Specification of $\boldsymbol{E2}$ – a 128-bit Block Cipher

Nippon Telegraph and Telephone Corporation

June 14, 1998

Contents

1 Description of Algorithm

1.1 Notations and Conventions

The following notations are used in this document.

1. Let $\mathbf{Z}$ denote the set of all integers.

2. Let A, B, and C be sets. Let $A \times B := \{(a,b) | a \in A, b \in B\}$ represent the Cartesian product of A and B. An element in $A \times B \times C$ is identified as follows: $(a,b,c) = ((a,b),c) = (a,(b,c))$. Moreover, let $A^1 := A$, and $A^n := A \times A^{n-1}$ for $n \geq 2$.

3. For an element $(a_{n-1}, a_{n-2}, \ldots, a_0)$ of set A^n, let a_{n-1} be the left most element, and a_0 be the right most element.

4. Let $\mathcal{K}$ be a field and $n \geq 1$. Let $\mathcal{K}^n$ be the n-dimensional vector space over $\mathcal{K}$. For $a = (a_{n-1}, a_{n-2}, \ldots, a_0)$, $b = (b_{n-1}, b_{n-2}, \ldots, b_0) \in \mathcal{K}^n$, and $\lambda \in \mathcal{K}$, the following equations hold.

$$\begin{aligned} a + b &= (a_{n-1} + b_{n-1}, a_{n-2} + b_{n-2}, \ldots, a_0 + b_0) \\ \lambda a &= (\lambda a_{n-1}, \lambda a_{n-2}, \ldots, \lambda a_0) \end{aligned}$$

5. When $\mathcal{K} = \mathrm{GF}(2) = \{0,1\}$, the Exclusive-Or operation, $\oplus$, is considered as the addition operation. Operation $\oplus$ is called the XOR operation simply.

6. A row vector $r = (r_{n-1}, r_{n-2}, \ldots, r_0)$ is identified with the column vector ${}^T r$.

7. Let $\mathbf{B}$ represent a vector space of 8-bit (byte[1]) elements, that is, $\mathbf{B} := \mathrm{GF}(2)^8$.

8. Let $\mathbf{W}$ represent a vector space of 32-bit (word) elements, that is, $\mathbf{W} := \mathbf{B}^4$.

9. Let $\mathbf{H}$ represent a vector space of 64-bit (half block) elements, that is, $\mathbf{H} := \mathbf{B}^8$.

10. An element of the field $\mathrm{GF}(2^8)$ is identified with a polynomial $p(X)$ in $\mathrm{GF}(2)[X]$ whose degree is less than 8, where $\mathrm{GF}(2^8)$ is isomorphic to $\mathrm{GF}(2)[X]/(r(X))$ and $r(X) = X^8 + X^4 + X^3 + X + 1$ which is an irreducible polynomial in $\mathrm{GF}(2)[X]$. Thus the complete set of representatives is $\{p(X) \bmod r(X) \in \mathrm{GF}(2^8) | \deg p(X) < 8\}$.

[1]In this document, a byte means octet.

11. An element $p(X)$ of the set $\mathrm{GF}(2)[X]/(r(X))$ represented by $p(X) = \sum_{i=0}^{7} a_i X^i$ is identified with $(a_7, a_6, \ldots, a_0) \in \mathbf{B}$.

12. An element $(a_7, a_6, \ldots, a_0)$ in the set $\mathbf{B}$, where $a_i \in \mathrm{GF}(2)$, is identified with

$$\sum_{i=0}^{7} \tilde{a}_i 2^i \bmod 2^8\mathbf{Z} \in \mathbf{Z}/2^8\mathbf{Z},$$

where $a_i \in \mathrm{GF}(2)$ $(i = 0, 1, \ldots, 7)$ corresponds to $\tilde{a}_i \in \{0, 1\} \subset \mathbf{Z}$ in a canonical way, i.e., a_7 is the most significant (left most) bit and a_0 is the least significant (right most) bit.

