional Feistel structure. Generalized Feistel structure has three or more data lines as opposed to two data lines in traditional Feistel structure. There are many types of generalized Feistel structures depending on the connected positions of the input and the output of F-functions to the data lines. Among them, we choose one structure which is known as "Generalized Type-2 transformation" defined in Zheng et al.'s paper [87]. Figure 2.1 shows the 4-branch case of Type-2 structure. Since the block length of the cipher is 128 bits, the width of each data line is 32 bits. The type-2 structure has two F-functions in one round in the 4 data lines case. The first F-function is applied to the first data line and the other is applied to the third data line.

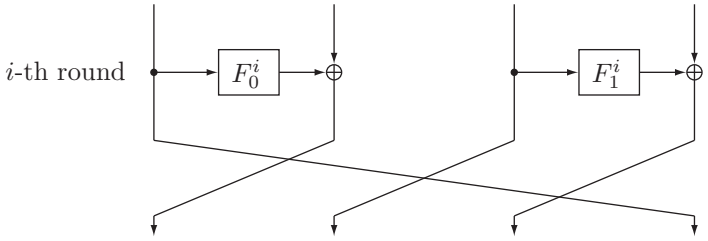

Figure 2.1: One round of the 4-branch Type-2 generalized Feistel structure

The Type-2 structure has the following features:

- Two F-functions can be processed simultaneously
- The size of F-functions is smaller than that in traditional Feistel structure
- The structure tends to require more rounds than traditional Feistel structure

The first feature is suitable to high-performance hardware implementations, and the second is of great advantage to software and hardware implementations. The last feature is a disadvantage of 4-branch structure because the diffusion speed of smaller F-functions is slower. But we succeeded to get rid of this disadvantage by introducing a new design technique, called DSM, explained in this document later. Consequently, CLEFIA mainly benefits from the first two advantages.

Pioneer research on the generalized Feistel structures is done by Zheng, Matsumoto and Imai [87], followed by the work by Nyberg [52]. The block-cipher RC6 also employs the Type-2 structure with slight modification, and it achieves good efficiency performance partially due to this structure [59].

In the security aspects, Moriai and Vaudenay treated pseudo-random property of the generalized Feistel structures [49]. Furthermore, Knudsen and Wagner presented with regard to integral cryptanalysis [36], and Kim et al. discussed with regard to impossible differential cryptanalysis [34].

2.1.2 F-functions

The F-function of CLEFIA is the so-called SP-type F-function which means Substitution layer and Permutation (Diffusion) layer are applied in this order after a round key addition [71]. This type of F-function is used in many blockcipher designs including Camellia [3] and Twofish [62]. CLEFIA uses four 8-bit S-boxes in the Substitution layer and a 4×4 diffusion matrix in the Permutation layer. This F-function can be implemented efficiently in software by using the table-lookup technique [22].

2.1.3 Key Whitenings

CLEFIA employs key whitenings at the beginning and the end of the data processing part [71]. The whitening operation at each part is done for only half of 128-bit data (i.e. two of four data lines), because these partial whitenings provide enough key information (entropy) for the data processing part. This is explained by using an equivalent transformation of round keys of generalized Feistel structure. Figure 2.2 shows the two generalized Feistel structures in which key addition layer is explicitly described out of the F-function. The two structures are equivalent. This figure shows that the full key whitening can be always converted into half key whitening and vice versa. Therefore, we designed CLEFIA using half key whitening to reduce the cost of key additions.

2.1.4 Diffusion Matrices

CLEFIA employs two different diffusion matrices M_0 and M_1 to enhance the immunity against the differential attack and the linear attack by using the Diffusion Switching Mechanism (DSM). This concept was first proposed by Shirai and Shibutani in 2004 followed by extended works by Shirai and Preneel, but it was applied to only the traditional Feistel structures [63–65]. We customized this technique suitable to the Type-2 generalized Feistel structures, which is one of the unique selling propositions of this cipher. By using this technique, we can prevent difference cancellations and linear mask cancellations in the neighborhood rounds in the cipher. As a result the guaranteed number of active S-boxes is increased.

To explain the mechanism, we introduce the following definitions.

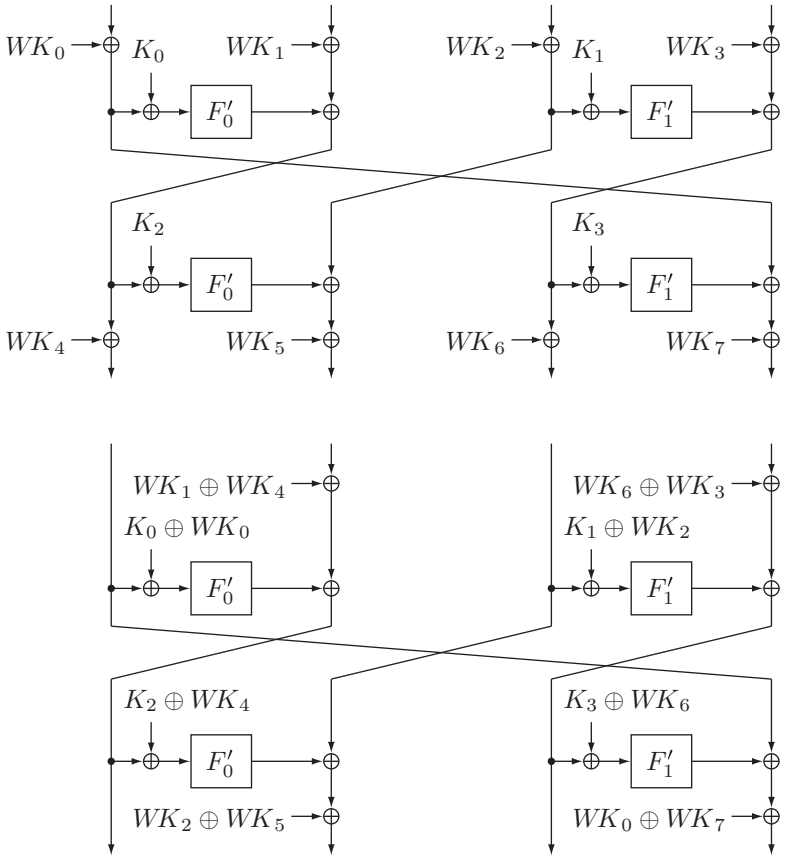

Figure 2.2: Equivalent Structures

Definition 2.1. *Let $x \in \{0,1\}^{pl}$ be represented as $x = (x_0 x_1 \ldots x_{p-1})$ where $x_i \in \{0,1\}^l$, then the bundle weight $w_l(x)$ is defined as*

$$w_l(x) = \#\{i \mid 0 \leq i \leq p-1, x_i \neq 0\} \ .$$

Definition 2.2. *Let $P : \{0,1\}^{pl} \rightarrow \{0,1\}^{ql}$. The branch number of P is defined as*

$$\mathcal{B}_l(P) = \min_{a \neq 0} \{w_l(a) + w_l(P(a))\} \ .$$

To utilize the DSM technique we need at least two matrices which satisfy certain branch number conditions. In CLEFIA's case, the two 4×4 matrices M_0 and M_1 whose elements are in $\mathrm{GF}(2^8)$ hold following conditions.

$$\mathcal{B}_8(M_0) = \mathcal{B}_8(M_1) = 5 \ .$$

This is an optimal branch number for matrices with this size. Besides that, the branch numbers of combined matrices $M_0|M_1$ and ${}^tM_0^{-1}|{}^tM_1^{-1}$ are also 5, which is also an optimal case as:

$$\mathcal{B}_8(M_0|M_1) = \mathcal{B}_8({}^tM_0^{-1}|{}^tM_1^{-1}) = 5 \ .$$

When M_0 and M_1 are put in the F-functions of CLEFIA as Figure 2.3, it is expected to hold good diffusion property by the synergy of these two matrices in neighboring F-functions [71]. In the figure, the data lines of Feistel structure are untwisted, accordingly positions of the F-functions are moved to correct positions. This technique is called the Diffusion Switching Mechanism (DSM), and detailed mechanism and the effects are described in Section 2.3.

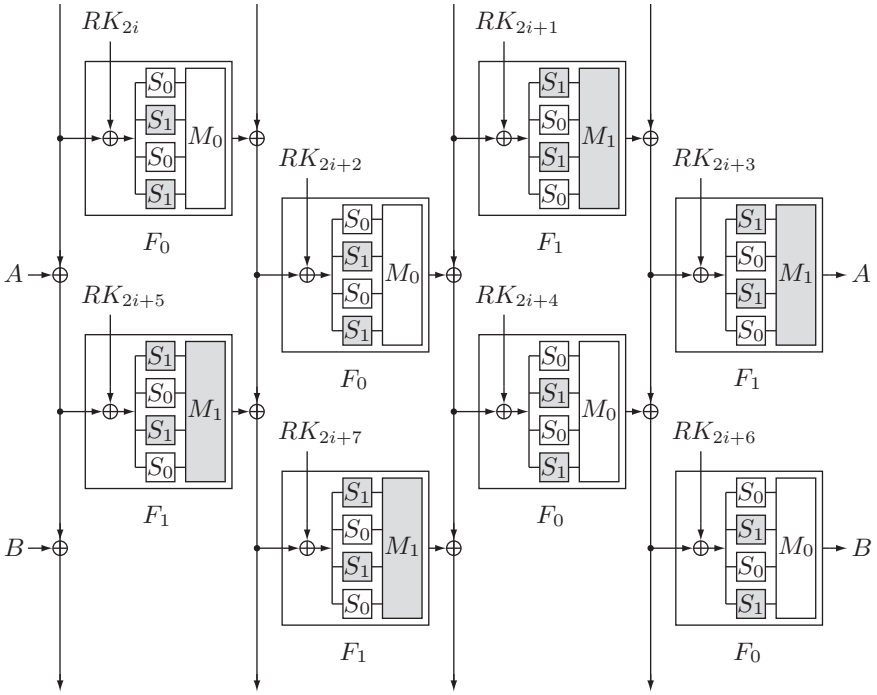

Figure 2.3: Allocation of M_0, M_1, S_0 and S_1

Table 2.1 lists the effect of the DSM by showing the guaranteed number of active S-boxes of CLEFIA. These values are obtained by a computer simulation using a weight-based evaluation method.

The columns indicated by 'Normal' show the guaranteed number of active S-boxes for generalized Feistel network without using the DSM technique while employing a single optimal diffusion mapping for all F-functions. The columns indicated by 'DSM(D)' show the guaranteed number of differential active S-boxes when using the DSM with optimal matrices M_0 and

Table 2.1: Guaranteed Numbers of Active S-boxes

r	Normal	DSM (D)	DSM (L)	r	Normal	DSM (D)	DSM (L)
1	0	0	0	14	25	34	34
2	1	1	1	15	26	36	36
3	2	2	5	16	30	38	39
4	6	6	6	17	32	40	42
5	8	8	10	18	36	44	46
6	12	12	15	19	36	46	48
7	12	14	16	20	37	50	50
8	13	18	18	21	38	52	52
9	14	20	20	22	42	55	55
10	18	22	23	23	44	56	58
11	20	24	26	24	48	59	62
12	24	28	30	25	48	62	64
13	24	30	32	26	49	65	66

M_1. Similarly, 'DSM(L)' means the guaranteed number of linear active S-boxes for a corresponding round. From this table we can confirm the effect of the DSM when $r \geq 3$, and these guaranteed numbers increase about $20\% - 40\%$ than the 'Normal'.

The search algorithm for the above estimation is described in Sec. 2.3. We also have theoretical results of lowerbounds of DSM [67]. Theorems of DSM for Type-2 generalized Feistel structure are described in Sec. 2.3.

There are two side effects due to introducing the DSM technique: one is a partially destroyed involution property of generalized Feistel structure and the other is that additional cost for implementing two matrices is expected. But we have confirmed these side effects have limited impact on efficient implementation. With regard to the involution property, we can avoid the problem by only changing the swapping order of data in the encryption and the decryption. Moreover, the penalty due to using two matrices is limited, because the size of matrices is not too large.

Here we compare the effect of the DSM technique to Camellia and Twofish, which are also employing 8-bit S-boxes and a Feistel structure.

CLEFIA and Camellia can be viewed as Feistel ciphers using a diffusion matrix with the same branch number, 5. According to [3], there are 18, 21 and 22 differential active S-boxes for 9, 10 and 11 rounds, respectively. Also, there are 18, 20 and 22 linear active S-boxes for 9, 10 and 11 rounds of Camellia without FL/FL^{-1}. These numbers are larger than CLEFIA using a single matrix, but smaller than CLEFIA with the DSM technique. In other words, by using two diffusion matrices with the DSM technique, CLEFIA

has more immunity against differential/linear cryptanalysis. Although a generalized Feistel structure has a worse diffusion property due to smaller diffusion matrices, DSM compensates the shortage without big investment.

Twofish also employs two 4×4 matrices with maximum branch number 5 in the round function. The designers claimed that Twofish has 20 guaranteed active S-boxes in 12 rounds [62]. The claimed number of estimated guaranteed active S-boxes is also smaller than CLEFIA.

Consequently, it is expected that the diffusion performance of CLEFIA is better than that of Camellia and Twofish by observing the known active S-box estimations.

Choices of two Diffusion Matrices

Two matrices have to satisfy the aforementioned optimal branch number conditions. But there are huge number of matrices satisfying the conditions, so we chose actual two matrices taking a cost of hardware implementation into consideration.

Candidate matrices were 4×4 circulant and Hadamard-type ones. An Hadamard-type matrix is used in blockcipher Anubis [4], and each element in an $m \times m$ Hadamard matrix is defined as $h_{i,j} = a_{i \oplus j}$ for a certain set of $(a_0, \ldots, a_{m-1})$. We checked all circulant matrices and Hadamard-type matrices which have low hamming weights, then we found the best matrices which can be implemented efficiently in hardware because the number of XOR gates is very small. As a result, two matrices M_0 and M_1 for CLEFIA are decided as Hadamard-type matrices.

2.1.5 S-boxes

CLEFIA employs plural types of S-boxes as in Serpent and Camellia [1, 3]. We believe that the reason for choosing CLEFIA's plural S-boxes is based on the following criteria and expected effects.

1. Good immunity against known attacks
2. Suitability for efficient hardware implementation

Then we first decided to employ two types of S-boxes for the security reason, then we choose actual two types of S-boxes taking the above implementation property into consideration. By adopting two S-boxes, we expect the following effects with regard to security.

- To enhance the immunity against the byte-oriented saturation attacks [19], and
- To enhance the immunity against algebraic attacks including the XSL attack [18].

Table 2.2: Security Parameters of S_0 and S_1

	S_0	S_1
maximum difference prob.	$2^{-4.67}$	$2^{-6.00}$
maximum linear prob.	$2^{-4.38}$	$2^{-6.00}$
minimum degree (Boolean)	6	7
minimum number of terms over GF(2^8)	244	252

The reasons are explained in this section later. CLEFIA employs two different types of 8-bit S-boxes S_0 and S_1. These two S-boxes are categorized as:

- S_0 : 8-bit S-box based on randomly chosen 4-bit S-boxes
- S_1 : 8-bit S-box based on the inverse function over GF(2^8)

The ways to select concrete two S-boxes and the influence on the security are described in the following subsections.

S-box based on 4-bit S-boxes

The first S-box S_0 is based on 4-bit S-boxes. It consists of 4 different 4-bit S-boxes, and all the (4-bit) S-boxes are connected by a 2×2 matrix over GF(2^4) defined by a primitive polynomial $x^4 + x + 1$. The branch number of the matrix is equal to 3 which is an optimal diffusion. The four 4-bit S-boxes are selected from random bit strings generated by AES with the counter mode. Table 2.2 shows the several security parameters of S_0.

S-box based on inverse function over GF(2^8)

The second S-box S_1 is designed based on the inverse function in GF(2^8). The used irreducible polynomial is $x^8 + x^4 + x^3 + x^2 + 1$. Additionally, there are affine mappings before and after the inverse operation to enhance immunity against the interpolation attack [28]. Table 2.2 shows the several security parameters of S_1.

Enhancing immunity against byte-oriented saturation attacks

The first effect of using two different S-boxes is to avoid collisions of the output values of the S-boxes. Let $X_i \in \{0,1\}^8$ $(0 \leq i \leq 255)$ be 256 8-bit variables. Now we classify X_i into four groups depending on conditions satisfied by all elements in the set. X_i $(0 \leq i \leq 255)$ is called:

- `Const (C)` : if $\forall i, j \;\; X_i = X_j$,
- `All (A)` : if $\forall i, j \;\; i \neq j \Leftrightarrow X_i \neq X_j$,

- `Balance (B)` : if $\bigoplus_i X_i = 0$,
- `Unknown (U)` : unknown.

Then consider a toy example that an F-function contains only one 8-bit round-key addition layer and a substitution layer using one 8-bit S-box (see left of Figure 2.4).

