# 1. Header files

In order to use the cryptopp’s DES functions, in your C++ source program (e.g., test1.cpp), you need to include the right header files, and use the right namespace as follows.

#include "cryptopp/cryptlib.h"

#include "cryptopp/hex.h"

#include "cryptopp/filters.h"

#include "cryptopp/des.h"

#include "cryptopp/aes.h"

#include "cryptopp/modes.h"

using namespace CryptoPP;

# 2. Encrypt with DES

A text string can be encrypted with DES algorithm using the following tool function:

**string des\_encode(string & plain, byte key[])**

**{**

**string cipher;**

**try**

**{**

**cout << "plain text: " << plain << endl;**

**ECB\_Mode<DES>::Encryption enc;**

**enc.SetKey(key, sizeof(key));**

**StringSource(plain, true,**

**new StreamTransformationFilter(enc,**

**new StringSink(cipher))**

**); //add padding by StreamTransformationFilter**

**}**

**catch(const CryptoPP::Exception& e)**

**{**

**cerr << e.what() << endl;**

**exit(1);**

**}**

**return cipher;**

**}**

The input parameter "plain" is the plain text that you want to encrypt. The input parameter "key" is a byte array that stores the key you want to use during the DES encryption process.

The length of the key array is defined by constant DES::DEFAULT\_KEYLENGTH. The string returned by this function is the encryption output. Here we use DES block cipher in ECB (Electronic Code Book, this will become clear later) operation mode. The Input text are padded by StreamTransformationFilter tool.

# 3. Decryption with DES

Using the following tool function:

**string des\_decode(string & cipher, byte key[])**

**{**

**string plain;**

**//decode**

**try**

**{**

**ECB\_Mode< DES >::Decryption dec;**

**dec.SetKey(key, sizeof(key));**

**StringSource s(cipher, true,**

**new StreamTransformationFilter(dec,**

**new StringSink(plain))**

**);**

**cout << "recovered text: " << plain<< endl;**

**}**

**catch(const CryptoPP::Exception& e)**

**{**

**cerr << e.what() << endl;**

**exit(1);**

**}**

**return plain;**

**}**

The input parameter cipher is the ciphertext string. The output is the recovered (decrypted) plaintext.

# 4. How to use example code.

## 1>. How to compile your source program:

cryptog++  <sourcefile.cpp>  -lcryptopp -o des\_encode

-lcryptopp : link CryptoPP library.

3. How to execute your program:

cryptoexec ./ des\_encode

## 2> Encrypt a plain text file:

Command format:

cryptoexec ./des\_encode [input file] [output file]

for example:

cryptoexec ./des\_encode 01\_plain 01\_enc

the output should be:

**key:786D666E6833717A**

**plain text: High above the city, on a tall column, stood the statue of the Happy Prince. He was gilded all over with thin leaves of fine gold, for eyes he had two bright sapphires, and a large red ruby glowed on his sword-hilt.**

**cipher text stored in: 01\_enc**

## 3> Decrypt an encrypted file:

Command format:

cryptoexec ./des\_decode [input file] [output file]

for example:

cryptoexec ./des\_decode 01\_enc 01\_recover

the output should be:

**key:786D666E6833717A**

**recovered text: High above the city, on a tall column,stood the statue of the Happy Prince. He was gildedall over with thin leaves of fine gold, for eyes he had two bright sapphires, and a large red ruby glowed on his sword-hilt.**

**plain text stored in:01\_recover**

**4> compare the difference of two files:**

**diff 01\_plain 01\_recover**

The output should be empty when the two files are identical