Systems and Computers in Japan, Vol. 19, No. 7, 1988
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. 70-D, No. 7, July 1987, pp. 1413-1423

# FEAL—Fast Data Encipherment Algorithm

Akihiro Shimizu and Shoji Miyaguchi, Members

NTT Electrical Communications Laboratories, Yokosuka, Japan 238

SUMMARY

This paper discusses the design of the fast data encipherment algorithm (FEAL). FEAL is a conventional encipherment algorithm using the same key for enciphering and deciphering. It is a block cipher algorithm which produces 64 bit ciphertext from 64 plaintext, using a 64-bit key. The main feature of the cipher processing in FEAL is that it is based on 8-bit data manipulations such as addition with modulo 256, and data cycle and data transfer commands for 1-byte. The program performance of FEAL is improved greatly compared with that of DES. The security of FEAL is based on the characteristic value representing the trace of the plaintext and ciphering key remaining in the ciphertext, as well as on the algorithm structure.

## 1. Introduction

In the data communication system, a cipher is a useful means of ensuring the security of data on the communication channel or in the file. In the past, DES [1] has been used as the cipher algorithm for this purpose. In this system, however, DES hardware must be installed, which increases the cost. There is also a problem in its use due to the inadequate software performance. The authors, therefore, have developed a cipher algorithm with high security and performance in both hardware and software implementation, called FEAL.

FEAL is a block cipher algorithm, which produces 64 bit ciphertext from 64 bit plaintext using 64-bit cipher key. The security of FEAL is guaranteed by characteristic value representing the trace of the plaintext and the ciphering key remaining in the ciphertext.

The design principles of FEAL are as follows.

(a) The replacement of DES is emphasized with a 64-bit block cipher with 64-bit plain/cipher text. The key is defined as 64 bits.

(b) To provide high-speed processing using the terminal or the personal computer software, the system is based on 8-bit data manipulation commands.

(c) The cipher algorithm is based on two strengths: cipher strength, represented by the lack of relation between output (ciphertext) changes and the input (key and plaintext changes; and strength based on the algorithm structure. The latter is considered individually in the design procedure.

## 2. Expression of Cipher Strength

This section describes some important notions for evaluating the strength of the cipher and authentication algorithm adopted in FEAL.

### 2.1 Preliminaries

(1) Block cipher algorithm: The input to the cipher algorithm is the key block and the plaintext block, and the output is the ciphertext block. The number of bits in the ciphertext block is denoted by $n$.

(2) Ideal binomial distribution: The binomial distribution $B(n, 1/2)$ is called the ideal binomial distribution. The probability density function $f_B(r)$ of the ideal binomial distribution is represented as follows:

$$f_B(r)=2^{-n} \cdot nCr \quad (r=0,\ 1,\ \cdots,\ n)$$

(1) Stochastic variable $L_k$: The cipher process is regarded as a blackbox. For an arbitrary plaintext block, the key block is varied. Then the change in the ciphertext $C$ (from $C_1$ to $C_2$) is regarded as



ISSN0882-1666/88/0007-0020$7.50/0

stochastic, and is represented by a stochastic variable $L_k$. In other words,

$$L_k = H(C_1, C_2)$$

where $H(C_1, C_2)$ is the Hamming distance between $C_1$ and $C_2$.

(2) Independence of ciphertext changes and key changes: If the probability of each bit inversion in the ciphertext is always 0.5 when the key block is changed, independent of changes in the key block (i.e., if $L_k$ follows the ideal binomial distribution), then changes in the ciphertext block are independent of key block changes. In this case, the cipher algorithm provides the greatest protection against attempts to break the cipher by estimating the key.

(3) Key index $M_k$: The closeness of the distribution of $L_k$ to the ideal binomial distribution is represented by the key index $M_k$; $M_k$ takes a value in the range $0 \leqq M_k \leqq 1$, and increases when the distribution of $L_k$ approaches the ideal binomial distribution; $M_k$ represents the strength of the cipher scheme protection against attempts to estimate the key.

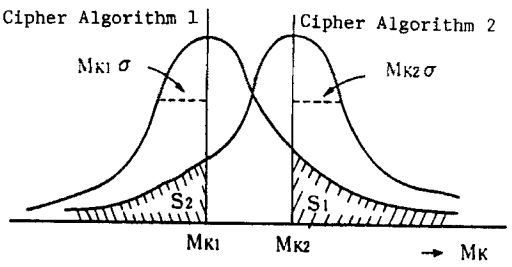
2.3 Plaintext variation

(1) Stochastic variable $L_p$: The cipher is regarded as a blackbox. For an arbitrarily selected key block, the plaintext block is varied. Then the change in the ciphertext block $C$ (from $C_1$ to $C_2$) is regarded as stochastic, and is represented by a stochastic variable $L_p$. In other words,

$$L_p = H(C_1, C_2)$$

(2) Independence of ciphertext changes and plaintext changes: If the probability of each bit inversion of the ciphertext is always 0.5 when the plaintext changes independent of changes in the plaintext (i.e., if the distribution of $L_p$ is the ideal binomial distribution), then changes in the ciphertext are independent of plaintext changes. In this case, the cipher algorithm provides the greatest protection against attempts to break the cipher by estimating the plaintext.

(3) Plaintext index $M_p$: The closeness of the distribution of $L_p$ to the ideal binomial distribution is represented by $M_p$ (called the plaintext index). $M_p$ is in the range $0 \leqq M_p \leqq 1$, and increases when the distribution of $L_p$ approaches the ideal binomial distribution; $M_p$ represents the strength of the cipher algorithm protection against attempts to estimate the plaintext.




note : MK distributions are approximated as normal distribution.

Fig. 1. Risk ratio when comparing Key indices.

2.4 Additional comments on the index

(1) Standard deviation of $M_k$ and $M_p$: Considering the distributions of $L_k$ and $L_p$ when the key block or the plaintext block is changed by the Hamming distance $\nu$, the degree of approximation $\theta_\nu$ is defined for the distributions of $L_k$ and $L_p$ in regard to the ideal binomial distribution (where $0 \leqq \theta_\nu \leqq 1$; see Appendix 1). Using a large number of key block and plaintext samples, $\nu$ is varied, and $M_k$ and $M_p$ are defined as the mean of $\theta_\nu$. The standard deviations $M_{k\sigma}$ and $M_{p\sigma}$ are also defined. These indices depend on the number of samples.

(2) Risk ratio: If $M_{k1} < M_{k2}$ applies to the two algorithms, one can decide that Cipher Algorithm 2 is better. However, $M_k$ is a statistical value and may contain an estimation error. The probability of a mistaken decision due to estimation error is called the risk ratio $R_k$ in the comparison of key index; $R_k$ is defined as the larger area of the two shaded areas $S_1$ and $S_2$ shown in Fig. 1; $R_k$ depends on the number of samples. The risk ratio $R_p$ for the plaintext is defined similarly.

## 3. Involution

This section describes the involution, which is the fundamental ciphering manipulation used in the design of FEAL [3].

Definition 1. $Z_{2N}$: The set of all bit sequences of length $2N$ composed of 0 and 1, is denoted by $Z_{2N}$.