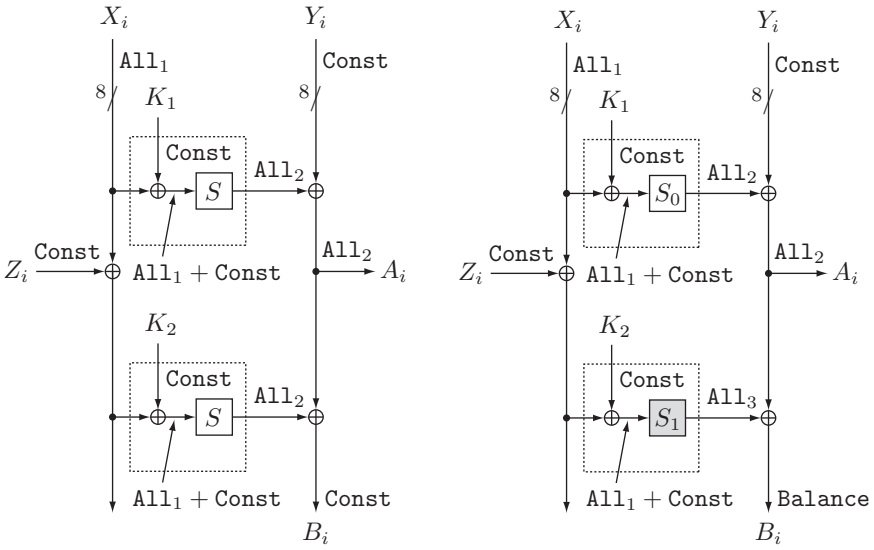

Figure 2.4: An Example of Saturation characteristic

Suppose that X_i is `All` and Y_i and Z_i are `Const`. Note that this assumption is reasonable especially in generalized Feistel structures like CLEFIA.

Then, B_i is expressed as:

$$B_i = S(X_i \oplus K_1) \oplus S(X_i \oplus Z_i \oplus K_2) \oplus Y_i \ .$$

Usually, we expect B_i be `Balance`, because two `All`s from the both of S-boxes are XORed. However, B_i can become `Const` in certain situations. When the constant values have relations $Z_i = K_1 \oplus K_2$, the outputs of two S-boxes always collide, as a result $B_i = Y_i$. This happens with probability $p = 1/256$ in this setting.

However in CLEFIA, two S-boxes S_0, S_1 satisfy the following condition,

$$\text{For any } c_1, c_2,\ \exists x \ \ S_0(x) \neq S_1(x \oplus c_1) \oplus c_2.$$

We can avoid the above cancellation of saturation characteristics. Putting S_0 and S_1 as the right of Figure 2.4, B_i won't be `Const` due to the S-box property, because two `All`s XORed will be never canceled due to the above condition.

Although this is a toy example, and an actual cipher employs more complicated matrices, we consider that a similar situation can also happen if all S-boxes in a cipher are the same S-box. Therefore we employed two S-boxes and changed the order of S-boxes in two F-functions to avoid the weak property explained above.

Enhancing immunity against algebraic attacks

Previous works on algebraic attacks [18, 28, 37] showed us that relying only on specific algebraic functions, e.g. the inversion function in a Galois Field, is not a good way from the view point of security against algebraic attacks. To resist algebraic attacks the designers of blockciphers have adopted several ideas, e.g. using "random" S-boxes [1, 31], mixed use of S-boxes with different sizes [47], or constructing from random 4-bit S-boxes [4, 29], some of which required much implementation cost. In designing CLEFIA, we adopt a novel countermeasure which increases the immunity against algebraic attacks without big penalty on implementation cost. The solution is to prepare two different 8-bit S-boxes and mixing up of these S-boxes in the cipher.

Two types of 8-bit S-boxes of CLEFIA are:

- 8-bit S-box based on randomly chosen 4-bit S-boxes
- 8-bit S-box based on the inverse function in $GF(2^8)$

We excluded a randomly chosen 8-bit S-box because the cost of hardware implementation is too large for CLEFIA. Both of above S-boxes are more advantageous than a randomly chosen 8-bit S-box with regard to efficient hardware implementations.

It is known that the inverse function based S-box is optimal with regard to differential probability and linear probability, but it is reported that there are simple algebraic relation over $GF(2)$ and $GF(2^8)$. If the cipher uses only the inverse function based S-boxes, then its immunity against the XSL attack over $GF(2^8)$ is considered to have potential weakness than that over $GF(2)$ [50]. Moreover, Daemen and Rijmen presented a new result on behavior of inverse function based S-boxes such that there are plateau trails in it [21]. That's why we don't want to use an inverse function based 8-bit S-box only.

On the other hand 8-bit S-boxes based on 4-bit S-boxes is not optimal regarding differential and linear properties, but the compactness in hardware implementation is very attractive. It is also known that there are simple relation over $GF(2)$ in 8-bit S-boxes based on 4-bit S-boxes, but simple quadratic relations over $GF(2^8)$ are not expected. Using an estimation method for complexity of the XSL attack, the immunity against the XSL attack over $GF(2)$ of this type of S-box is expected weaker than inverse based

S-box [18]. That's why we don't want to use a 8-bit S-boxes based on 4-bit S-boxes only.

Also, we saw trends of choice of S-boxes in literatures, in a certain period of time many blockcipher designers used the inverse function based 8-bit S-boxes as in AES/Rijndael, Camellia, Misty, Hierocrypt-3 and so on. Then 8-bit S-boxes based on 4-bit S-boxes are tend to be used as in Whirlpool[1], Anubis and FOX [3–5, 22, 29, 47, 53]. Our approach is different from the above trends.

In CLEFIA half of S-boxes are the inverse function based 8-bit S-boxes and the others are 8-bit S-boxes based on 4-bit S-boxes. This design makes the cipher stronger against the XSL attack in both over GF(2) and GF(2^8), though big penalty in hardware implementation isn't required as only randomly chosen 8-bit S-boxes are employed.

Positions for S_0 and S_1

The two S-boxes system is suitable for CLEFIA because DSM has been already employed in which two distinct F-functions exist. In the first F-function F_0, the four S-boxes are chosen as S_0, S_1, S_0, and S_1 in this order, then in the second F-function F_1 the order of S-boxes is S_1, S_0, S_1, and S_0. It is obvious that there are the same number of 4-bit based S-boxes and inverse based S-boxes in CLEFIA, and it is guaranteed that a certain byte in the data line of generalized Feistel structure is applied the both of S-boxes alternatively (see Figure 2.3). Thus this construction is enough to enhance the immunity against the byte-oriented saturation attack and the XSL attack. Note that there are two S_0 and two S_1 in the both of F_0 and F_1, this is good property for implementation aspect taking sharing resources into account.

2.2 Key Scheduling Part

In this section, we mention the design rationale of the key scheduling part of CLEFIA. Properties of the key scheduling part of CLEFIA are as follows:

1. Intermediate key L is generated from a key K by a permutation based on the data processing part of CLEFIA. As a result, strong immunity against related-key attacks is expected.

2. L is employed as round keys at certain rounds to exclude equivalent round keys.

3. $K \oplus L$ is employed as round keys at certain rounds to benefit from the property of the one-wayness $K \rightarrow K \oplus L$ which means it is difficult to recover K from $K \oplus L$.

[1] A hash function included in ISO/IEC 10118-3 standard.

4. Although the permutation function to generate L is comparatively heavy, the cost of generating round keys from a key K and an intermediate value L is kept light-weighted.

5. The above features are valid for the key scheduling steps for any key length.

Details for the above properties are explained in this section.

2.2.1 Employing $GFN_{4,12}$ for 128-bit key

$GFN_{4,12}$ is a 12-round CLEFIA without key scheduling part and key whitenings. The round keys for $GFN_{4,12}$ are fixed constants. $GFN_{4,12}$ is used in the key scheduling step of 128-bit key CLEFIA. We consider that $GFN_{4,12}$ has a good difference propagation property, it means that controlling the output difference of $GFN_{4,12}$ is very difficult even though attacker can control the input difference of it. If $GFN_{4,12}$ is used in the key scheduling part properly, we can construct a blockcipher for which related-key attacks will be very difficult.

From the previous evaluation result, we know that there are 28 differential active S-boxes and 30 linear active S-boxes in 12-round CLEFIA and highest DP_{max} is $2^{-4.67}$ and highest LP_{max} is $2^{-4.38}$ due to S_0. As a result, we can assure that there are no differential characteristics or linear approximation with probability more than 2^{-128}, because $28 \times 4.67 = 130.76$ and $30 \times 4.38 = 131.40$. This is only saying about characteristics but not about differential and linear hulls, thus we cannot conclude that there is no good differentials or linear hulls in $GFN_{4,12}$. However, CLEFIA uses S-boxes S_1 with $DP_{max} = LP_{max} = 2^{-6}$, the actual margin of characteristic probability is expected to be larger than this estimations. Detailed discussion on the margin of characteristic probability is presented by Daemen and Rijmen [20].

2.2.2 Employing $GFN_{8,10}$ for 192/256-bit keys

$GFN_{8,10}$ is a 10-round generalized Feistel structure with 8 data lines, the width of each data line is 32 bits. The round keys for $GFN_{8,10}$ are fixed constants determined by the key length. The input/output data length of $GFN_{8,10}$ is 256 bits. $GFN_{8,10}$ is used in the key scheduling step of 192/256-bit key CLEFIA. If $GFN_{8,10}$ is used in the key scheduling part properly, we can construct a blockcipher for which related-key attacks will be very difficult.

From the evaluation result shown in Table 2.3, we know that there are at least 29 differential active S-boxes in $GFN_{8,10}$. We can assure that there are no differential characteristics with probability more than 2^{-128} because $2^{29\times(-4.67)} = 2^{-135.43}$.

Table 2.3: Active S-boxes for 8-branch Generalized Feistel structure

rounds	1	2	3	4	5	6	7	8	9	10	11	12
active	0	1	2	6	8	12	14	21	24	29	34	39

2.2.3 Mixed Use of K and L

In the 128-bit key scheduling part of CLEFIA, a 128-bit intermediate value L is generated from the key K by using $GFN_{4,12}$. Then both of the K and L are mixed up in the generation steps of round keys. The advantage of this usage is noticed when conducting the exhaustive search for K and L. It is difficult to guess even one bit information in K from only partial information of L, and vice versa, because any single bit in K depends on the all bits in L by the permutation $GFN_{4,12}$. Consequently, if K and L are allocated appropriately to generate round keys, we can strengthen a cipher against such attackers.

Similarly in the key scheduling part of 192 and 256-bit keys, two 128-bit intermediate values L_L, L_R are generated from the key K_L, K_R by using $GFN_{8,10}$. The same effect will be expected also in 192-bit and 256-bit key cases.

2.2.4 *DoubleSwap* function

In the round key generation process of CLEFIA, the intermediate values L, L_L and L_R are updated by a *DoubleSwap* function in every two rounds repeatedly. One reason this is to destroy simple relation between round keys. Moreover, comparing to a rotation operation, the *DoubleSwap* function enables efficient hardware implementation.

2.2.5 Flexibility for Implementations

We designed the key scheduling algorithm for 128, 192 and 256-bit keys and data processing part to be able to share common components. All key scheduling algorithms use $GFN_{4,12}$ or $GFN_{8,10}$ which are based on the data processing part of CLEFIA. Consequently we expect that efficient hardware implementation can be achieved by sharing components of all key scheduling algorithms.

2.2.6 Constant Values

There are round constants used in key scheduling algorithm for each key length. The size of each constant is 32 bits and each value is made from one 16-bit initial values [71]. Moreover, these constants can be generated

sequentially from the first 16-bit constant by applying simple bit operations repeatedly. Therefore, cost for storing constant values in hardware is significantly reduced if these values are generated dynamically in the implementation.

2.3 Diffusion Switching Mechanism (DSM)

Here, we describe the search algorithm for the active S-box estimation and theoretical results of lowerbounds of DSM for Type-2 generalized Feistel structure [67][2].

2.3.1 Target Structure

First of all, "Type-2" generalized Feistel structure which operates d data branches ($d \geq 2$) are shown. Here, we call a class of structures *generalized* Feistel if it is identical with the conventional Feistel structure in case of $d = 2$. Our target is "Type-2" generalized Feistel structure. The structures are defined by Zheng *et al.* [87]. Several cryptographic properties of the generalized structure are studied in [34, 49].

Let n be an integer $d|n$ and $P_0, \ldots, P_{d-1}$ be n/d-bit plaintext words, and let $C_0, \ldots, C_{d-1}$ be n/d-bit ciphertext words. Type-2 structure uses a plural number of F-functions per round, and the number of branches d is even. Let $F_i^j(x, y)$ be the j-th F-function from the left in the i-th round. Type-2 generalized Feistel structure is defined as follows.

Step 1. $X_0 \leftarrow P_0, \ldots, X_{d-1} \leftarrow P_{d-1}$
Step 2. For $i = 1$ to r do the following:
  Step 2.1 For $j = 0$ to $d/2 - 1$ do the following:
    Step 2.1.1 $X_{2j+1} \leftarrow X_{2j+1} \oplus F_i^j(RK_i^j, X_{2j})$
  Step 2.2 $tmp \leftarrow X_{d-1}$,
    $X_j \leftarrow X_{j-1}$ (for $j = d - 1$ to 1),
    $X_0 \leftarrow tmp$
Step 3. $C_0 \leftarrow X_0, \ldots, C_{d-1} \leftarrow X_{d-1}$

In the above, RK_i ($1 \leq i \leq r$) are provided by a key scheduling part which is not defined here. Without loss of generality, a swap operation at the final round is included. Figure 2.5 shows a round function of the Type-2 structure in case of $d = 8$.

In this report, we assume that the type of F-functions used in these structures is the SP-type F-function which is one of the popular F-functions [30]. Let l be the size of S-boxes and let m be the dimension of a diffusion matrix,

[2]The copyright of this section is owned by The Institute of Electronics, Information and Communication Engineers (IEICE). ©2008 IEICE

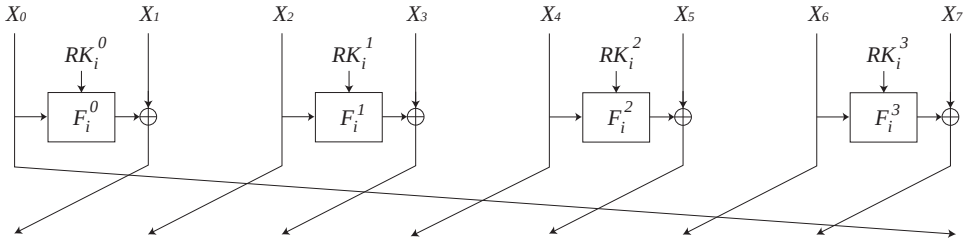

Figure 2.5: A Round Function of Type-II Feistel Structure ($d = 8$)

then an SP-type F-function taking an lm-bit round key RK, input data X and output data Y is defined as:

Step 1. $T \leftarrow RK \oplus X$
Step 2. Let $T = T_0 \mid T_1 \mid \cdots \mid T_{m-1},\ T_i \in \{0,1\}^l$
$T_i \leftarrow S(T_i)$ (for $i = 0$ to $m-1$)
Step 3. Let $Y = Y_0 \mid Y_1 \mid \cdots \mid Y_{m-1},\ Y_i \in \{0,1\}^l$
${}^t(Y_0, Y_1, \cdots, Y_{m-1}) = M\ {}^t(T_0, T_1, \cdots, T_{m-1})$

where $A|B$ denotes a concatenation of data A and B. $S(\cdot)$ denotes an l-bit bijective S-box and M denotes a non-singular $m \times m$ matrix over a chosen field GF(2^l). Hereafter M_i^j denotes diffusion matrices M used in F-functions F_i^j in generalized Feistel structures, respectively. Figure 2.6 shows an example of an SP-type F-function F_i^j in case of $m = 4$.

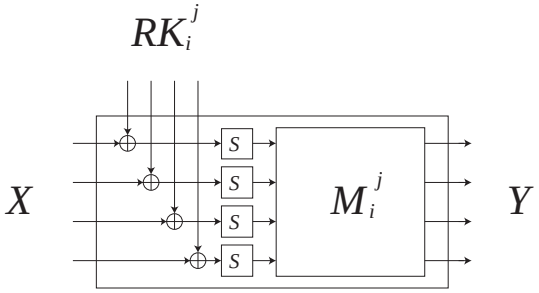

Figure 2.6: F-function F_j^i

Using the above definitions, the block length n is now determined by three parameters d, l and m as $n = dlm$.

