Specifications

128-bit blockcipher CLEFIA reference code specifications

Version 1.0.0 (January 29, 2010)

Sony Corporation

Change History

Version	Date	Description
1.0.0	January 29, 2010	Created

Contents

1. Overview

This document describes how to use a CLEFIA reference code.

2. Notation

The byte order used in this document is big-endian.
In other words, all of data including plaintexts, ciphertexts, secret keys and extended keys are assumed to input or output from MSB (most significant byte) to LSB (least significant byte).

MSB LSB

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

e.g.)

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

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

This document also assumes that all of memory areas for output are allocated and freed by user (caller).

3. CLEFIA reference code API specifications

3.1 CLEFIA extended key setup function

```
int ClefiaKeySet(
                unsigned char *rk,          // output: extended key (18/22/26 x 8 + 16 [bytes])
                const unsigned char *skey,  // input: secret key (key_bitlen / 8 [bytes])
                const int key_bitlen        // input: bit size of secret key (128, 192 or 256)
);
```

returns: 18 (key_bitlen=128), 22 (key_bitlen=192), 26 (key_bitlen=256)
or 0 (otherwise: invalid key_bitlen)
(This function returns the total number of rounds “r”, or 0 when the input key size is invalid.)

3.2 CLEFIA encryption function

```
void ClefiaEncrypt(
                unsigned char *ct,          // output: ciphertext (16 [bytes])
                const unsigned char *pt,    // input: plaintext (16 [bytes])
                const unsigned char *rk,    // input: extended key (18/22/26 x 8 + 16 [bytes])*1
                const int r                 // input: total number of rounds (18, 22 or 26)*1
);
```

3.3 CLEFIA decryption function

```
void ClefiaDecrypt(
                unsigned char *pt,          // output: plaintext (16 [bytes])
                const unsigned char *ct,    // input: ciphertext (16 [bytes])
                const unsigned char *rk,    // input: extended key (18/22/26 x 8 + 16 [bytes])*1
                const int r                 // input: total number of rounds (18, 22 or 26)*1
);
```

*1: Two inputs rk and r, which are inputs of the functions ClefiaEncrypt() and ClefiaDecrypt(), must be set by the function ClefiaKeySet().
The usage of these APIs is described in Section 4.

4. CLEFIA reference code API usage

4.1 CLEFIA API usage

The basic usage of the reference code (clefia_ref.c) and its header file (clefia_ref.h) is as follows.
(The required areas for a plaintext[ciphertext] and extended keys are assumed to be set in advance.)

1. Input a secret key size (key_bitlen = 128, 192 or 256) and a secret key (skey = 16, 24 or 32 [bytes]) to the extended key setup function ClefiaKeySet() to setup extended keys rk.

2. Input one-block plaintext[ciphertext], the extended keys and the total number of rounds which were calculated in Step 1 to the encryption[decryption] function ClefiaEncrypt()[ClefiaDecrypt()], then obtain one-block ciphertext[plaintext].

4.2 Sample code

```
/* sample C code for 128-bit blockcipher CLEFIA reference code */
#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 key CLEFIA) encryption/decryption */
  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 key CLEFIA) encryption/decryption */
  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 key CLEFIA) encryption/decryption */
  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;
);
```