Definition 2. Involution: If a mapping $\pi$ from inside of one $Z_{2N}$ to inside another $Z_{2N}$ satisfies the following condition, $\pi$ is called an involution.

$\pi(\pi(x)) = x$, where $x$ is a point inside of $Z_{2N}$.

Theorem 1. If $\pi$ is an involution, $\pi$ is a one-to-one mapping.

Proof. Let $x_1 \neq x_2$. Then either of the following relations applies:

$$\pi(x_1)=\pi(x_2)$$
$$\pi(x_1)\neq\pi(x_2)$$

Assume that the equality holds. Applying the mapping $h$ on both sides,

$$\pi(\pi(x_1))=\pi(\pi(x_2))$$
$$x_1=x_2$$

This contradicts the assumption that $x_1$ and $x_2$ are different. Consequently, the following properties are shown:

$$\text{if } x_1=x_2,\ \pi(x_1)=\pi(x_2)$$
$$\text{if } x_1\neq x_2,\ \pi(x_1)\neq\pi(x_2)$$

A similar proof can be applied to the inverse of $\pi$.

Corollary of Theorem 1. Product of involutions: $\pi_2(\pi_1(x))$ is defined as the product of involutions, and is denoted by $\pi_2 \times \pi_1(x)$. Let $\phi = \pi_m \times \cdots \times \pi_2 \times \pi_1$ (where $m$ is an integer); then the set of $\phi$ forms a group. Then,

$$\pi^{-1}=\pi$$
$$\phi^{-1}=\pi_1\times\pi_2\times\cdots\times\pi_m .$$

(3) Examples of involution: Three involution examples are shown in Fig. 2. It is obvious from $x \oplus x = 0$ and other relations that these examples satisfy the

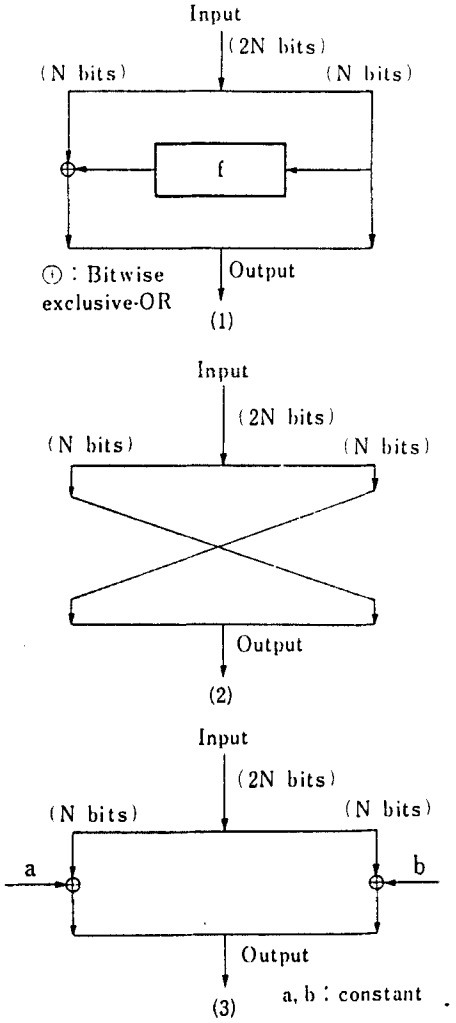



Fig. 2. Involution.

definition of the involution. In the foregoing, $\oplus$ represents the bitwise exclusive-or. In this paper, $N$ is set at 32 for $Z_{2N}$.

## 4. Basic Organization of FEAL

### 4.1 Whole organization

FEAL is composed of a data randomization part and a key scheduling part (Fig. 3). Involution units (GB) are placed at the input and the output of the data randomization part. Four stages of data randomization (DRE) are placed inside the data randomization part. The notations ($K_0$, $K_1$, $\cdots$) in Fig. 3 ① - ③ indicate concatenation of $K_0$, $K_1$, $\cdots$.

### 4.2 Organization of the data randomization part

(1) Number of cipher stages: Excluding the input and the output involution units,

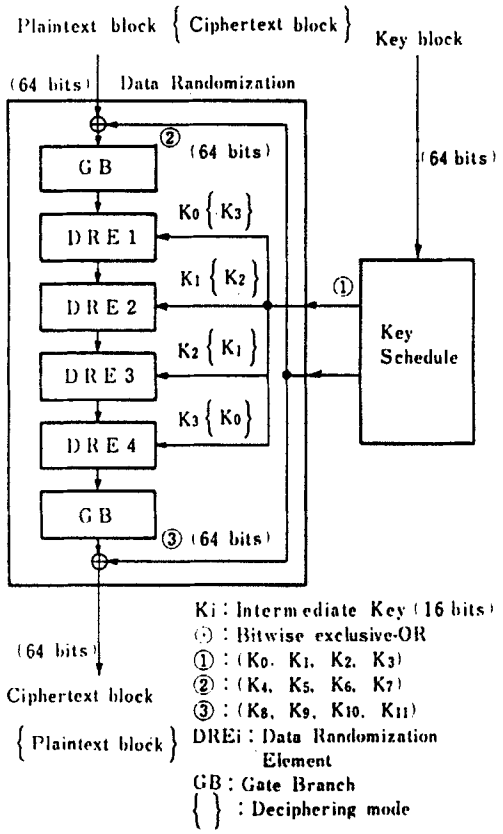



Fig. 3. FEAL organization.

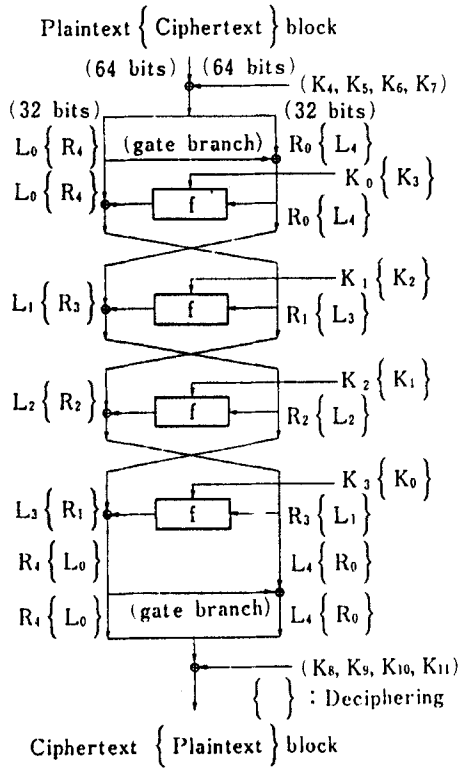



Fig. 4. Data randomization part of FEAL.

there are 4 stages of cipher processing, considerably less than the 16 stages required by DES. In FEAL, when the number of cipher stages exceeds 3, the plaintext index and the key index exceed those of DES and almost saturate. Consequently, the number of stages is set at 4, with one excess stage.

(2) Unknown 128 bit information: 128 bit unknown information is inserted at the input and the output (② and ③ in Fig. 2). Through this scheme, the amount of analysis of the internal structure of the data randomization part required to break the cipher will be increased drastically; ② and ③ in Fig. 3 represent the insertion method in the cipher procedure. For deciphering, procedures ② and ③ are reversed.