2.3.2 Basic Concepts

Basic concept of DSM is explained using Fig. 2.7. Let M be a non-singular $m \times m$ matrix, and $\boldsymbol{a}, \boldsymbol{b} \in \{\{0,1\}^l\}^m$ are m-dimensional vectors. The left side of Fig. 2.7 shows that the two output vectors through the same matrix M

are XORed to the data line. Suppose that x and y are fixed for 0, it is shown that $w_l(\boldsymbol{a})+w_l(\boldsymbol{b}) = 2$ is a possible value because $M(\boldsymbol{a}+\boldsymbol{b}) = \mathbf{0}$ is realized by $\boldsymbol{a} = \boldsymbol{b}, w_l(\boldsymbol{a}) = 1$ for any fixed M. However, if two different matrices M_1 and M_2 are used as in the right side of Fig 2.7, then $w_l(\boldsymbol{a})+w_l(\boldsymbol{b}) \geq \mathcal{B}_l([M_1|M_2])$, where $[A|B]$ denotes an $m \times 2m$ matrix obtained by concatenating matrices A and B. From Definition 2.2, $\mathcal{B}_l([M_1|M_2])$ can be $m+1$ at most, which is optimal diffusion [22, 65]. If we put S-boxes just before the matrices as an SP-type F-function, $w_l(\boldsymbol{a})+w_l(\boldsymbol{b})$ is regarded as the number of active S-boxes in this case. From this observation, the latter construction can guarantee larger numbers of active S-boxes if the above conditions are satisfied. DSM incorporates this property in the whole Feistel structure to raise the lower bounds.

2.3.3 Type-2 Generalized Feistel Structure using DSM

Next, the DSM is applied to Type-2 generalized Feistel structure. For example, Type-2 generalized Feistel structure where $d = 6$ is illustrated as an untwisted form as in Fig. 2.8. To use the DSM, two matrices M_i^j in F_i^j and M_{i+2}^{j-1} in F_{i+2}^{j-1} for all possible i and j should satisfy the following DSM branch number conditions. Note that the indices at the upper right of M and F are taken mod $d/2$, i.e. $d/2 = 0$, and $-1 = d/2 - 1$.

We define B_1^D, B_2^D and B_2^L as follows:

Definition 2.3.

$$B_1^D = \min_{1\leq i\leq r,\ 0\leq j<d/2}(\mathcal{B}_l(M_i^j)) \ ,$$

$$B_2^D = \min_{1\leq i\leq r-2,\ 0\leq j<d/2}(\mathcal{B}_l([M_i^j \mid M_{i+2}^{j-1}]) \ .$$

$$B_2^L = \min_{1\leq i\leq r-2,\ 0\leq j<d/2}(\mathcal{B}_l([{}^t(M_i^j)^{-1} \mid {}^t(M_{i+2}^{j-1})^{-1}])) \ .$$

The above definition directly implies $B_1^D \geq B_2^D$.

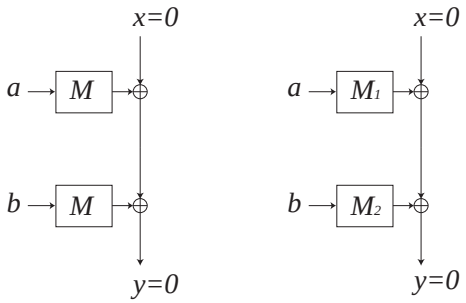

Figure 2.7: Concept of DSM

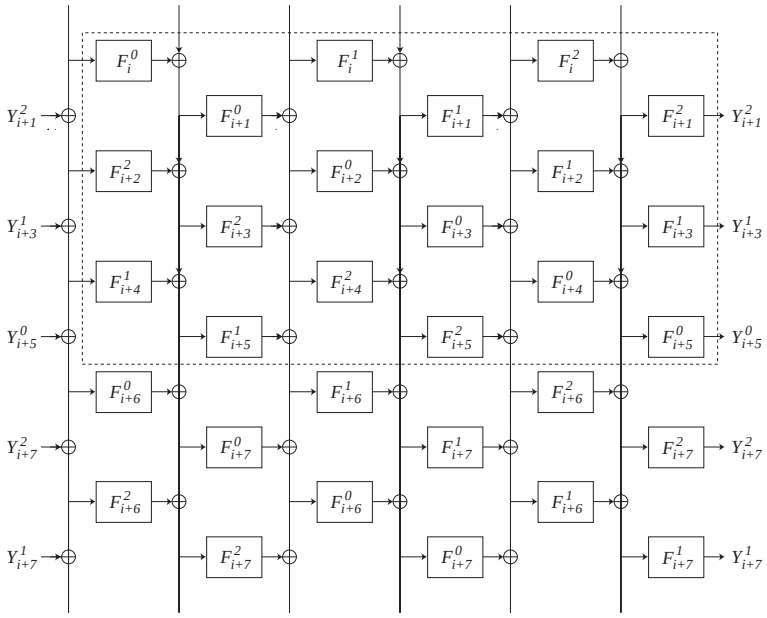

Figure 2.8: Type-2 generalized Feistel Structure($d = 6$, Untwisted)

Using these definitions, proven lower bounds of differential and linear active S-boxes for Type-2 generalized Feistel structure are shown.

Differential Active S-boxes in Type-2 generalized Feistel Structure

Let X_i^j and K_i^j be an input, a round-key. D_i^j denotes a number of differential active S-boxes in F_i^j, respectively. If non-zero difference is input to Type-2 generalized Feistel structure, we use the following properties:

Property 2.1. *There is at least one F-function which contains at least one active S-box in any consecutive 2 rounds.*

This property is due to the invertibility of the structure.

Property 2.2. *If* $D_i^j = 0$, *then* $D_{i-1}^{j+1} = D_{i+1}^j$, *and if* $D_i^j \neq 0$, *then* $D_i^j + D_{i-1}^{j+1} + D_{i+1}^j \geq B_1^D$.

This property is implied by the equation

$$F_i^j(K_i^j, X_i^j) = X_{i-1}^{j+1} \oplus X_{i+1}^j.$$

Property 2.3. *If* $D_i^j \neq 0$ *or* $D_{i+2}^{j-1} \neq 0$, *then* $D_i^j + D_{i+2}^{j-1} + D_{i-1}^{j+1} + D_{i+3}^{j-1} \geq B_2^D$.

This property is implied by the equation

$$F_i^j(K_i^j, X_i^j) \oplus F_{i+2}^{j-1}(K_{i+2}^{j-1}, X_{i+2}^{j-1}) = X_{i-1}^{j+1} \oplus X_{i+3}^{j-1}.$$

Using these properties, we obtain

Theorem 2.1. *Let $d \geq 4$. Any consecutive 6 rounds of d-branch Type-2 generalized Feistel Structure using SP-type F-functions guarantee $B_1^D + B_2^D$ differential active S-boxes.*

Proof: We consider 6 consecutive rounds that starts from the a-th round. To make the proof easy to understand, we put $3d$ F-functions in the 6 rounds into the arranged boxes as in Figure 2.9. The width of the boxes is d. F-functions in the same round are found in the boxes in the same row, and F-functions in the next rounds are found in the next columns. The region indicated by a dashed line in Fig. 2.8 shows 6 rounds from i-th round for $d = 6$ case.

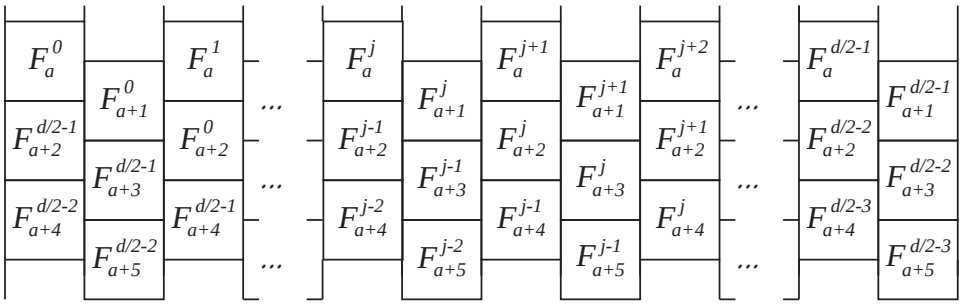

Figure 2.9: Type-2 generalized Feistel Structure (Box form)

Prop. 2.1 guarantees at least one F-function which has a non-zero difference in the 3rd or 4th rounds, i.e. $(a+2)$-th round or $(a+3)$-th round.

CASE 1 (Any non-zero difference exists in the 3rd round)
Let $D_{a+2}^j \neq 0$. Then Prop. 2.2 and 2.3 imply,

$$D_{a+2}^j + D_{a+1}^{j+1} + D_{a+3}^j \geq B_1^D \ , \tag{2.1}$$

$$D_{a+2}^j + D_{a+4}^{j-1} + D_{a+1}^{j+1} + D_{a+5}^{j-1} \geq B_2^D \ . \tag{2.2}$$

1A If $D_{a+3}^{j-1} \neq 0$, then Prop. 2.3 implies $D_{a+1}^j + D_{a+3}^{j-1} + D_a^{j+1} + D_{a+4}^{j-1} \geq B_2^D$. By combining it and (2.1), we obtain $\sum_{i=a}^{a+5} \sum_{j=0}^{d/2-1} D_i^j \geq B_1^D + B_2^D$.

1B If $D_{a+3}^j \neq 0$, then Prop. 2.2 implies $D_{a+3}^j + D_{a+2}^{j+1} + D_{a+4}^j \geq B_1^D$. By combining (2.2), we obtain $\sum_{i=a}^{a+5} \sum_{j=0}^{d/2-1} D_i^j \geq B_1^D + B_2^D$.

1C If $D_{a+3}^{j-1} = D_{a+3}^j = 0$, then the condition of $D_{a+3}^{j-1} = 0$ and Prop. 2.2 imply $D_{a+4}^{j-1} = D_{a+2}^j \neq 0$. Using the Prop. 2.2 for D_{a+4}^j, we obtain

$$D_{a+4}^{j-1} + D_{a+3}^j + D_{a+5}^{j-1} \geq B_1^D. \tag{2.3}$$

Eqs. (2.1) and (2.3) have an overlapping term D^j_{a+3}, but we assumed $D^j_{a+3} = 0$. As a result, we obtain $\sum_{i=a}^{a+5}\sum_{j=0}^{d/2-1} D^j_i \geq 2 \times B^D_1 \geq B^D_1 + B^D_2$.

The cases for the non-zero difference at other than D^j_{a+2} can also be proved in the same way.

CASE 2 (Any non-zero difference exists in the 4th round.)
We can prove the same lowerbounds for this case as CASE 1 due to the symmetry of the structure. □

Linear Active S-boxes in Type-2 generalized Feistel Structure

Similar to the differential case, we write a number of linear active S-boxes for F^j_i as L^j_i. If a non-zero linear mask is input to Type-2 generalized Feistel structure, we can use the following properties:

Property 2.4. *There is at least one F-function which contains at least one linear active S-box in any consecutive 2 rounds.*

This property is due to the invertibility of the structure.

Property 2.5. *For any set of L^j_i, L^j_{i+1} and L^{j-1}_{i+2},*

- $L^j_i = L^j_{i+1} = L^{j-1}_{i+2} = 0$, *or*
- $L^j_i + L^j_{i+1} + L^{j-1}_{i+2} \geq B^L_2$, *and two of the three terms are non-zero.*

Using the above properties, we show the following theorem.

Theorem 2.2. *Let $d \geq 4$. Any consecutive 6 rounds of d-branch Type-2 generalized Feistel structure using SP-type F-functions guarantee at least $2 \times B^L_2$ linear active S-boxes.*

Proof: Similar to Theorem 2.1, we prove that a guaranteed number of active S-boxes in 6 consecutive rounds which starts from the a-th round.
Prop. 2.4 guarantees at least one F-function which has non-zero linear mask in the 3rd or 4th rounds, i.e. $(a+2)$-th round or $(a+3)$-th round.

CASE 1 (Any non-zero linear mask exists in the 3rd round.)
$L^j_{a+2} \neq 0$. From Prop. 2.5, we obtain $L^j_{a+1} + L^j_{a+2} + L^{j-1}_{a+3} \geq B^L_2$. Assume that each term in the inequality is non-zero, we can additionally say

- $L^j_a + \boldsymbol{L^j_{a+1}} + L^{j-1}_{a+2} \geq B^L_2$
- $L^{j+1}_a + L^{j+1}_{a+1} + \boldsymbol{L^j_{a+2}} \geq B^L_2$
- $\boldsymbol{L^{j-1}_{a+3}} + L^{j-1}_{a+4} + L^{j-2}_{a+5} \geq B^L_2$

Note that these three terms under consideration are emphasized in a bold type, and there is no overlapped term in the above three inequalities. Prop. 2.5 implies that at least two of the three terms are non-zero, therefore two of the above inequalities are valid. As a result, we obtain $\sum_{i=a}^{a+5}\sum_{j=0}^{d/2-1} L_i^j \geq 2\times B_2^L$

The cases for the non-zero linear mask at other than L_{a+2}^j can also be proved in the same way.

CASE 2 (Any non-zero linear mask exists in the 4th round.)
Similarly, we can prove the same lower bounds for this case as CASE 1 due to the symmetry of the structure. □

2.3.4 Computational Evaluation

In this section we show the other approach to show lower bounds of generalized Feistel structures. We improve a known search algorithm to fit to generalized Feistel structures [64].

2.3.5 Basic Search Algorithm

The basic search algorithm counting active S-boxes is introduced in [64].

1. For each candidate in all possible combinations of weight values D_i^j (or L_i^j), $(1 \leq i \leq r)$ do:
   - Check inconsistency between given D_i^js (or L_i^js) determined by aforementioned properties. If they are inconsistent, discard the candidate, else calculate and store a sum of D_i^js $(1 \leq i \leq r)$.
2. Output the smallest sum as the lower bound of the target structure.

Properties shown in Sect. 2.3.3 are used to rule out wrong combinations of weight values. For example, in the properties of Type-2 Feistel structure, letting $B_2^D = 5$ and $D_i^j \neq 0$, then the case of $D_i^j + D_{i-1}^{j+1} + D_{i+1}^j < 5$ is an impossible combination implied by Prop. 2.1.

An actual search algorithm is as follows. Let $\mathcal{ST}_R$ be an R-round generalized structure to be evaluated, and NF_i be a total number of F-functions in the first i rounds of $\mathcal{ST}_R$. Then we define alias names of F-functions $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_{NF_R}$ as $\mathcal{F}_{di+j} = F_i^j$ for Type-2 generalized Feistel structures. Moreover, $\mathcal{D}_i$ and $\mathcal{L}_i$ denote numbers of differential and linear active S-boxes for $\mathcal{F}_i$, respectively. The basic algorithm to find the lower bounds of active S-boxes is shown in Table 2.4.

In the above $Func(x)$ is a recursive function call. To find lower bounds of linear active S-boxes, $\mathcal{D}_i$ is replaced with $\mathcal{L}_i$, and applied properties are changed for linear masks.

Table 2.4: Basic Search Algorithm

INPUT: R (a number of rounds), $\mathcal{ST}_R$ (target structure)
OUTPUT: a guaranteed number of active S-boxes for $\mathcal{ST}_R$
Main:
Step 1. Set global variable $LB = \infty$
Step 2. Call $Func(1)$
Step 3. Output LB
$Func(x)$
Step 4. If $x = NF_R + 1$ do the following:
If $LB > \sum_{p=1}^{NF_R} \mathcal{D}_p$, $LB \leftarrow \sum_{p=1}^{NF_R} \mathcal{D}_p$.
Step 5. If $x \neq NF_R + 1$. For $j = 0$ to m do the following:
Set $\mathcal{D}_x = j$ and check whether all properties
for the target structure are satisfied or not.
If check is OK, call $Func(x + 1)$

We confirmed that this algorithm works well only for small sized parameters. Our experimental result shows that even searching for 16-round Type-I Feistel structures $m = 4$, $d = 4$ requires more than one day. This huge calculation cost may sometimes be an obstacle to estimate the larger size of generalized Feistel structures.

2.3.6 Improved Search Algorithm

We speed up the basic algorithm by introducing an additional branch cutting technique. The improved algorithm is shown in Table 2.5. The major difference between the basic and the improved algorithms is that the improved algorithm makes use of the already obtained lower bounds for smaller rounds.

At Step 5.2., if the total of the sum of determined active S-boxes in the first z F-functions and the known lower bound for the rest of rounds already exceeds temporary lower bounds LB_i of the current target number i further searches are aborted, because this situation never gives a better lower bound. The branch cutting with this early-abort approach can significantly reduce the search effort. Our implementation result shows that a search for 50-round Type-2 Feistel structures when $m = 4$, $d = 4$ can be obtained within a few tens of seconds by the improved algorithm. This improvement enables us to evaluate larger structures.

Table 2.5: Improved Search Algorithm

INPUT: R (a number of rounds), $\mathcal{ST}_R$ (target structure)
OUTPUT: a guaranteed number of active S-boxes for $\mathcal{ST}_R$

Main:
Step 1. Set global variable $LB_i = \infty$ $(1 \leq i \leq R)$
Step 2. For $i = 1$ to R do the following:
Call $Func(1, i)$
Step 3. Output LB_R

$Func(x, r)$
Step 4. If $x = NF_r + 1$ do the following:
If $LB_r > \sum_{p=1}^{NF_r} \mathcal{D}_p$, $LB_r \leftarrow \sum_{p=1}^{NF_r} \mathcal{D}_p$.
Step 5. If $x \neq NF_r + 1$. For $j = 0$ to m do the following:
Set $\mathcal{D}_x = j$ and check whether all properties
for the target structure are satisfied or not.
If the check is OK, do the following:
Step 5.1. If $x \notin \{NF_k | 1 \leq k \leq r - 1\}$,
call $Func(x + 1, r)$.
Step 5.2. If $x \in \{NF_k | 1 \leq k \leq r - 1\}$
Let z be an integer satisfying $x = NF_z$.
If $\sum_{p=1}^{NF_z} \mathcal{D}_p + LB_{r-z} \leq LB_r$, call $Func(x + 1, r)$.

