



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

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

\$ man Crypt::Mac::Pelican.3pm

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

NAME

Crypt::Mac::Pelican - Message authentication code Pelican (AES based MAC)

SYNOPSIS

Functional interface:

```
use Crypt::Mac::Pelican qw( pelican pelican_hex );
```

```
# calculate MAC from string/buffer
```

```
$pelican_raw = pelican($key, 'data buffer');
```

```
$pelican_hex = pelican_hex($key, 'data buffer');
```

```
$pelican_b64 = pelican_b64($key, 'data buffer');
```

```
$pelican_b64u = pelican_b64u($key, 'data buffer');
```

OO interface:

```
use Crypt::Mac::Pelican;
```

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

EXPORT

Nothing is exported by default.

You can export selected functions:

```
use Crypt::Mac::Pelican qw(pelican pelican_hex );
```

Or all of them at once:

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

FUNCTIONS

pelican

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

```
$pelican_raw = pelican($key, 'data buffer');
```

#or

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

pelican_hex

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

```
$pelican_hex = pelican_hex($key, 'data buffer');
```

#or

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

pelican_b64

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

```
$pelican_b64 = pelican_b64($key, 'data buffer');
```

#or

```
$pelican_b64 = pelican_b64($key, 'any data', 'more data', 'even more data');
```

pelican_b64u

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

```
$pelican_b64url = pelican_b64u($key, 'data buffer');
```

#or

```
$pelican_b64url = pelican_b64u($key, 'any data', 'more data', 'even more data');
```

METHODS

new

```
$d = Crypt::Mac::Pelican->new($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

? <<http://eprint.iacr.org/2005/088.pdf>>

perl v5.34.0

2022-02-06

Crypt::Mac::Pelican(3pm)