(3) Gatebranch: The gatebranch (GB in Fig. 3) is introduced to ensure the homogeneity of the operational conditions of the first cipher stage (DRE 1) and the further randomization of the key processing (Sect. 5.5).

(4) IP (initial inversion) and $IP^{-1}$ of DES does not improve $M_k$ or $M_p$ at the input/output of the data randomization part (Theorem 1 in [2]). Consequently, IP or $IP^{-1}$ is not adopted.

### 4.3 Construction of the key scheduling part

(1) The key information (intermediate key) produced as the output from the key schedule part is composed of 16 × 12 bits = 192 bits. This is larger than 64 bits of key information, and should make the cipher more resistant to attack.

(2) The key is passed to the data randomization part after a sufficient randomization in the key scheduling part. By this randomization, the key index $M_k$ is made sufficiently large.

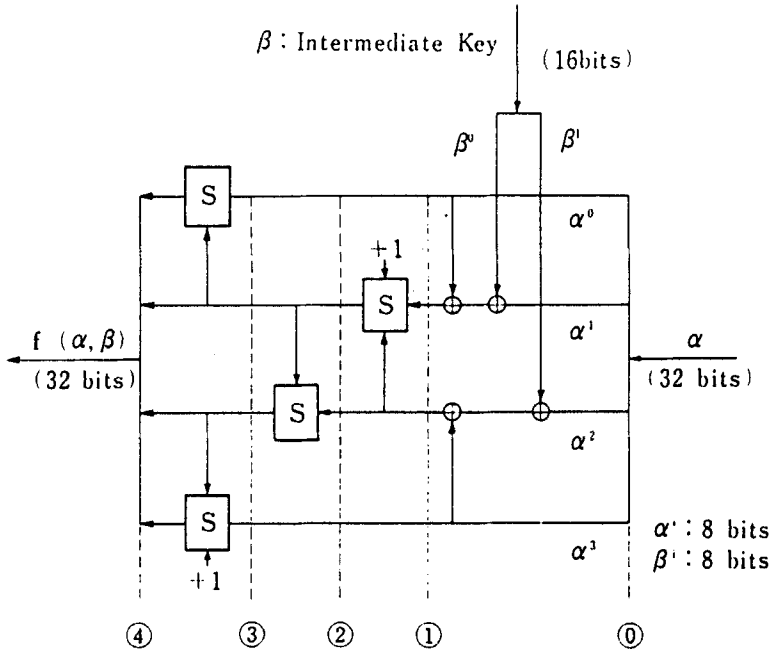
## 5. Data Randomization

### 5.1 Involution representation

The "FEAL" function of the data randomization part (Fig. 4) can be represented as follows using involution:

$$\mathrm{FEAL} = K_{891011} \times G \times \Pi_3 \times \theta \times \Pi_2 \times \theta \times \Pi_1 \times \theta \times \Pi_0 \times G \times K_{4567}$$

where




note : ⓪ to ④ show flow numbers (see Fig. 6).

Fig. 5. Function $f$ in data randomization part.

$K_{4567}$; input by exclusive-or of ($K_4$, $K_5$, $K_6$, $K_7$) (② of Fig. 3).

$K_{891011}$: input by exclusive-or of ($K_8$, $K_9$, $K_{10}$, $K_{11}$) (③ of Fig. 3).

$G$: gatebranch

$\Pi_i$: (1) of Fig. 2, function is $f(\alpha, \beta)$.

$\theta$: (2) of Fig. 2, $N = 32$.

$K_{4567}$ corresponds to (3) of Fig. 2, where the constants ($a$ and $b$) are set as $K_4$, $K_5$, $K_6$ and $K_7$. Similarly, for $K_{891011}$; corresponds to (1) of Fig. 2, where $f$ is deleted and the arrow is reversed (from left to right). Function $f(\alpha, \beta)$ is discussed in Sect. 5.2.

The inverse $\mathrm{FEAL}^{-1}$ of FEAL is represented as follows by the corollary of Theorem 1:

$$\mathrm{FEAL}^{-1} = K_{4567} \times G \times \Pi_0 \times \theta \times \Pi_1 \times \theta \times \Pi_2 \times \theta \times \Pi_3 \times G \times K_{891011}$$

Consequently, the intermediate key in decipherment is obtained by reversing the order of $K_0$, $K_1$, $K_2$ and $K_3$ from the cipher, and by exchanging the positions of $K_{4567}$ and $K_{891011}$.

5.2 Function $f$

Function $f$ in the data randomization part has an interchanging network structure, where each byte in the output changes, if any byte of the input changes. To realize this, the following manipulations were employed (Fig. 5).

(a) Concatenation of 1 byte of data by exclusive-or.

(b) Concatenation of 1 byte of data and data randomization by function $S$. Function $S$ employs the following operations:

① Addition modulo 256.

② 2-bit left-cyclic shift.

(c) Mixing of the result with $S$ with and without input +1.

The nonlinear property of function $f$ is realized by combining the exclusive-or and function $S$. These operations can be realized in principle by inter-register operations in the microprocessor which are useful for improving the speed of FEAL program processing and reducing the program size (no

| N | $\alpha$ | TEXT (Hexa) | | | | $\delta$ |
|---|---|---|---|---|---|---|
| 0 | $\alpha 1$: | 0F | F0 | 0F | F0 | 0 |
| | $\alpha 2$: | 0F | F0 | 0F | F0 | |
| | $\alpha 3$: | 0F | F0 | 0F | D0 | 1 |
| 1 | $\alpha 1$: | 0F | 4C | 33 | F0 | 1 |
| | $\alpha 2$: | 0F | CC | 33 | F0 | |
| | $\alpha 3$: | 0F | CC | 13 | D0 | 2 |
| 2 | $\alpha 1$: | 0F | 02 | 33 | F0 | 1 |
| | $\alpha 2$: | 0F | 00 | 33 | F0 | |
| | $\alpha 3$: | 0F | 83 | 13 | D0 | 5 |
| 3 | $\alpha 1$: | 0F | 02 | D4 | F0 | 3 |
| | $\alpha 2$: | 0F | 00 | CC | F0 | |
| | $\alpha 3$: | 0F | 83 | 5A | D0 | 8 |
| 4 | $\alpha 1$: | 44 | 02 | D4 | 17 | 11 |
| | $\alpha 2$: | 3C | 00 | CC | F6 | |
| | $\alpha 3$: | 4A | 83 | 5A | AC | 16 |
| | $\beta 1$ : | B3 | CC | | | 1 |
| | $\beta 2$ : | 33 | CC | | | |
| | $\beta 3$ : | 33 | CC | | | 0 |

note : Ns (Flow numbers) are shown in Fig.5.
$\delta$ : Hamming distance to $\alpha 2$

Fig. 6. Example of α variation of $f$ (α, β).

memory is required); (c) is used as the randomization factor. When, for example, inputs α and β of $f$ are both 0, the output of $f$ cannot be 0. This makes the analysis of $f$ more difficult in attempts to break the cipher.

The structure of function $f$ was determined by the cut-and-try method so that the following four conditions are satisfied:

(1) Data manipulations are performed by 8-bit commands.

(2) The plaintext index $M_p$ of FEAL is large.

(3) The number of program steps should be small.

(4) The circuit should easily be realized by LSI.

