The 128-bit Blockcipher CLEFIA

Design Rationale

Revision 1.0

June 1, 2007

Sony Corporation

NOTICE

THIS DOCUMENT IS PROVIDED "AS IS," WITH NO WARRANTIES WHATSOVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Acknowledgments

Tetsu Iwata of Nagoya University has contributed to the development of CLEFIA.

Contact

E-MAIL :
clefia-q@jp.sony.com

Postal Address :
Information Technologies Laboratories
Sony Corporation
1-7-1 Konan, Minato-ku, Tokyo 108-0075 Japan

Revision History

June 1, 2007	1.0 revision

Contents

1 Introduction

This document describes design rationale of blockcipher CLEFIA [28, 29].

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 [5, 17]
2. Compact F-functions realized by employing a 4-branch generalized Feistel structure
3. Enhanced immunity against certain class of attacks by using a two S-boxes system
4. Using only lightweight components for efficient software and hardware implementations
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 this document.

2 Data Processing Part

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

2.1 Fundamental Structure

CLEFIA employs a generalized Feistel structure which is an extension of the traditional Feistel structure. Generalized Feistel structures have 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 transformations" defined in Zheng et al.'s paper [30]. Figure 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.

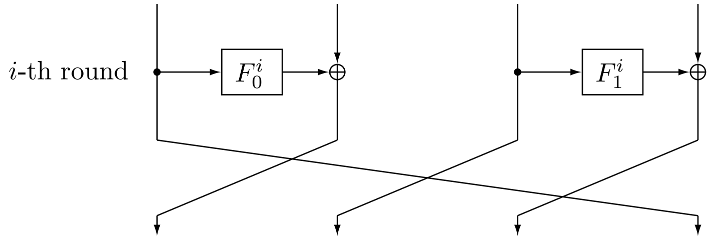

Figure 1: One round of the 4-branch type-2 generalized Feistel structure

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 benefited from the first two advantages.

Pioneer research on the generalized Feistel structures is done by Zheng, Matsumoto and Imai [30], followed by the work by Nyberg [21]. The blockcipher RC6 also employs the type-2 structure with slight modification, and it achieves good efficiency performance partially due to this structure [23]. In the security aspects, Moriai and Vaudenay treated pseudo-random property of the generalized Feistel structures [19]. Furthermore, Knudsen and Wagner presented with regard to integral cryptanalysis [15], and Kim et al. discussed with regard to impossible cryptanalysis [14].

2.2 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 [29]. This type of F-function is used in many blockcipher designs including Camellia [2] and Twofish [24]. 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 [10].

2.3 Key Whitenings

CLEFIA employs key whitenings at the beginning and the end of the data processing part [29]. 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 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.

2.4 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 [25–27]. 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.

Definition 1 *Let $x \in \{0,1\}^{pn}$ be represented as $x = [x_0 x_1 \dots x_{p-1}]$ where $x_i \in \{0,1\}^n$, then the bundle weight $w_n(x)$ is defined as*

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

Definition 2 *Let $P : \{0,1\}^{pn} \rightarrow \{0,1\}^{qn}$. The branch number of P is defined as*

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

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 GF(2^8) hold following conditions.

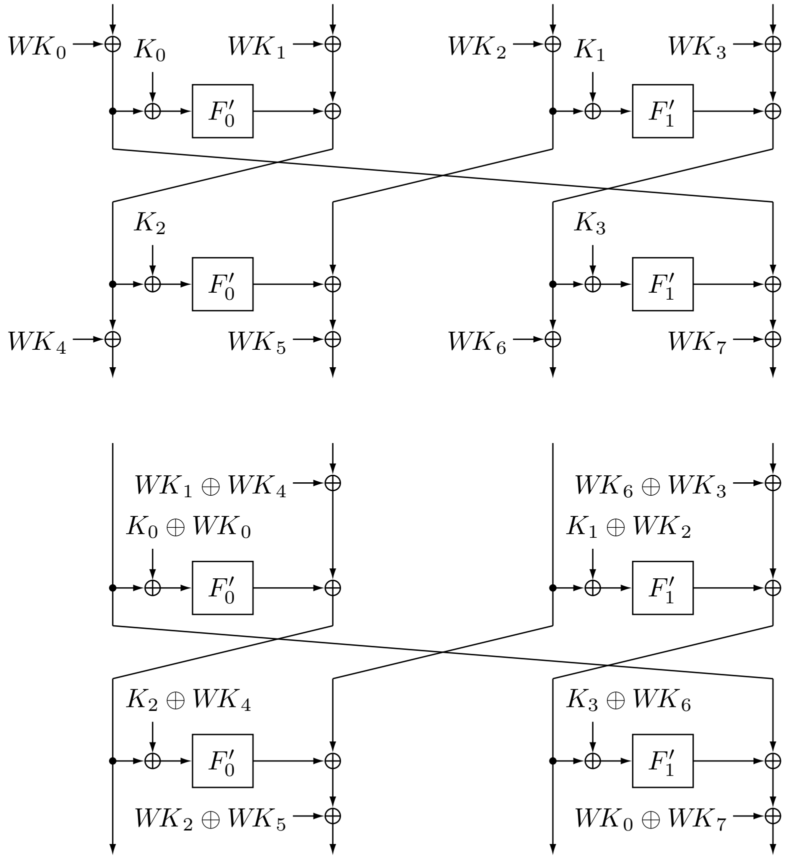

Figure 2: Equivalent Structures

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

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.

