NAME

crypto - OpenCrypto algorithms

DESCRIPTION

The in-kernel OpenCrypto framework supports several different encryption and authentication algorithms. This document describes the parameters and requirements of these algorithms. Unless otherwise noted, all sizes listed below are in bytes.

Authenticators

Authenticators compute a value (also known as a digest, hash, or tag) over an input of bytes. In-kernel requests can either compute the value for a given input, or verify if a given tag matches the computed tag for a given input. The following authentication algorithms are supported:

Name	Nonce	Key Sizes	Digest	Description
CRYPTO_AES_CCM_CBC_MAC	12	16, 24, 32	16	Authentication-only mode of AES-CCM
CRYPTO_AES_NIST_GMAC	12	16, 24, 32	16	Galois message authentication code
CRYPTO_BLAKE2B		0, 64	64	Blake2b
CRYPTO_BLAKE2S		0, 32	32	Blake2s
CRYPTO_NULL_HMAC			12	IPsec NULL HMAC
CRYPTO_POLY1305		32	16	Poly1305 authenticator
CRYPTO_RIPEMD160			20	RIPE Message Digest-160
CRYPTO_RIPEMD160_HMAC		64	20	RIPE Message Digest-160 HMAC
CRYPTO_SHA1			20	SHA-1
CRYPTO_SHA1_HMAC		64	20	SHA-1 HMAC
CRYPTO_SHA2_224			28	SHA-2 224
CRYPTO_SHA2_224_HMAC		64	28	SHA-2 224 HMAC
CRYPTO_SHA2_256			32	SHA-2 256
CRYPTO_SHA2_256_HMAC		64	32	SHA-2 256 HMAC
CRYPTO_SHA2_384			48	SHA-2 384
CRYPTO_SHA2_384_HMAC		128	48	SHA-2 384 HMAC
CRYPTO_SHA2_512			64	SHA-2 512
CRYPTO_SHA2_512_HMAC		128	64	SHA-2 512 HMAC

Block Ciphers

Block ciphers in OCF can only operate on messages whose length is an exact multiple of the cipher’s block size. OCF supports the following block ciphers:

Name	IV Size	Block Size	Key Sizes	Description
CRYPTO_AES_CBC	16	16	16, 24, 32	AES-CBC

CRYPTO_AES_XTS	8	16	32, 64	AES-XTS
CRYPTO_CAMELLIA_CBC	16	16	16, 24, 32	Camellia CBC
CRYPTO_NULL_CBC	0	4	0-256	IPsec NULL cipher

CRYPTO_AES_XTS implements XEX Tweakable Block Cipher with Ciphertext Stealing as defined in NIST SP 800-38E. OCF consumers provide the first 8 bytes of the IV. The remaining 8 bytes are defined to be a block counter beginning at 0.

NOTE: The ciphertext stealing part is not implemented in all backends which is why this cipher requires input that is a multiple of the block size.

Stream Ciphers

Stream ciphers can operate on messages with arbitrary lengths. OCF supports the following stream ciphers:

Name	IV Size	Key Sizes	Description
CRYPTO_AES_ICM	16	16, 24, 32	AES Counter Mode
CRYPTO_CHACHA20	16	16, 32	ChaCha20

The IV for each request must be provided in *crp_iv* via the CRYPTO_F_IV_SEPARATE flag.

CRYPTO_AES_ICM uses the entire IV as a 128-bit big endian block counter. The IV sets the initial counter value for a message. If a consumer wishes to use an IV whose value is split into separate nonce and counter fields (e.g., IPsec), the consumer is responsible for splitting requests to handle counter rollover.

CRYPTO_CHACHA20 accepts a 16 byte IV. The first 8 bytes are used as a nonce. The last 8 bytes are used as a 64-bit little-endian block counter.

Authenticated Encryption with Associated Data Algorithms

AEAD algorithms in OCF combine a stream cipher with an authentication algorithm to provide both secrecy and authentication. AEAD algorithms accept additional authentication data (AAD) in addition to the ciphertext or plaintext. AAD is passed to the authentication algorithm as input in a method defined by the specific AEAD algorithm.

AEAD algorithms in OCF accept a nonce that is combined with an algorithm-defined counter to construct the IV for the underlying stream cipher. This nonce must be provided in *crp_iv* via the CRYPTO_F_IV_SEPARATE flag. Some AEAD algorithms support multiple nonce sizes. The first size listed is the default nonce size.

The following AEAD algorithms are supported:

Name	Nonce	Key Sizes	Tag	Description
CRYPTO_AES_NIST_GCM_16	12	16, 24, 32	16	AES Galois/Counter Mode
CRYPTO_AES_CCM_16	12, 7-13	16, 24, 32	16	AES Counter with CBC-MAC
CRYPTO_CHACHA20_POLY1305	12, 8	32	16	ChaCha20-Poly1305
CRYPTO_XCHACHA20_POLY1305	24	32	16	XChaCha20-Poly1305

SEE ALSO

crypto(4), crypto(9)

HISTORY

The **crypto** manual page first appeared in FreeBSD 10.1.