



**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::BLAKE2b.3pm'***

**\$ man Crypt::Mac::BLAKE2b.3pm**

Crypt::Mac::BLAKE2b(3pm) User Contributed Perl Documentation Crypt::Mac::BLAKE2b(3pm)

**NAME**

Crypt::Mac::BLAKE2b - Message authentication code BLAKE2b MAC (RFC 7693)

**SYNOPSIS**

### Functional interface:

```
use Crypt::Mac::BLAKE2b qw( blake2b blake2b_hex );
```

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

```
$blake2b_raw = blake2b($size, $key, 'data buffer');
```

```
$blake2b_hex = blake2b_hex($size, $key, 'data buffer');
```

```
$blake2b_b64 = blake2b_b64($size, $key, 'data buffer');
```

```
$blake2b_b64u = blake2b_b64u($size, $key, 'data buffer');
```

### OO interface:

```
use Crypt::Mac::BLAKE2b;
```

```
$d = Crypt::Mac::BLAKE2b->new($size, $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 BLAKE2b message authentication code (MAC) algorithm.

## EXPORT

Nothing is exported by default.

You can export selected functions:

```
use Crypt::Mac::BLAKE2b qw(blake2b blake2b_hex );
```

Or all of them at once:

```
use Crypt::Mac::BLAKE2b ':all';
```

## FUNCTIONS

### blake2b

Logically joins all arguments into a single string, and returns its BLAKE2b message authentication code encoded as a binary string.

```
$blake2b_raw = blake2b($size, $key, 'data buffer');
```

#or

```
$blake2b_raw = blake2b($size, $key, 'any data', 'more data', 'even more data');
```

### blake2b\_hex

Logically joins all arguments into a single string, and returns its BLAKE2b message authentication code encoded as a hexadecimal string.

```
$blake2b_hex = blake2b_hex($size, $key, 'data buffer');
```

#or

```
$blake2b_hex = blake2b_hex($size, $key, 'any data', 'more data', 'even more data');
```

#### blake2b\_b64

Logically joins all arguments into a single string, and returns its BLAKE2b message authentication code encoded as a Base64 string.

```
$blake2b_b64 = blake2b_b64($size, $key, 'data buffer');
```

```
#or
```

```
$blake2b_b64 = blake2b_b64($size, $key, 'any data', 'more data', 'even more data');
```

#### blake2b\_b64u

Logically joins all arguments into a single string, and returns its BLAKE2b message authentication code encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$blake2b_b64url = blake2b_b64u($size, $key, 'data buffer');
```

```
#or
```

```
$blake2b_b64url = blake2b_b64u($size, $key, 'any data', 'more data', 'even more data');
```

## METHODS

### new

```
$d = Crypt::Mac::BLAKE2b->new($size, $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://tools.ietf.org/html/rfc7693>>