Specifications in 2001 MISTY1

(updated : May 13, 2002)

September 27, 2001
Mitsubishi Electric Corporation

Block Cipher Algorithm MISTY1

This document shows a complete description of encryption algorithm MISTY1, which are secret-key cipher with 64-bit data block and 128-bit secret key. The number of rounds *n* of MISTY1 is variable under the condition that *n* is a multiple of four. Recommendation value is *n* = 8. In the following description, it is defined that the most left bit is Most Significant Bit (MSB), the most right bit is Least Significant Bit (LSB).

Data Randomizing Part

- Figure 1a and 1b show the data randomizing part of MISTY1 for encryption and decryption, respectively. The plaintext/ciphertext is divided into two 32-bit data, which are transformed by bitwise XOR operations denoted by the symbol $\oplus$ and sub-functions $FO_i(1 \le i \le n)$, $FL_i(1 \le i \le n+2)$ and $FL_i^{-1}(1 \le i \le n+2)$. FO_i uses a 64-bit subkey KO_i and a 48-bit subkey KI_i. FL_i and FL_i^{-1} are used in encryption and decryption, respectively, both of which use a 32-bit subkey KL_i.
- Figure 2 shows the structure of FO_i. The input is divided into two 16-bit data, which are transformed by bitwise XOR operations denoted by the symbol $\oplus$ and sub-functions $FI_{ij}(1 \le j \le 3)$, where $KO_{ij}(1 \le j \le 4)$ and $KI_{ij}(1 \le j \le 3)$ are the *j*-th (from left) 16-bit data of KO_i and KI_i , respectively.
- Figure 3 shows the structure of FI_i. The input is divided into left 9-bit data and right 7-bit data, which are transformed by bitwise XOR operations denoted by the symbol $\oplus$ and substitution tables S_7 and S_9. In the first and third XORs, the 7-bit data is zero-extended to 9 bits, and on the second XOR, the 9-bit data is truncated to 7 bits by discarding its highest two bits. KI_{ij1} and KI_{ij2} are left 7-bit data and right 9-bit data of KI_{ij} , respectively.
- Figure 4a and 4b show the structure of FL_i and FL_i^{-1}, respectively. The input is divided into two 16-bit data, which are transformed by bitwise XOR operations denoted by the symbol $\oplus$, a bitwise AND operation denoted by the symbol $\cap$ and a bitwise OR operation denoted by the symbol $\cup$, where $KL_{ij}(1 \le j \le 2)$ is the *j*-th (from left) 16-bit data of KL_i.
- Tables 1 and 2 show decimal representation of the substitution tables S_7 and S_9, respectively.

Key Scheduling Part

- Figure 5 shows the Key Scheduling part of MISTY1. Let $K_i(1 \le i \le 8)$ be the *i*-th (from left) 16-bit data of the secret key K, and let $K_i^{'}(1 \le i \le 8)$ be the output of FI_{ij} where the input of FI_{ij} is K_i and the key KI_{ij} is K_{i+1}. Also, identify K_9 with K_1.
- The correspondence between the symbols KO_{ij}, KI_{ij}, KL_{ij} and the actual key is as follows:

Symbol	KO_{i1}	KO_{i2}	KO_{i3}	KO_{i4}	KI_{i1}	KI_{i2}	KI_{i3}	KL_{i1}	KL_{i2}
Key	K_i	K_{i+2}	K_{i+7}	K_{i+4}	$K_{i+5}^{'}$	$K_{i+1}^{'}$	$K_{i+3}^{'}$	$K_{\frac{i+1}{2}}$ (*odd i*) $K_{\frac{i}{2}+2}^{'}$ (*even i*)	$K_{\frac{i+1}{2}+6}^{'}$ (*odd i*) $K_{\frac{i}{2}+4}$ (*even i*)

Where K_i and $K_i^{'}$ are identified with K_{i-8} and $K_{i-8}^{'}$, respectively, when i exceeds 8.

