INTERNATIONAL STANDARD

ISO/IEC 18033-3

First edition
2005-07-01

Information technology — Security techniques — Encryption algorithms —

Part 3: Block ciphers

Technologies de l'information — Techniques de sécurité — Algorithmes de chiffrement —

Partie 3: Chiffrement par blocs

Reference number
ISO/IEC 18033-3:2005(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword v

Introduction vi

1 Scope 1

2 Terms and definitions 1

3 Symbols 2

4 64-bit block ciphers 2
4.1 TDEA 3
4.1.1 TDEA encryption/decryption 3
4.1.2 TDEA keying options 3
4.2 MISTY1 3
4.2.1 MISTY1 encryption 3
4.2.2 MISTY1 decryption 4
4.2.3 MISTY1 functions 4
4.2.4 MISTY1 key schedule 9
4.3 CAST-128 10
4.3.1 CAST-128 encryption 10
4.3.2 CAST-128 decryption 10
4.3.3 CAST-128 functions 10
4.3.4 CAST-128 key schedule 17

5 128-bit block ciphers 20
5.1 AES 20
5.1.1 AES encryption 20
5.1.2 AES decryption 21
5.1.3 AES transformations 21
5.1.4 AES key schedule 26
5.2 Camellia 27
5.2.1 Camellia encryption 27
5.2.2 Camellia decryption 29
5.2.3 Camellia functions 32
5.2.4 Camellia key schedule 38
5.3 SEED 42
5.3.1 SEED encryption 42
5.3.2 SEED decryption 42
5.3.3 SEED functions 43
5.3.4 SEED key schedule 46

Annex A (normative) **Description of DES** 47
A.1. DES encryption 47
A.2. DES decryption 47
A.3. DES functions 47
A.3.1 Initial permutation *IP* 47
A.3.2 Inverse initial permutation IP^{-1} 48
A.3.3 Function *f* 49
A.3.4 Expansion permutation *E* 49
A.3.5 Permutation *P* 50
A.3.6 S-Boxes 50
A.4 DES key schedule (*KS*) 51

Annex B (normative) **ASN.1 module** 53

Annex C (informative) **Algebraic forms of MISTY1 and Camellia S-boxes** 55
C.1 MISTY1 S-boxes 55

C.1.1 MISTY1 S-box S_7 ..... **55**
C.1.2 MISTY1 S-box S_9 ..... **55**
C.2 Camellia S-box **55**

Annex D (informative) **Test vectors** **57**
D.1 TDEA test vectors **57**
D.1.1 TDEA encryption **57**
D.1.2 DES encryption and decryption **58**
D.2 MISTY1 test vectors **59**
D.3 CAST-128 test vectors **60**
D.4 AES test vectors **60**
D.4.1 AES encryption **60**
D.4.2 Key expansion example **61**
D.4.3 Cipher example **63**
D.5 Camellia test vectors **65**
D.5.1 Camellia encryption **65**
D.6 SEED test vectors **68**

Annex E (informative) **Feature table** **70**

Bibliography **71**

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 18033-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

ISO/IEC 18033 consists of the following parts, under the general title *Information technology — Security techniques — Encryption algorithms*:

— *Part 1: General*

— *Part 2: Asynnetric ciphers*

— *Part 3: Block ciphers*

— *Part 4: Stream ciphers*

Introduction

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

The ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the ISO and IEC that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with the ISO and IEC. Information may be obtained from:

ISO/IEC JTC 1/SC 27 Standing Document 8 (SD8) "Patent Information"

Standing Document 8 (SD8) is available at http://www.ni.din.de/sc27

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Security techniques — Encryption algorithms —

Part 3:
Block ciphers

1 Scope

This part of ISO/IEC 18033 specifies block ciphers. A block cipher maps blocks of *n* bits to blocks of *n* bits, under the control of a key of *k* bits. A total of six different block ciphers are defined. They are categorized in Table 1.

