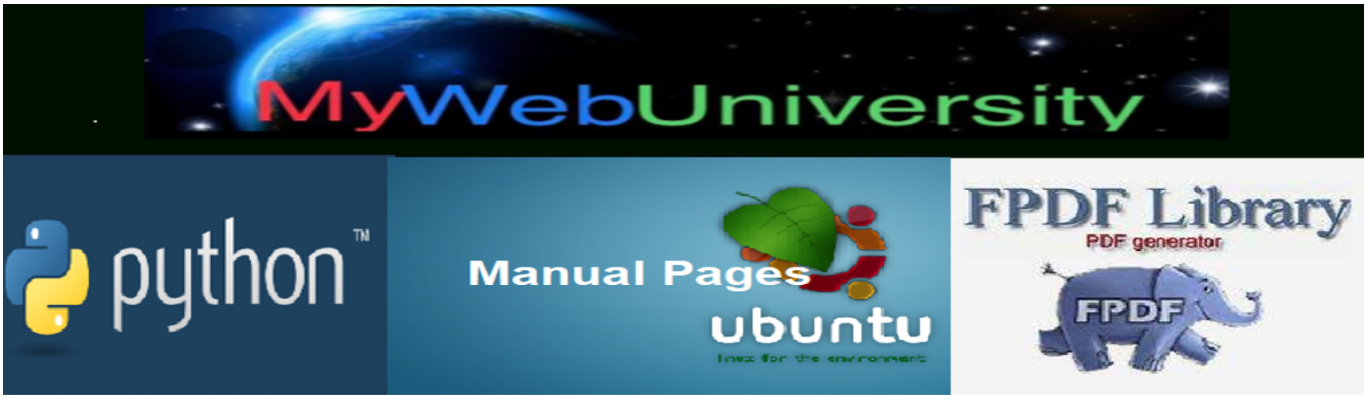

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Cipher::DES_EDE.3pm'

$ man Crypt::Cipher::DES_EDE.3pm

Crypt::Cipher::DES_EDE(3pm) User Contributed Perl Documentation Crypt::Cipher::DES_EDE(3pm)

NAME

Crypt::Cipher::DES_EDE - Symmetric cipher DES_EDE (aka Triple-DES, 3DES), key size: 192[168] bits

SYNOPSIS

```
### example 1
use Crypt::Mode::CBC;

my $key = '...'; # length has to be valid key size for this cipher
my $iv = '...';  # 16 bytes
my $cbc = Crypt::Mode::CBC->new('DES_EDE');
my $ciphertext = $cbc->encrypt("secret data", $key, $iv);

### example 2 (slower)
use Crypt::CBC;
use Crypt::Cipher::DES_EDE;

my $key = '...'; # length has to be valid key size for this cipher
my $iv = '...';  # 16 bytes
my $cbc = Crypt::CBC->new( -cipher=>'Cipher::DES_EDE', -key=>$key, -iv=>$iv );
my $ciphertext = $cbc->encrypt("secret data");
```

DESCRIPTION

This module implements the DES_EDE cipher. Provided interface is compliant with Crypt::CBC module.

BEWARE: This module implements just elementary "one-block-(en|de)cryption" operation - if you want to encrypt/decrypt generic data you have to use some of the cipher block modes - check for example Crypt::Mode::CBC, Crypt::Mode::CTR or Crypt::CBC (which will be slower).

METHODS

new

```
$c = Crypt::Cipher::DES_EDE->new($key);
#or
$c = Crypt::Cipher::DES_EDE->new($key, $rounds);
```

encrypt

```
$ciphertext = $c->encrypt($plaintext);
```

decrypt

```
$plaintext = $c->decrypt($ciphertext);
```

keysize

```
$c->keysize;
#or
Crypt::Cipher::DES_EDE->keysize;
#or
Crypt::Cipher::DES_EDE::keysize;
```

blocksize

```
$c->blocksize;
#or
Crypt::Cipher::DES_EDE->blocksize;
#or
```

```
Crypt::Cipher::DES_EDE::blocksize;

max_keysize
    $c->max_keysize;
    #or
    Crypt::Cipher::DES_EDE->max_keysize;
    #or
    Crypt::Cipher::DES_EDE::max_keysize;

min_keysize
    $c->min_keysize;
    #or
    Crypt::Cipher::DES_EDE->min_keysize;
    #or
    Crypt::Cipher::DES_EDE::min_keysize;

default_rounds
    $c->default_rounds;
    #or
    Crypt::Cipher::DES_EDE->default_rounds;
    #or
    Crypt::Cipher::DES_EDE::default_rounds;
```

SEE ALSO

- CryptX, Crypt::Cipher
- <https://en.wikipedia.org/wiki/Triple_DES>