Test Data

- The following is sample data for MISTY1 with eight rounds in hexadecimal form:

Secret Key $(K_1 \cdots K_8)$	00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
Plaintext	01 23 45 67 89 ab cd ef
Extended Key $(K_1^{'} \cdots K_8^{'})$	cf 51 8e 7f 5e 29 67 3a cd bc 07 d6 bf 35 5e 11
Ciphertext	8b 1d a5 f5 6a b3 d0 7c

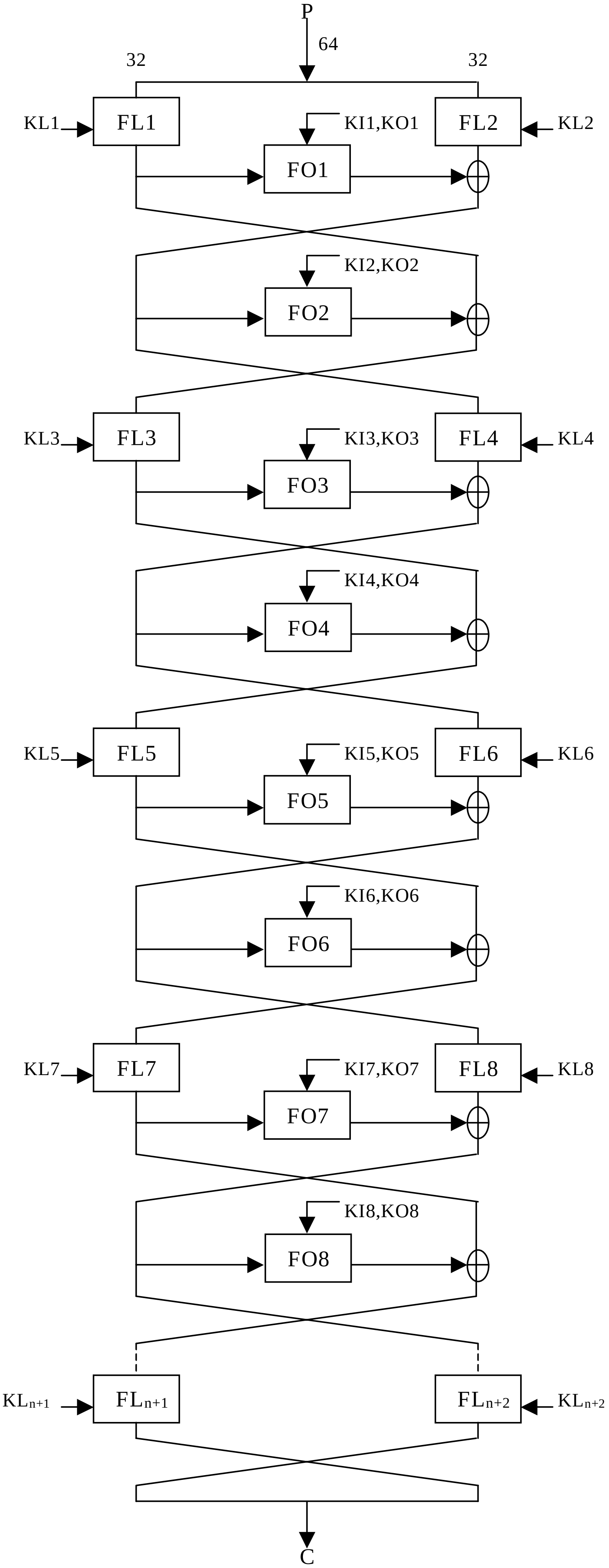


Figure 1a. MISTY1 (Encryption)

