libDES Documentation

Release 0.7.0

ANSSI

Oct 10, 2019

CONTENTS

Contents

The libdes project aims at implementing the DES (Data Encryption Standard) and TDES (Triple Data Encryption Standard) algorithms.

By now, these algorithms are only fully software based (i.e. they don't depend on any hardware cryptographic accelerator), but future work include adding an hardware acceleration when available.

CHAPTER

ONE

OVERVIEW

1.1 Principles

The implementation of the library follows the design principles described here:

https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf

The TDES implementation follows an EDE (Encrypt-Decrypt-Encrypt) design with two or three different keys.

1.2 Limitations

The libdes only implements the ECB mode. Future work includes adding other modes such as CBC and CTR. Also, the library does not handle padding: all the input and output data are supposed to be aligned on a DES block size, i.e. 8 bytes (64 bits).

CHAPTER

TWO

THE LIBDES API

2.1 DES data (de/en)cryption

Data encryption and decryption using DES algorithm is done using the following API:

```
#include "libdes.h"

int des_set_key(      des_context   *ctx,
                const unsigned char  k[8],
                      des_direction  dir);
int des_exec(const des_context    *ctx,
             const unsigned char   input[8],
                   unsigned char   output[8]);
```

Encrypting or decrypting data is done in two times:

- setting the DES key and the algorithm direction, using `des_set_key()`
- Encrypting or decrypting successive data chunks of 8 bytes, using `des_exec()`

The `des_context` structure contains the following fields:

- **dir**: the algorithm direction, DES_ENCRYPTION or DES_DECRYPTION
- **sk**: the DES subkeys after key schedule

These two fields are set by the `des_set_key()` function, based on the two other arguments:

- **k**: the DES key to use
- **dir**: the DES algorithm direction to use

The DES context can be keept by the caller task in order to use it during each successive DES execution through `des_exec()`.

Hint: It is possible to use multiple DES contexts for multiple cryptographic actions in the same time

Warning: When executing des_exec(), the input content must be padded to 8 bytes by the user

2.2 TDES data (de/en)cryption

Data encryption and decryption using Triple-DES algorithm is done using the following API

```
#include "libdes.h"

int des3_set_key(      des_context   *ctx,
                 const unsigned char  k1[8],
                 const unsigned char  k2[8],
                 const unsigned char  k3[8],
                       des_direction  dir);
int des3_exec(const des_context    *ctx,
              const unsigned char   input[8],
                    unsigned char   output[8]);
```

Encrypting or decrypting data is done in two times:

- setting the three DES keys and the algorithm direction, using `des3_set_key()`
- Encrypting or decrypting successive data chunk of 8 bytes, using `des3_exec()`

The TDES context is made of three DES contexts (see the DES documentation). The two TDES functions can be used in the same way the DES API is.