

**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>

BIO_METHOD * BIO_f_cipher(void);
void BIO_set_cipher(BIO *b, const EVP_CIPHER *cipher,
    unsigned char *key, 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 signalled 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()* does not return a value.

*BIO\_get\_cipher\_status()* returns 1 for a successful decrypt and 0 for failure.

*BIO\_get\_cipher\_ctx()* currently always returns 1.

**EXAMPLES**

TBA

**SEE ALSO**

TBA