Table 1. Block ciphers specified

Block length	Algorithm name (Clause #)	Key length
64 bits	TDEA (4.1)	128 or 192 bits
	MISTY1 (4.2)	128 bits
	CAST-128 (4.3) [1]	
128 bits	AES (5.1)	128, 192 or 256 bits
	Camellia (5.2)	
	SEED (5.3)	128 bits

The algorithms specified in this part of ISO/IEC 18033 have been assigned object identifiers in accordance with ISO/IEC 9834. The list of assigned object identifiers is given in Annex B. Any changes to the specification of the algorithms resulting in a change of functional behaviour will result in a change of the object identifier assigned to the algorithm.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

2.1
block
string of bits of defined length. [ISO/IEC 18033-1:2004]

NOTE – In this part of ISO/IEC 18033, the block length is either 64 or 128 bits.

2.2
block cipher
symmetric encipherment system with the property that the encryption algorithm operates on a block of plaintext, i.e. a string of bits of a defined length, to yield a block of ciphertext. [ISO/IEC 18033-1:2004]

2.3
ciphertext
data which has been transformed to hide its information content. [ISO/IEC 9798-1:1997]

[1] The key length of the original version of CAST-128 is variable from 40 bits to 128 bits. This part of ISO/IEC 18033, however, specifies its use only with keys of 128 bits.

2.4
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. encipherment, decipherment). [ISO/IEC 11770-1:1996]

NOTE – In all the ciphers specified in this part of ISO/IEC18033, keys consist of a sequence of bits.

2.5
***n*-bit block cipher**
block cipher with the property that plaintext blocks and ciphertext blocks are *n* bits in length. [ISO/IEC 10116:1997]

2.6
plaintext
unenciphered information. [ISO/IEC 9797-1:1999]

3 Symbols

n – plaintext/ciphertext bit length for a block cipher.

E_K – encryption function with key *K*.

D_K – decryption function with key *K*.

Nr – the number of rounds for the AES algorithm, which is 10, 12 or 14 for the choices of key length 128, 192 or 256 bits respectively.

Nk – the number of 32-bit words comprising a key for the AES algorithm, which is 4, 6 or 8 for the choices of key length 128, 192 or 256 bits respectively.

$\oplus$ – the bit-wise logical exclusive-OR operation on bit-strings, i.e., if *A*, *B* are strings of the same length then $A \oplus B$ is the string equal to the bit-wise logical exclusive-OR of *A* and *B*.

$\wedge$ –the bit-wise logical AND operation on bit-strings, i.e., if *A*, *B* are strings of the same length then $A \wedge B$ is the string equal to the bit-wise logical AND of *A* and *B*.

$\vee$ – the bit-wise logical OR operation on bit-strings, i.e., if *A, B* are strings of the same length then $A \vee B$ is the string equal to the bit-wise logical OR of *A* and *B*.

|| – concatenation of bit strings.

$\bullet$ – finite field multiplication.

$<<<_i$ – the left circular rotation of the operand by *i* bits.

$>>>_i$ – the right circular rotation of the operand by *i* bits.

$\overline{x}$ – the bitwise complement of *x*.

4 64-bit block ciphers

In this clause, three 64-bit block ciphers are specified; TDEA (or 'Triple DES') in clause 4.1, MISTY1 in clause 4.2 and CAST-128 in clause 4.3.

Users authorized to access data that has been enciphered must have the key that was used to encipher the data in order to decipher it. The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 128- (or 192-) bit key. Deciphering must be accomplished using the same key as for enciphering.

4.1 TDEA

The Triple Data Encryption Algorithm (TDEA) is a symmetric cipher that can process data blocks of 64 bits, using cipher keys with length of 128 (or 192) bits, of which 112 (or 168) bits can be chosen arbitrarily, and the rest may be used for error detection. The TDEA is commonly known as Triple DES (Data Encryption Standard).

A TDEA encryption/decryption operation is a compound operation of DES encryption and decryption operations, where the DES algorithm is specified in Annex A. A TDEA key consists of three DES keys.

