NAME

blowfish, BF_set_key, BF_encrypt, BF_decrypt, BF_ecb_encrypt, BF_cbc_encrypt, BF_cfb64_encrypt, BF_ofb64_encrypt, BF_options - Blowfish encryption

SYNOPSIS

```
#include <openssl/blowfish.h>

void BF_set_key(BF_KEY *key, int len, const unsigned char *data);

void BF_ecb_encrypt(const unsigned char *in, unsigned char *out,
BF_KEY *key, int enc);
void BF_cbc_encrypt(const unsigned char *in, unsigned char *out,
long length, BF_KEY *schedule, unsigned char *ivec, int enc);
void BF_cfb64_encrypt(const unsigned char *in, unsigned char *out,
long length, BF_KEY *schedule, unsigned char *ivec, int *num,
int enc);
void BF_ofb64_encrypt(const unsigned char *in, unsigned char *out,
long length, BF_KEY *schedule, unsigned char *ivec, int *num);
const char *BF_options(void);

void BF_encrypt(BF_LONG *data,const BF_KEY *key);
void BF_decrypt(BF_LONG *data,const BF_KEY *key);
```

DESCRIPTION

This library implements the Blowfish cipher, which was invented and described by Counterpane (see http://www.counterpane.com/blowfish.html).

Blowfish is a block cipher that operates on 64 bit (8 byte) blocks of data. It uses a variable size key, but typically, 128 bit (16 byte) keys are considered good for strong encryption. Blowfish can be used in the same modes as DES (see *des_modes(7)*). Blowfish is currently one of the faster block ciphers. It is quite a bit faster than DES, and much faster than IDEA or RC2.

Blowfish consists of a key setup phase and the actual encryption or decryption phase.

BF_set_key() sets up the **BF_KEY key** using the **len** bytes long key at **data**.

BF_ecb_encrypt() is the basic Blowfish encryption and decryption function. It encrypts or decrypts the first 64 bits of **in** using the key **key**, putting the result in **out**. **enc** decides if encryption (**BF_ENCRYPT**) or decryption (**BF_DECRYPT**) shall be performed. The vector pointed at by **in** and **out** must be 64 bits in length, no less. If they are larger, everything after the first 64 bits is ignored.

The mode functions *BF_cbc_encrypt()*, *BF_cfb64_encrypt()* and *BF_ofb64_encrypt()* all operate on variable length data. They all take an initialization vector **ivec** which needs to be passed along into the next call of the same function for the same message. **ivec** may be initialized with anything, but the recipient needs to know what it was initialized with, or it won't be able to decrypt. Some programs and protocols simplify this, like SSH, where **ivec** is simply initialized to zero. *BF_cbc_encrypt()* operates on data that is a multiple of 8 bytes long, while *BF_cfb64_encrypt()* and *BF_ofb64_encrypt()* are used to encrypt an variable number of bytes (the amount does not have to be an exact multiple of 8). The purpose of the latter two is to simulate stream ciphers, and therefore, they need the parameter **num**, which is a pointer to an integer where the current offset in **ivec** is stored between calls. This integer must be initialized to zero when **ivec** is initialized.

BF_cbc_encrypt() is the Cipher Block Chaining function for Blowfish. It encrypts or decrypts the 64 bits chunks of **in** using the key **schedule**, putting the result in **out**. **enc** decides if encryption (BF_ENCRYPT) or decryption (BF_DECRYPT) shall be performed. **ivec** must point at an 8 byte long initialization vector.

BF_cfb64_encrypt() is the CFB mode for Blowfish with 64 bit feedback. It encrypts or decrypts the bytes in **in** using the key **schedule**, putting the result in **out**. **enc** decides if encryption (**BF_ENCRYPT**) or decryption (**BF_DECRYPT**) shall be performed. **ivec** must point at an 8 byte long initialization vector. **num** must point at an integer which must be initially zero.

BF_ofb64_encrypt() is the OFB mode for Blowfish with 64 bit feedback. It uses the same parameters as *BF_cfb64_encrypt()*, which must be initialized the same way.

BF_encrypt() and *BF_decrypt()* are the lowest level functions for Blowfish encryption. They encrypt/decrypt the first 64 bits of the vector pointed by **data**, using the key **key**. These functions should not be used unless you implement 'modes' of Blowfish. The alternative is to use *BF_ecb_encrypt()*. If you still want to use these functions, you should be aware that they take each 32-bit chunk in host-byte order, which is little-endian on little-endian platforms and big-endian on big-endian ones.

RETURN VALUES

None of the functions presented here return any value.

NOTE

Applications should use the higher level functions *EVP_EncryptInit(3)* etc. instead of calling the blowfish functions directly.

SEE ALSO

des_modes(7)

HISTORY

The Blowfish functions are available in all versions of SSLeay and OpenSSL.