Chapter 3

Security

This chapter describes security of CLEFIA. To estimate security of CLEFIA, all known attacks of blockciphers are considered. After checking the applicability of each attack, immunity of CLEFIA against each attack is evaluated in detail by estimating how many rounds of CLEFIA can be attacked. The twenty types of attacks considered for CLEFIA are listed below:

1. Differential Cryptanalysis
2. Linear Cryptanalysis
3. Differential-Linear Cryptanalysis
4. Boomerang Attack
5. Amplified Boomerang Attack
6. Rectangle Attack
7. Truncated Differential Cryptanalysis
8. Truncated Linear Cryptanalysis
9. Impossible Differential Cryptanalysis
10. Saturation Cryptanalysis
11. Collision Attack
12. Higher Order Differential Cryptanalysis
13. Interpolation Cryptanalysis
14. XSL Attack
15. χ^2 Cryptanalysis

16. Slide Attack
17. Related-Cipher Cryptanalysis
18. Related-Key Cryptanalysis
19. Related-Key Boomerang Cryptanalysis
20. Related-Key Rectangle Cryptanalysis

These evaluated results are shown in this order in the following sections. In Section 3.1, security with regard to the data processing part of CLEFIA is described. Then in Section 3.2, security of CLEFIA including the key scheduling part is described.

As a result, CLEFIA with 128-bit keys can be attacked up to 12 rounds (out of 18 rounds), CLEFIA with 192-bit keys can be attacked up to 13 rounds (out of 22 rounds), and CLEFIA with 256-bit keys can be attacked up to 14 rounds (out of 26 rounds), by impossible differential attacks. However, full-round CLEFIA for each key length can not be broken even with the best attacking technique available to date, with less complexity than that of exhaustive key search.

3.1 Cryptanalysis I — Data Processing Part

In this section, the following attacks are considered to evaluate the security of the data processing part of CLEFIA.

1. Differential Cryptanalysis
2. Linear Cryptanalysis
3. Differential-Linear Cryptanalysis
4. Boomerang Attack
5. Amplified Boomerang Attack
6. Rectangle Attack
7. Truncated Differential Cryptanalysis
8. Truncated Linear Cryptanalysis
9. Impossible Differential Cryptanalysis
10. Saturation Cryptanalysis
11. Collision Attack
12. Higher Order Differential Cryptanalysis
13. Interpolation Cryptanalysis
14. XSL Attack
15. χ^2 Cryptanalysis

3.1.1 Differential Cryptanalysis

Differential cryptanalysis is a general technique for the analysis of blockciphers, which was proposed by Biham and Shamir [10, 11]. There are two ways to evaluate the immunity of blockciphers against differential attack,

1. To show there is no differential which can be used to distinguish from random permutations
2. To show there is no differential characteristic which can be used to distinguish from random permutations

So far, it is known that it is difficult to achieve the first goal for many ciphers. Although there is a useful theory proposed by Hong *et al.* to evaluate maximal differential probability of SPN structure, we cannot use the theory for the evaluations of CLEFIA as AES and FOX because CLEFIA does not use SPN type structure [22, 26, 29].

We adopt the remaining approach to estimate probabilities of differential characteristics. It is known that this can be achieved by counting guaranteed numbers of active S-boxes. This approach of counting active S-boxes is adopted by AES, Camellia and other well-known blockciphers as well [3, 22]. The gap between the maximal differential probability and the maximum differential characteristic probability was not very clear so far, but the relationship between them was discussed in detail by Daemen and Rijmen [20]. From their result, there is a certain statistical relationship between them if we accept some statistical assumptions. Thus the characteristics based approach can be considered as a reasonable way to estimate the immunity against differential cryptanalysis.

Active S-boxes

In general, there are two ways to show the guaranteed number of differential active S-boxes of blockciphers. One is to use proved lower bounds of active S-boxes, the other is to estimate the lower bounds by using a search algorithm. We checked the both methods for CLEFIA to see which approach has tighter lower bound. As a result, we found that the implied bounds by theoretical proofs are not tighter than the search based bounds. Therefore, we use the results obtained from computer search for the security estimation of CLEFIA.

Table 2.1 shows the guaranteed numbers of active S-boxes of CLEFIA obtained by the computer search. Now we focus on the columns indexed by 'DSM(D)' which show the guaranteed numbers of differential active S-boxes corresponding to the round numbers in columns indexed by 'r'.

Differential Probability of S-boxes used for Estimation

We need the differential probabilities of S-boxes to estimate the immunity against differential cryptanalysis. There are two S-boxes S_0 and S_1, each S-box has maximum differential probability $DP_{max}^{S_0} = 2^{-4.67}$ and $DP_{max}^{S_1} = 2^{-6.0}$, respectively. From designers' point of view, CLEFIA's security against differential cryptanalysis should be estimated assuming all S-boxes to be S_0, which has a larger maximum differential probability.

Differential Attack

Combining guaranteed 28 differential active S-boxes for 12-round CLEFIA (Table 2.1) and $DP_{max}^{S_0} = 2^{-4.67}$, it is shown that the maximum differential

characteristic probability $DCP_{max}^{12\text{-round}} \leq 2^{28\times(-4.67)} = 2^{-130.76}$. This means there is no useful 12-round differential characteristic for an attacker. Additionally, there are two reasons that the actual values of DCP_{max} is expected to be smaller than the above estimation. The first reason is that it is very difficult to construct a differential characteristic in which all the 28 active S-boxes use the highest differential probability $2^{-4.67}$ simultaneously. The second reason is that CLEFIA also employs S_1 with a smaller maximum differential probability. Consequently, it is considered to be difficult for an attacker to find 12-round differentials which can be used to distinguish CLEFIA from random permutations. From this observation, we believe that the full-round CLEFIA is secure against differential cryptanalysis taking the most efficient key recovery attack into consideration.

3.1.2 Linear Cryptanalysis

Linear cryptanalysis is a general technique for the analysis of blockciphers, which was proposed by Matsui [43]. In order to evaluate the immunity against linear cryptanalysis, a similar method to differential attack can be used, which is a method utilizing the knowledge of the guaranteed number of linear active S-boxes and maximum linear probability of S-boxes.

The columns indexed by 'DSM(L)' in Table 2.1 show the guaranteed numbers of linear active S-boxes corresponding to the number of rounds in columns indexed by 'r'. Since the maximum linear probability $LP_{max}^{S_0} = 2^{-4.38}$ and $LP_{max}^{S_1} = 2^{-6.00}$ respectively, we assume all S-boxes in CLEFIA are S_0. Combining 30 active S-boxes for 12-round CLEFIA and S-box property, the maximum linear characteristic probability $LCP_{max}^{12\text{-round}} \leq 2^{30\times-4.38} = 2^{-131.40}$. It is difficult to construct a linear approximation in which all the 30 active S-boxes use the highest linear probability $2^{-4.38}$ simultaneously. Moreover, CLEFIA employs stronger S-box S_1 with $DP_{max}^{S_1} = 2^{-6.00}$, so we believe that it is difficult for an attacker to find 12-round linear-hulls which can be used to distinguish CLEFIA from random permutations.

3.1.3 Differential-Linear Cryptanalysis

Differential-Linear cryptanalysis is a general technique for the analysis of blockciphers, which was proposed by Langford and Hellman [41]. This cryptanalysis uses both of differential characteristics and linear approximations. Letting p be the probability of the differential characteristic used for the attack and letting q be the probability of the linear approximation used for the attack, the complexity of the differential-linear cryptanalysis would have the complexity order of about p^2q^2. Based on characteristic based analysis, an 8-round distinguisher consists of a 3-round differential characteristic holding 2-active S-boxes with probability $2^{2\times(-4.67)} = 2^{-9.34}$ and a 5-round linear approximation holding 10 active S-boxes with probability $2^{10\times(-4.38)}$

is the best combination. However, the attack complexity by using this distinguisher is higher than those by using the best differential distinguisher or the best linear distinguisher. Therefore, we consider that full-round CLEFIA has strong immunity against differential-linear cryptanalysis.

3.1.4 Boomerang Attack

Boomerang attack is an adaptive chosen plaintext and ciphertext attack proposed by Wagner [82]. It is based on a pair of short differential characteristics used in a specially built quartet. The main idea behind the boomerang attack is to use two short differentials with high probabilities instead of one differential of more rounds with low probability.

Let n be the block size in bits and k be the key length in bits. We assume that CLEFIA $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ can be described as a cascade $E = E_1 \circ E_0$, such that for E_0 there exists a differential $\alpha \to \beta$ with probability p, and for E_1 there exists a differential $\gamma \to \delta$ with probability q. The boomerang attack uses the first characteristic $(\alpha \to \beta)$ for E_0 with respect to the pairs (P_0, P_1) and (P_2, P_3), and uses the second characteristic $(\gamma \to \delta)$ for E_1 with respect to the pairs (C_0, C_2) and (C_1, C_3). The attack is based on the following boomerang process:

- Ask for the encryption of a pair of plaintexts (P_0, P_1) such that $P_0 \oplus P_1 = \alpha$ and denote the corresponding ciphertexts by (C_0, C_1).
- Calculate $C_2 = C_0 \oplus \delta$ and $C_3 = C_1 \oplus \delta$, and ask for the decryption of the pair (C_2, C_3). Denote the corresponding plaintexts by (P_2, P_3).
- Check whether $P_2 \oplus P_3 = \alpha$.

It is easy to see that for a random permutation the probability that the last condition is satisfied is 2^{-n}. For E, however, the probability that the pair (P_0, P_1) is a right pair with respect to the first differential $(\alpha \to \beta)$ is p. The probability that both pairs (C_0, C_2) and (C_1, C_3) are right pairs with respect to the second differential is q^2. If all these are right pairs, then they satisfy

$$E_1^{-1}(C_2) \oplus E_1^{-1}(C_3) = \beta = E_0(P_2) \oplus E_0(P_3),$$

and thus, with probability p also $P_2 \oplus P_3 = \alpha$. Therefore, the total probability of this quartet of plaintexts and ciphertexts to satisfy the boomerang conditions is $(pq)^2$. Therefore,

$$pq > 2^{-64}$$

must hold for the boomerang attack to work on CLEFIA.

We found two types of 9-round boomerang distinguishers for CLEFIA, which are called distinguisher I and distinguisher II. As shown in Table 3.1,

CLEFIA with at most 9 rounds can be distinguished from a random permutation by using distinguishers I and II. The case III is not actually a distinguisher which is referred to show the evidence that 10-round extension from distinguisher I has too low probability. Since it is expected that key recovery attacks can be mounted for CLEFIA with up to 9 or a few more rounds, the full-round CLEFIA has enough security against the boomerang attacks.

Table 3.1: Boomerang Distinguishers

Case	$E_1 \circ E_0$		E
I	(E_0, E_1) (p, q)	(3.5-round, 5.5-round) $(\leq (2^{-4.68})^2, \leq (2^{-4.67})^8)$	9-round $(pq)^2 \leq 2^{-93.40}$
II	(E_0, E_1) (p, q)	(4.5-round, 4.5-round) $(\leq (2^{-4.68})^6, \leq (2^{-4.67})^6)$	9-round $(pq)^2 \leq 2^{-112.08}$
III	(E_0, E_1) (p, q)	(3.5-round, 6.5-round) $(\leq (2^{-4.68})^2, \leq (2^{-4.67})^{12})$	10-round $(pq)^2 \leq 2^{-130.76}$

Some of differential characteristics used in the above distinguisher are shown here. For E_0 in I, the 3.5-round differential characteristic shown in Figure 3.1 is used, where $a \in \{0,1\}^{32}$, $b \in \{0,1\}^{32}$, $c \in \{0,1\}^{32}$ is a non-zero value such that $w_8(a) = 1$, $w_8(b) = 4$, $w_8(c) = 4$, respectively.

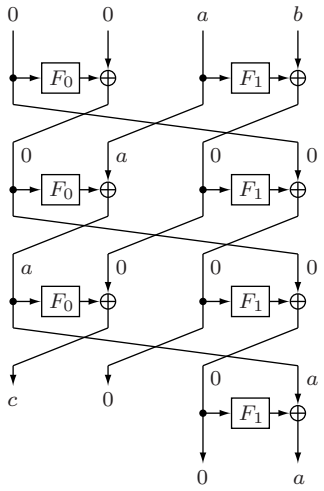

Figure 3.1: 3.5-round Differential Characteristic for E_0 (Distinguisher I)

For E_1 in distinguisher I, the 5.5-round differential characteristic shown in Figure 3.2 (Left) is used, where $d \in \{0,1\}^{32}$, $e \in \{0,1\}^{32}$, $f \in \{0,1\}^{32}$,

$g \in \{0,1\}^{32}$ $h \in \{0,1\}^{32}$ is a non-zero value such that $w_8(d) = 1$, $w_8(e) = 4$, $w_8(f) = 4$, $w_8(g) = 1$, $w_8(h) = 4$, respectively.

Moreover, E_1 in distinguisher III can be obtained from E_1 in distinguisher I by simply adding one round at the last iteration (right, Figure 3.2).

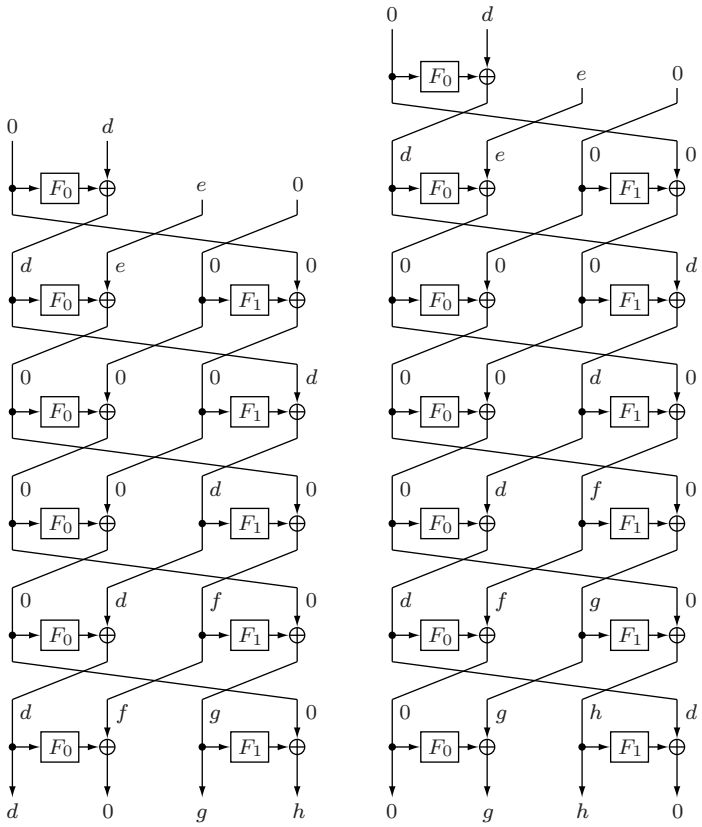

Figure 3.2: 5.5-round and 6.5-round Characteristics for E_1 (Distinguisher I, III)

3.1.5 Amplified Boomerang Attack

Amplified boomerang attack is a chosen plaintext variant of the boomerang attack [32]. The key idea behind the transformation is to encrypt many plaintext pairs with input difference α, and to look for quartets that conform to the requirements of the boomerang process.

The analysis in [32] shows that out of N plaintext pairs, the number of right quartets is expected to be $N^2 2^{-(n+1)} p^2 q^2$, where n is the block size in bits. Therefore, in the case of 9-round CLEFIA, it is expected to see one right quartet from $N = 2^{111.30}$ plaintext pairs. These plaintext pairs can

be obtained from $2^{92.30}$ structures of $2^8 \times 4$ plaintext pairs. Therefore, the attack requires $2^{92.30} \times 2^8 \times 4 = 2^{102.30}$ chosen plaintexts.

In the amplified boomerang attack scenario, CLEFIA with at most 9 rounds can be distinguished from a random permutation. Since it is expected that key-recovery attacks can be mounted for CLEFIA with up to 9 or a few more rounds, the full-round CLEFIA has enough security against amplified boomerang attacks.

3.1.6 Rectangle Attack

The rectangle attack shows that it is possible to use all the possible β and γ simultaneously, and presents additional improvements over the amplified boomerang attack. These improvements increase the probability of a quartet to be a right quartet and N plaintext pairs with input difference α are expected to produce $N^2 2^{-n} \hat{p}^2 \hat{q}^2$ right quartets, where $\hat{p}$ and $\hat{q}$ are as defined as:

$$\hat{p} = \sqrt{\sum_{\beta} \Pr^2[\alpha \rightarrow \beta]}, \qquad \hat{q} = \sqrt{\sum_{\gamma} \Pr^2[\gamma \rightarrow \delta]}. \tag{3.1}$$

By using the above observation, the existence of 10-round distinguisher is strongly implied, because the current characteristic probability of 10-round CLEFIA for the boomerang attack is slightly smaller than the threshold 2^{-128} (see Table 3.2). However, it is expected that key-recovery attacks can be mounted for CLEFIA with up to 10 or a few more rounds, so the full-round CLEFIA has enough security against rectangle attacks.

