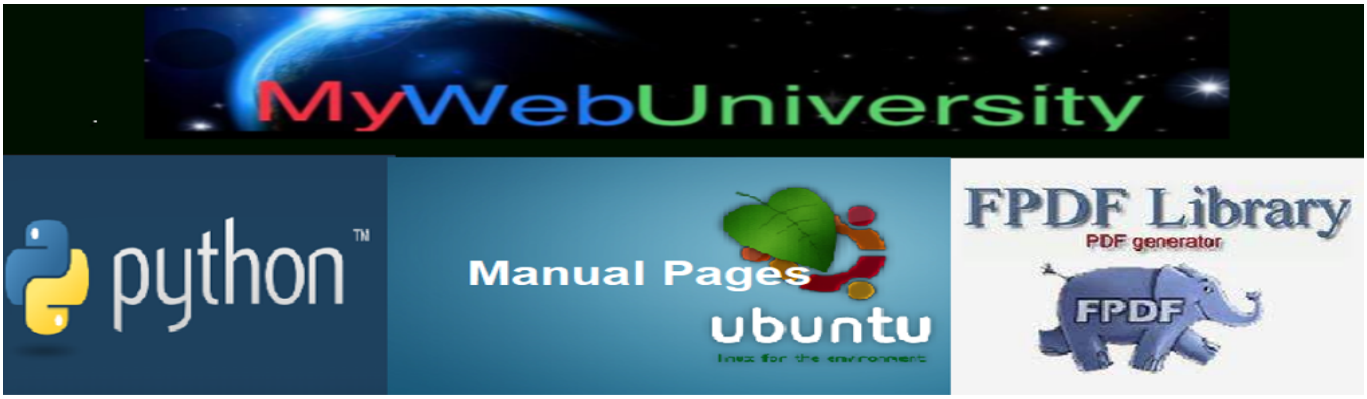

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Misc.3pm'

$ man Crypt::Misc.3pm

NAME

Crypt::Misc - miscellaneous functions related to (or used by) CryptX

SYNOPSIS

This module contains a collection of mostly unsorted functions loosely-related to CryptX distribution but not implementing cryptography.

Most of them are also available in other perl modules but once you utilize CryptX you might avoid dependencies on other modules by using functions from Crypt::Misc.

DESCRIPTION

```
use Crypt::Misc ':all';

# Base64 and Base64/URL-safe functions
$base64    = encode_b64($rawbytes);
$rawbytes  = decode_b64($base64);
$base64url = encode_b64u($encode_b64u);
$rawbytes  = decode_b64u($base64url);

# read/write file
$rawdata = read_rawfile($filename);
```

```
write_rawfile($filename, $rawdata);

# convert PEM/DER
$der_data = pem_to_der($pem_data);
$pem_data = der_to_pem($der_data);

 # others
 die "mismatch" unless slow_eq($str1, $str2);
```

FUNCTIONS

By default, Crypt::Misc doesn't import any function. You can import individual functions like this:

```
 use Crypt::Misc qw(read_rawfile);
```

Or import all available functions:

```
 use Crypt::Misc ':all';
```

read_rawfile

Since: 0.029

```
 $rawdata = read_rawfile($filename);
```

Read file $filename into a scalar as a binary data (without decoding/transformation).

write_rawfile

Since: 0.029

```
 write_rawfile($filename, $rawdata);
```

Write $rawdata to file $filename as binary data.

slow_eq

Since: 0.029

```
if (slow_eq($data1, $data2)) { ... }
```

Constant time compare (to avoid timing side-channel).

pem_to_der

Since: 0.029

```
$der_data = pem_to_der($pem_data);
#or
$der_data = pem_to_der($pem_data, $password);
```

Convert PEM to DER representation. Supports also password protected PEM data.

der_to_pem

Since: 0.029

```
$pem_data = der_to_pem($der_data, $header_name);
#or
$pem_data = der_to_pem($der_data, $header_name, $password);
#or
$pem_data = der_to_pem($der_data, $header_name, $passord, $cipher_name);

# $header_name e.g. "PUBLIC KEY", "RSA PRIVATE KEY" ...
# $cipher_name e.g. "DES-EDE3-CBC", "AES-256-CBC" (DEFAULT) ...
```