The tradeoff between (2) and (3) is determined to balance the requirement for security (7.1) against that of program performance (7.2).

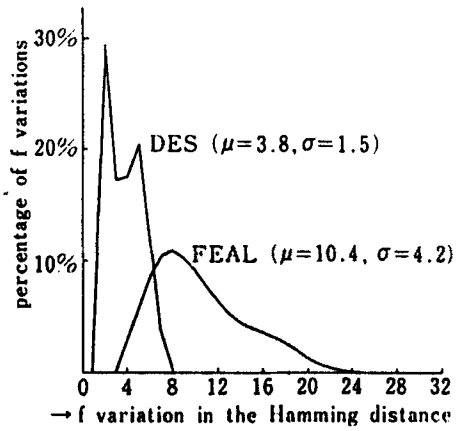



Fig. 7. $f$ Variation with 1 bit variation of α.

### 5.3 Function $S$

Function $S$ used in function $f$ is determined as follows. Note that $S$ is also used in the key scheduling to be discussed later.

Definition 3. Function $S$:

$$S(x_1,\ x_2,\ \delta)=\mathrm{Rot}\,2\,(w)$$

where

$$w=(x_1+x_2+\delta) \mod 256$$

and δ = 0 or 1 (constant)

In the foregoing, $x1$ and $x2$ are 1-byte blocks; δ is a constant, which is either 0 or 1; $W$ is a 1-byte block obtained by regarding $x1$ and $x2$ as binary nonnegative integers and calculating ($x1$ + $x2$ + δ) mod 256. Rot 2 ($W$) is a 1-byte block obtained by the left 2-bit cyclic shift (left 2-bit rotation) of 1-byte block $W$.

Example: Rot 2 (11011100) = 01110011

The value of δ is determined in function $f$.

### 5.4 Data randomization effect of function $f$

Figure 6 is an example of the course of change of α during the processing of $f$(α, β). It is seen that when α or β changes by 1 bit, $f$ (α, β) changes greatly. Considering the situation where α is changed by 1 bit for various values of α and β, the change of the Hamming distance of function $f$ is shown in Fig. 7 (the number of samples is $2^{16}$). In this distribution, the mean μ is 10.4 and the standard deviation σ is approximately

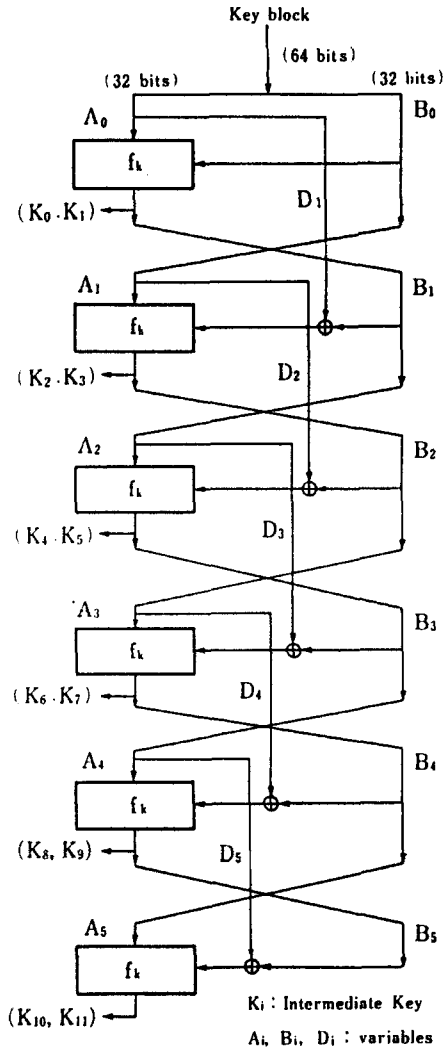



Fig. 8. FEAL Key Schedule.

4.2. The $f$ function in DES has approximately $\mu = 3.8$ and $\sigma = 1.5$, which indicates that the data randomization of the $f$ function in FEAL is considerably better.

### 5.5 Gatebranch

Let the left 32 bits of the plaintext, before entering the gatebranch, be $L_0$ and the right 32 bits be $R_0$ (Fig. 4). Let the change of $L_0$ be $\delta L_0$ and that of $R_0$ be $\delta R_0$.

(a) When there is no gatebranch:

If $\delta R_0 = 0$, DRE 1 does not operate.

The number of such cases is $\sum_{\gamma=1}^{32} {}_{32}C_r$.

(b) When there is a gatebranch:

If $\delta L_0 = \delta R_0$, $\delta L_0 \oplus \delta R_0 = 0$ and DRE 1 do not operate. This situation is restricted to the case where $r$ is even, and the number of such cases is $\sum_{\gamma=1}^{32} {}_{32}C_{r/2}$.

There is no difference between the number of operations of DRE 1 in (a) and (b). If there is a gatebranch, however, $\delta L_0 \neq \delta R_0$. Consequently, DRE 1 never fails to operate.

Furthermore, when there is a gatebranch, the intermediate keys ($K_4$, $K_5$) and ($K_6$, $K_7$) are combined by the exclusive-or, and the result is passed to DRE 1. This greatly increases the complexity of the key information, and makes it much harder to guess the key.

## 6. Key Scheduling

### 6.1 Basic structure

Figure 8 shows the structure of the key scheduling part. The structure of the key schedule is designed so that it is difficult to determine one intermediate key from another. From the values of $K_4$, $K_5$, $K_6$ and $K_7$, for example, the values of $K_0$, $K_1$, $K_2$ and $K_3$ cannot be determined. This makes it virtually impossible to break the cipher.

### 6.2 Structure of function $f_k$

Function $f_k$ (Fig. 9) adopts an interchanging network structure that is almost the same as that of function $f$ (Fig. 5).

### 6.3 Properties of function $f_k$

This section considers the properties of functions $S$ and $f_k$ required to thwart attempts to break the cipher.

Theorem 2. Assume that $\delta$ in function $y = S(x_1, x_2, \delta)$ is given ($\delta = 0$ or 1).

Then if two of $y$, $x_1$ and $x_2$ are given, the last is determined uniquely.

Proof. (1) When $x_1$ and $x_2$ are given, $w$ in definition 3 is determined uniquely. Consequently, $y$ is determined.

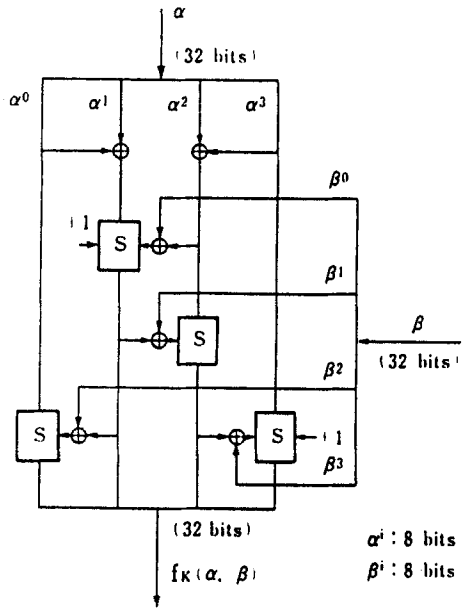



Fig. 9. Function $f$ in FEAL Key schedule.