4.1.1 TDEA encryption/decryption

The TDEA is defined in terms of DES operations, where E_K is the DES encryption operation for the key K and D_K is the DES decryption operation for the key K.

4.1.1.1 TDEA encryption

The transformation of a 64-bit block P into a 64-bit block C is defined as follows:

$$C = E_{K_3}(D_{K_2}(E_{K_1}(P))).$$

4.1.1.2 TDEA decryption

The transformation of a 64-bit block C into a 64-bit block P is defined as follows:

$$P = D_{K_1}(E_{K_2}(D_{K_3}(C))).$$

4.1.2 TDEA keying options [2]

This part of ISO/IEC 18033 specifies the following keying options for TDEA. The TDEA key comprises the triple (K_1, K_2, K_3).

1. Keying Option 1: K_1, K_2 and K_3 are different DES keys;

2. Keying Option 2: K_1 and K_2 are different DES keys and $K_3 = K_1$.

NOTE – The option that $K_1 = K_2 = K_3$, the single-DES equivalent, is not recommended. Furthermore, the use of keying option 1 is preferred over keying option 2 since it provides additional security at the same performance level.

4.2 MISTY1

The MISTY1 algorithm is a symmetric block cipher that can process data blocks of 64 bits, using a cipher key with length of 128 bits.

4.2.1 MISTY1 encryption

The encryption operation is as shown in Figure 1. The transformation of a 64-bit block P into a 64-bit block C is defined as follows (*KL*, *KO* and *KI* are keys):

[2] The Keying Option 2 is approved only through the year 2009 by NIST.

(1) $P = L_0 \,||\, R_0$

$KL = KL_1 \,||\, KL_2 \,||\, \ldots \,||\, KL_{10}$

$KO = KO_1 \,||\, KO_2 \,||\, \ldots \,||\, KO_8$

$KI = KI_1 \,||\, KI_2 \,||\, \ldots \,||\, KI_8$

(2) for i = 1, 3, …, 7 (increment in steps of 2 because the loop body consists of two rounds):

$R_i = \mathrm{FL}(L_{i-1}, KL_i)$

$L_i = \mathrm{FL}(R_{i-1}, KL_{i+1}) \oplus \mathrm{FO}(R_i, KO_i, KI_i)$

$L_{i+1} = R_i \oplus \mathrm{FO}(L_i, KO_{i+1}, KI_{i+1})$

$R_{i+1} = L_i$

for i = 9 :

$R_i = \mathrm{FL}(L_{i-1}, KL_i)$

$L_i = \mathrm{FL}(R_{i-1}, KL_{i+1})$

(3) $C = L_9 \,||\, R_9$

4.2.2 MISTY1 decryption

The decryption operation is as shown in Figure 2, and is identical in operation to encryption apart from the following two modifications.

(1) All FL functions are replaced by their inverse functions FL^{-1}.

(2) The order in which the subkeys are applied is reversed.

4.2.3 MISTY1 functions

The MISTY1 algorithm uses a number of functions, namely S_7, S_9, FI, FO, FL and FL^{-1}, which are now defined.

4.2.3.1 Function FL

The FL function is used in encryption only and is shown in Figure 3. The FL function is defined as follows (X and Y are data, KL is a key):