3.1.7 Truncated Differential Cryptanalysis

Truncated differential cryptanalysis is a general technique for the analysis of blockciphers, which was proposed by Knudsen [37]. Truncated differentials are differentials where only a part of the difference can be predicted. Due to the strong byte oriented design of CLEFIA, it is natural that truncated differential cryptanalysis using '0' and '1' to represent each byte data depending on existence of difference. This approach was adopted to evaluate many blockciphers including E2 and Camellia [30, 46, 48, 72]. So far, it is still an open problem to find systematic ways to evaluate immunity of Feistel-type blockciphers with SP-type F-functions against the truncated differential attack. This makes the evaluation of CLEFIA difficult.

However, we can learn from the results of E2 and Camellia because the both algorithms and CLEFIA have Feistel-type structure. The big difference with regard to truncated differential attack between two algorithms is that E2 has SPS-type F-functions but Camellia has SP-type F-functions. Best known results for them are as follows : E2 has a known 7-round truncated

differential, and Camellia without FL/FL^{-1} has a 9-round truncated differential [30, 46, 48, 72]. The lack of the second S-layer in Camellia can be considered to be a reason for producing the difference of estimated immunity against truncated differential attack.

We call modified CLEFIA whose F-function are SPS-type CLEFIA+S. Moreover we call modified CLEFIA+S whose F-functions do not use diffusion switching mechanism CLEFIA+S-D, in which only a single diffusion matrix is repeatedly used. We confirmed by a computer simulation that CLEFIA+S-D with 10 rounds (or more) does not have any useful truncated differentials. Suppose that there is a 9-round truncated differential for CLEFIA+S-D, we expect that full-round CLEFIA-D is expected to be strong against truncated differential cryptanalysis from the observation of the difference between E2 and Camellia. Moreover, if the DSM is enabled, the immunity is expected to be stronger than the above.

Since these are partial analysis of truncated differential attack obtained so far, we consider a more convincing evaluation method is required for the full-spec CLEFIA.

3.1.8 Truncated Linear Cryptanalysis

Truncated linear cryptanalysis is a general technique for the analysis of blockciphers, which was proposed by the designers of Camellia in the course of the evaluation of the cipher [2]. Due to the duality between differential and linear cryptanalysis, the security against truncated linear cryptanalysis by using a similar algorithm to truncated differential cryptanalysis can be evaluated [44]. Consequently, we believe that full-round CLEFIA even without DSM is strong against truncated linear cryptanalysis. Moreover, if the DSM is enabled, the immunity is expected to be stronger than the above.

3.1.9 Impossible Differential Cryptanalysis

The impossible differential is the differential which holds with probability zero, i.e., the differential which never happens. Using such impossible differentials, it is possible to eliminate wrong key candidates and thus find the correct key [8].

We found following 9-round impossible differential characteristics for CLEFIA [70].

- $(0, \alpha, 0, 0) \overset{9r}{\not\rightarrow} (0, \alpha, 0, 0)$
- $(0, 0, 0, \alpha) \overset{9r}{\not\rightarrow} (0, 0, 0, \alpha)$

where $\alpha \in \{0, 1\}^{32}$ is a non-zero value. Figure 3.3 shows the first 9-round impossible differential characteristic. In the figure, '+' denotes a non-zero

difference and '*' denotes an unknown difference. The second characteristic is obtained by rotating the positions of all differences in Figure 3.3.

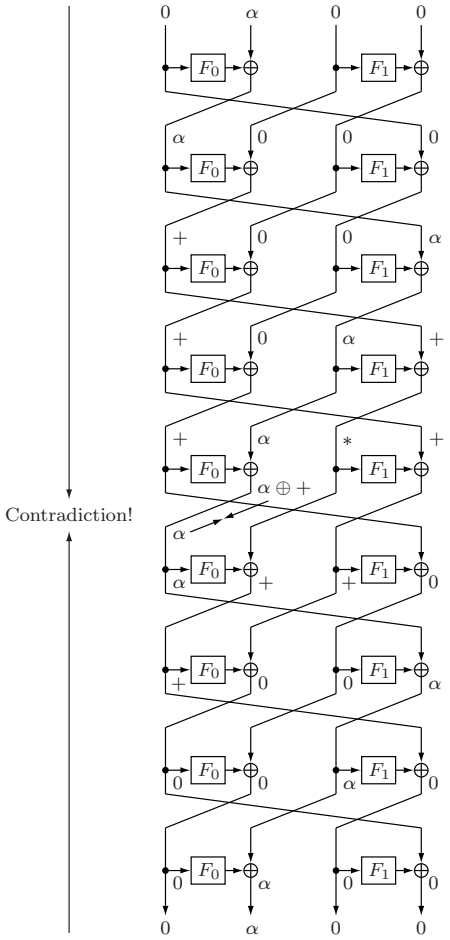

Figure 3.3: A 9-round Impossible Differential Characteristic

The designers of CLEFIA show key recovery attacks using the 9-round impossible differential characteristic in [70]. Table 3.2 shows the summary of the complexity required for the impossible differential attacks.

After the evaluation by the designers of CLEFIA [70], many results on improved impossible differential attacks on CLEFIA have been published [73, 78, 79, 83, 86].

Wang *et al.*'s attack [83] is based on the same 9-round impossible differential as [70] and decreased the complexity of recovering subkeys by some table lookups and sieving less subkey space. Their attack is applicable to CLEFIA-128 with up to 12 rounds, CLEFIA-192 with up to 13 rounds, and the CLEFIA-256 with up to 14 rounds.

Table 3.2: Summary of Impossible Differential Cryptanalysis in [70]

# of rounds	key length (bits)	key whitening	# of chosen plaintexts	time complexity
10	128, 192, 256	w/	$2^{101.7}$	2^{102}
11	192, 256	w/	$2^{103.5}$	2^{188}
12	256	w/o	$2^{103.8}$	2^{252}

Tsunoo *et al.* [79] improved the search method for impossible differentials and found the following 9-round impossible differentials, which utilize properties of DSM matrices used in CLEFIA:

- $(0, \alpha_{in}, 0, 0) \overset{9r}{\not\rightarrow} (0, \alpha_{out}, 0, 0)$
- $(0, 0, 0, \alpha_{in}) \overset{9r}{\not\rightarrow} (0, 0, 0, \alpha_{out})$

where α_{in} and $\alpha_{out} \in \{0,1\}^{32}$ take the differential values shown in Table 3.3. The a and b in Table 3.3 are 8-bit arbitrary non-zero values.

Table 3.3: Differential values for α_{in}, α_{out} [79]

α_{in}	α_{out}		
(0,0,0,a)	(0,0,b,0),	(0,b,0,0),	(b,0,0,0)
(0,0,a,0)	(0,0,0,b),	(0,b,0,0),	(b,0,0,0)
(0,a,0,0)	(0,0,0,b),	(0,0,b,0),	(b,0,0,0)
(a,0,0,0)	(0,0,0,b),	(0,0,b,0),	(0,b,0,0)

Using the 9-round impossible differential above, Tsunoo *et al.* [79] reduced the complexity for the attack with additional techniques including movement of the whitening keys and use of a difference distribution table of the S-box. Their attack is applicable to 128-bit key CLEFIA with up to 12 rounds, 192-bit key CLEFIA with up to 13 rounds, and 256-bit key CLEFIA with up to 14 rounds. Furthermore, their attack is extended to a more efficient attack which recovers more subkey bits with less complexity by Tsujihara *et al.* [78]. Their attack is based on the new 9-round impossible differential where α_{in} and $\alpha_{out} \in \{0,1\}^{32}$ take the differential values shown in Table 3.4. The a, b and c in Table 3.4 are 8-bit arbitrary non-zero values.

Table 3.5 shows the current best results of the impossible differential attacks [78].

Independently, Sun *et al.* presented their results of impossible differential attack [73], but the paper was withdrawn afterwards.

Table 3.4: Differential values for α_{in}, α_{out} [78]

α_{in}	α_{out}
(0,0,0,a)	(0,0,b,c), (0,b,0,c), (b,0,0,c)
(0,0,a,0)	(0,0,b,c), (0,b,c,0), (b,0,c,0)
(0,a,0,0)	(0,b,0,c), (0,b,c,0), (b,c,0,0)
(a,0,0,0)	(b,0,0,c), (b,0,c,0), (b,c,0,0)
(0,0,b,c)	(0,0,0,a), (0,0,a,0)
(0,b,0,c)	(0,0,0,a), (0,a,0,0)
(b,0,0,c)	(0,0,0,a), (a,0,0,0)
(0,b,c,0)	(0,0,a,0), (0,a,0,0)
(b,0,c,0)	(0,0,a,0), (a,0,0,0)
(b,c,0,0)	(0,a,0,0), (a,0,0,0)

Table 3.5: Current Best Results on Impossible Differential Cryptanalysis [78]

# of rounds	key length (bits)	key whitening	# of chosen plaintexts	time complexity	memory (blocks)
12	128, 192, 256	w/	$2^{111.0}$	2^{111}	2^{81}
13	192, 256	w/	$2^{111.8}$	2^{155}	2^{112}
14	256	w/	$2^{112.3}$	2^{220}	2^{113}

At Inscrypt 2008, Zhang *et al.* [86] claimed that CLEFIA-128 without whitening key layers can be attacked up to 14 rounds by using the same 9-round impossible differential shown in [79]. However, there was a flaw in the time complexity in the pre-proceedings [86] and we pointed out it [85]. As a result, in the final proceeding version published from Springer [86], the authors deleted calculation details on the time complexity which appeared in the pre-proceedings, and described that whether their attack scenario is successful is waiting to be proved.

So far, impossible differential attacks on CLEFIA have been extensively studied by the designers and external experts. Neverthless Table 3.5 shows that it is expected that full-round CLEFIA has enough security against impossible differential attacks.

3.1.10 Saturation Cryptanalysis

Saturation cryptanalysis was first proposed by Daemen *et al.* [19] as a dedicated attack called "square attack", which was applied to the blockcipher

Square. This type of attack is also known as multiset cryptanalysis.

Distinguisher based on Byte Saturation

Typically, the saturation cryptanalysis makes use of byte-oriented structure of blockciphers. This type of attack works well on AES [23]. Since CLEFIA also has strong byte oriented structure, we first consider the byte-based saturation cryptanalysis.

Let $X = \{X_i | X_i \in \{0,1\}^8\}$ $(0 \leq i < 2^8)$ be a set of all 8-bit values. Now we categorize status of the set X_i into four groups depending on conditions defined as follows.

- `Const (C)` : if $\forall i, j \;\; X_i = X_j$,
- `All (A)` : if $\forall i, j \;\; i \neq j \Leftrightarrow X_i \neq X_j$,
- `Balance (B)` : if $\bigoplus_i X_i = 0$,
- `Unknown (U)` : unknown.

Then consider the situation that there are 256 plaintexts in which all bytes are `Const` except only one `All` byte. Using the conditions, saturation relationship between input and output for 5-round CLEFIA can be written as follows.

- `((C C C C) (C C C A) (C C C C) (C C C C))`
  $\xrightarrow{5r}$ `((U U U U) (U U U U) (B B B B) (U U U U))`
- `((C C C C) (C C C C) (C C C C) (C C C A))`
  $\xrightarrow{5r}$ `((B B B B) (U U U U) (U U U U) (U U U U))`

Moreover, if `(C C C A)` is replaced by one of `(C C A C)`, `(C A C C)`, `(A C C C)`, the above output conditions are valid. As a result, there are eight saturation paths in total. The first saturation path is depicted in Figure 3.4. These paths make 5-round CLEFIA distinguishable from a random permutation since `Balance` is found at the output.

Key Recovery Attack on 7-round CLEFIA

We consider key recovery attacks using the above saturation characteristics. We assume that there are additional 2 rounds after the 5-round saturation characteristic (see Figure 3.5), and RK_{10} and RK_{13} are the keys to be recovered. This attack uses the fact that only a part of ciphertext $C_0^{(7)}, C_2^{(7)}, C_3^{(7)}$ and round keys RK_{10}, RK_{13} are required to decrypt the 32-bit data which is saturated as `(B B B B)`.

In this setting, round keys $(RK_{10}, RK_{13}) \in \mathcal{K}$ can be derived as follows:

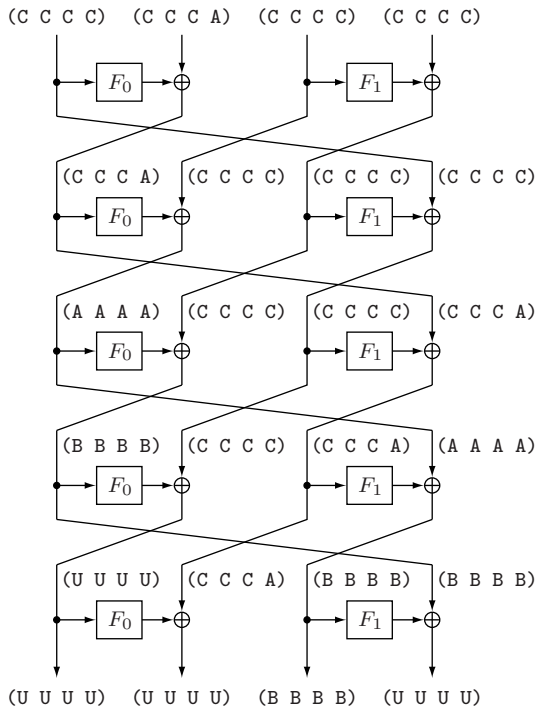

Figure 3.4: A Saturation Characteristic for 5-round CLEFIA

1. Guess an element $k_{guess10}$ for RK_{10} and $k_{guess13}$ for RK_{13} .
2. For each guessed key value,
   - For each ciphertext, compute
   $$Z_i = F_0(k_{guess13}, C_2^{(7)}) \oplus C_3^{(7)} \ ,$$
   then compute
   $$Y_i = F_1(k_{guess10}, Z_i) \oplus C_0^{(7)} \ .$$
   Compute the exclusive-or of all Y_i, $Y = \bigoplus_i Y_i$.
3. If $Y = 0$, then $k_{guess10}, k_{guess13}$ is a candidate for RK_{10}, RK_{13}, respectively. If $Y \neq 0$, then the guess is not the correct value for RK_{10}, RK_{13}, respectively.

The probability that a key candidate in the key space survives the above discarding step is expected to be 2^{-32}. Therefore, three sets of 256 plaintexts is enough for narrowing down to one correct key value. The time complexity is $2^{64} \times 2^8 \times 2^8$ calculations of F-function which is about 2^{80} F-function

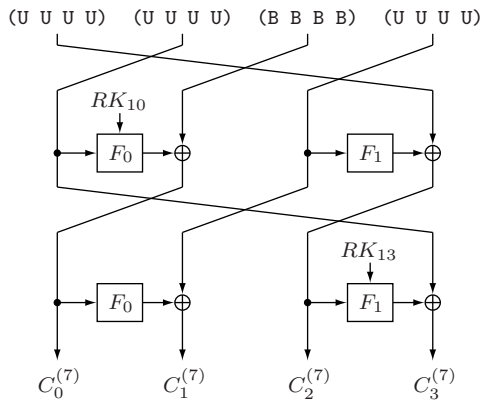

Figure 3.5: Key Recovery Attack on 7-round CLEFIA

executions. Consequently, this attack is applicable to 128-bit, 192-bit and 256-bit keys.

Attacks for additional round may be possible, but we expect that the number of additional rounds is limited as far as the same saturation characteristic is used. However, if we extend the size of saturated word from 8-bit to 32-bit, we can attack longer rounds of CLEFIA. It is explained below.

Distinguisher based on 32-bit Word Saturation

We consider the 32-bit word oriented saturation attack as follows.

Let $X = \{X_i | X_i \in \{0,1\}^{32}\}$ be a set of all 32-bit values, then we categorize status of the set of X_i into four groups in the same fashion.

- `Const (C)` : if $\forall i, j \;\; X_i = X_j$,
- `All (A)` : if $\forall i, j \;\; i \neq j \Leftrightarrow X_i \neq X_j$,
- `Balance (B)` : if $\bigoplus_i X_i = 0$,
- `Unknown (U)` : unknown.

Using these extended conditions, the saturation relationship between input and output for 6-round CLEFIA can be written as follows.

- `(C A C C)` $\stackrel{6r}{\rightarrow}$ `(B U U U)`
- `(C C C A)` $\stackrel{6r}{\rightarrow}$ `(U U B U)`

The first saturation path is depicted in Figure 3.6. These paths make 6-round CLEFIA distinguishable from a random permutation since `Balance` is found at the output.

