

Programming Assignment 2 - BLOWFISH (64bit blocks and 128-bit keys)

Use the OpenSSL library (www.openssl.org) to write the following two functions for encryption and decryption in a file `fsencrypt.cc`.

You should use block cipher method blowfish for encryption. Blowfish uses 64-bit blocks and typically 128-bit keys.

```
// put the following lines in fsencrypt.h
#include "openssl/blowfish.h"

// encrypt plaintext of length bufsz. Use keystr as the key.
const int BLOCKSIZE = 8;           // Block size for blowfish
void *fs_encrypt(void *plaintext, int bufsz, char *keystr,
int *resultlen);

// decrypt ciphertext of length bufsz. Use keystr as the key.
void *fs_decrypt(void *ciphertext, int bufsz, char *keystr,
int *resultlen);
```

Both functions allocate the result buffer of at least the required size (using `new()`) and return a pointer to it. Both functions also return the number of valid bytes in the result buffer in `resultlen`.

The application code is responsible for deleting the buffer.

Use CBC mode of encryption. For padding, pad with length of the pad in all the padded characters.

Assume that the initialization vector contains NULL characters (all 0's).

Description of blowfish functions can be found at
<http://www.openssl.org/docs/crypto/blowfish.html>

Use the following functions to facilitate your work:

`BF_set_key`: use all characters of the `keystr`, excluding NULL terminator. Valid `keystr` is assumed to be a string.

`BF_cbc_encrypt` and `BF_ecb_encrypt`

You should use `BF_ecb_encrypt` to implement the CBC mode on your own.

However, you will get 15 bonus points if you submit an additional separate program, which uses only `BF_cbc_encrypt`.

You will need to include `"openssl/blowfish.h"` from the openssl package) and link with the `"crypto"` library.

Below is a small test code (`main.cc`).

You can compile it with your code in `fsencrypt.cc` using
`gcc (or g++) main.cc fsencrypt.cc -lcrypto`

Submit your `fsencrypt.cc`, which uses only `BF_ecb_encrypt`.
If you want to get bonus points, submit a different file `fsencrypt2.cc`, which contains only `BF_cbc_encrypt`.

```
=====

#include <assert.h>
#include <stdio.h>
#include <string.h>
#include "fscrypt.h"

int main()
{
    char s[] = "hello world";
    char *outbuf, *recvbuf;
    char pass[] = "top secret";
    int len = 0;
    int recvlen = 0;

    outbuf = (char *) fs_encrypt((void *) s, strlen(s)+1, pass, &len);
    printf("%s %d\n", "length after encryption = ", len);

    int i = 0;
    printf("ciphertext = ");
    for (i = 0; i < len; i++)
        printf("%02x", outbuf[i]);
    printf("\n");

    recvbuf = (char *) fs_decrypt((void *) outbuf, len, pass, &recvlen);
    assert(memcmp(s, recvbuf, recvlen) == 0);
    assert(recvlen == (strlen(s) + 1));
    printf("plaintext = %s\n", recvbuf);
}

=====
```