(1) $X_{32} = X_L \,||\, X_R$, $KL_i = KL_{iL} \,||\, KL_{iR}$

(2) $Y_R = (X_L \wedge KL_{iL}) \oplus X_R$

(3) $Y_L = X_L \oplus (Y_R \vee KL_{iR})$

(4) $Y_{32} = Y_L \,||\, Y_R$

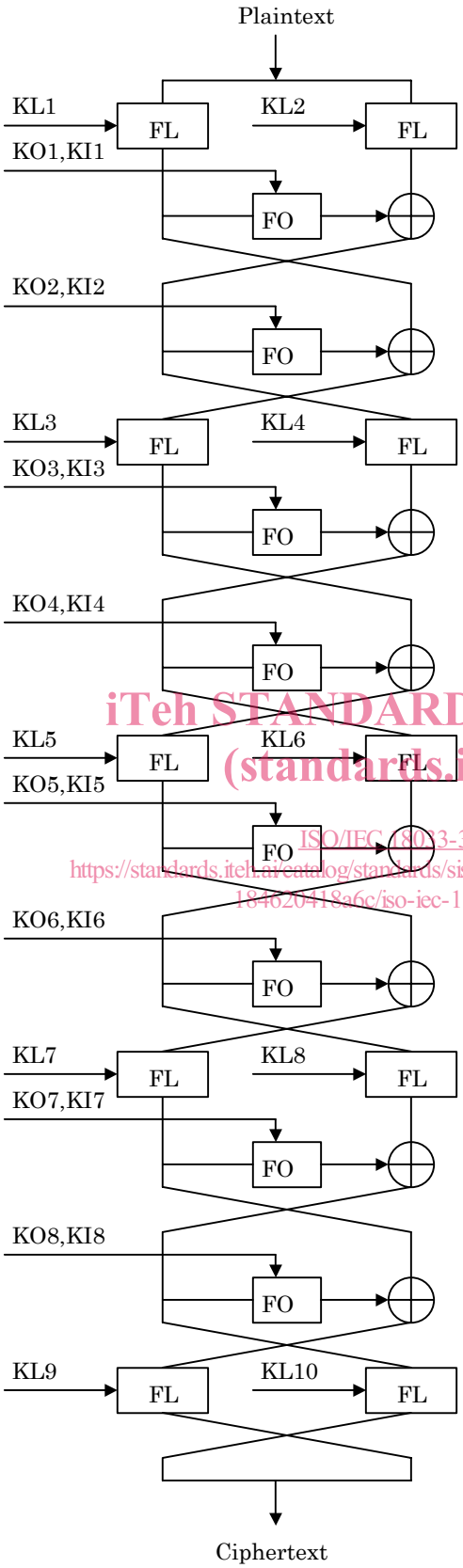

Figure 1. The Encryption Procedure

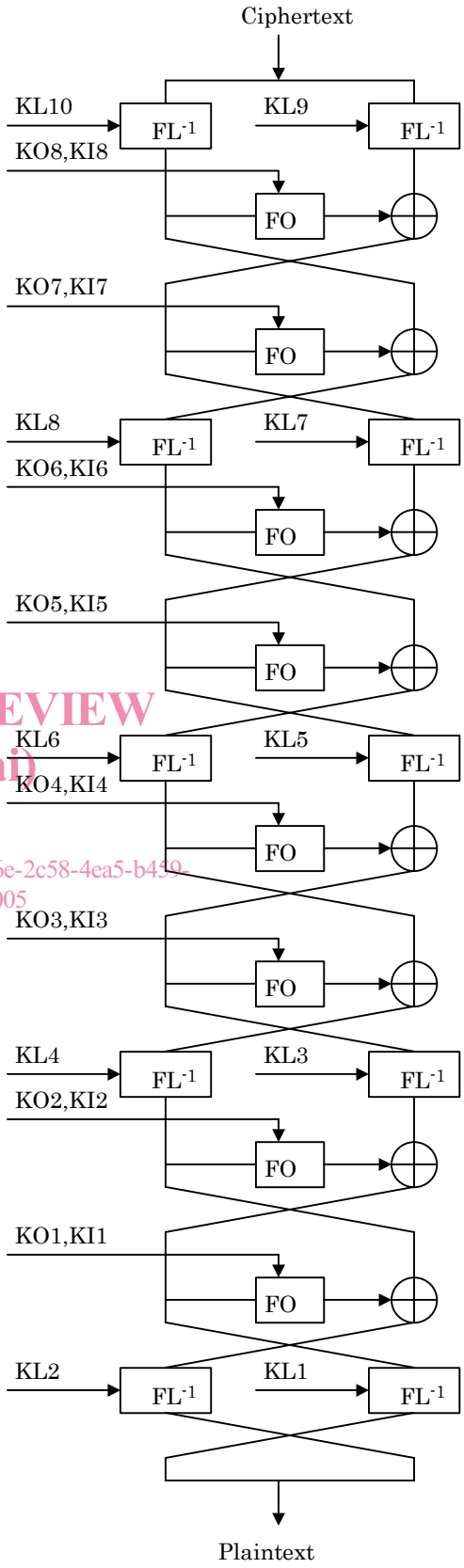

Figure 2. The Decryption Procedure

4.2.3.2 Function FL^{-1}

The FL^{-1} function, which is the inverse to the FL function, is used in decryption only and is shown in Figure 4. The FL^{-1} function is defined as follows (X and Y are data, KL is a key):

(1) $Y_{32} = Y_L \,||\, Y_R$, $KL_i = KL_{iL} \,||\, KL_{iR}$

(2) $X_L = Y_L \oplus (Y_R \vee KL_{iR})$

(3) $X_R = (X_L \wedge KL_{iL}) \oplus Y_R$

(4) $X_{32} = X_L \,||\, X_R$

4.2.3.3 Function FO

The FO function is used in encryption and decryption, and is shown in Figure 5. The FO function is defined as follows (X and Y are data, KO and KI are keys):

(1) $X_{32} = L_0 \,||\, R_0$

$KO_i = KO_{i1} \,||\, KO_{i2} \,||\, KO_{i3} \,||\, KO_{i4}$, $KI_i = KI_{i1} \,||\, KI_{i2} \,||\, KI_{i3}$

(2) for j = 1 to 3 :

$R_j = \mathrm{FI}(L_{j-1} \oplus KO_{ij}, KI_{ij}) \oplus R_{j-1}$

$L_j = R_{j-1}$

(3) $Y_{32} = (L_3 \oplus KO_{i4}) \,||\, R_3$

4.2.3.4 Function FI

The FI function is used for encryption, decryption and the key schedule, and is shown in Figure 6, where Extnd is the operation zero-extended from 7 bits to 9 bits by the concatenation of two bits on the left side, and Trunc is the operation truncated by two bits on the left side. The FI function is defined as follows (X and Y are data, KI is a key):

(1) $X_{16} = L_0$ (9 bits) $\,||\, R_0$ (7 bits), $KI_{ij} = KI_{ijL} \,||\, KI_{ijR}$