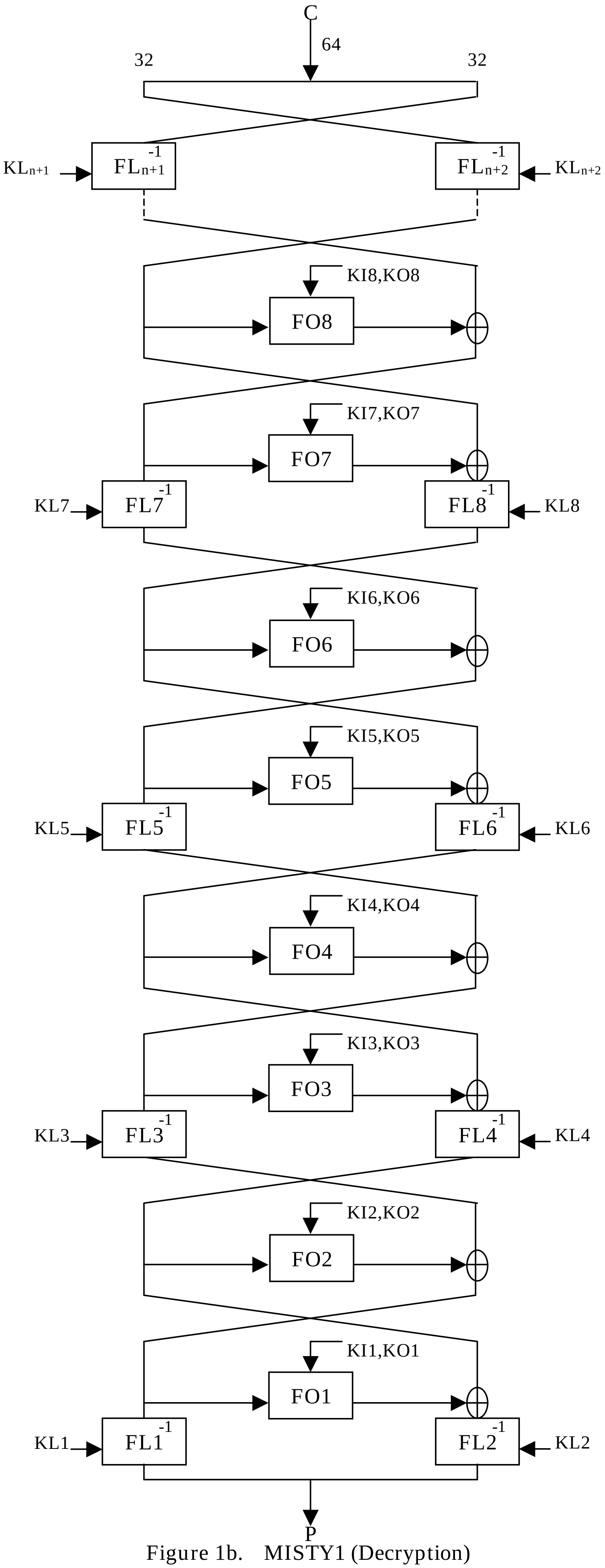


Figure 1b. MISTY1 (Decryption)

