Specifications

128 ビットブロック暗号 CLEFIA 参照ソースコード仕様書

Version 1.0.0 (2010年1月29日)

ソニー株式会社

変更履歴

バージョン	日付	説明
1.0.0	2010年1月29日	初版作成

目次

1. 概要

本仕様書では，CLEFIA 参照ソースコードの仕様について述べる.

2. 表記

本仕様書で扱うデータ(平文，暗号文，秘密鍵)はMSB 側から順次バイト単位で
入出力されるものとする(big endian).

MSB LSB

bit number	31	30	29	28	27	26	25	24	23	…	16	15	…	8	7	…	0
byte number	0								1			2			3		

ex)

MSB LSB
data = 00010203 04050607 08090a0b 0c0d0e0f (hex)

```
unsigned char data[] = {
        0x00U, 0x01U, 0x02U, 0x03U, 0x04U, 0x05U, 0x06U, 0x07U,
        0x08U, 0x09U, 0x0aU, 0x0bU, 0x0cU, 0x0dU, 0x0eU, 0x0fU
};
```

また，本仕様書は以下の前提で記述されている.

•output のメモリ領域は全て呼び出し側が確保，解放するものとする.
（関数内では，メモリの確保，解放を行わず，全てその領域が確保されているものとして処理を行う）

3. CLEFIA 参照ソースコード API 仕様

3.1 CLEFIA 拡大鍵生成関数

```
int ClefiaKeySet(
        unsigned char *rk,          // output: 拡大鍵 (18/22/26 x 8 + 16 [bytes])
        const unsigned char *skey,  // input: 秘密鍵 (key_bitlen / 8 [bytes])
        const int key_bitlen        // input: 秘密鍵ビットサイズ (128, 192 or 256)
);
```

returns: 18 (key_bitlen=128), 22 (key_bitlen=192), 26 (key_bitlen=256)
or 0 (otherwise: invalid key_bitlen)
(補足: 処理ラウンド数 r を返す. 不正鍵サイズの場合は 0 を返す)

3.2 CLEFIA 暗号化関数

```
void ClefiaEncrypt(
        unsigned char *ct,          // output: 暗号文 (16 [bytes])
        const unsigned char *pt,    // input: 平文 (16 [bytes])
        const unsigned char *rk,    // input: 拡大鍵 (18/22/26 x 8 + 16 [bytes])*1
        const int r                 // input: 処理ラウンド数 (18, 22 or 26)*1
);
```

3.3 CLEFIA 復号関数

```
void ClefiaDecrypt(
        unsigned char *pt,          // output: 平文 (16 [bytes])
        const unsigned char *ct,    // input: 暗号文 (16 [bytes])
        const unsigned char *rk,    // input: 拡大鍵 (18/22/26 x 8 + 16 [bytes])*1
        const int r                 // input: 処理ラウンド数 (18, 22 or 26)*1
);
```

*1: CLEFIA 暗号化関数 ClefiaEncrypt(), および CLEFIA 復号関数 ClefiaDecrypt()
の入力 rk, r は CLEFIA 拡大鍵生成関数 ClefiaKeySet() より得られる rk, r を入力するものとし,
それら以外の値が入力された場合の動作は保証しない.
これら API の詳しい使用法については 4 章に記述する.

4. CLEFIA 参照ソースコード API 使用法

4.1 CLEFIA API 使用法

参照ソースコード(clefia_ref.c), 参照ソースコード用ヘッダファイル(clefia_ref.h) の基本的な使用法は以下の通り (必要な平文, 暗号文, および拡大鍵の領域は予め用意する).

1. 拡大鍵生成関数に秘密鍵長 (key_bitlen = 128, 192 or 256), および
   秘密鍵 (skey = 16, 24 or 32 [bytes]) を入力し, 拡大鍵 rk をセットアップする

2. 暗号化関数[復号関数]に 1-block (= 16-byte) の平文[暗号文], 1 で得られた拡大鍵,
   および処理ラウンド数を入力, 1-block (= 16-byte) 暗号文[平文]を得る

4.2 サンプルコード

```
/* 128-bit ブロック暗号 CLEFIA のサンプル C コード */
#include <clefia_ref.h>

int main(void)
{
  const unsigned char pt[16];
  const unsigned char ct[16];
  const unsigned char rk[26 * 8 + 16]; /* max */
  unsigned char *dst[16]; /* output */
  int r;

  /* CLEFIA-128 (128-bit 鍵 CLEFIA) を用いた暗号化/復号の例 */
  r = ClefiaKeySet(rk, skey, 128); /* key setup for CLEFIA-128 */
  ClefiaEncrypt(dst, pt, rk, r); /* CLEFIA-128 encryption */
  ClefiaDecrypt(dst, ct, rk, r); /* CLEFIA-128 decryption */

  /* CLEFIA-192 (192-bit 鍵 CLEFIA) を用いた暗号化/復号の例 */
  r = ClefiaKeySet(rk, skey, 192); /* key setup for CLEFIA-192 */
  ClefiaEncrypt(dst, pt, rk, r); /* CLEFIA-192 encryption */
  ClefiaDecrypt(dst, ct, rk, r); /* CLEFIA-192 decryption */

  /* CLEFIA-256 (256-bit 鍵 CLEFIA) を用いた暗号化/復号の例 */
  r = ClefiaKeySet(rk, skey, 256); /* key setup for CLEFIA-256 */
  ClefiaEncrypt(dst, pt, rk, r); /* CLEFIA-256 encryption */
  ClefiaDecrypt(dst, ct, rk, r); /* CLEFIA-256 decryption */

  return 0;
);
```