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

When M_0 and M_1 are put in the F-functions of CLEFIA as Figure 3, it is expected to hold good diffusion property by the synergy of these two matrices in neighboring F-functions [29]. 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 effects are described in Shirai et al.'s papers [25–27] .

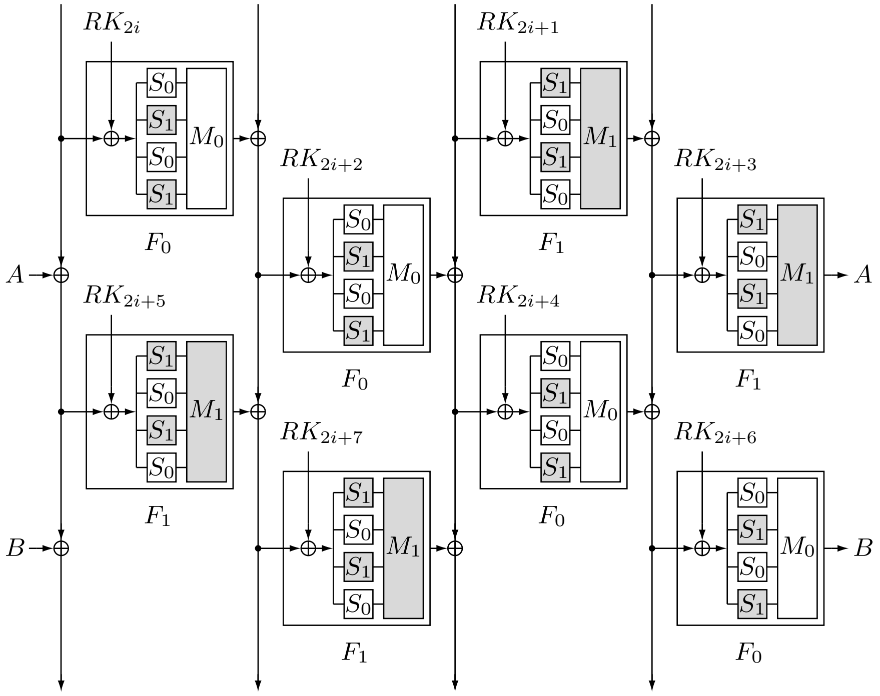

Figure 3: Allocation of M_0, M_1, S_0 and S_1

Table 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 branch number matrices M_0 and M_1. Similarly, 'DSM(L)' means the guaranteed number of linear active S-boxes for 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'.

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. As a result we are able to reduce

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

Table 1: Guaranteed Numbers of Active S-boxes

the number of rounds keeping immunity against differential the attack and the linear attack.

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 [2], there are 18, 21 and 22 differential active S-boxes for 9, 10 and 11 rounds, respectively, also 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-round [24]. 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.

2.4.1 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 matrices. An Hadamard-type matrix is used in blockcipher Anubis, and its elements are defined as $a_{i,j} = A_{i \oplus j}$ for a certain set of A_k [3]. 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 small. As a result, two matrices M_0 and M_1 for CLEFIA are decided as Hadamard-type matrices.

2.5 S-boxes

CLEFIA employs plural types of S-boxes as in Serpent and Camellia. However, we believe that the reason for choosing CLEFIA's plural S-boxes is based on novel criteria and expected effects [1, 2]. Basically, we respect the following properties for choosing the S-boxes.

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 [7], and
- To enhance the immunity against algebraic attacks including the XSL attack [6].

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 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_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

Table 2: Security Parameters of S_0 and S_1

2.5.1 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 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 shows the several security parameters of S_0.

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

The second S-box S_1 is designed based on 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. Table 2 shows the several security parameters of S_1.

2.5.3 Enhancing Immunity against the Byte-oriented Saturation Attack

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` if $\forall i, j \;\; X_i = X_j$,
- `All` if $\forall i, j \;\; i \neq j \Leftrightarrow \; X_i \neq X_j$,
- `Balance` if $\bigoplus_{i=0}^{255} X_i = 0$, but not `Const` nor `All`, and
- `Unknown` unknown .

Then consider a toy example that 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 4).

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.