(2) When $x_1$ and $y$ are given, $x_2$ is determined since $x_1$ and $w$ are determined uniquely.

(3) When $x_2$ and $y$ are given, $x_1$ is determined since $x_2$ and $w$ are determined uniquely. (Q.E.D.)

Theorem 3. Let function $f_k$ be $y = f_k(\alpha, \beta)$. If two of $y$, $\alpha$ and $\beta$ are given, the last is determined uniquely. In other words, the following single-valued functions $P$ and $Q$ exist:

$$\alpha = P(y, \beta)$$
$$\beta = Q(y, \alpha)$$

Proof. In the exclusive-or, $c = a \oplus b$ if two of $a$, $b$ and $c$ are given, the last is determined uniquely. Since function $f_k$ is a combination of function $S$ and exclusive-or, Theorem 3 holds, as seen by following the internal flow and its reverse flow in function $f_k$ of Fig. 9 using Theorem 2. (Q.E.D.)

Corollary of Theorem 3. Let function $f$ be $y = f(\alpha, \beta)$. If two of $y$, $\alpha$ and $\beta$ are given, the last is determined uniquely. In other words, there exist the following two single-valued functions $P^*$ and $Q^*$:

$$\alpha = P^*(y, \beta)$$
$$\beta = Q^*(y, \alpha)$$
$$K_{01} = (K_0, K_1),\ K_{23} = (K_2, K_3)$$

Definition 4.

$$K_{45} = (K_4, K_5),\ K_{67} = (K_6, K_7)$$

where $(K_0, K_1)$ represents the concatenation of $K_0$ and $K_1$ (similar in other cases).

Theorem 4. If three of $K_{01}$, $K_{23}$, $K_{45}$ and $K_{67}$ are given, the last is determined uniquely.

Proof. It follows from Fig. 8 that

$$K_{67} = f_k(K_{23}, K_{01} \oplus K_{45})$$

Applying Theorem 3 to determine $K_{01}$, $K_{23}$ and $K_{45}$,

$$K_{01} = K_{45} \oplus Q(K_{67}, K_{23})$$
$$K_{23} = P(K_{67}, K_{01} \oplus K_{45})$$
$$K_{45} = K_{01} \oplus Q(K_{67}, K_{23})$$ (Q.E.D.)

Theorem 5. If three of $K_{01}$, $K_{23}$, $K_{45}$ and $K_{67}$ are given, the key block $K$ is determined uniquely. Consequently, any three of the foregoing contain 64 bits of information of the key block.

Proof. It follows from Fig. 8 that

$$K_{23} = f_k(B_0, A_0 \oplus K_{01}) \quad (1)$$
$$K_{45} = f_k(K_{01}, B_0 \oplus K_{23}) \quad (2)$$

Determining $A_0$ and $B_0$ from Eqs. (1) and (2),

$$A_0 = K_{01} \oplus Q(K_{23}, K_{23} \oplus Q(K_{45}, K_{01}))$$
$$B_0 = K_{23} \oplus Q(K_{45}, K_{01})$$

Consequently, the key $K = (A_0, B_0)$ is determined by $K_{01}$, $K_{23}$ and $K_{45}$. Then Theorem 4 is applied. (Q.E.D.)

Corollary 1 of Theorem 5. If three of $K_{01}$, $K_{23}$, $K_{45}$ and $K_{67}$ are given, $K_{89} = (K_8, K_9)$ and $K_{1011} = (K_{10}, K_{11})$ are determined.

Corollary 2 of Theorem 5. If the key block $K$ of FEAL changes, the intermediate key $(K_{01}, K_{23}, K_{45}, K_{67}, K_{89}, K_{1011})$ always changes.

## 7. Discussion

### 7.1 Security of the cipher

#### 7.1.1 Independence of ciphertext change

(1) Comparison of FEAL and DES

It is seen from Table 1 that the ciphertext of FEAL has a greater degree of

Table 1. Comparison of Key strength (number of samples = $16^3 \times 64$, in%)

| | | FEAL | DES | Ideal algorithm |
|---|---|---|---|---|
| Key index | $M_K$ | 97.1 | 93.4 | 96.5 |
| | $M_K\ \sigma$ | 2.1 | 4.9 | 2.6 |
| Plaintext index | $M_P$ | 96.8 | 95.5 | 96.5 |
| | $M_P\ \sigma$ | 2.4 | 3.4 | 2.6 |

Note: For ideal algorithm, see Appendix 2.

Table 2. Dependence of index on ciphering Process

| No. of ciphering stages | | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Key index | $M_K$ | 88.2 | 94.6 | 96.8 | 97.1 |
| | $M_K\ \sigma$ | 8.6 | 4.0 | 2.4 | 2.1 |
| Plaintext index | $M_P$ | 42.9 | 77.9 | 96.3 | 96.8 |
| | $M_P\ \sigma$ | 57.9 | 16.4 | 2.7 | 2.4 |

Note: number of samples = $16^3$ x 64, in %. By index for R ciphering stages is meant the index measured by setting the number of ciphering elements (DRE) in data randomization part as R.

independence than that of DES. The probabilities of misdecision are $R_k$ = 23% and $R_p$ = 35%. The index of FEAL is almost equal to that of the ideal algorithm (for the ideal algorithm, see Appendix 2). DES has an unsatisfactory $M_k$. The reason is that the output (intermediate key) of the key schedule of DES is composed of rearrangements of only 48 bits extracted from 64 bits of the key, and the randomization of the key is insufficient.

(2) Dependence on the number of cipher stages: Table 2 shows the dependence of the indices on the number of cipher stages. The index of FEAL exceeds that of DES for 3 cipher stages ($R_k$ = 24%, $R_p$ = 41%), indicating that the change of the ciphertext in FEAL is more independent. Figure 10 shows the cipher process for a plaintext given as the input to the data randomization part.

### 7.1.2 Estimating the key using the plaintext and ciphertext

(1) Data representation on branch $S$:

| R | P | TEXT (Hexa) | | | | | | | | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | P1: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0 |
| | P2: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| | P3: | 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 1 |
| 1 | P1: | 3F | A6 | EE | 90 | DB | F8 | 2A | FC | 35 |
| | P2: | 58 | 64 | E3 | 79 | FB | 47 | 35 | 47 | |
| | P3: | D8 | 64 | E3 | 79 | 71 | 45 | 2D | 27 | 9 |
| 2 | P1: | DB | F8 | 2A | FC | 65 | 1D | 5B | 5A | 40 |
| | P2: | FB | 47 | 35 | 47 | 72 | EB | F4 | 04 | |
| | P3: | 71 | 45 | 2D | 27 | 20 | A9 | 8F | 2B | 24 |
| 3 | P1: | 65 | 1D | 5B | 5A | 7D | BC | E4 | 58 | 38 |
| | P2: | 72 | EB | F4 | 04 | 66 | B2 | 93 | E9 | |
| | P3: | 20 | A9 | 8F | 2B | 69 | A3 | 36 | 3A | 31 |
| 4 | P1: | 37 | 03 | 46 | F1 | ED | B0 | AF | 9C | 39 |
| | P2: | A8 | F8 | E0 | B1 | 1A | BA | BC | 6E | |
| | P3: | 10 | D9 | F4 | 34 | AD | 8A | OD | 38 | 27 |
| KEY1: | | 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 1 |
| KEY2: | | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| KEY3: | | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 0 |

(note) R : Round number
$\delta$ : Hamming distance to P2

Fig. 10. Data randomization process.