Convert DER to PEM representation. Supports also password protected PEM data.

random_v4uuid

Since: 0.031

```
 my $uuid = random_v4uuid();
```

Returns cryptographically strong Version 4 random UUID:

"xxxxxxxx-xxxx-4xxx-Yxxx-xxxxxxxxxxxx" where "x" is any hexadecimal digit and "Y" is one of 8, 9, A, B (1000, 1001, 1010, 1011) e.g. "f47ac10b-58cc-4372-a567-0e02b2c3d479".

is_v4uuid

Since: 0.031

```
 if (is_v4uuid($uuid)) {
   ...
 }
```

Checks the given $uuid string whether it matches V4 UUID format and returns 0 (mismatch) or 1 (match).

increment_octets_le

Since: 0.048

```
 $octects = increment_octets_le($octets);
```

Take input $octets as a little-endian big number and return an increment.

increment_octets_be

Since: 0.048

```
 $octects = increment_octets_be($octets);
```

Take input $octets as a big-endian big number and return an increment.

encode_b64

Since: 0.029

$base64string = encode_b64($rawdata);

Encode $rawbytes into Base64 string, no line-endings in the output string.

decode_b64

Since: 0.029

$rawdata = decode_b64($base64string);

Decode a Base64 string.

encode_b64u

Since: 0.029

$base64url_string = encode_b64($rawdata);

Encode $rawbytes into Base64/URL-Safe string, no line-endings in the output string.

decode_b64u

Since: 0.029

$rawdata = decode_b64($base64url_string);

Decode a Base64/URL-Safe string.

encode_b32r

Since: 0.049

$string = encode_b32r($rawdata);

Encode bytes into Base32 (rfc4648 alphabet) string, without "=" padding.

decode_b32r

Since: 0.049

$rawdata = decode_b32r($string);

Decode a Base32 (rfc4648 alphabet) string into bytes.

encode_b32b

Since: 0.049

$string = encode_b32b($rawdata);

Encode bytes into Base32 (base32hex alphabet) string, without "=" padding.

decode_b32b

Since: 0.049

$rawdata = decode_b32b($string);

Decode a Base32 (base32hex alphabet) string into bytes.

encode_b32z

Since: 0.049

$string = encode_b32z($rawdata);

Encode bytes into Base32 (zbase32 alphabet) string.

decode_b32z

Since: 0.049

$rawdata = decode_b32z($string);

Decode a Base32 (zbase32 alphabet) string into bytes.

encode_b32c

Since: 0.049

```
$string = encode_b32c($rawdata);
```

Encode bytes into Base32 (crockford alphabet) string.

decode_b32c

Since: 0.049

```
$rawdata = decode_b32c($string);
```

Decode a Base32 (crockford alphabet) string into bytes.

encode_b58b

Since: 0.049

```
$string = encode_b58b($rawdata);
```

Encode bytes into Base58 (Bitcoin alphabet) string.

decode_b58b

Since: 0.049

```
$rawdata = decode_b58b($string);
```

Decode a Base58 (Bitcoin alphabet) string into bytes.

encode_b58f

Since: 0.049

```
$string = encode_b58f($rawdata);
```

Encode bytes into Base58 (Flickr alphabet) string.

decode_b58f

Since: 0.049

```
$rawdata = decode_b58f($string);
```

Decode a Base58 (Flickr alphabet) string into bytes.

encode_b58r

Since: 0.049

```
$string = encode_b58r($rawdata);
```

Encode bytes into Base58 (Ripple alphabet) string.

decode_b58r

Since: 0.049

```
$rawdata = decode_b58r($string);
```

Decode a Base58 (Ripple alphabet) string into bytes.

encode_b58t

Since: 0.049

```
$string = encode_b58t($rawdata);
```

Encode bytes into Base58 (Tipple alphabet) string.

decode_b58t

Since: 0.049

```
$rawdata = decode_b58t($string);
```

Decode a Base58 (Tipple alphabet) string into bytes.

encode_b58s

Since: 0.049

```
$string = encode_b58s($rawdata);
```

Encode bytes into Base58 (Stellar alphabet) string.

decode_b58s

Since: 0.049

```
$rawdata = decode_b58s($string);
```

Decode a Base58 (Stellar alphabet) string into bytes.

SEE ALSO

? CryptX