13. An element (b_3, b_2, b_1, b_0) in the set $\mathbf{W}$, where $b_i \in \mathbf{B}$, is identified with

$$\sum_{i=0}^{3} \tilde{b}_i 2^{8i} \bmod 2^{32}\mathbf{Z} \in \mathbf{Z}/2^{32}\mathbf{Z},$$

where $b_i \in \mathbf{B}$ $(i = 0, 1, 2, 3)$ corresponds to $\tilde{b}_i \in \{0, 1, \ldots, 2^8 - 1\} \subset \mathbf{Z}$. The correspondence of b_i to $\tilde{b}_i$ is defined in item 12.

1.2 Outline

Let

$$\begin{array}{lll} M & \text{be a plaintext} & (M \in \mathbf{H}^2) \\ K & \text{be a secret-key} & (K \in \mathbf{H}^2, \mathbf{H}^3 \text{ or } \mathbf{H}^4), \text{ and} \\ C & \text{be a ciphertext} & (C \in \mathbf{H}^2). \end{array}$$

The encryption algorithm ***E2*** is defined as:

$$C = \mathrm{E}(M, K)$$
$$M = \mathrm{D}(C, K),$$

where E is the encryption function of ***E2***, which is described in Section 1.3, and D is the decryption function of ***E2***, which is described in Section 1.4. The following equations hold.

$$\begin{aligned} M &= \mathrm{D}(\mathrm{E}(M, K), K) \\ C &= \mathrm{E}(\mathrm{D}(C, K), K) \end{aligned}$$

1.3 Encryption

The data randomizing part consists of an initial transformation IT, a 12-round Feistel cipher structure with F-Function, and a final transformation FT. The key scheduling part generates 16 subkeys $\{k_1, k_2, \ldots, k_{16}\}$ ($k_i \in \mathbf{B}^{16}$), from a secret-key K before encryption.

First, calculate

$$M' = IT(M, k_{13}, k_{14})$$

where M is a plaintext. Next, M' is separated into L_0 and R_0 of equal length, i.e., M'=(L_0, R_0), where $L_0 \in \mathbf{H}$ and $R_0 \in \mathbf{H}$. Then, calculate the following from $r = 1$ to 12.

$$\begin{aligned} R_r &= L_{r-1} \oplus F(R_{r-1}, k_r) \\ L_r &= R_{r-1} \end{aligned}$$

Let C' be the concatenation of R_{12} and L_{12}, i.e., $C' = (R_{12}, L_{12})$.

Finally, calculate

$$C = FT(C', k_{16}, k_{15}).$$

The result C is a ciphertext.

The encryption is shown in Figure 1. IT-Function is described in Section 2.1, F-Function is described in Section 2.2, and FT-Function is described in Section 2.3.

1.4 Decryption

Similarly to encryption, the data randomizing part consists of an initial transformation IT, a 12-round Feistel structure with F-Function, and a final transformation FT. The key scheduling part generates 16 subkeys $\{k_1, k_2, \ldots, k_{16}\}$ ($k_i \in \mathbf{B}^{16}$), from a secret-key K before decryption.

First, calculate

$$C' = IT(C, k_{16}, k_{15})$$

where C is a ciphertext. Next, C' is separated into R_{12} and L_{12} of equal length, i.e., $C' = (R_{12}, L_{12})$ where $R_{12} \in \mathbf{H}$, $L_{12} \in \mathbf{H}$. Then, calculate the following from $r = 12$ down to 1.

$$\begin{aligned} L_{r-1} &= R_r \oplus F(L_r, k_r) \\ R_{r-1} &= L_r \end{aligned}$$

