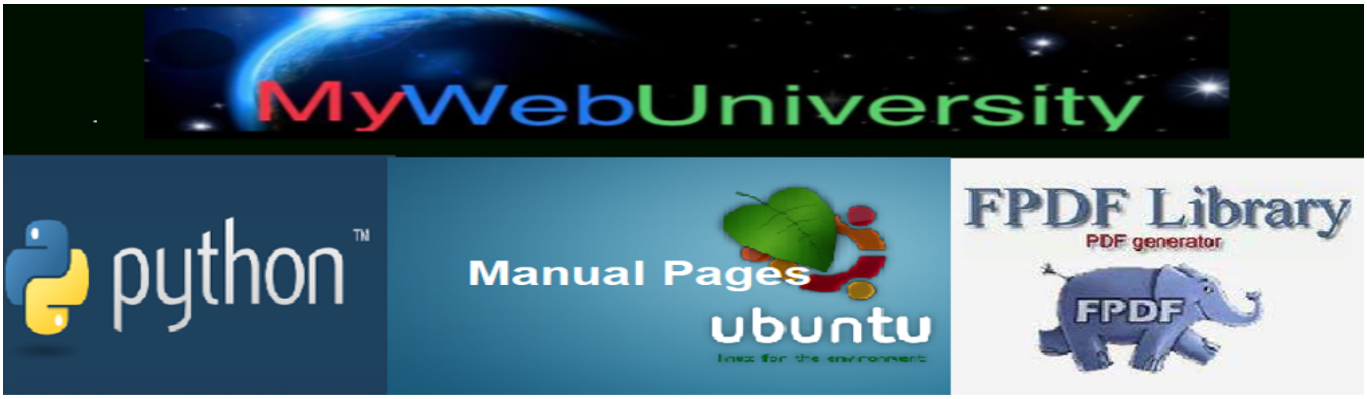

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Mac::Poly1305.3pm'

$ man Crypt::Mac::Poly1305.3pm

NAME

Crypt::Mac::Poly1305 - Message authentication code Poly1305 (RFC 7539)

SYNOPSIS

```
### Functional interface:
use Crypt::Mac::Poly1305 qw( poly1305 poly1305_hex );

# calculate MAC from string/buffer
$poly1305_raw  = poly1305($key, 'data buffer');
$poly1305_hex  = poly1305_hex($key, 'data buffer');
$poly1305_b64  = poly1305_b64($key, 'data buffer');
$poly1305_b64u = poly1305_b64u($key, 'data buffer');

### OO interface:
use Crypt::Mac::Poly1305;

$d = Crypt::Mac::Poly1305->new($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 Poly1305 message authentication code (MAC) algorithm.

EXPORT

Nothing is exported by default.

You can export selected functions:

```
use Crypt::Mac::Poly1305 qw(poly1305 poly1305_hex );
```

Or all of them at once:

```
use Crypt::Mac::Poly1305 ':all';
```

FUNCTIONS

poly1305

Logically joins all arguments into a single string, and returns its Poly1305 message authentication code encoded as a binary string.

```
$poly1305_raw = poly1305($key, 'data buffer');
#or
$poly1305_raw = poly1305($key, 'any data', 'more data', 'even more data');
```

poly1305_hex

Logically joins all arguments into a single string, and returns its Poly1305 message authentication code encoded as a hexadecimal string.

```
$poly1305_hex = poly1305_hex($key, 'data buffer');
#or
```

```
$poly1305_hex = poly1305_hex($key, 'any data', 'more data', 'even more data');
```

poly1305_b64

Logically joins all arguments into a single string, and returns its Poly1305 message authentication code encoded as a Base64 string.

```
$poly1305_b64 = poly1305_b64($key, 'data buffer');
#or
$poly1305_b64 = poly1305_b64($key, 'any data', 'more data', 'even more data');
```

poly1305_b64u

Logically joins all arguments into a single string, and returns its Poly1305 message authentication code encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$poly1305_b64url = poly1305_b64u($key, 'data buffer');
#or
$poly1305_b64url = poly1305_b64u($key, 'any data', 'more data', 'even more data');
```

METHODS

new

```
$d = Crypt::Mac::Poly1305->new($key);
```

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();
```

SEE ALSO

- CryptX
- <https://www.ietf.org/rfc/rfc7539.txt>