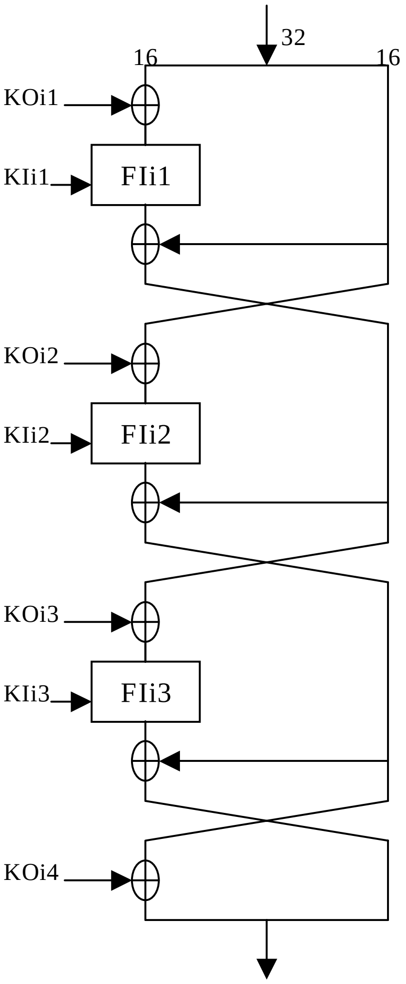


Figure 2. FOi

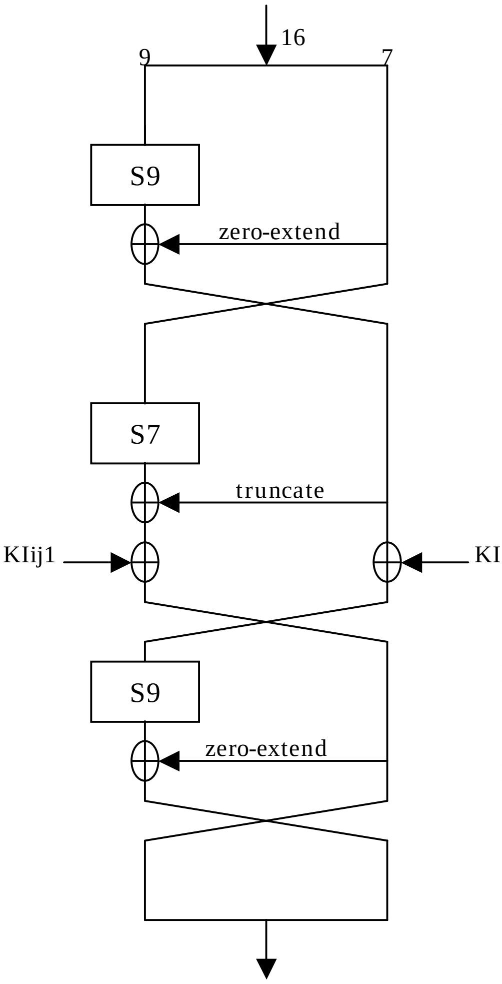


Figure 3. FIij

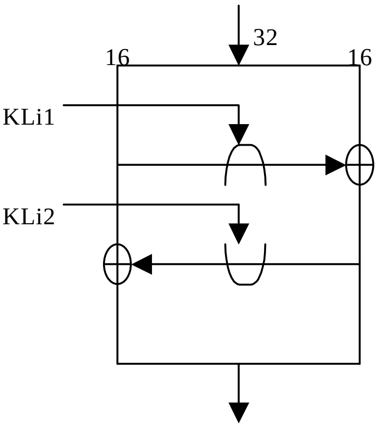


Figure 4a. FLi

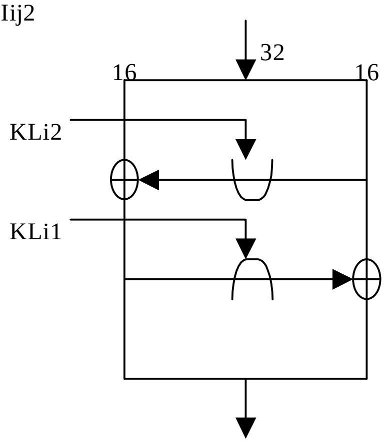


Figure 4b. Fli^{-1}

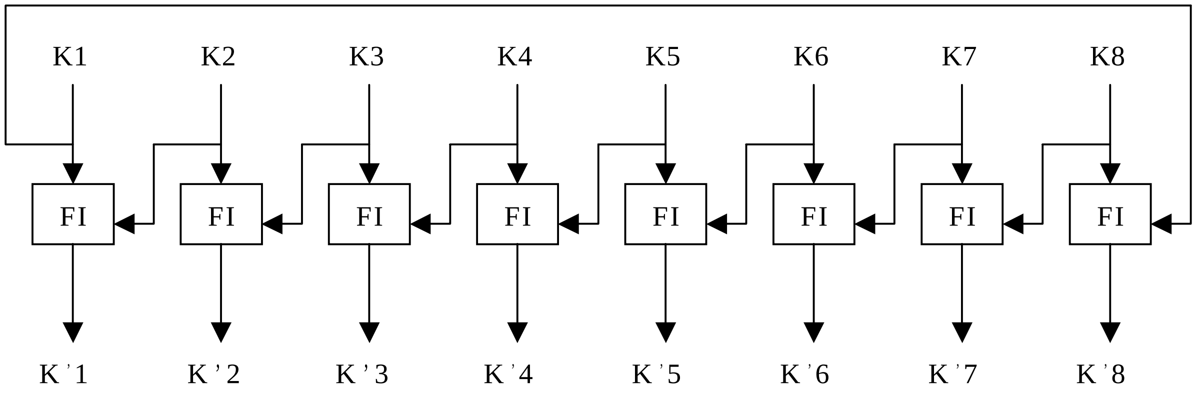


Figure 5. Key Scheduling

```
27, 50, 51, 90, 59, 16, 23, 84, 91, 26,114,115,107, 44,102, 73,
31, 36, 19,108, 55, 46, 63, 74, 93, 15, 64, 86, 37, 81, 28,  4,
11, 70, 32, 13,123, 53, 68, 66, 43, 30, 65, 20, 75,121, 21,111,
14, 85,  9, 54,116, 12,103, 83, 40, 10,126, 56,  2,  7, 96, 41,
25, 18,101, 47, 48, 57,  8,104, 95,120, 42, 76,100, 69,117, 61,
89, 72,  3, 87,124, 79, 98, 60, 29, 33, 94, 39,106,112, 77, 58,
 1,109,110, 99, 24,119, 35,  5, 38,118,  0, 49, 45,122,127, 97,
80, 34, 17,  6, 71, 22, 82, 78,113, 62,105, 67, 52, 92, 88,125
```

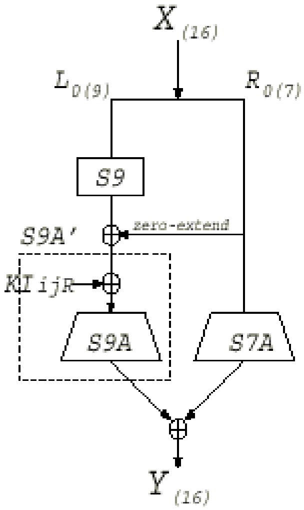
Table 1. The table of S_7

```
451,203,339,415,483,233,251, 53,385,185,279,491,307,  9, 45,211,
199,330, 55,126,235,356,403,472,163,286, 85, 44, 29,418,355,280,
331,338,466, 15, 43, 48,314,229,273,312,398, 99,227,200,500, 27,
  1,157,248,416,365,499, 28,326,125,209,130,490,387,301,244,414,
467,221,482,296,480,236, 89,145, 17,303, 38,220,176,396,271,503,
231,364,182,249,216,337,257,332,259,184,340,299,430, 23,113, 12,
 71, 88,127,420,308,297,132,349,413,434,419, 72,124, 81,458, 35,
317,423,357, 59, 66,218,402,206,193,107,159,497,300,388,250,406,
481,361,381, 49,384,266,148,474,390,318,284, 96,373,463,103,281,