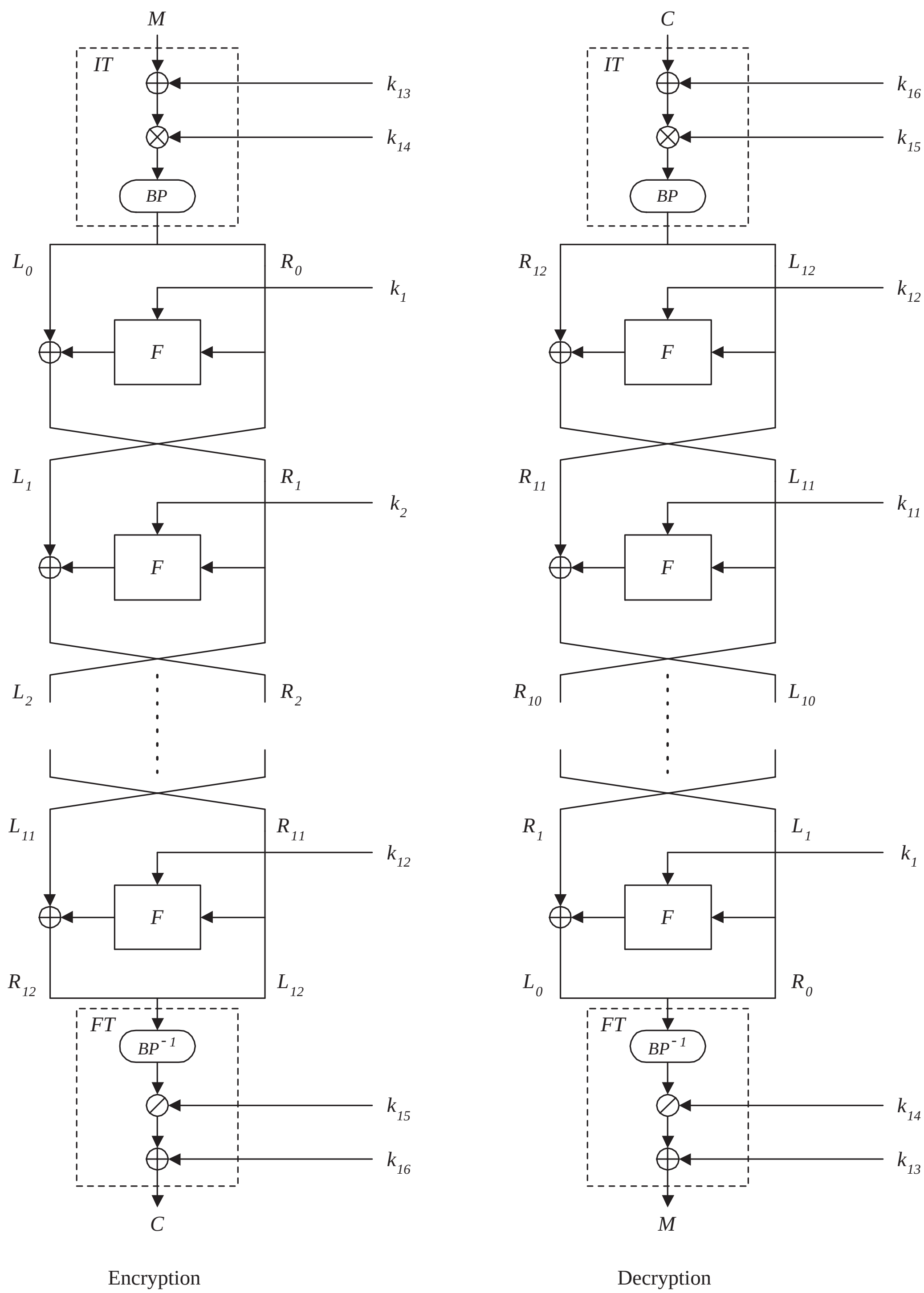

Figure 1: Encryption and Decryption Procedures

Let M' be the concatenation of L_0 and R_0, i.e., $M' = (L_0, R_0)$.