(2) $R_1 = S_9(L_0) \oplus \mathrm{Extnd}(R_0)$

(3) $L_1 = R_0$

(4) $R_2 = S_7(L_1) \oplus \mathrm{Trunc}(R_1) \oplus KI_{ijL}$

(5) $L_2 = R_1 \oplus KI_{ijR}$

(4) $R_3 = S_9(L_2) \oplus \mathrm{Extnd}(R_2)$

(5) $L_3 = R_2$

(6) $Y_{16} = L_3 \,||\, R_3$

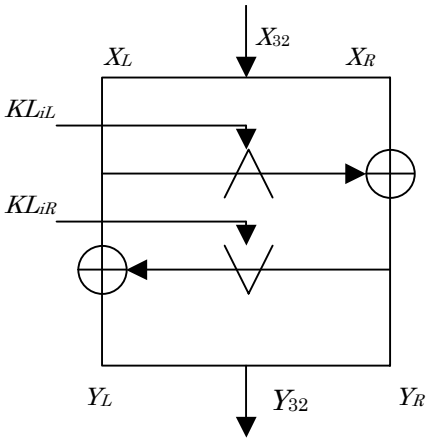

Figure 3. The Function FL

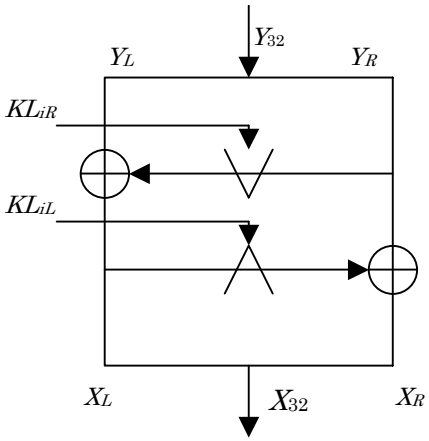

Figure 4. The Function FL^{-1}

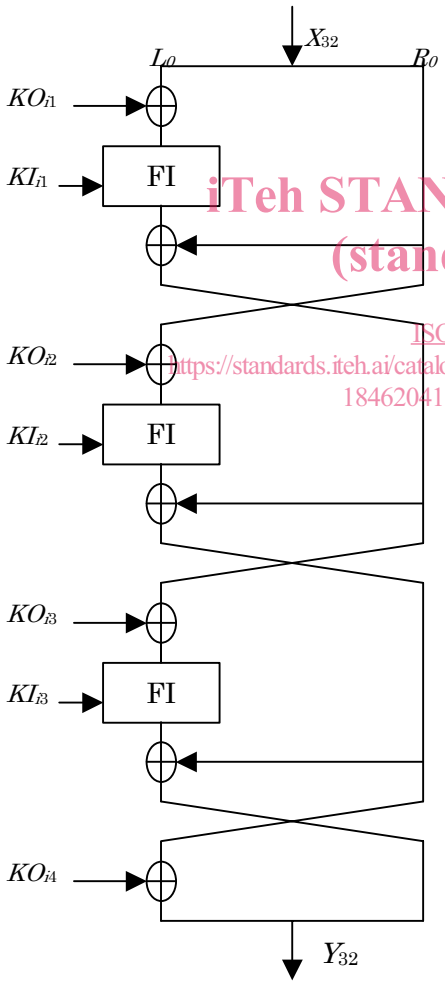

Figure 5. The Function FO

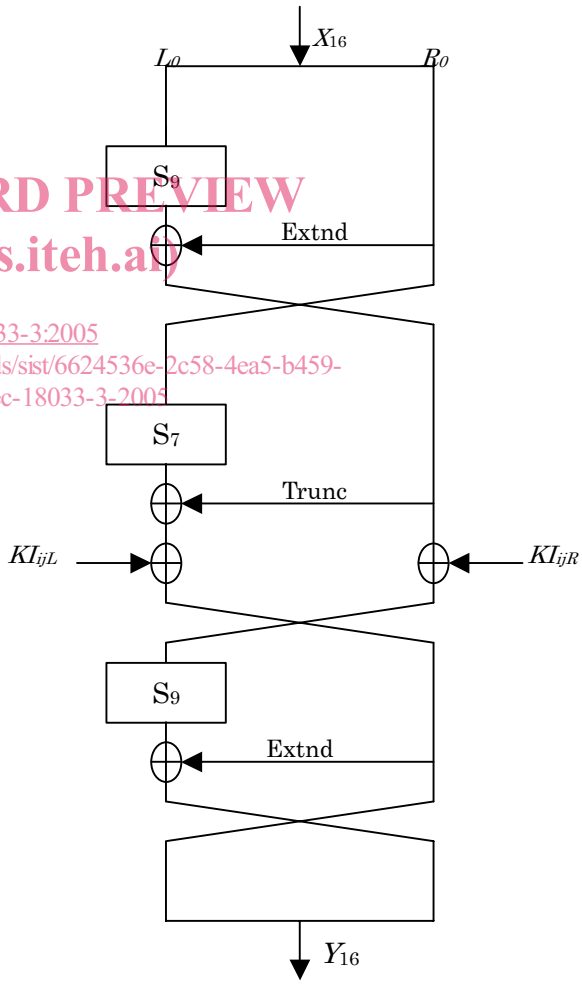

Figure 6. The Function FI

4.2.3.5 Lookup Tables S_7 and S_9

S_7 is a bijective lookup table that accepts a 7-bit input and yields a 7-bit output. S_9 is a bijective lookup table that accepts a 9-bit input and yields a 9-bit output. Tables 2 and 3 define these lookup tables in a hexadecimal form. S_7 and S_9 can be also described in a simple algebraic form over GF(2) as shown in Clause C.1.

