NAME

BIO_f_cipher, BIO_set_cipher, BIO_get_cipher_status, BIO_get_cipher_ctx - cipher BIO filter

SYNOPSIS

```
 #include <openssl/bio.h>
 #include <openssl/evp.h>

 const BIO_METHOD *BIO_f_cipher(void);
 int BIO_set_cipher(BIO *b, const EVP_CIPHER *cipher,
                    const unsigned char *key, const unsigned char *iv, int enc);
 int BIO_get_cipher_status(BIO *b);
 int BIO_get_cipher_ctx(BIO *b, EVP_CIPHER_CTX **pctx);
```

DESCRIPTION

BIO_f_cipher() returns the cipher BIO method. This is a filter BIO that encrypts any data written through it, and decrypts any data read from it. It is a BIO wrapper for the cipher routines **EVP_CipherInit()**, **EVP_CipherUpdate()** and **EVP_CipherFinal()**.

Cipher BIOs do not support **BIO_gets()** or **BIO_puts()**.

BIO_flush() on an encryption BIO that is being written through is used to signal that no more data is to be encrypted: this is used to flush and possibly pad the final block through the BIO.

BIO_set_cipher() sets the cipher of BIO **b** to **cipher** using key **key** and IV **iv**. **enc** should be set to 1 for encryption and zero for decryption.

When reading from an encryption BIO the final block is automatically decrypted and checked when EOF is detected. **BIO_get_cipher_status()** is a **BIO_ctrl()** macro which can be called to determine whether the decryption operation was successful.

BIO_get_cipher_ctx() is a **BIO_ctrl()** macro which retrieves the internal BIO cipher context. The retrieved context can be used in conjunction with the standard cipher routines to set it up. This is useful when **BIO_set_cipher()** is not flexible enough for the applications needs.

NOTES

When encrypting **BIO_flush()** **must** be called to flush the final block through the BIO. If it is not then the final block will fail a subsequent decrypt.

When decrypting an error on the final block is signaled by a zero return value from the read operation. A successful decrypt followed by EOF will also return zero for the final read. **BIO_get_cipher_status()**

should be called to determine if the decrypt was successful.

As always, if **BIO_gets()** or **BIO_puts()** support is needed then it can be achieved by preceding the cipher BIO with a buffering BIO.

RETURN VALUES

BIO_f_cipher() returns the cipher BIO method.

BIO_set_cipher() returns 1 for success and 0 for failure.

BIO_get_cipher_status() returns 1 for a successful decrypt and <=0 for failure.

BIO_get_cipher_ctx() returns 1 for success and <=0 for failure.

COPYRIGHT