Finally, calculate

$$M \quad = \quad FT(M', k_{13}, k_{14}).$$

The result M is a plaintext.

The decryption is shown in Figure 1. F-Function is described in Section 2.2, IT-Function is described in Section 2.1, and FT-Function is described in Section 2.3.

1.5 Key Scheduling

For secret-key $K = (K_1, K_2, K_3, K_4)$ ($K_i \in \mathbf{H}$, $i = 1, 2, 3, 4$), which is given as input to ***E****2* (E or D), the subkeys $k_i \in \mathbf{B}^{16}$ ($i = 1, 2, \ldots, 16$) are generated as follows using G- and S-Functions defined later.

$$\begin{aligned}
&v_{-1} = \texttt{0123456789abcdef}_{\text{(hex)}} \\
&(L_0, (Y_0, v_0)) = G(K, v_{-1}) \\
&(L_{i+1}, (Y_{i+1}, v_{i+1})) = G(Y_i, v_i) \quad (i = 0, 1, 2, \ldots, 7) \\
&(l_{4i}, l_{4i+1}, l_{4i+2}, l_{4i+3}) = L_{i+1} \quad (i = 0, 1, \ldots, 7) \\
&(t_i^{(0)}, t_i^{(1)}, \ldots, t_i^{(7)}) = l_i \quad (i = 0, 1, \ldots, 31) \\
&k_{i+1} = (t_{0+(i \bmod 2)}^{(\lfloor i/2 \rfloor)}, t_{2+(i \bmod 2)}^{(\lfloor i/2 \rfloor)}, \ldots, t_{30+(i \bmod 2)}^{(\lfloor i/2 \rfloor)}) \quad (i = 0, 1, \ldots, 15)
\end{aligned}$$

where $L_i, Y_i \in \mathbf{H}^4, l_i, v_i \in \mathbf{H}$, and $t_i^{(j)} \in \mathbf{B}$.

The procedure for generating subkeys is the same when the secret-key is 128-, 192-, or 256-bits. When the secret-key is 128-bits, constant values are set on K_3 and K_4: $K_3 = S(S(S(v_{-1})))$, $K_4 = S(S(S(S(v_{-1}))))$, respectively. When the secret-key is 192-bits, a constant value is set on K_4: $K_4 = S(S(S(S(v_{-1}))))$.

S-Function is described in Section 2.5, and G-Function is described in Section 2.8 and shown in Figure 3.

2 Functions

Let variables denoted by small letters, e.g., x, y, x_i, y_i, be elements of $\mathbf{B}$ or $\mathbf{W}$, and variables denoted by capital letters, e.g., X, Y, be elements of $\mathbf{H}$ or $\mathbf{H}^2$ hereafter if not stated explicitly otherwise. Figures are represented as decimals without an explicit description.

2.1 *IT*-Function

IT-Function, which we call the initial transformation, is defined as follows:

$$IT : \mathbf{H}^2 \times \mathbf{H}^2 \times \mathbf{H}^2 \longrightarrow \mathbf{H}^2;\ (X, A, B) \longmapsto BP((X \oplus A) \otimes B)$$

The binary operator $\otimes$ is described in Section 2.10, and BP-Function, which we call the byte permutation, is described in Section 2.12.

2.2 *F*-Function

F-Function is defined as follows:

$$\begin{array}{rcl} F : \mathbf{H} \times \mathbf{H}^2 & \longrightarrow & \mathbf{H} \\ (X, (K^{(1)}, K^{(2)})) & \longmapsto & Y = BRL(S(P(S(X \oplus K^{(1)})) \oplus K^{(2)})). \end{array}$$

F-Function is shown in Figure 2. S-Function is described in Section 2.5, P-Function is described in Section 2.7, and BRL-Function is described in Section 2.4.

