Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'EVP_aes_128_cbc_hmac_sha256.3ossl' command

$ man EVP_aes_128_cbc_hmac_sha256.3ossl

EVP_AES_128_GCM(3ossl) OpenSSL EVP_AES_128_GCM(3ossl)

NAME

EVP_aes_128_cbc, EVP_aes_192_cbc, EVP_aes_256_cbc, EVP_aes_128_cfb, EVP_aes_192_cfb, EVP_aes_256_cfb, EVP_aes_128_cfb1, EVP_aes_192_cfb1, EVP_aes_256_cfb1, EVP_aes_128_cfb8, EVP_aes_192_cfb8, EVP_aes_256_cfb8, EVP_aes_128_cfb128, EVP_aes_192_cfb128, EVP_aes_256_cfb128, EVP_aes_128_ctr, EVP_aes_192_ctr, EVP_aes_256_ctr, EVP_aes_128_ecb, EVP_aes_192_ecb, EVP_aes_256_ecb, EVP_aes_128_ofb, EVP_aes_192_ofb, EVP_aes_256_ofb, EVP_aes_128_cbc_hmac_sha1, EVP_aes_256_cbc_hmac_sha1, EVP_aes_128_cbc_hmac_sha256, EVP_aes_256_cbc_hmac_sha256, EVP_aes_128_ccm, EVP_aes_192_ccm, EVP_aes_256_ccm, EVP_aes_128_gcm, EVP_aes_192_gcm, EVP_aes_256_gcm, EVP_aes_128_ocb, EVP_aes_192_ocb, EVP_aes_256_ocb, EVP_aes_128_wrap, EVP_aes_192_wrap, EVP_aes_256_wrap, EVP_aes_128_wrap_pad, EVP_aes_192_wrap_pad, EVP_aes_256_wrap_pad, EVP_aes_128_xts, EVP_aes_256_xts - EVP AES cipher

SYNOPSIS

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

const EVP_CIPHER *EVP_ciphername(void)
```

EVP_ciphername is used a placeholder for any of the described cipher

functions, such as EVP_aes_128_cbc.

DESCRIPTION

The AES encryption algorithm for EVP.

EVP_aes_128_cbc(), EVP_aes_192_cbc(), EVP_aes_256_cbc(),
EVP_aes_128_cfb(), EVP_aes_192_cfb(), EVP_aes_256_cfb(),
EVP_aes_128_cfb1(), EVP_aes_192_cfb1(), EVP_aes_256_cfb1(),
EVP_aes_128_cfb8(), EVP_aes_192_cfb8(), EVP_aes_256_cfb8(),
EVP_aes_128_cfb128(), EVP_aes_192_cfb128(), EVP_aes_256_cfb128(),
EVP_aes_128_ctr(), EVP_aes_192_ctr(), EVP_aes_256_ctr(),
EVP_aes_128_ecb(), EVP_aes_192_ecb(), EVP_aes_256_ecb(),
EVP_aes_128_ofb(), EVP_aes_192_ofb(), EVP_aes_256_ofb()

AES for 128, 192 and 256 bit keys in the following modes: CBC, CFB with 128-bit shift, CFB with 1-bit shift, CFB with 8-bit shift, CTR, ECB, and OFB.

EVP_aes_128_cbc_hmac_sha1(), EVP_aes_256_cbc_hmac_sha1()

Authenticated encryption with AES in CBC mode using SHA-1 as HMAC, with keys of 128 and 256 bits length respectively. The authentication tag is 160 bits long.

WARNING: this is not intended for usage outside of TLS and requires calling of some undocumented ctrl functions. These ciphers do not conform to the EVP AEAD interface.

EVP_aes_128_cbc_hmac_sha256(), EVP_aes_256_cbc_hmac_sha256()

Authenticated encryption with AES in CBC mode using SHA256 (SHA-2, 256-bits) as HMAC, with keys of 128 and 256 bits length respectively. The authentication tag is 256 bits long.

WARNING: this is not intended for usage outside of TLS and requires calling of some undocumented ctrl functions. These ciphers do not

conform to the EVP AEAD interface.

EVP_aes_128_ccm(), EVP_aes_192_ccm(), EVP_aes_256_ccm(),
EVP_aes_128_gcm(), EVP_aes_192_gcm(), EVP_aes_256_gcm(),
EVP_aes_128_ocb(), EVP_aes_192_ocb(), EVP_aes_256_ocb()

AES for 128, 192 and 256 bit keys in CBC-MAC Mode (CCM), Galois Counter Mode (GCM) and OCB Mode respectively. These ciphers require additional control operations to function correctly, see the "AEAD Interface" in EVP_EncryptInit(3) section for details.

EVP_aes_128_wrap(), EVP_aes_192_wrap(), EVP_aes_256_wrap(),
EVP_aes_128_wrap_pad(), EVP_aes_128_wrap(), EVP_aes_192_wrap(),
EVP_aes_256_wrap(), EVP_aes_192_wrap_pad(), EVP_aes_128_wrap(),
EVP_aes_192_wrap(), EVP_aes_256_wrap(), EVP_aes_256_wrap_pad()

AES key wrap with 128, 192 and 256 bit keys, as according to RFC 3394 section 2.2.1 ("wrap") and RFC 5649 section 4.1 ("wrap with padding") respectively.

EVP_aes_128_xts(), EVP_aes_256_xts()

AES XTS mode (XTS-AES) is standardized in IEEE Std. 1619-2007 and described in NIST SP 800-38E. The XTS (XEX-based tweaked-codebook mode with ciphertext stealing) mode was designed by Prof. Phillip Rogaway of University of California, Davis, intended for encrypting data on a storage device.

XTS-AES provides confidentiality but not authentication of data. It also requires a key of double-length for protection of a certain key size. In particular, XTS-AES-128 (EVP_aes_128_xts) takes input of a 256-bit key to achieve AES 128-bit security, and XTS-AES-256 (EVP_aes_256_xts) takes input of a 512-bit key to achieve AES 256-bit security.

The XTS implementation in OpenSSL does not support streaming. That

is there must only be one EVP_EncryptUpdate(3) call per EVP_EncryptInit_ex(3) call (and similarly with the "Decrypt" functions).

The iv parameter to EVP_EncryptInit_ex(3) or EVP_DecryptInit_ex(3) is the XTS "tweak" value.

RETURN VALUES

These functions return an EVP_CIPHER structure that contains the implementation of the symmetric cipher. See EVP_CIPHER_meth_new(3) for details of the EVP_CIPHER structure.

SEE ALSO

evp(7), EVP_EncryptInit(3), EVP_CIPHER_meth_new(3)

COPYRIGHT