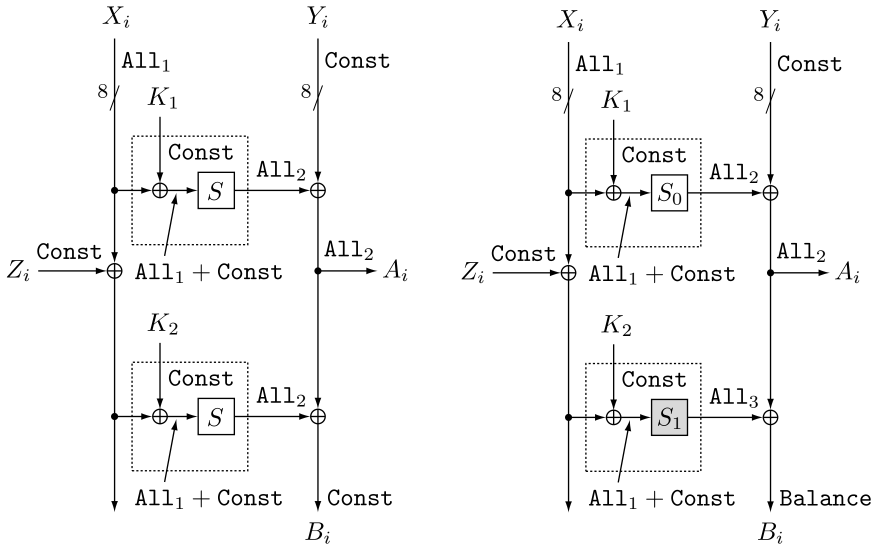

Figure 4: An Example of Saturation characteristic

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 Alls 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,

$$\text{For any } c_1, c_2,\ \exists x \quad S_0[x] \neq S_1[x \oplus c_1] \oplus c_2.$$

We can avoid the above cancellation of saturation characteristics. Putting S_0 and S_1 as the right of Figure 4, B_i won't be Const due to the S-box property, because two Alls XORed will be never canceled due to the above condition.

Although this is a toy example, and actual cipher employs more complicated matrices, we consider 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.

2.5.4 Enhancing Immunity against Algebraic Attacks

Previous works on algebraic attacks [6, 11, 16] 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, 13], mixed use of S-boxes with different sizes [18], or constructing from random 4-bit S-boxes [3, 12], 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 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) [20]. Moreover, Daemen and Rijmen presented a new result on behavior of inverse function based S-boxes such that there are plateau trails in it [9]. That’s why we don’t want to use an inverse function based 8-bit S-box only.

On the other hand 4-bit S-box based 8-bit S-box 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 4-bit S-box based 8-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 [6]. That’s why we don’t want to use a 4-bit S-box based 8-bit S-box only.

Also we saw trends of choice of S-boxes in literatures, in a certain period of time many blockcipher designers used inverse function based 8-bit S-boxes as in AES/Rijndael, Camellia, Misty, Hierocrypt-3 and so on, then recently 4-bit S-box based 8-bit S-boxes are tend to be used as in Whirlpool[1], Anubis and FOX [2–4,10,12,18,22]. Our approach is different from the above trends.

In CLEFIA half of S-boxes are inverse function based 8-bit S-boxes and the others are 4-bit S-box based 8-bit S-boxes. This design makes the cipher

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

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. In CLEFIA, S_0 is chosen as a 4-bit S-box based 8-bit S-box and S_1 is chosen as an inverse based S-box.

2.5.5 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, 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, 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 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.

3 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 $K \oplus L$.

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.

Table 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

3.1 Employing $\boldsymbol{GFN}_{4,12}$ 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 lowest DP_{max} is $2^{-4.67}$ and lowest 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 [8].

3.2 Employing $\boldsymbol{GFN}_{8,10}$ 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. We consider that $GFN_{8,10}$ has a good difference propagation property, it means that controlling the output difference of $GFN_{8,10}$ is very difficult even though attacker can control the input difference of it. 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 3, we know that there are at least 29 differential active S-boxes in $GFN_{8,10}$. Since the highest DP_{max} is $2^{-4.67}$ due to S_0, we can assure that there are no differential characteristics with probability more than 2^{-128} because $2^{29\times(-4.67)} = 2^{-135.43}$. Although this is only saying about characteristics but not about differential, CLE-

FIA 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 [8].

3.3 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 way of $GFN_{4,12}$. 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 way of $GFN_{8,10}$. The same effect will be expected also in 192-bit and 256-bit key cases.

3.4 *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 rotation shift operation, the *DoubleSwap* function offers efficient hardware implementation.

3.5 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 efficient hardware implementation can be achieved by sharing components of all key scheduling algorithms.

3.6 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 [29]. 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.

References

[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." 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.

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

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

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

[6] 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.

[7] 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.

[8] J. Daemen and V. Rijmen, "Statistics of correlation and differentials in block ciphers." in *IACR ePrint archive 2005/212*, 2005.

[9] J. Daemen and V. Rijmen, "Two-round AES differentials." in *IACR ePrint archive 2006/039*, 2006.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] S. Murphy and M. Robshaw, "Essential algebraic structure within the AES." in *Proceedings of Crypto'02* (M. Yung, ed.), no. 2442 in LNCS, pp. 1–16, Springer-Verlag, 2002.

[21] 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.

[22] 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.

[23] 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/.

[24] 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/.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] "The 128-bit blockcipher CLEFIA : Algorithm specification." On-line document, 2007. Sony Corporation.

[30] 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.