2.3 *FT*-Function

FT-Function, which we call the final transformation, is defined as follows:

$$FT : \mathbf{H}^2 \times \mathbf{H}^2 \times \mathbf{H}^2 \longrightarrow \mathbf{H}^2;\ (X, A, B) \longmapsto (BP^{-1}(X) \oslash B) \oplus A$$

The binary operator $\oslash$ is described in Section 2.11. BP- and BP^{-1}-Function are described in Section 2.12.

Note that FT-Function is the inverse of IT-Function, i.e.,

$$X = FT(IT(X, A, B), A, B).$$

2.4 *BRL*-Function

BRL-Function, which we call the byte rotate left function, is a part of F-Function and is defined as follows:

$$BRL : \mathbf{H} \longrightarrow \mathbf{H};\ (b_1, b_2, b_3, \ldots, b_8) \longmapsto (b_2, b_3, \ldots, b_8, b_1).$$

BRL-Function is shown in Figure 2.

2.5 S-Function

S-Function is a part of F-Function, and is defined as follows using s-boxes:

$$S : \mathbf{H} \longrightarrow \mathbf{H};\ (x_1, x_2, \ldots, x_8) \longmapsto (s(x_1), s(x_2), \ldots, s(x_8)).$$

s-box is described in Section 2.6.

2.6 s-box

The definition of s-box in S-Function is described as follows:

$$s : \mathbf{B} \longrightarrow \mathbf{B};\ x \longmapsto \mathrm{Affine}(\mathrm{Power}(x, 127), 97, 225),$$

where

$$\begin{aligned} \mathrm{Power}(x, e) &= x^e \quad \text{in } \mathrm{GF}(2^8) \\ \mathrm{Affine}(y, a, b) &= ay + b \quad (\mathrm{mod}\ 2^8\mathbf{Z}). \end{aligned}$$

The following canonical identification among <u>sets</u> is adopted here:

$$\mathrm{GF}(2^8) = \mathrm{GF}(2)[X]/(r(X)) = \mathrm{GF}(2)^8 = \mathbf{Z}/2^8\mathbf{Z}, \tag{1}$$

where the first equality = is given in item 10 in Section 1.1, the second one is given in item 11, and the third one is given in item 12. The calculation result of Power-Function in $\mathrm{GF}(2^8)$ is considered to be an element in $\mathbf{Z}/2^8\mathbf{Z}$, which is input to Affine-Function, as given in the above relation. The table expression of s-box is given as follows. This means that $s(0) = 225, s(1) = 66, \ldots, s(16) = 204, \ldots,$ and $s(255) = 42$.

225	66	62	129	78	23	158	253	180	63	44	218	49	30	224	65
204	243	130	125	124	18	142	187	228	88	21	213	111	233	76	75
53	123	90	154	144	69	188	248	121	214	27	136	2	171	207	100
9	12	240	1	164	176	246	147	67	99	134	220	17	165	131	139
201	208	25	149	106	161	92	36	110	80	33	128	47	231	83	15
145	34	4	237	166	72	73	103	236	247	192	57	206	242	45	190
93	28	227	135	7	13	122	244	251	50	245	140	219	143	37	150
168	234	205	51	101	84	6	141	137	10	94	217	22	14	113	108
11	255	96	210	46	211	200	85	194	35	183	116	226	155	223	119
43	185	60	98	19	229	148	52	177	39	132	159	215	81	0	97
173	133	115	3	8	64	239	104	254	151	31	222	175	102	232	184
174	189	179	235	198	107	71	169	216	167	114	238	29	126	170	182
117	203	212	48	105	32	127	55	91	157	120	163	241	118	250	5
61	58	68	87	59	202	199	138	24	70	156	191	186	56	86	26
146	77	38	41	162	152	16	153	112	160	197	40	193	109	20	172
249	95	79	196	195	209	252	221	178	89	230	181	54	82	74	42

2.7 *P*-Function

P-Function is a part of F-Function, and is defined as follows using a matrix expression.

$$P : \mathbf{H} \to \mathbf{H}; \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_8 \end{pmatrix} \longmapsto \begin{pmatrix} z'_1 \\ z'_2 \\ \vdots \\ z'_8 \end{pmatrix} = P \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_8 \end{pmatrix}$$

where matrix P is given as follows:

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

We can calculate P-Function using Figure 2, for example.