Consider Fig. 11. The numbers in the figure are the values of $i$ of the intermediate key $K_i$ which can determine uniquely the data on the branches (connections), when the plaintext block ($P$) and the ciphertext block ($C$) are given. The numbers in parenthesis indicate the intermediate key numbers when $C$ is given. The data on $A1$, for example, can uniquely be determined when $P$ and $K_4$, $K_5$ are given. The data on $D2$ can uniquely be determined using the corollary of Theorem 3, when $C$ as well as $K_8$, $K_9$, $K_{10}$, $K_{11}$ and $K_3$ are given.

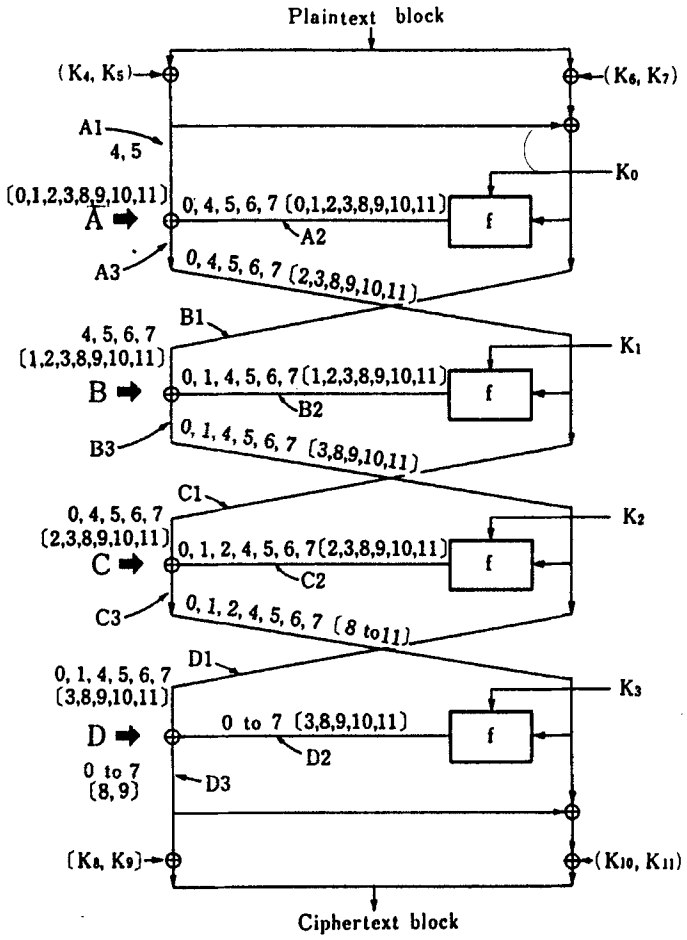
The data on a branch are indicated by the following notation:

$$D_b(P/C,\ i\cdots)$$

where $b$ is the branch name, $P$ indicates the case where $P$ is given, $C$ indicates the case where $C$ is given, and $i$ is the intermediate key number $K_i$.

(2) Estimating the intermediate key:

The data on the branch can be determined not only from $P$ and the intermediate key, but also from $C$ and the intermediate




Note : Numbers show intermediate key numbers included in branches. { } designates key numbers gained from the ciphertext block, others from the plaintext.

Fig. 11. Seeking Key block of FEAL.

key. The data determined by the two methods must be the same. This is illustrated in Table 3. Consider branch $A3$, and the calculation of the intermediate key by which $D_{A3}(P, 0, 4, 5, 6, 7)$ and $D_{A3}(C, 2, 3, 8, 9, 10, 11)$ are made equal.

For this purpose, it suffices to vary all $K_0$ - $K_{11}$ $K_0$, $K_2$, $K_3$, $K_4$, $K_5$, $K_6$, $K_7$, $K_8$, $K_9$, $K_{10}$ and $K_{11}$ intermediate keys which determine $D_{A3}$, and to establish the condition $D_{A3}(P, \cdots) = D_{A3}(C, \cdots)$. Among the intermediate keys, $K_2$, $K_3$, $K_4$, $K_5$, $K_6$ and $K_7$ contain all the information of the 64-bit key $K$ (Theorem 5). The situation is similar for other expressions in Table 3, since any expression contains three of $K_{01} = (K_0, K_1)$, $K_{23} = (K_2, K_3)$, $K_{45} = (K_4, K_5)$ and $K_{67} = (K_6, K_7)$. Thus, it is seen that 64 bits of key information are required to determine the intermediate key to obtain the data on all branches $A1$ to $D3$ in the data randomization part.

(3) When $K_4$ - $K_{11}$ are missing:

When ($K_4$, $K_5$, $K_6$, $K_7$) in ② of Fig. 3 and ($K_8$, $K_9$, $K_{10}$, $K_{11}$) of ③ are missing, the following relation applies:

$$D_{A3}(P,\ 0) = D_{A3}(C,\ 2,\ 3)$$

Then, ($K_0$, $K_2$, $K_3$) can be determined by at most $2^{48}$ trials, checking for all possible cases, and $K_1$ is determined by at most $2^{16}$ trials. Consequently, it is quite easy to break the cipher. Thus, it is verified that ② and ③ in Fig. 3 are effective in maintaining the integrity of the cipher.

Table 3. Relation among plaintext block, ciphertext block and intermediate key

| Intermediate key no. | | Relation |
|---|---|---|
| $D_{A1}$ (P, 4,5) | = $D_{A1}$( C, 0,1,2,3,8,9,a,b) | 0,1,2,3,4,5, 8,9,a,b |
| $D_{A2}$ (P, 0,4,5,6,7) | = $D_{A2}$( C, 0,1,2,3,8,9,a,b) | 0,1,2,3,4,5,6,7,8,9,a,b |
| $D_{A3}$ (P, 0,4,5,6,7) | = $D_{A3}$( C, 2,3,8,9,a,b) | 0, 2,3,4,5,6,7,8,9,a,b |
| $D_{B1}$ (P, 4,5,6,7) | = $D_{B1}$( C, 1,2,3,8,9,a,b) | 1,2,3,4,5,6,7,8,9,a,b |
| $D_{B2}$ (P, 0,1,4,5,6,7) | = $D_{B2}$ (C, 1,2,3,8,9,a,b) | 0,1,2,3,4,5,6,7,8,9,a,b |
| $D_{B3}$ (P, 0,1,4,5,6,7) | = $D_{B3}$ (C, 3,8,9,a,b) | 0,1, 3,4,5,6,7,8,9,a,b |
| $D_{C1}$ (P, 0,4,5,6,7) | = $D_{C1}$ (C, 2,3,8,9,a,b) | 0, 2,3,4,5,6,7,8,9,a,b |
| $D_{C2}$ (P, 0,1,2,4,5,6,7) | = $D_{C2}$ (C, 2,3,8,9,a,b) | 0,1,2,3,4,5,6,7,8,9,a,b |
| $D_{C3}$ (P, 0,1,2,4,5,6,7) | = $D_{C3}$ (C, 8,9,a,b) | 0,1,2, 4,5,6,7,8,9,a,b |
| $D_{D1}$ (P, 0,1,4,5,6,7) | = $D_{D1}$ (C, 3,8,9,a,b) | 0,1, 3,4,5,6,7,8,9,a,b |
| $D_{D2}$ (P, 0,1,2,3,4,5,6,7) | = $D_{D2}$ (C, 3,8,9,a,b) | 0;1,2,3,4,5,6,7,8,9,a,b |
| $D_{D3}$ (P, 0,1,2,3,4,5,6,7) | = $D_{D3}$ (C, 8,9) | 0,1,2,3,4,5,6,7,8,9 |

