NAME

crypto - API for cryptographic services in the kernel

SYNOPSIS

#include <opencrypto/cryptodev.h>

DESCRIPTION

crypto is a framework for in-kernel cryptography. It permits in-kernel consumers to encrypt and decrypt data and also enables userland applications to use cryptographic hardware through the */dev/crypto* device.

crypto supports encryption and decryption operations using block and stream ciphers as well as computation and verification of message authentication codes (MACs). Consumers allocate sessions to describe a transform as discussed in crypto_session(9). Consumers then allocate request objects to describe each transformation such as encrypting a network packet or decrypting a disk sector. Requests are described in crypto_request(9).

Device drivers are responsible for processing requests submitted by consumers. crypto_driver(9) describes the interfaces drivers use to register with the framework, helper routines the framework provides to facilitate request processing, and the interfaces drivers are required to provide.

Callbacks

Since the consumers may not be associated with a process, drivers may not sleep(9). The same holds for the framework. Thus, a callback mechanism is used to notify a consumer that a request has been completed (the callback is specified by the consumer on a per-request basis). The callback is invoked by the framework whether the request was successfully completed or not. Errors are reported to the callback function.

Session initialization does not use callbacks and returns errors synchronously.

Session Migration

Operations may fail with a specific error code, EAGAIN, to indicate that a session handle has changed and that the request may be re-submitted immediately with the new session. The consumer should update its saved copy of the session handle to the value of *crp_session* so that future requests use the new session.

Supported Algorithms

More details on some algorithms may be found in crypto(7).

The following authentication algorithms are supported:

CRYPTO_AES_CCM_CBC_MAC
CRYPTO_AES_NIST_GMAC
CRYPTO_BLAKE2B
CRYPTO_BLAKE2S
CRYPTO_NULL_HMAC
CRYPTO_POLY1305
CRYPTO_RIPEMD160
CRYPTO_RIPEMD160_HMAC
CRYPTO_SHA1
CRYPTO_SHA1_HMAC
CRYPTO_SHA2_224
CRYPTO_SHA2_224_HMAC
CRYPTO_SHA2_256
CRYPTO_SHA2_256_HMAC
CRYPTO_SHA2_384
CRYPTO_SHA2_384_HMAC
CRYPTO_SHA2_512
CRYPTO_SHA2_512_HMAC

The following encryption algorithms are supported:

CRYPTO_AES_CBC
CRYPTO_AES_ICM
CRYPTO_AES_XTS
CRYPTO_CAMELLIA_CBC
CRYPTO_CHACHA20
CRYPTO_NULL_CBC

The following authenticated encryption with additional data (AEAD) algorithms are supported:

CRYPTO_AES_CCM_16
CRYPTO_AES_NIST_GCM_16
CRYPTO_CHACHA20_POLY1305

The following compression algorithms are supported:

CRYPTO_DEFLATE_COMP

FILES

sys/opencrypto/crypto.c most of the framework code

SEE ALSO

crypto(4), ipsec(4), crypto(7), crypto_driver(9), crypto_request(9), crypto_session(9), sleep(9)

HISTORY

The cryptographic framework first appeared in OpenBSD 2.7 and was written by Angelos D. Keromytis <*angelos@openbsd.org*>.

BUGS

The framework needs a mechanism for determining which driver is best for a specific set of algorithms associated with a session. Some type of benchmarking is in order here.