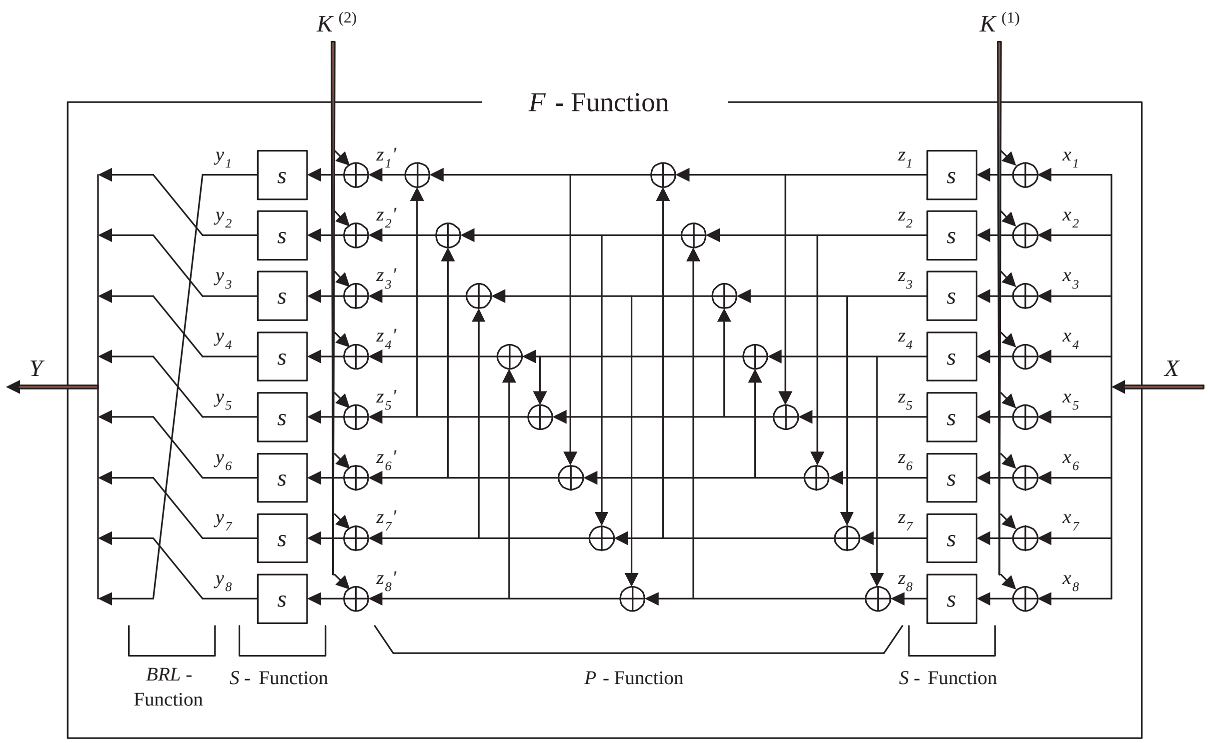

Figure 2: F-Function

2.8 *G*-Function

G-Function is defined as follows:

$$\begin{array}{rcl} G : \mathbf{H}^4 \times \mathbf{H} & \longrightarrow & \mathbf{H}^4 \times (\mathbf{H}^4 \times \mathbf{H}) \\ ((X_1, X_2, X_3, X_4), U_0) & \longmapsto & ((U_1, U_2, U_3, U_4), ((Y_1, Y_2, Y_3, Y_4), V)) \\ \text{where} & & \\ Y_i & = & f(X_i) \quad (i = 1, 2, 3, 4) \\ U_i & = & f(U_{i-1}) \oplus Y_i \quad (i = 1, 2, 3, 4) \\ V & = & U_4 \end{array}$$

G-Function is shown in Figure 3, and f-Function is described in Section 2.9.