Note: $D_{A1}$ (P, 4, 5) indicates the data on A1 which can be determined by plaintext block P and the intermediate keys $K_4$, $K_5$; C indicates the data which can be determined by the ciphertext block and the intermediate key; a and b indicate 10 and 11, respectively (in hexadecimal notation). The data on A3, C1 and B3, D1 are the same, but they are written as above. The intermediate key number indicates all of those appearing in the expression (OR condition).

(4) Summary

All 64 bits of key information are required to determine the data on the branches. It is impossible to guess the key in FEAL even if a hypothetical interloper were in possession of the plaintext block P, ciphertext C and less than 64 bits of key information.

7.1.3 Key block length

To guarantee security, the key block length of FEAL is set at 64 bits. In DES, the length of the key block is also 64 bits but 8 bits are used for parity check, and only 56 bits are used for cipher. Furthermore, in DES, the complements of the plaintext and ciphertext blocks (i.e., the inversion of all bits) generates the complement of the ciphertext block, which reduces the effective key information by 1 bit, leaving only 55 bits for the key. Consequently, FEAL is much more secure against attempted branches of the cipher by examining all keys.

7.2 Performance of the program

Using a 16-bit μP (8086), the performance of FEAL and DES are compared. It was verified that compared with the DES program under the same conditions (i.e., the same μP, assembler and programmer), FEAL can achieve several tens of times the processing volume at only one-sixth the program size.

7.3 Realization by LSI

The exclusive-or and the addition modulo 256 in the ciphering units in FEAL can be realized by fundamental logic circuits and can easily be realized by LSI. The constant δ(0 or 1) given as the input to the adder in function $S$ can be considered as the array input to the least-significant bit. The 2-bit left-cyclic shift and the network structure of function $f$ can also be realized easily as connections on an LSI chip.

7.4 Processing unit in ciphering algorithm

Table 4 compares the processing units in FEAL and DES. Noteworthy features of FEAL are that substitution is not employed and the addition modulo 256 is used. Those schemes are useful for improving the speed of the FEAL program.

7.5 Error detection in the key block

In FEAL, the key block does not have a parity bit; the parity check method of error detection thus cannot be used. Key block error detection is performed by implementation of an enciphering/deciphering chip in the FEAL hardware, as described in the following paragraph.

Table 4. Comparison of cipherment algorithm processing elements

| Object | Kind of function | FEAL | DES |
|---|---|---|---|
| Data randomization | Coordinate permutation | 2-bit left cyclic shift Interchanging structure☆ | IP, $IP^{-1}$ P, E ★ |
| | Substitution | Not adopted | S box |
| Data concatenation | Addition modulo 256 | Newly adopted | —— (none) |
| | Exclusive-or | Adopted | Adopted |

☆Function to transfer data byte-wise in function $f$ and $f_k$.

★Although the coordinate permutation cannot be mathematically considered for a nonsquare matrix, it is written for convenience.

Error detection method (example): Two key block data are always prepared, which are read and compared. If there is a discrepancy, an error is detected. Usually, the ciphertext is much longer than the key, and the overhead by reading the key twice can be neglected. This method is used in some commercial ciphering devices.

## 8. Conclusions

This paper described the design of a secure cipher algorithm called FEAL, which can realize high-speed processing on a microprocessor. The independency of the ciphertext of FEAL (a kind of security) for the change of the key and the plaintext, is better than that of DES.

The method ensures that all key information (64 bits) is used on all branches in the cipher procedure This arrangement is desirable for preventing branches of the cipher. The performance of the microprocessor program was verified to be several tens of times that of DES, with an approximate program size one-sixth that of DES. In this paper, only one method of breaking the cipher has been considered. Other potential attacks will have to be thoroughly considered in the future. Another subject for further study is to investigate specific FEAL applications; for example, a simple key delivery system for personal computers.

Acknowledgement. In FEAL, a modification of the specification was made, where the security is guaranteed by adding intermediate keys $K_8$ - $K_{11}$. For this suggestion, the authors acknowledge the discussions and comments made by Dr. T. Shiraishi, Hitachi System Dev. Res. Lab., Dr. K. Ohta, Chief of Researcher, NTT Elect. Comm. Lab. and other members of 1986 Inf. Security Workshop. They also thank Dr. M. Hirano, NTT Elect. Comm. Lab. for constructing the strength index evaluation program for the cipher/verification algorithm.

## REFERENCES

1. NBS. Data Encryption Standard, FIPS-PUB-45 (1977).
2. S. Miyaguchi and M. Hirano. Strength evaluation index for cipherment/verification algorithm, Trans. (A), I.E.C.E., Japan, J69-A, 10, pp. 1252-1259 (Oct. 1986).
3. A.G. Kohnheim, Cryptography. A primer, clause 6.6 INVOLUTIONS, pp. 236-240, A Wiley Interscience Publication, John Wiley & Sons, New York (Jan. 1986).

# APPENDIX

## 1. Approximation Ratio $\theta_\nu$ for Ideal Binomial Distribution

The approximation ratio $\theta_\nu$ of the distribution of the stochastic variable $L_k$ to the ideal binomial distribution is described in the following (the description is similar for $L_p$). For various key blocks and plaintext blocks, the key block $K$ is varied under the following conditions, and the mean $X_\nu$ and the variance $Y_\nu$ of the stochastic variable $L_k$ are determined:

$$H(K_a,\ K_b)=\nu(\nu=1,\ 2,\ 3,\ \cdots)$$

where key $K$ is varied from $K_a$ to $K_b$.

By approximating the distribution of $L_k$ by a binomial distribution $B(n'', p'')$ the following relations apply:

$$X_\nu = n'' \cdot p''$$
$$Y_\nu = n'' \cdot p'' \cdot (1 - p'')$$

Determining $n''$ and $p''$ from these relations,

$$n'' = X^2/(X - Y), \quad p'' = (X - Y)/X$$

$\theta_\nu$ is defined as follows:

$$\theta_\nu = (n'/n) \cdot (p'/p)$$
$n = 64$ (64 -bit block cipherment)
$p = 0.5$
$$a = n - |n - n''|, \quad b = p - |p - p''|$$
$$n' = a \text{ (when } a \geqq 0), \quad 0 \text{ (when } a < 0)$$
$$p' = b \text{ (when } b \geqq 0), \quad 0 \text{ (when } b < 0)$$

## 2. Ideal Algorithm

The ideal algorithm is the cipher algorithm, in which $L_k$ and $L_p$ of the stochastic variable follows the ideal binomial distribution. Even in the ideal algorithm, $M_k$ and $M_p$ $R_p$ are not 100 percent, unless whole samples are examined. The parameters approach 100 percent as the number of samples is increased (Sect. 7 of [2]); $M_k$ and $M_p$ of FEAL in Table 1 are larger than those of the ideal algorithm, which is due to measurement error as was mentioned in the discussion of risk factor $R_p$ in comparison with the plaintext index. The measurement error decreases as the number of samples increases.