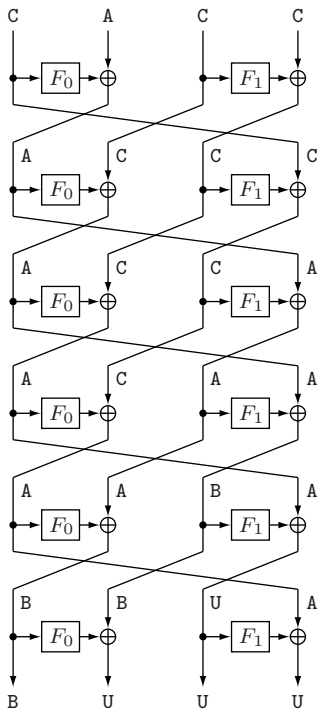

Figure 3.6: Saturation Characteristic (32-bit)

These 6-round distinguishers can be extended into 8-round distinguishers. We first explain how to extend to 7-round distinguishers. Let $\mathtt{A}_{(64)}$ be an `All` state of 64-bit words, and it is divided into two segments as $\mathtt{A}_{(64)} = \mathtt{A}_{0(64)} \mid \mathtt{A}_{1(64)}$. Using this, we can get the 7-round distinguisher as:

- $(\mathtt{C}\ \mathtt{C}\ \mathtt{A}_{0(64)}\ \mathtt{A}_{1(64)}) \xrightarrow{7r} (\mathtt{B}\ \mathtt{U}\ \mathtt{U}\ \mathtt{U})$
- $(\mathtt{A}_{0(64)}\ \mathtt{A}_{1(64)}\ \mathtt{C}\ \mathtt{C}) \xrightarrow{7r} (\mathtt{U}\ \mathtt{U}\ \mathtt{B}\ \mathtt{U})$

These distinguishers require 2^{64} plaintexts.

This is explained as follows. After the first round $(\mathtt{C}\ \mathtt{C}\ \mathtt{A}_{0(64)}\ \mathtt{A}_{1(64)})$ becomes $(\mathtt{C}\ \mathtt{A}_{0(64)}\ \mathtt{A'}_{1(64)}\ \mathtt{C})$ where the concatenated segment $\mathtt{A}_{0(64)} \mid \mathtt{A'}_{1(64)}$ is also `All`. It can be viewed that $(\mathtt{C}\ \mathtt{A}_{0(64)}\ \mathtt{A'}_{1(64)}\ \mathtt{C})$ contains 2^{32} structures `(C A C C)` where the second rightmost constant takes all possible 2^{32} values. Therefore `Balance` is kept at the output which is directly suggested by the above 6-round distinguishers.

Extension to 8-round distinguisher is obtained in a similar way. Let $\mathtt{A}_{(96)}$ be an `All` state of 96-bit words, and it is divide into three segments as $\mathtt{A}_{(96)} = \mathtt{A}_{0(96)} \mid \mathtt{A}_{1(96)} \mid \mathtt{A}_{2(96)}$. Using this, we can get the 8-round distinguishers as:

- $(\mathtt{A}_{0(96)}\ \mathtt{C}\ \mathtt{A}_{1(96)}\ \mathtt{A}_{2(96)}) \xrightarrow{8r} (\mathtt{B}\ \mathtt{U}\ \mathtt{U}\ \mathtt{U})$

- $(\mathtt{A}_{0(96)}\ \mathtt{A}_{1(96)}\ \mathtt{A}_{2(96)}\ \mathtt{C}) \xrightarrow{8r} (\mathtt{U}\ \mathtt{U}\ \mathtt{B}\ \mathtt{U})$

These distinguishers require 2^{96} plaintexts.

In the following, a 9-round attack and a 10-round attack are described.

Key Recovery Attack on 9-round CLEFIA

We assume that there is an additional round after the 8-round distinguishers, and RK_{17} is the key to be recovered.

1. Input a set of 2^{96} plaintexts which has the following format: $(\mathtt{A}_{0(96)}\ \mathtt{C}\ \mathtt{A}_{1(96)}\ \mathtt{A}_{2(96)})$

2. For the output word $C_2^{(9)}$, count the frequencies of the values. Then make a list, $LIST$, of the 32-bit values with odd frequencies.

3. For the output word $C_3^{(9)}$, compute the exclusive-or of the all 2^{96} values, denoted Y.

4. For all candidates $l_i \in LIST$ and each guessed key value k_{guess} for RK_{17}, compute

$$Z = \bigoplus_i F_1(k_{guess}, l_i) \ .$$

   - If $Z = Y$, then k_{guess} is a candidate for RK_{17}. If $Z \neq Y$, then k_{guess} is not the correct value for RK_{17}.

The probability that a key candidate in the key space survives the above discarding step is expected to be 2^{-32}. Therefore, three sets of 2^{32} plaintexts are enough for narrowing down to one correct key value. The time complexity is about 2^{31} calculations of F-function for $LIST$. Consequently, this attack is applicable to 128-bit, 192-bit and 256-bit keys.

Key Recovery Attack on 10-round CLEFIA

We assume that there are two additional rounds after the 8-round distinguishers, and RK_{17} and RK_{18} are the keys to be recovered. Basically, the scenario is the same as that of the 7-round attack on CLEFIA based on byte-oriented saturation described in Section 3.1.10.

In this setting, round keys (RK_{17}, RK_{18}) can be derived as follows:

1. Guess an element $k_{guess17}$ for RK_{17} and $k_{guess18}$ for RK_{18} .

2. For each guessed key value,

- For each ciphertext, compute

$$Z_i = F_0(k_{guess18}, C_0^{(10)}) \oplus C_1^{(10)} \ ,$$

then compute

$$Y_i = F_1(k_{guess17}, Z_i) \oplus C_2^{(10)} \ .$$

Compute the exclusive-or of all Y_i, $Y = \bigoplus_i Y_i$.

3. If $Y = 0$, then $k_{guess17}, k_{guess18}$ is a candidate for RK_{17}, RK_{18}, respectively. If $Y \neq 0$, then the guess is not the correct value for RK_{17}, RK_{18}, respectively.

The probability that a key candidate in the key space survives the above discarding step is expected to be 2^{-32}. Therefore, more than two sets of 2^{32} plaintexts are required for narrowing down to one correct key value. The time complexity is about 2^{128} calculations of F-function because at most 2^{32} F_0 calculations and 2^{32} F_1 calculations are required for each guessed key (64 bits). Consequently, this attack is applicable to 128-bit, 192-bit and 256-bit keys.

Although we have shown several versions of key recovery attacks for reduced-round CLEFIA, the attackable numbers of rounds which will be extended in the future is expected within a few more rounds. Therefore, we believe full-round CLEFIA holds strong immunity against saturation cryptanalysis.

3.1.11 Gilbert-Minier Collision Attack

Gilbert-Minier Collision attack is a kind of saturation attack, which is proposed by Gilbert and Minier [25]. In their original paper, this attack utilized a special type of 4-round distinguisher customized for Rijndael. It seems to be difficult to apply the same distinguishing function to CLEFIA. Using this technique, 7-round Rijndael can be attacked as opposed to 6-round attack by the normal saturation attack. We also expect that similar technique can be applied to CLEFIA, but expected gaining is within a few rounds as well. Therefore, we believe full-round CLEFIA holds strong immunity against Gilbert-Minier Collision attack.

3.1.12 Higher Order Differential Cryptanalysis

This type of attack was developed in [28, 37, 40] and works well for blockciphers for which the nonlinear components can be represented as Boolean polynomials of low degree. The attack is based on the following fact: if the intermediate bits of the blockcipher are represented by Boolean polynomials of degree d, then the $(d+1)$-st order differential of the polynomial is zero.

For CLEFIA S-boxes S_0 and S_1, their degrees are 6 and 7, respectively. In more detail, the S-box S_0 consists of four smaller S-boxes, SS_0, SS_1, SS_2, and SS_3. Let $SS_i : \{0,1\}^4 \rightarrow \{0,1\}^4$ where $SS_i(x_0, x_1, x_2, x_3) = (y_0, y_1, y_2, y_3)$, and if we write y_j as a Boolean polynomial in (x_0, x_1, x_2, x_3), then we verified

$$\deg(y_j) = 3$$

holds for all $0 \leq j \leq 3$. Furthermore, let $S_0 : \{0,1\}^8 \rightarrow \{0,1\}^8$ where $S_0(x_0, x_1, \ldots, x_7) = (y_0, y_1, \ldots, y_7)$, and if we write y_j as a Boolean polynomial in $(x_0, x_1, \ldots, x_7)$, then we verified

$$\deg(y_j) = 6$$

holds for all $0 \leq j \leq 7$ by deriving concrete Boolean expressions.

Next, the S-box S_1 is based on the inversion function over GF(2^8), and it has the highest possible degree of 8-bit S-boxes. That is, let $S_1 : \{0,1\}^8 \rightarrow \{0,1\}^8$ where $S_1(x_0, x_1, \ldots, x_7) = (y_0, y_1, \ldots, y_7)$ and if we write y_j as a Boolean polynomial in $(x_0, x_1, \ldots, x_7)$, then we verified

$$\deg(y_j) = 7$$

for all $0 \leq j \leq 7$ by deriving concrete Boolean expressions.

Therefore, it is expected that the degree of an intermediate bit increases exponentially as the data passes through the S-boxes, whose degree is at least 6.

We expect that after passing three S-boxes, it is difficult to collect data for taking the $(d+1)$-st order differential since $6^3 > 128$. We believe that the higher order differential cryptanalysis has limited applications on CLEFIA, and the full-round CLEFIA is strong enough against this attack.

3.1.13 Interpolation Cryptanalysis

This type of attack was proposed by Jakobsen and Knudsen in [28] and it works for blockciphers for which the nonlinear components have a simple mathematical description. The principle of interpolation attack is that, if the ciphertext is represented as a polynomial or rational expression of the plaintext whose number of unknown coefficients is N, then the polynomial or rational expression can be constructed using N pairs of plaintext and ciphertext. If the attacker constructs the polynomial or rational expression, then it is possible to encrypt any plaintext into the corresponding ciphertext or decrypt any ciphertext into the corresponding plaintext without knowing the key. Since N determines the complexity and the number of pairs required for the attack, it is important to make N as large as possible. If N is so large that it is impractical for the attackers to collect N plaintext-ciphertext pairs, the blockcipher is secure against interpolation attack.

For CLEFIA S-boxes S_0 and S_1, we evaluate the number of terms to represent them as polynomials in GF(2^8). Let $S_0 : \mathrm{GF}(2^8) \rightarrow \mathrm{GF}(2^8)$ where $S_0(x) = y$, and if we write y as a polynomial in x, then we verify that the minimum number of terms is 244, where the minimum is taken over all irreducible polynomials.

Next, for S-box $S_1 : \mathrm{GF}(2^8) \rightarrow \mathrm{GF}(2^8)$ where $S_1(x) = y$, we verify that the minimum number of terms is 252, where the minimum is taken over all irreducible polynomials.

In both cases, the number of terms is close to the maximum value, 255, for a permutation over GF(2^8). Furthermore the use of two S-boxes S_0 and S_1 is likely to destroy any mathematical structure from the individual S-box in few rounds.

We believe it is very unlikely that the interpolation attack will be of any threat to CLEFIA.

3.1.14 XSL Cryptanalysis

A pure algebraic construction for the S-boxes has many interesting non-linear properties. However, they may lead to the possible expression of a blockcipher as a system of sparse, over-defined low-degree multivariate polynomial equations over GF(2) or GF(2^8), and this fact may lead to attacks, as argued by Courtois and Pieprzyk in [18].

In what follows, we estimate the complexity of an XSL attack against modified version of CLEFIA, called CLEFIA-I, by following the very same methodology than [18]. We consider the first XSL attack as in [29], where the goal of the attack is to derive the round keys. Thus, in this scenario, we do not consider the key scheduling part.

CLEFIA-I is obtained from CLEFIA by simply replacing all 4-bit S-boxes, SS_0, SS_1, SS_2, and SS_3, by an identity function I, i.e., $I : \{0,1\}^4 \rightarrow \{0,1\}^4$, where $I(x) = x$. Notice that attacking CLEFIA-I is much easier than attacking CLEFIA since the replaced 4-bit S-boxes do not contribute to increasing the security against XSL attacks.

According to Courtois and Pieprzyk [18], the complexity of the XSL attack can be estimated to

$$T^{\omega} \text{ with } T \approx (t-\rho)^P \binom{S_{\text{inv}}}{P}, \tag{3.2}$$

where:

- T is the total number of terms,
- ω is the complexity exponent of a Gaussian reduction,
- t is the number of monomials to represent the S-box S_1,
- ρ is the number of equations,

- S_{inv} is the number of S-boxes considered during the attack, and
- The integer, P, is the parameter of the attack.

Now if the S-box on s bits is an affine transformation of the inverse function in $\mathrm{GF}(2^s)$, then it will give $\rho = 3s - 1$ bi-affine equations true with probability 1, and one additional equation true with probability $1 - 2^{-s}$ [18]. Based on the similar observation than in [18], we have $t = 81$ and $\rho = 23$.

Next S_{inv} is the total number of S-boxes S_1 considered during an attack. Then for $r = 18$ we have

$$S_{\text{inv}} = 2 \times 2 \times 18 \times 2 = 144,$$

since each F_i is built from two S-boxes, there are two F-functions in one round, there are 18 rounds, and we need 2 plaintext-ciphertext pairs. Note that we need 2 known plaintext-ciphertext pairs to uniquely determine the round keys, since round keys involve both K and L. Similarly, we have $S_{\text{inv}} = 2 \times 2 \times 22 \times 4 = 352$ for $r = 22$, and $S_{\text{inv}} = 2 \times 2 \times 26 \times 4 = 416$ for $r = 26$, since we need at least 4 known plaintext-ciphertext pairs to uniquely determine the round keys, as round keys involve K_L, K_R, L_L, and L_R.

There are conditions on the parameter of the attack, P [18]. According to [29], P is given by

$$P = \frac{t - \rho}{s + \frac{t'}{S_{\text{inv}}}},$$

where $t' = 25$ in our case. This gives $P = 8$ for $r = 18$, 22, and 26.

Courtois and Pieprzyk [18] assume that $\omega = 2.376$, which is the best known value obtained by Coppersmith and Winograd [17]. According to [18], the constant factor in this algorithm is unknown to the authors of [17], and is expected to be very big. Accordingly, it is disputed whether such an algorithm can be applied efficiently in practice. For this reason, we will consider both $\omega = 2.376$ and $\omega = 3$ in our estimations.

Given above values and based on Eq.(3.2), the total number of terms can be estimated as $T = 81^8 \binom{144}{8} > 2^{50+41} = 2^{91}$ for CLEFIA-I with $r = 18$, which gives the complexity $T^{2.376} = 2^{216}$ and $T^3 = 2^{273}$. For CLEFIA-I with $r = 22$, we have $T = 81^8 \binom{352}{8} > 2^{50+52} = 2^{102}$, and thus $T^{2.376} = 2^{242}$ and $T^3 = 2^{306}$. Finally, for CLEFIA-I with $r = 26$, we have $T = 81^8 \binom{416}{8} > 2^{50+54} = 2^{104}$, $T^{2.376} = 2^{247}$, and $T^3 = 2^{312}$.

A summary of our estimations is given in Table 3.6. At the light of the previous discussion, we should interpret these figures with an extreme care: on the one hand, the real complexity of XSL attacks is by no means clear at the time of writing and is the subject of much controversy [42, 51].

Furthermore, we are eliminating S_0, which is the highly impractical and most pessimistic hypotheses that S_0 has no contribution on the strength against XSL attacks. We believe the actual attack against original CLEFIA must be much harder than this estimation.

Table 3.6: Estimations of Complexity of XSL Attacks against CLEFIA-I.

	$\omega = 2.376$	$\omega = 3$
$r = 18$	2^{216}	2^{273}
$r = 22$	2^{242}	2^{306}
$r = 26$	2^{247}	2^{312}

So far, we have considered the XSL attack on GF(2), and the XSL technique on GF(2^8) may lead to an efficient key recovery attack [18]. At the time of this writing, it is not possible to check the effectiveness of this approach as pointed out in [42], and we believe that the XSL estimates do not have the accuracy needed to substantiate claims of the existence of an efficient key recovery attack based on the XSL technique. Still, we believe two S-boxes, S_0 and S_1, make the applicability of the XSL technique substantially harder than the single S-box case, and prevent possible attacks based on the XSL attacks.

3.1.15 χ^2 Cryptanalysis

The χ^2 cryptanalysis is a kind of statistical attack for the analysis of block-ciphers. This attack was originally proposed by Vaudenay [80], and was applied to RC6 by Gilbert et al. [24] and Knudsen and Meier [35], independently. In [35], Knudsen and Meier attacked up to 15-round RC6 with general keys and 17-round RC6 with weak keys. The vulnerability of RC6 against χ^2 cryptanalysis consists in the use of data dependent rotations in the design of RC6. We consider there is no correlation useful for χ^2 attacks in the design of CLEFIA. Therefore, we believe the χ^2 cryptanalysis does not work on the full-round CLEFIA.

3.2 Cryptanalysis II — Key Scheduling Part

In this section, security of CLEFIA including the key scheduling part is evaluated.