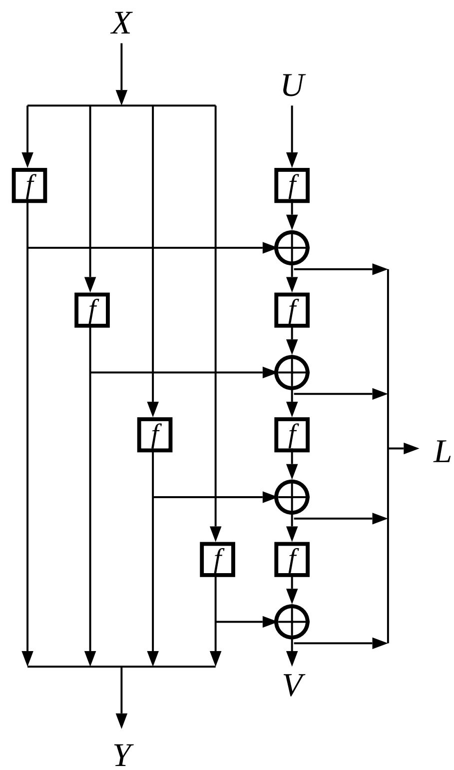

Figure 3: G-Function

2.9 f-Function

f-Function is a part of G-Function, and is defined as follows:

$$f : \mathbf{H} \longrightarrow \mathbf{H};\ X \longmapsto P(S(X)).$$

2.10 Binary Operator $\otimes$

The binary operator $\otimes$ is defined as follows.

$$\begin{aligned}
Y &= X \otimes B \quad (X, Y, B \in \mathbf{H}^2) \\
\text{where} & \\
(x_1, x_2, x_3, x_4) &= X \quad (x_i \in \mathbf{W},\ i = 1, 2, 3, 4) \\
(b_1, b_2, b_3, b_4) &= B \quad (b_i \in \mathbf{W},\ i = 1, 2, 3, 4) \\
y_i &= x_i(b_i \vee 1) \bmod 2^{32}\mathbf{Z} \quad (i = 1, 2, 3, 4) \\
Y &= (y_1, y_2, y_3, y_4)
\end{aligned}$$

Let $\vee 1$ denote bitwise logical OR with $1 \in 2^{32}\mathbf{Z}$.

2.11 Binary Operator $\oslash$

The binary operator $\oslash$ is defined as follows.

$$\begin{aligned}
X &= Y \oslash B \quad (X, Y, B \in \mathbf{H}^2) \\
\text{where} & \\
(y_1, y_2, y_3, y_4) &= Y \quad (y_i \in \mathbf{W},\ i = 1, 2, 3, 4) \\
(b_1, b_2, b_3, b_4) &= B \quad (b_i \in \mathbf{W},\ i = 1, 2, 3, 4) \\
x_i &= y_i(b_i \vee 1)^{-1} \bmod 2^{32}\mathbf{Z} \quad (i = 1, 2, 3, 4) \\
X &= (x_1, x_2, x_3, x_4)
\end{aligned}$$

Let $\vee 1$ denote bitwise logical OR with $1 \in 2^{32}\mathbf{Z}$.