101,104,153,336,  8,  7,380,183, 36, 25,222,295,219,228,425, 82,
265,144,412,449, 40,435,309,362,374,223,485,392,197,366,478,433,
195,479, 54,238,494,240,147, 73,154,438,105,129,293, 11, 94,180,
329,455,372, 62,315,439,142,454,174, 16,149,495, 78,242,509,133,
253,246,160,367,131,138,342,155,316,263,359,152,464,489,  3,510,
189,290,137,210,399, 18, 51,106,322,237,368,283,226,335,344,305,
327, 93,275,461,121,353,421,377,158,436,204, 34,306, 26,232,  4,
391,493,407, 57,447,471, 39,395,198,156,208,334,108, 52,498,110,
202, 37,186,401,254, 19,262, 47,429,370,475,192,267,470,245,492,
269,118,276,427,117,268,484,345, 84,287, 75,196,446,247, 41,164,
 14,496,119, 77,378,134,139,179,369,191,270,260,151,347,352,360,
215,187,102,462,252,146,453,111, 22, 74,161,313,175,241,400, 10,
426,323,379, 86,397,358,212,507,333,404,410,135,504,291,167,440,
321, 60,505,320, 42,341,282,417,408,213,294,431, 97,302,343,476,
114,394,170,150,277,239, 69,123,141,325, 83, 95,376,178, 46, 32,
469, 63,457,487,428, 68, 56, 20,177,363,171,181, 90,386,456,468,
 24,375,100,207,109,256,409,304,346,  5,288,443,445,224, 79,214,
319,452,298, 21,  6,255,411,166, 67,136, 80,351,488,289,115,382,
188,194,201,371,393,501,116,460,486,424,405, 31, 65, 13,442, 50,
 61,465,128,168, 87,441,354,328,217,261, 98,122, 33,511,274,264,
448,169,285,432,422,205,243, 92,258, 91,473,324,502,173,165, 58,
459,310,383, 70,225, 30,477,230,311,506,389,140,143, 64,437,190,
120,  0,172,272,350,292,  2,444,162,234,112,508,278,348, 76,450
```

