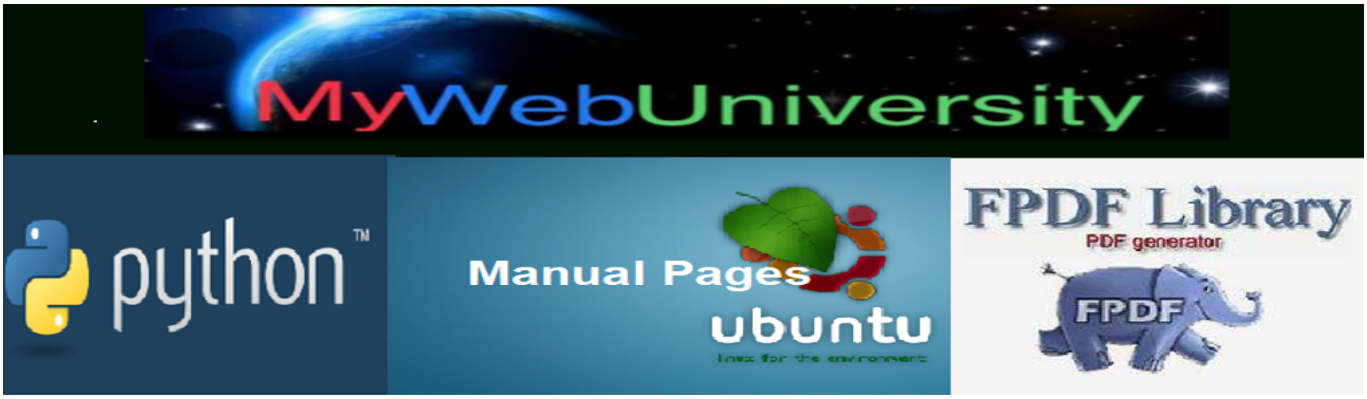

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Mac::HMAC.3pm'

$ man Crypt::Mac::HMAC.3pm

Crypt::Mac::HMAC(3pm) User Contributed Perl Documentation Crypt::Mac::HMAC(3pm)

NAME

Crypt::Mac::HMAC - Message authentication code HMAC

SYNOPSIS

```
### Functional interface:
use Crypt::Mac::HMAC qw( hmac hmac_hex );

# calculate MAC from string/buffer
$hmac_raw  = hmac('SHA256', $key, 'data buffer');
$hmac_hex  = hmac_hex('SHA256', $key, 'data buffer');
$hmac_b64  = hmac_b64('SHA256', $key, 'data buffer');
$hmac_b64u = hmac_b64u('SHA256', $key, 'data buffer');

### OO interface:
use Crypt::Mac::HMAC;

$d = Crypt::Mac::HMAC->new('SHA256', $key);
$d->add('any data');
$d->addfile('filename.dat');
$d->addfile(*FILEHANDLE);
$result_raw  = $d->mac;     # raw bytes
```

```
$result_hex  = $d->hexmac;  # hexadecimal form
$result_b64  = $d->b64mac;  # Base64 form
$result_b64u = $d->b64umac; # Base64 URL Safe form
```

DESCRIPTION

Provides an interface to the HMAC message authentication code (MAC) algorithm.

EXPORT

Nothing is exported by default.

You can export selected functions:

```
use Crypt::Mac::HMAC qw(hmac hmac_hex );
```

Or all of them at once:

```
use Crypt::Mac::HMAC ':all';
```

FUNCTIONS

hmac

Logically joins all arguments into a single string, and returns its HMAC message authentication code encoded as a binary string.

```
$hmac_raw = hmac($hash_name, $key, 'data buffer');
#or
$hmac_raw = hmac($hash_name, $key, 'any data', 'more data', 'even more data');

# $hash_name ... any <NAME> for which there exists Crypt::Digest::<NAME>
# $key ..... the key (octets/bytes)
```

hmac_hex

Logically joins all arguments into a single string, and returns its HMAC message authentication code encoded as a hexadecimal string.

```
$hmac_hex = hmac_hex($hash_name, $key, 'data buffer');
#or
$hmac_hex = hmac_hex($hash_name, $key, 'any data', 'more data', 'even more data');

# $hash_name ... any <NAME> for which there exists Crypt::Digest::<NAME>
# $key ..... the key (octets/bytes, not hex!)
```

hmac_b64

Logically joins all arguments into a single string, and returns its HMAC message authentication code encoded as a Base64 string.

```
$hmac_b64 = hmac_b64($hash_name, $key, 'data buffer');
#or
$hmac_b64 = hmac_b64($hash_name, $key, 'any data', 'more data', 'even more data');

# $hash_name ... any <NAME> for which there exists Crypt::Digest::<NAME>
# $key ..... the key (octets/bytes, not Base64!)
```

hmac_b64u

Logically joins all arguments into a single string, and returns its HMAC message authentication code encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$hmac_b64url = hmac_b64u($hash_name, $key, 'data buffer');
#or
$hmac_b64url = hmac_b64u($hash_name, $key, 'any data', 'more data', 'even more data');

# $hash_name ... any <NAME> for which there exists Crypt::Digest::<NAME>
# $key ..... the key (octets/bytes, not Base64url!)
```

METHODS

new

```
$d = Crypt::Mac::HMAC->new($hash_name, $key);
```

```
    # $hash_name ... any <NAME> for which there exists Crypt::Digest::<NAME>
    # $key ..... the key (octets/bytes)

clone
    $d->clone();

reset
    $d->reset();

add
    $d->add('any data');
    #or
    $d->add('any data', 'more data', 'even more data');

addfile
    $d->addfile('filename.dat');
    #or
    $d->addfile(*FILEHANDLE);

mac
    $result_raw = $d->mac();

hexmac
    $result_hex = $d->hexmac();

b64mac
    $result_b64 = $d->b64mac();

b64umac
    $result_b64url = $d->b64umac();
```

- CryptX
- <https://en.wikipedia.org/wiki/Hmac>
- <https://tools.ietf.org/html/rfc2104>