For example, if the input to S_7 is {53}, then the substitution value would be determined by the intersection of the row with index '5' and the column with index '3' in Table 2. This would result in S_7 having a value of {57}.

Table 2. S_7

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	1b	32	33	5a	3b	10	17	54	5b	1a	72	73	6b	2c	66	49
1	1f	24	13	6c	37	2e	3f	4a	5d	0f	40	56	25	51	1c	04
2	0b	46	20	0d	7b	35	44	42	2b	1e	41	14	4b	79	15	6f
3	0e	55	09	36	74	0c	67	53	28	0a	7e	38	02	07	60	29
4	19	12	65	2f	30	39	08	68	5f	78	2a	4c	64	45	75	3d
5	59	48	03	57	7c	4f	62	3c	1d	21	5e	27	6a	70	4d	3a
6	01	6d	6e	63	18	77	23	05	26	76	00	31	2d	7a	7f	61
7	50	22	11	06	47	16	52	4e	71	3e	69	43	34	5c	58	7d

Table 3. S_9

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	1c3	0cb	153	19f	1e3	0e9	0fb	035	181	0b9	117	1eb	133	009	02d	0d3
01	0c7	14a	037	07e	0eb	164	193	1d8	0a3	11e	055	02c	01d	1a2	163	118
02	14b	152	1d2	00f	02b	030	13a	0e5	111	138	18e	063	0e3	0c8	1f4	01b
03	001	09d	0f8	1a0	16d	1f3	01c	146	07d	0d1	082	1ea	183	12d	0f4	19e
04	1d3	0dd	1e2	128	1e0	0ec	059	091	011	12f	026	0dc	0b0	18c	10f	1f7
05	0e7	16c	0b6	0f9	0d8	151	101	14c	103	0b8	154	12b	1ae	017	071	00c
06	047	058	07f	1a4	134	129	084	15d	19d	1b2	1a3	048	07c	051	1ca	023
07	13d	1a7	165	03b	042	0da	192	0ce	0c1	06b	09f	1f1	12c	184	0fa	196
08	1e1	169	17d	031	180	10a	094	1da	186	13e	11c	060	175	1cf	067	119
09	065	068	099	150	008	007	17c	0b7	024	019	0de	127	0db	0e4	1a9	052
0a	109	090	19c	1c1	028	1b3	135	16a	176	0df	1e5	188	0c5	16e	1de	1b1
0b	0c3	1df	036	0ee	1ee	0f0	093	049	09a	1b6	069	081	125	00b	05e	0b4
0c	149	1c7	174	03e	13b	1b7	08e	1c6	0ae	010	095	1ef	04e	0f2	1fd	085
0d	0fd	0f6	0a0	16f	083	08a	156	09b	13c	107	167	098	1d0	1e9	003	1fe
0e	0bd	122	089	0d2	18f	012	033	06a	142	0ed	170	11b	0e2	14f	158	131
0f	147	05d	113	1cd	079	161	1a5	179	09e	1b4	0cc	022	132	01a	0e8	004
10	187	1ed	197	039	1bf	1d7	027	18b	0c6	09c	0d0	14e	06c	034	1f2	06e
11	0ca	025	0ba	191	0fe	013	106	02f	1ad	172	1db	0c0	10b	1d6	0f5	1ec
12	10d	076	114	1ab	075	10c	1e4	159	054	11f	04b	0c4	1be	0f7	029	0a4
13	00e	1f0	077	04d	17a	086	08b	0b3	171	0bf	10e	104	097	15b	160	168
14	0d7	0bb	066	1ce	0fc	092	1c5	06f	016	04a	0a1	139	0af	0f1	190	00a
15	1aa	143	17b	056	18d	166	0d4	1fb	14d	194	19a	087	1f8	123	0a7	1b8