Table 2. The table of S_9

Implementation Methods

The reference C source code attached to the submission package shows an example of straightforward implementation based on the specifications of MISTY1. There are however many more techniques for implementing MISTY1 in software balancing speed and memory requirements. The following shows how to speed up MISTY1 in software.



It is not difficult to see that the FI-function can be also written as shown in the left figure, where S9A and S9B are newly introduced look-up tables that transform 9 and 7 input bits into 16 output bits, respectively. Note that the subkey Kij1 can be "embedded" into other subkeys. Though implementing this form requires more memory than the straightforward method, but faster speed is expected.

Moreover, noting that the number of possible varieties of Kij2 is only eight, we can even remove the subkey by introducing eight different S9A tables (S9A') if a target processor has on-chip cache of 16K bytes or more.

Version Information

MISTY1 has been proposed in the following standardization activities, where the proposed specification is exactly the same as the specification described in this document.

ISO / SC27　NESSIE　IETF-TLS

Also, KASUMI, which was modified on the basis of MISTY1, has been adopted as the world standard of the forthcoming W-CDMA systems. KASUMI is a variant of MISTY1, but is not compatible with MISTY1.

Object Identifier

The object identifier of MISTY1 is described in RFC2994 "A Description of the MISTY1 Encryption Algorithm". The following is extracted from the RFC2994 document.

The Object Identifier for MISTY1 in Cipher Block Chaining (CBC) mode is as follows:

```
MISTY1-CBC OBJECT IDENTIFIER ::=
   {iso(1) member-body(2) jisc(392)
   mitsubishi-electric-corporation(200011) isl(61) security(1)
   algorithm(1) symmetric-encryption-algorithm(1) misty1-cbc(1)}
```

MISTY1-CBC needs Initialization Vector (IV) as like as other algorithms, such as DES-CBC, DES-EDE3-CBC and so on. To determine the value of IV, MISTY1-CBC takes parameter as:

```
MISTY1-CBC Parameter ::= IV
```

where IV ::= OCTET STRING -- 8 octets.

When this Object Identifier is used, plaintext is padded before encrypt it. At least 1 padding octet is appended at the end of the plaintext to make the length of the plaintext to the multiple of 8 octets. The value of these octets is as same as the number of appended octets. (e.g., If 5 octets are needed to pad, the value is 0x05.)

Applications and Products

MISTY1 has been used in various applications and products as follows; most of the information can be found at http://www.security.melco.co.jp/

[Software Products]
Encryption library <PowerMISTY>, PKI library <CertMISTY>, PKI server system <CERTMANAGER>, File encryption software <CRYPTOFILE>, Secure web access <TRUSTWEB>, Message encryption software <CRYPTOSIGN>, Digital contents secure distribution system <DIGICAPSULE>, Email security enhancement tool <MAHOUBIN-II> (by NTT Electronics), File encryption tool <SecureStaff> (by Mitsubishi Control Software)
[Hardware Products]
LAN encryption hardware <MELWALL>, Key management hardware <MISTYKEYPER> (by Mitsubishi Electric Engineering), Encryption LSI CDI2050 (by Cognitive Design, Inc), Encryption Algorithm IP for LSI development.