2.12 BP-Function

BP-Function, which we call the byte permutation, is a part of IT- and FT-Function. It is defined as follows.

$$BP : \mathbf{W}^4 \quad \longrightarrow \quad \mathbf{W}^4$$

$$
\begin{array}{rcl}
(x_1, x_2, x_3, x_4) & \longmapsto & (y_1, y_2, y_3, y_4) \\
\text{where} & & \\
(x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, x_i^{(4)}) & = & x_i \quad (x_i^{(j)} \in \mathbf{B},\ i = 1,2,3,4,\ j = 1,2,3,4) \\
y_i & = & (x_i^{(1)}, x_{i+1}^{(2)}, x_{i+2}^{(3)}, x_{i+3}^{(4)}) \quad (i = 1,2,3,4) \\
 & & \quad (x_{i+4}^{(j)} \text{ is identified with } x_i^{(j)},\ i = 0,1,2,3,\ j = 1,2,3,4) \\
Y & = & (y_1, y_2, y_3, y_4)
\end{array}
$$

We can calculate BP^{-1} as follows.

$$
\begin{array}{rcl}
BP^{-1} : \mathbf{W}^4 & \longrightarrow & \mathbf{W}^4 \\
(y_1, y_2, y_3, y_4) & \longmapsto & (x_1, x_2, x_3, x_4) \\
\text{where} & & \\
(y_i^{(1)}, y_i^{(2)}, y_i^{(3)}, y_i^{(4)}) & = & y_i \quad (y_i^{(j)} \in \mathbf{B},\ i = 1,2,3,4,\ j = 1,2,3,4) \\
x_i & = & (y_i^{(1)}, y_{i-1}^{(2)}, y_{i-2}^{(3)}, y_{i-3}^{(4)}) \quad (i = 1,2,3,4) \\
 & & \quad (y_{i-4}^{(j)} \text{ is identified with } y_i^{(j)},\ i = 1,2,3,4,\ j = 1,2,3,4) \\
X & = & (x_1, x_2, x_3, x_4)
\end{array}
$$

BP-Function is shown in Figure 4.

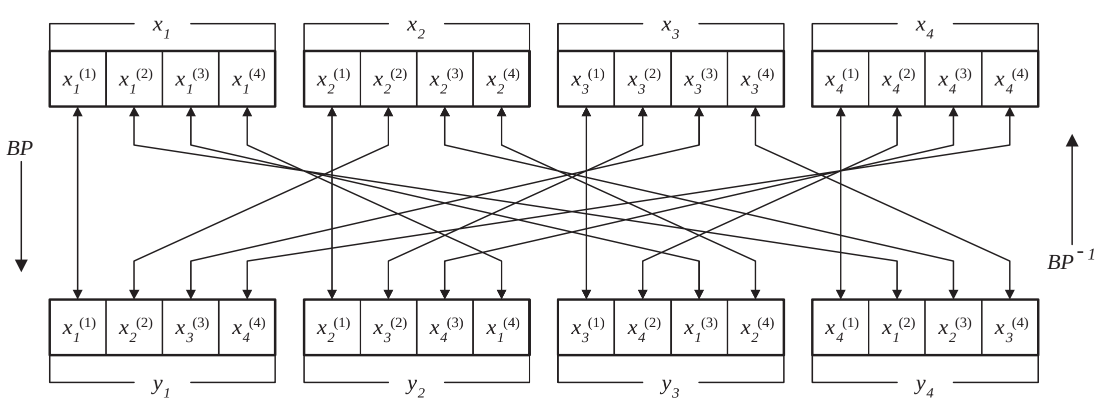

Figure 4: BP-Function

A Test Data for *E2*

Sample data are shown in hexadecimal notation.

Case 1) The key length is 128-bits long

$K =$ 00000000000000000000000000000000

$M =$ 00000000000000000000000000000000

$C =$ c2883490b9d9d5e5a03f216edb815fff

Case 2) The key length is 192-bits long

$K =$ 000000000000000000000000000000000000000000000000

$M =$ 00000000000000000000000000000000

$C =$ 882f80269d3c146d6ebb9addc4715b4c

Case 3) The key length is 256-bits long

$K =$ 0000000000000000000000000000000000000000000000000000000000000000

$M =$ 00000000000000000000000000000000

$C =$ 5002cb8cd878f26fbab9f52e6c96501e