3.2.1 Slide Attack

Slide attack is a general technique for the analysis of a key scheduling part of blockciphers, which was proposed by Biryukov and Wagner [12]. So far, it is known that there is a good countermeasure against slide attack using round constants independent of each round. In CLEFIA there are round constants shown in the specification. Therefore CLEFIA is expected enough immunity against the slide attack.

3.2.2 Related-Cipher Attack

Related-cipher attack is a general technique for the analysis of a key scheduling part of blockciphers, which was proposed by Wu [84]. Consider a blockcipher whose numbers of rounds vary depending on the lengths of key, and whose round keys set for all key length are identical expect the round keys for additional rounds. In this case, these blockciphers with different number of rounds are called 'related', and the longer-round cipher will be easily crypt-analyzed by using shorter-round encryption results if round keys of them are the same. Due to the similarity between the key scheduling algorithms of 192-bit key and 256-bit key of CLEFIA, there is a risk for this attack. In order to avoid the related-cipher attack, CLEFIA uses different sets of round constants for each key length [71]. This is the same case as the slide attack. Therefore CLEFIA is expected to have enough immunity against the related-cipher attack.

3.2.3 Related-Key Cryptanalysis

Biham proposed related-key cryptanalysis [7]. This attack considers the information that can be extracted from two encryptions using related keys. The concept was used by Kelsey et al. in [33] to present the idea of related-key differentials. These differentials study the development of differences in two encryptions under two related keys.

A related-key differential is a triplet of a plaintext difference ΔP, a ciphertext difference ΔC, and a key difference ΔK, such that

$$\Pr_{P,K}[E_K(P) \oplus E_{K \oplus \Delta K}(P \oplus \Delta P) = \Delta C]$$

is high enough (or zero).

As for CLEFIA with 128-bit keys, L is generated by using $GFN_{4,12}$ where $GFN_{4,12}$ is a 12-round generalized Feistel structure with 4 data lines.

As in Table 2.1, $GFN_{4,12}$ has at least 28 active S-boxes, and we have $DCP_{max} \leq 2^{28\times(-4.67)} = 2^{-130.76}$. Therefore, for any ΔK and ΔL, a differential characteristic probability of $(\Delta K \rightarrow \Delta L)$ is expected to be less than 2^{-128}, i.e., no useful differential $(\Delta K \rightarrow \Delta L)$ exists. This implies the probability of any related-key differential $(\Delta P, \Delta C, \Delta K)$ is less than 2^{-128}, if all the information on ΔL is needed, since all the bits in L are used as round keys in 2 consecutive rounds. Other types of distinguishers may use $(\Delta K \rightarrow \Delta L)$, where some of the words in ΔL are unknown. We consider these types of distinguishers have limited effect since the unknown word propagates to all words in at least 3 rounds. Also we are not aware of any related-key differential with probability zero.

For CLEFIA with 192 and 256-bit keys, (L_L, L_R) is generated by applying $GFN_{8,10}$, where $GFN_{8,10}$ is a 10-round generalized Feistel structure with 8 data lines. The round keys for $GFN_{8,10}$ are fixed constants determined by the key length. From Table 2.3, it has at least 29 differential active S-boxes, which implies there are no differential characteristics with probability more than 2^{-128}. That is, for any $(\Delta K_L, \Delta K_R)$ and $(\Delta L_L, \Delta L_R)$, a differential characteristic probability of $((\Delta K_L, \Delta K_R) \rightarrow (\Delta L_L, \Delta L_R))$ is expected to be less than 2^{-128}. Therefore, the probability of any related-key differential $(\Delta P, \Delta C, (\Delta K_L, \Delta K_R))$ is less than 2^{-128}, if all the values of $(\Delta L_L, \Delta L_R)$ are needed, since all the bits in L_L and L_R are used as round keys in at least 6 consecutive rounds.

Other types of distinguishers may use $((\Delta K_L, \Delta L_R) \rightarrow (\Delta L_L, \Delta L_R))$, where some of the words in $(\Delta L_L, \Delta L_R)$ are unknown, but this is not effective as in the case for CLEFIA with 128-bit keys.

Therefore, we believe full-round CLEFIA holds strong immunity against related-key cryptanalysis.

3.2.4 Related-Key Boomerang Cryptanalysis

The main idea behind the attack is to use two short related-key differentials with high probabilities instead of one long related-key differential with a low probability.

Let n be the block size in bits and k be the key length in bits. As in the case for the boomerang attack, we assume that CLEFIA $E : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ can be described as a cascade $E = E_1 \circ E_0$, such that for E_0 there exists a related-key differential $\alpha \rightarrow \beta$ under a key difference ΔK_0 with probability p, and for E_1 there exists a related-key differential $\gamma \rightarrow \delta$ under a key difference ΔK_1 with probability q.

The related-key boomerang process involves four different unknown (but

related) keys K_a, K_b, K_c, K_d:

$$\begin{aligned} & K_a, \\ K_b &= K_a \oplus \Delta K_0, \\ K_c &= K_a \oplus \Delta K_1, \\ K_d &= K_a \oplus \Delta K_0 \oplus \Delta K_1 \end{aligned}$$

The attack is performed by the following algorithm:

- Choose a plaintext P_a at random and compute $P_b = P_a \oplus \alpha$.
- Ask for the encryption of P_a under K_a ($C_a = E_{K_a}(P_a)$)) and P_b under K_b ($C_b = E_{K_b}(P_b)$).
- Compute $C_c = C_a \oplus \delta$ and $C_d = C_b \oplus \delta$.
- Ask for the decryption of C_x under K_x, i.e., $P_c = E_{K_c}^{-1}(C_c)$ and $P_d = E_{K_d}^{-1}(C_d)$.
- Test whether $P_c \oplus P_d = \alpha$.

It is easy to see that for a random permutation the probability that the last condition is satisfied is 2^{-n}. For E the probability that this condition is satisfied is p^2q^2 just like for a regular boomerang attack, and we need $(pq)^2 > 2^{-128}$, i.e., $pq > 2^{-64}$, in order to attack CLEFIA.

As we have seen in the previous section, since we employ $GFN_{4,12}$ for 128-bit key schedule and $GFN_{8,10}$ for 192/256-bit key schedules in CLEFIA, it is very hard to achieve the condition $pq > 2^{-64}$. Indeed, we are not aware of E_0 and E_1 with more than a few rounds satisfying this condition.

Therefore, we believe it is very unlikely that the attack will be of any threat to CLEFIA.

3.2.5 Related-Key Rectangle Cryptanalysis

The transformation of the related-key boomerang attack into a related-key rectangle attack is similar to the transformation of the boomerang attack into the rectangle attack. Assume that E can be decomposed as before, where α, δ, $\hat{p}$, $\hat{q}$, K_a, K_b, K_c, and K_d have the same meaning. Then, the related-key rectangle distinguisher is as follows:

- Choose N plaintext pairs (P_a, P_b), where $P_b = P_a \oplus \alpha$, at random and ask for the encryption of P_a under K_a and of P_b under K_b.
- Choose N plaintext pairs (P_c, P_d), where $P_d = P_c \oplus \alpha$, at random and ask for the encryption of P_c under K_c and of P_d under K_d.
- Search for quartets of plaintexts (P_a, P_b, P_c, P_d) and the corresponding ciphertexts (C_a, C_b, C_c, C_d), satisfying $C_a \oplus C_c = C_b \oplus C_d = \delta$.

Then starting with N plaintext pairs with input difference α to be encrypted under K_a and K_b, we expect $N^2 2^{-n}(\hat{p}\hat{q})^2$ right quartets.

The attack requires $(\hat{p}\hat{q})^2 > 2^{-128}$ in order to apply to CLEFIA, however, we have not found E_0 and E_1 more than a few rounds satisfying this condition. Therefore, we believe CLEFIA is strong enough against this attack.

Chapter 4

Performance Evaluations

This chapter describes performance evaluations of software implementations and hardware implementations of CLEFIA, and security against side channel attacks for CLEFIA.

4.1 Software Implementations

This section describes performance evaluations of software implementations of CLEFIA. We present the evaluation results on current software performance of CLEFIA both in C language and in assembly language. The platforms we evaluated are summarized in Table 4.1.

Table 4.2 shows the evaluation results of CLEFIA in C language. We measured software processing speed of encryption/decryption and key setup using the rdtsc instruction.

We estimated the memory usage of CLEFIA software implementations. The results are shown in Table 4.3. We measured the stack usage by using the checkstack.pl script, which is included in recent Linux kernel sources, and counted the maximum stack usage based on the output of objdump.

Table 4.4 shows software performance results in assembly language on

Table 4.1: Evaluation platforms

Platform	Processor	Clock [GHz]	OS	Compiler
1	AMD Opteron	2.6	Red Hat Enterprise Linux 3	gcc 3.2.3
2	Intel Core2 Duo	2.4	Windows Vista (32-bit)	Intel C++ Compiler 11.0
3	Intel Core2 Duo	2.4	Windows Vista (64-bit)	Intel C++ Compiler 11.0
4	AMD Athlon 64 4000+	2.4	Windows XP (64-bit)	Microsoft Visual Studio 2005

Table 4.2: Results on Software Performance of CLEFIA (C language)

Platform	Key Length [bit]	Encryption [cycles/byte]	Decryption [cycles/byte]	Key Setup (Encryption) [cycles]	Key Setup (Decryption) [cycles]
1	128	17.7	18.0	442	517
	192	21.5	21.8	683	789
	256	25.2	25.6	734	859
2	128	18.7	19.7	304	385
	192	22.6	23.7	545	653
	256	26.4	28.0	590	722
3	128	17.6	18.5	325	446
	192	21.4	22.1	460	616
	256	25.0	25.8	493	683
4	128	19.0	19.1	386	452
	192	23.0	23.0	583	681
	256	26.8	27.0	627	720

Table 4.3: memory usage

code size [byte]	stack usage [byte]
17955	224

platform 4. We measured software processing speed of encryption/decryption and key setup for two type of implementations: the single-block (common) implementation and the two-block parallel implementation [45]. In the single-block implementation, 12.9 cycles/byte (1.48 Gbps on the processor) is achieved. The two-block parallel implementation, which is suitable for parallelizable modes such as CTR mode and CBC decryption, achieves 11.1 cycles/byte.

4.2 Hardware Implementations

This section describes performance evaluations of hardware implementations of CLEFIA. We implement two types of architecture for CLEFIA with 128-bit key and a type of architecture for CLEFIA with 192/256-bit key. We evaluate gate size and throughput of each implementation using an ASIC library.

For CLEFIA with 128-bit key, two types of architecture are implemented: the loop architecture and the compact architecture. The loop architecture is straightforward hardware implementation taking 1 cycle per round, where both F-functions F_0 and F_1 are implemented in parallel. In the compact architecture, the F-functions F_0 and F_1 are merged into an F_0/F_1 circuit in

Table 4.4: Results on Software Performance of CLEFIA (assembly language)

Type of Implementation	Key Length [bit]	Encryption [cycles/byte]	Decryption [cycles/byte]	Key Setup (Encryption) [cycles]	Key Setup (Decryption) [cycles]
single-block	128	12.9	13.3	217	229
	192	15.8	16.2	272	293
	256	18.3	18.4	328	357
two-block parallel encryption	128	11.1	11.1	217	229
	192	13.3	13.3	272	293
	256	15.6	15.6	328	357

order to reduce circuit area. The F_0/F_1 circuit is used as F_0 in one cycle and F_1 in another cycle, so that it takes 2 cycles per round. The latency of loop and compact architecture for encryption/decryption is 18 and 36, respectively.

For CLEFIA with 192/256-bit key, only the loop architecture is implemented. The latency of encryption/decryption for CLEFIA with 192-bit and 256-bit key is 22 and 26, respectively.

The environment of our hardware design and evaluation is as follows:

Language	Verilog-HDL
Design library	0.09 μm CMOS ASIC library
Simulator	VCS version 2005.06
Logic synthesis	Design Compiler version 2006.06

One gate is equivalent to a 2-way NAND and the speed is evaluated under the worst-case conditions.

Table 4.5 represents the evaluation results. For each implementation, two types of circuit are synthesized by specifying either area or speed optimization. We also show, for comparison, the best known results of hardware performance of AES and Camellia [61]. The synthesized circuit of CLEFIA with 128-bit key in loop architecture occupies only 5,979 gates with efficiency of 268.63 Kbps/gate. Moreover, less than 5 Kgates is achieved for CLEFIA with 128-bit key in compact architecture. Although we take into account the difference of ASIC libraries, these figures indicate that CLEFIA satisfies both low cost and high efficiency in hardware implementation compared to AES and Camellia.

4.3 Security against Side Channel Attacks

This section describes security against side channel attacks for CLEFIA.

Since CLEFIA has similar SP-type F-function with 8-bit S-boxes to AES, it was reported that similar cache-based timing attacks as AES [6, 54] was

Table 4.5: Results on Hardware Performance of CLEFIA

	Key Length	Enc/Dec (cycles)	Key Setup (cycles)	Area (gates)	Freq. (MHz)	Speed (Mbps)	Speed/Area (Kbps/gate)
CLEFIA (0.09μm)	128	18	12	5,979	225.83	1,605.94	268.63
				12,009	422.29	3,003.00	250.06
		36	24	4,950	201.28	715.69	144.59
				9,377	389.55	1,385.10	147.71
	192	22	20	8,536	206.56	1,201.85	140.81
				15,718	391.08	2,275.39	144.76
	256	26	20	8,482	206.56	1,016.95	119.89
				15,542	391.08	1,925.33	123.88
AES [61] (0.13μm)	128	11	N/A	12,454	145.35	1,691.35	135.81
				21,337	224.22	2,609.11	122.28
		54	N/A	5,398	131.24	311.09	57.63
				9,227	220.75	523.26	56.71
Camellia [61] (0.13μm)	128	22	N/A	10,993	166.94	971.29	88.36
				16,905	256.41	1,491.84	88.25
		44	N/A	6,511	111.98	325.76	50.03
				12,231	238.10	692.65	56.63

For each implementation, the above and below columns show the results of the synthesized circuits by area and speed optimization, respectively.

applicable to CLEFIA [57, 60]. On the flip side, it is likely that the same countermeasures, such as bit-slicing techniques, which are employed to protect AES from cache-based timing attacks will also applicable to CLEFIA. It was also reported that similar differential fault analysis as AES [55] was applicable to CLEFIA [16, 75, 76].

As for power analysis [39] and electromagnetic analysis [56], many logic-level countermeasures such as dual-rail logics [77] and masking logics [74] are applicable to implementations of CLEFIA.

Chapter 5

Evaluations by External Researchers

CLEFIA had been evaluated by external researchers in 2007. Evaluators are Prof. Alex Biryukov, Prof. Vincent Rijmen and Prof. Serge Vaudenay, and it is also evaluated by Prof. Lars R. Knudsen, and Prof. Bart Preneel of ABT. We received their evaluation reports between July and September 2007, and now the reports can be obtained from the CLEFIA's web site (http://www.sony.net/clefia) [13, 38, 58, 81]. These reports show evaluation results and additional comments on the security aspects of the design of CLEFIA based on input documents to the evaluators. The input documents are also available on the web site, they consists of specifications [68], design rationale [69] and self-evaluations [70] written by designers.

In the course of evaluation, we asked some questions including that "Do you think SONY's cipher algorithm is secure against the current cryptanalytic techniques?". All of the evaluators answered in the affirmative. Please see the evaluation reports for the details.

Bibliography

[1] R. Anderson, E. Biham, and L. R. Knudsen, "Serpent: A proposal for the advanced encryption standard." Primitive submitted to AES, 1998. Available at http://www.cs.technion.ac.il/~biham/Reports/Serpent/.

[2] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-bit block cipher suitable for multiple platforms." 2000. Available at http://info.isl.ntt.co.jp/crypt/camellia/dl/support.pdf.

[3] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-bit block cipher suitable for multiple platforms." in *Proceedings of Selected Areas in Cryptography – SAC '00* (D. R. Stinson and S. E. Tavares, eds.), no. 2012 in LNCS, pp. 41–54, Springer-Verlag, 2001.

[4] P. S. L. M. Barreto and V. Rijmen, "The Anubis block cipher." Primitive submitted to NESSIE, Sept. 2000. Available at http://www.cryptonessie.org/.

[5] P. S. L. M. Barreto and V. Rijmen, "The Whirlpool hashing function." Primitive submitted to NESSIE, Sept. 2000. Available at http://www.cryptonessie.org/.

[6] D. J. Bernstein, "Cache-timing Attacks on AES." 2005. Available at `http://cr.yp.to/antiforgery/cachetiming-20050414.pdf`.

[7] E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys." *Journal of Cryptology*, vol. 7, no. 4, pp. 229–246, 1994.

[8] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials." in *Proceedings of Eurocrypt'99* (J. Stern, ed.), no. 1592 in LNCS, pp. 12–23, Springer-Verlag, 1999.

[9] E. Biham, O. Dunkelman, and N. Keller, "Related-Key Impossible Differential Attacks on 8-Round AES-192." in *Topics in Cryptology – CT-RSA 2006, The Cryptographers' Track* (D. Pointcheval, ed.), no. 3860 in LNCS, pp. 21–33, Springer-Verlag, 2006.