16	141	03c	1f9	140	02a	155	11a	1a1	198	0d5	126	1af	061	12e	157	1dc
17	072	18a	0aa	096	115	0ef	045	07b	08d	145	053	05f	178	0b2	02e	020
18	1d5	03f	1c9	1e7	1ac	044	038	014	0b1	16b	0ab	0b5	05a	182	1c8	1d4
19	018	177	064	0cf	06d	100	199	130	15a	005	120	1bb	1bd	0e0	04f	0d6
1a	13f	1c4	12a	015	006	0ff	19b	0a6	043	088	050	15f	1e8	121	073	17e
1b	0bc	0c2	0c9	173	189	1f5	074	1cc	1e6	1a8	195	01f	041	00d	1ba	032
1c	03d	1d1	080	0a8	057	1b9	162	148	0d9	105	062	07a	021	1ff	112	108
1d	1c0	0a9	11d	1b0	1a6	0cd	0f3	05c	102	05b	1d9	144	1f6	0ad	0a5	03a
1e	1cb	136	17f	046	0e1	01e	1dd	0e6	137	1fa	185	08c	08f	040	1b5	0be
1f	078	000	0ac	110	15e	124	002	1bc	0a2	0ea	070	1fc	116	15c	04c	1c2

4.2.4 MISTY1 key schedule

The key scheduling part accepts a 128-bit key K and yields another 128-bit subkey K', as shown below. The figure of the key scheduling part is described in Figure 7.

The key scheduling operation is thus defined as follows.

(1) $K = K_1 \| K_2 \| K_3 \| K_4 \| K_5 \| K_6 \| K_7 \| K_8$

(2) for i = 1 to 7 :

$K'_i = \mathrm{FI}(K_i, K_{i+1})$

(3) $K'_8 = \mathrm{FI}(K_8, K_1)$

(4) $K' = K'_1 \| K'_2 \| K'_3 \| K'_4 \| K'_5 \| K'_6 \| K'_7 \| K'_8$

(5) $KO_{i1} = K_i$, $KO_{i2} = K_{i+2}$, $KO_{i3} = K_{i+7}$, $KO_{i4} = K_{i+4}$, $KI_{i1} = K'_{i+5}$, $KI_{i2} = K'_{i+1}$, $KI_{i3} = K'_{i+3}$,

$KL_{iL} = K_{(i+1)/2}$ (odd i) or $K_{(i/2)+2}$ (even i), $KL_{iR} = K'_{(i+1)/2+6}$ (odd i) or $K'_{(i/2)+4}$ (even i)

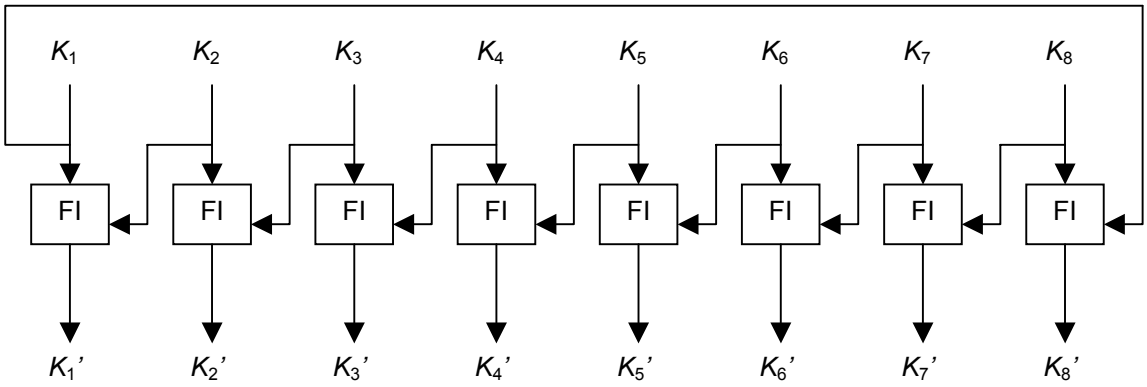

Figure 7. MISTY1 Key Scheduling