## 3. Outline of FEAL Specifications

### 1. Notations

(1) Block: $A$, $A_r$, $\cdots$ denote blocks with more than one byte.

(2) Byte: $A^j$ and $Ar^j$ denote the $j$-th ($j$ = 0, 1, $\cdots$) one byte in blocks $A$ and $A_r$, respectively.

(3) Concatenation: $(A, B, \cdots)$ indicates the concatenation in this order.

(4) $\oplus$: $A \oplus B$ indicates the bitwise exclusive-or of blocks $A$ and $B$.

(5) The equality indicates that the right-hand side is substituted into the left-hand side.

### 2. Function

#### 2.1 Function $S$

$S$ is a function with a one-byte block.

$$S(X1, X2, \delta) = \text{Rot } 2(W)$$
$$W = X1 + X2 + \delta \bmod 256$$

where $X1$ and $X2$ are one-byte blocks; $\delta$ is a constant which is either 0 or 1; $W$ is a one-byte block, which is obtained by operation $X1 + X2 + \delta$ mod 256 regarding $X1$ and $X2$ as binary nonnegative integers. Rot2($W$) is a one-byte block obtained by the left 2-bit cyclic shift (left 2-bit rotation) of one-byte block $W$.

Example: Rot 2(11011100) = 01110011

#### 2.2 Function $f_k$

For simplicity, $f_k(\alpha, \beta)$ is written as $f_k$; $f_k = (f_k^0, f_k^1, f_k^2, f_k^3)$ is calculated as follows:

$$f_k^1 = \alpha^1 \oplus \alpha^0$$
$$f_k^2 = \alpha^2 \oplus \alpha^3$$
$$f_k^1 = S(f_k^1, (f_k^2 \oplus \beta^0), 1)$$
$$f_k^2 = S(f_k^2, (f_k^1 \oplus \beta^1), 0)$$
$$f_k^0 = S(\alpha^0, (f_k^1 \oplus \beta^2), 0)$$
$$f_k^3 = S(\alpha^3, (f_k^2 \oplus \beta^3), 1)$$

#### 2.3 Function $f$

For simplicity, $f(\alpha, \beta)$ is written as $f$; $f = (f^0, f^1, f^2, f^3)$ is calculated as follows:

$$f^1 = \alpha^1 \oplus \beta^0 \oplus \alpha^0$$
$$f^2 = \alpha^2 \oplus \beta^1 \oplus \alpha^3$$
$$f^1 = S(f^1, f^2, 1)$$
$$f^2 = S(f^2, f^1, 0)$$
$$f^0 = S(\alpha^0, f^1, 0)$$
$$f^3 = S(\alpha^3, f^2, 1)$$

### 3. Key Schedule

The intermediate keys $K_i$ ($i$ = 0 - 11) (each being 2 bytes) are determined from the 64-bit key block. Let the left and the right of the key block be $A_0$ and $B_0$, respectively.

As the first step, let $D_0 \neq \phi$, where $\phi$ represents the null block (all elements being 0) of 4 bytes. Then $K_i$ ($i$ = 0 - 11) are determined for $r$ = 1 to 6:

$$D_r = A_{r-1}$$
$$A_r = B_{r-1}$$
$$B_r = f_k(A_{r-1},\ B_{r-1} \oplus D_{r-1})$$
$$K_{2(r-1)} = (B_r^0,\ B_r^1)$$
$$K_{2(r-1)+1} = (B_r^2,\ B_r^3)$$

where $A_r$, $B_r$ and $D_r$ are auxiliary variables.

4. Cipher procedure

Let the left and the right 4-byte blocks of the plaintext block be $L_0$ and $R_0$, respectively. Let

$$(L_0,\ R_0) = (L_0,\ R_0) \oplus (K_4,\ K_5,\ K_6,\ K_7)$$
$$(L_0,\ R_0) = (L_0,\ R_0) \oplus (\phi,\ L_0)$$

Then letting $r$ = 1 to 4, $R_r$ and $L_r$ are successively calculated:

$$R_r = L_{r-1} \oplus f(R_{r-1},\ K_{r-1})$$
$$L_r = R_{r-1}$$

The ciphertext block is obtained as $(R_4,\ L_4)$:

$$(R_4,\ L_4) = (R_4,\ L_4) \oplus (\phi,\ R_4)$$
$$(R_4,\ L_4) = (R_4,\ L_4) \oplus (K_8,\ K_9,\ K_{10},\ K_{11})$$

5. Decipherment procedure

Let the left and the right 4-byte blocks of the ciphertext block be $R_4$ and $L_4$, respectively. Let

$$(R_4,\ L_4) = (R_4,\ L_4) \oplus (K_8,\ K_9,\ K_{10},\ K_{11})$$
$$(R_4,\ L_4) = (R_4,\ L_4) \oplus (\phi,\ R_4)$$

Then $L_{r-1}$ and $R_{r-1}$ are calculated successively for $r$ = 4 to 1:

$$L_{r-1} = R_r \oplus f(L_r,\ K_{r-1})$$
$$R_{r-1} = L_r$$

Finally, for $L_0$ and $R_0$,

$$(L_0,\ R_0) = (L_0,\ R_0) \oplus (\phi,\ L_0)$$
$$(L_0,\ R_0) = (L_0,\ R_0) \oplus (K_4,\ K_5,\ K_6,\ K_7)$$

The plaintext block is obtained as $(L_0,\ R_0)$.

6. Example of manipulation (hexadecimal)

(1) Key: $K$ = 4B 45 59 42 4C 4F 43 4B.

(2) Intermediate key:

$(K_0,\ K_1,\ K_2,\ K_3)$ = 94 99 C7 3F F4 6F EE 7F

$(K_4,\ K_5,\ K_6,\ K_7)$ = A0 39 44 C2 E9 05 D8 96

$(K_8,\ K_9,\ K_{10},\ K_{11})$ = 0A D4 11 EE EF 47 E3 99

(3) Plaintext = F0 75 05 8B 07 68 C4 00.

(4) Ciphertext = 38 EA 14 81 96 4C B2.

AUTHORS (from left to right)

Akihiro Shimizu graduated in 1981 from Dept. Electrical Eng., Fac. Eng., Ehime Univ. and affiliated with NTT Yokosuka Elect. Comm. Lab. Engaged in research on image processing, information security and Japanese language processing. Presently, Research Engineer Chief, In-House Intell. Device Lab., In-House Device Div., NTT Composite Comm. Lab.

Shoji Miyaguchi graduated in 1965 from Dept. Electrical Eng., Niigata Univ. Completed Master's program 1967 Kyushu Univ., and affiliated with NTT. Engaged in research on microprocessor assist systems and network security. Senior Research Engineer, Supervisor, NTT Inf. Comm. & Inf. Proc. Labs.