[10] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems." *Journal of Cryptology*, vol. 4, pp. 3–72, 1991.

[11] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, 1993.

[12] A. Biryukov and D. Wagner, "Slide attack." in *Proceedings of Fast Software Encryption – FSE'99* (L. R. Knudsen, ed.), no. 1636 in LNCS, pp. 245–259, Springer-Verlag, 1999.

[13] A. Biryukov, "Review of CLEFIA." 2007. Available at `http://www.sony.net/clefia`.

[14] A. Biryukov and D. Khovratovich, "Related-Key Cryptanalysis of the Full AES-192 and AES-256." in *Advances in Cryptology – ASIACRYPT 2009* (M. Matsui, ed.), no. 5912 in LNCS, pp. 1–18, Springer-Verlag, 2009.

[15] A. Biryukov, D. Khovratovich, and I. Nikolić, "Distinguisher and Related-Key Attack on the Full AES-256." in *Advances in Cryptology – CRYPTO 2009* (S. Halevi, ed.), no. 5677 in LNCS, pp. 231–249, Springer-Verlag, 2009.

[16] H. Chen, W. Wu, and D. Feng, "Differential Fault Analysis on CLEFIA." in *Proceedings of ICICS 2007*, no. 4861 in LNCS, pp. 284–295, Springer-Verlag, 2007.

[17] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions." *Journal of Symbolic Computation*, vol. 9, no. 3, pp. 251–280, 1990.

[18] N. Courtois and J. Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations." in *Proceedings of ASIACRYPT'02* (Y. Zheng, ed.), no. 2501 in LNCS, pp. 267–287, Springer-Verlag, 2002.

[19] J. Daemen, L. R. Knudsen, and V. Rijmen, "The block cipher SQUARE." in *Proceedings of Fast Software Encryption – FSE'97* (E. Biham, ed.), no. 1267 in LNCS, pp. 149–165, Springer-Verlag, 1997.

[20] J. Daemen and V. Rijmen, "Statistics of Correlation and Differentials in Block Ciphers." in *IACR ePrint archive 2005/212*, 2005.

[21] J. Daemen and V. Rijmen, "Two-Round AES Differentials." in *IACR ePrint archive 2006/039*, 2006.

[22] J. Daemen and V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard (Information Security and Cryptography)*. Springer, 2002.

[23] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved Cryptanalysis of Rijndael." in *Proceedings of Fast Software Encryption – FSE 2000* (B. Schneier, ed.), vol. 1978 of *Lecture Notes in Computer Science*, pp. 213–230, Springer, 2001.

[24] H. Gilbert, H. Handshuh, A. Joux, and S. Vaudenay, "A stastical attack on RC6." in *Proceedings of Fast Software Encryption – FSE'00* (B. Schneier, ed.), no. 1978 in LNCS, pp. 64–74, Springer-Verlag, 2001.

[25] H. Gilbert and M. Minier, "A collision attack on 7 rounds of Rijndael." in *Proceedings of 3rd AES candidate conference*, pp. 230–241, 2001.

[26] S. Hong, S. Lee, J. Lim, J. Sung, D. H. Cheon, and I. Cho, "Provable security against differential and linear cryptanalysis for the SPN structure." in *Proceedings of Fast Software Encryption – FSE'00* (B. Schneier, ed.), no. 1978 in LNCS, pp. 273–283, Springer-Verlag, 2001.

[27] IPA and TAO, "CRYPTREC report 2002." 2003. Available at `http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/c02_2.pdf` (in Japanese) and `http://www2.nict.go.jp/y/y213/cryptrec_publicity/c02_report_english.pdf`(in English).

[28] T. Jakobsen and L. R. Knudsen, "The interpolation attack on block ciphers." in *Proceedings of Fast Software Encryption – FSE'97* (E. Biham, ed.), no. 1267 in LNCS, pp. 28–40, Springer-Verlag, 1997.

[29] P. Junod and S. Vaudenay, "FOX : A new family of block ciphers." in *Proceedings of Selected Areas in Cryptography – SAC'04* (H. Handschuh and M. A. Hasan, eds.), no. 3357 in LNCS, pp. 114–129, Springer-Verlag, 2004.

[30] M. Kanda, "Practical security evaluation against differential and linear cryptanalyses for Feistel ciphers with SPN round function." in *Proceedings of Selected Areas in Cryptography – SAC'00* (D. R. Stinson and S. E. Tavares, eds.), no. 2012 in LNCS, pp. 324–338, Springer-Verlag, 2001.

[31] M. Kanda, S. Moriai, K. Aoki, H. Ueda, Y. Takashima, K. Ohta, and T. Matsumoto, "E2 — A New 128-bit Block Cipher." *IEICE. Trans. Fundamentals, E83A*, no. 1, pp. 48–59, 2000.

[32] J. Kelsey, T. Kohno, and B. Schneier, "Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent." in *Proceedings of Fast Software Encryption – FSE'00* (B. Schneier, ed.), no. 1978 in LNCS, pp. 75–93, Springer-Verlag, 2001.

[33] J. Kelsey, B. Schneier, and D. Wagner, "Related-key cryptanalysis of 3-Way, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA." in *Proceedings of Information and Communication Security '97*, no. 1334 in LNCS, pp. 233–246, Springer-Verlag, 1997.

[34] J. Kim, S. Hong, J. Sung, C. Lee, and S. Lee, "Impossible differential cryptanalysis for block cipher structure." in *Proceedings of Indocrypt 2003* (T. Johansson and S. Maitra, eds.), no. 2904 in LNCS, pp. 82–96, Springer-Verlag, 2003.

[35] L. R. Knudsen and W. Meier, "Correlations in RC6 with a reduced number of rounds." in *Proceedings of Fast Software Encryption – FSE'00* (B. Schneier, ed.), no. 1978 in LNCS, pp. 94–108, Springer-Verlag, 2001.

[36] L. R. Knudsen and D. Wagner, "Integral cryptanalysis." in *Proceedings of Fast Software Encryption – FSE'02* (J. Daemen and V. Rijmen, eds.), no. 2365 in LNCS, pp. 112–127, Springer-Verlag, 2002.

[37] L. R. Knudsen, "Truncated and higher order differentials." in *Fast Software Encryption: Second International Workshop* (B. Preneel, ed.), no. 1008 in LNCS, pp. 196–211, Springer-Verlag, 1994.

[38] L. R. Knudsen and B. Preneel, "Evaluation of CLEFIA." 2007. Available at `http://www.sony.net/clefia`.

[39] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis." in *Proceedings of CRYPTO 99* (M. J. Wiener, ed.), no. 1666 in LNCS, pp. 388–397, Springer-Verlag, 1999.

[40] X. Lai, "Higher order derivatives and differential cryptanalysis." in *Proceedings of symposium on communication, coding and cryptography, in honor of J. L. Massey on the occation of his 60th birthday*, 1994.

[41] S. K. Langford and M. E. Hellman, "Differential-linear cryptanalysis." in *Proceedings of CRYPTO 94* (Y. Desmedt, ed.), no. 839 in LNCS, pp. 17–25, Springer-Verlag, 1994.

[42] C. Lim and K. Khoo, "An analysis of XSL applied to BES." in *Preproceedings of Fast Software Encryption – FSE'07* (A. Biryukov, ed.), pp. 253–265, 2007.

[43] M. Matsui, "Linear cryptanalysis of the data encryption standard." in *Proceedings of Eurocrypt'93* (T. Helleseth, ed.), no. 765 in LNCS, pp. 386–397, Springer-Verlag, 1994.

[44] M. Matsui, "On correlation between the order of s-boxes and the strength of DES." in *Proceedings of Eurocrypt'94* (A. D. Santis, ed.), no. 950 in LNCS, pp. 366–375, Springer-Verlag, 1995.

[45] M. Matsui, "How far can we go on the x64 processors?." in *Proceedings of Fast Software Encryption – FSE'06* (M. Robshaw, ed.), no. 4047 in LNCS, pp. 341–358, Springer-Verlag, 2006.

[46] M. Matsui and T. Tokita, "Cryptanalysis of reduced version of the block cipher E2." in *Proceedings of Fast Software Encryption – FSE'99* (L. R. Knudsen, ed.), no. 1636 in LNCS, pp. 71–80, Springer-Verlag, 1999.

[47] M. Matusi, "New block encryption algorithm MISTY." in *Proceedings of Fast Software Encryption – FSE'97* (E. Biham, ed.), no. 1267 in LNCS, pp. 54–68, Springer-Verlag, 1997.

[48] S. Moriai, M. Sugita, K. Aoki, and M. Kanda, "Security of E2 against truncated differential cryptanalysis." in *Proceedings of Selected Areas in Cryptography – SAC'99* (H. M. Heys and C. M. Adams, eds.), no. 1758 in LNCS, pp. 106–117, Springer-Verlag, 2000.

[49] S. Moriai and S. Vaudenay, "On the pseudorandomness of top-level schemes of block ciphers." in *Proceedings of ASIACRYPT'00* (T. Okamoto, ed.), no. 1976 in LNCS, pp. 289–302, Springer-Verlag, 2000.

[50] S. Murphy and M. Robshaw, "Essential algebraic structure within the AES." in *Proceedings of CRYPTO 2002* (M. Yung, ed.), no. 2442 in LNCS, pp. 1–16, Springer-Verlag, 2002.

[51] S. Murphy and M. Robshaw, "Comments on the security of the AES and the XSL technique." *Electronic Letters*, vol. 39, no. 1, pp. 36–38, 2003.

[52] K. Nyberg, "Generalized Feistel network." in *Proceedings of ASIACRYPT'96* (K. Kim and T. Matsumoto, eds.), no. 1163 in LNCS, pp. 91–104, Springer-Verlag, 1996.

[53] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura, "The block cipher Hierocrypt." in *Proceedings of Selected Areas in Cryptography – SAC'00* (D. R. Stinson and S. E. Tavares, eds.), no. 2012 in LNCS, pp. 72–88, Springer-Verlag, 2001.

[54] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: The case of aes." in *Proceedings of CT-RSA 2006* (D. Pointcheval, ed.), no. 2860 in LNCS, pp. 1–20, Springer-Verlag, 2006.

[55] G. Piret and J.-J. Quisquater, "A Differential Fault Attack Technique against SPN Structure, with Application to the AES and KHAZAD."

in *Proceedings of CHES 2003*, no. 2779 in LNCS, pp. 77–88, Springer-Verlag, 2003.

[56] J. J. Quisquater and D. Samyde, "ElectroMagnetic Analysis (EMA):Measures and Countermeasures for Smart Cards." in *Proceedings of E-smart 2001*, 2001.

[57] C. Rebeiro, D. Mukhopadhyay, J. Takahashi, and T. Fukunaga, "Cache Timing Attacks on Clefia." in *Proceedings of Indocrypt 2009*, no. 5922 in LNCS, pp. 104–118, Springer-Verlag, 2009.

[58] V. Rijmen, "Evaluation of CLEFIA." 2007. Available at `http://www.sony.net/clefia`.

[59] R. L. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L. Yin, "The RC6 block cipher." Primitive submitted to AES, 1998. Available at http://www.rsasecurity.com/.

[60] C. Robeiro and D. Mukhopadhyay, "Differnce Cache Trace Attack against CLEFIA." in *IACR ePrint archive 2010/012*, 2010.

[61] A. Satoh and S. Morioka, "Hardware-focused performance comparison for the standard block ciphers AES,Camellia, and Triple-DES." in *Proceedings of ISC 2003* (C. Boyd and W. Mao, eds.), no. 2851 in LNCS, pp. 252–266, Springer-Verlag, 2003.

[62] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-bit block cipher." Primitive submitted to AES, 1998. Available at http://www.schneier.com/.

[63] T. Shirai and B. Preneel, "On Feistel ciphers using optimal diffusion mappings across multiple rounds." in *Proceedings of ASIACRYPT'04* (P. J. Lee, ed.), no. 3329 in LNCS, pp. 1–15, Springer-Verlag, 2004.

[64] T. Shirai and K. Shibutani, "Improving immunity of Feistel ciphers against differential cryptanalysis by using multiple MDS matrices." in *Proceedings of Fast Software Encryption – FSE'04* (B. Roy and W. Meier, eds.), no. 3017 in LNCS, pp. 260–278, Springer-Verlag, 2004.

[65] T. Shirai and K. Shibutani, "On Feistel structures using a diffusion switching mechanism." in *Proceedings of Fast Software Encryption – FSE'06* (M. Robshaw, ed.), no. 4047 in LNCS, pp. 41–56, Springer-Verlag, 2006.

[66] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA." in *Proceedings of Fast Software Encryption – FSE'07* (A. Biryukov, ed.), no. 4593 in LNCS, pp. 181–195, Springer-Verlag, 2007.

[67] T. Shirai and K. Araki, "On generalized Feistel structures using the diffusion switching mechanism." *IEICE. Trans. Fundamentals*, vol. E91A, no. 8, pp. 2120–2129, 2008.

[68] Sony Corporation, "The 128-bit Blockcipher CLEFIA : Algorithm Specification, Rev 1.0." 2007. Available at http://www.sony.net/clefia.

[69] Sony Corporation, "The 128-bit Blockcipher CLEFIA : Design Rationale, Rev 1.0." 2007. Available at http://www.sony.net/clefia.

[70] Sony Corporation, "The 128-bit Blockcipher CLEFIA : Security and Performance Evaluations." 2007. Available at http://www.sony.net/clefia.

[71] Sony Corporation, "The 128-bit Blockcipher CLEFIA: Specification, Version 1.0 (Japanese/English)." 2010. Submission to CRYPTREC (Application for Cryptographic Techniques towards the Revision of the e-Government Recommended Ciphers List).

[72] M. Sugita, K. Kobara, and H. Imai, "Security of reduced version of the block cipher Camellia against truncated and impossible differential cryptanalysis." in *Proceedings of ASIACRYPT'01* (C. Boyd, ed.), no. 2248 in LNCS, pp. 193–207, Springer-Verlag, 2001.

[73] B. Sun, R. Li, M. Wang, P. Li, and C. Li, "Impossible Differential Cryptanalysis of CLEFIA." in *IACR ePrint archive 2008/151*, 2008.

[74] D. Suzuki, M. Saeki, and T. Ichikawa, "Random Switching Logic: A Countermeasure against DPA based on Transition Probability." in *IACR ePrint archive 2004/236*, 2004.

[75] J. Takahashi and T. Fukunaga, "Differential Fault Analysis on CLEFIA." in *Proceedings of Symposium on Cryptography and Information Security 2009 –SCIS 2009 2A3-4, (in Japanese)*, 2009.

[76] J. Takahashi and T. Fukunaga, "Improved Differential Fault Analysis on CLEFIA." in *Proceedings of FDTC 2008*, LNCS, pp. 25–34, IEEE Computer Society, 2008.

[77] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for A Secure DPA Resistant ASIC or FPGA Implementation." in *Proceedings of DATE 2004*, pp. 246–251, 2004.

[78] E. Tsujihara, M. Shigeri, T. Suzaki, T. Kawabata, and Y. Tsunoo, "New Impossible Differentials of CLEFIA." in *IEICE Technical Report – ISEC2008-3 (in Japanese)*, 2008.

[79] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzaki, and H. Kubo, "Impossible Differential Cryptanalysis of CLEFIA." in *Proceedings of Fast Software Encryption – FSE 2008* (K. Nyberg, ed.), no. 5086 in LNCS, pp. 398–411, Springer-Verlag, 2008.

[80] S. Vaudenay, "An experimental on DES statistical cryptanalysis." in *3rd ACM conference on computer and communications security*, pp. 139–147, ACM Press, 1996.

[81] S. Vaudenay, "Evaluation report on CLEFIA." 2007. Available at `http://www.sony.net/clefia`.

[82] D. Wagner, "The boomerang attack." in *Proceedings of Fast Software Encryption – FSE'99* (L. R. Knudsen, ed.), no. 1636 in LNCS, pp. 156–170, Springer-Verlag, 1999.

[83] W. Wang and X. Wang, "Improved Impossible Differential Cryptanalysis of CLEFIA." in *IACR ePrint archive 2007/466*, 2007.

[84] H. Wu, "Related-Cipher Attacks." in *Proceedings of Information and Communications Security – ICICS 2002* (R. H. Deng, S. Qing, F. Bao, and J. Zhou, eds.), no. 2513 in LNCS, pp. 447–455, Springer-Verlag, 2002.

[85] W. Zhang. Private Communication, 2008.

[86] W. Zhang and J. Han, "Impossible Differential Analysis of Reduced Round CLEFIA." in *Pre-Proceedings of Inscrypt 2008* (M. Yung, P. Liu, and D. Lin, eds.), pp. 143–154, 2008.

[87] Y. Zheng, T. Matsumoto, and H. Imai, "On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses." in *Proceedings of CRYPTO 89* (G. Brassard, ed.), no. 435 in LNCS, pp. 461–480, Springer-Verlag, 1989.

Copyright