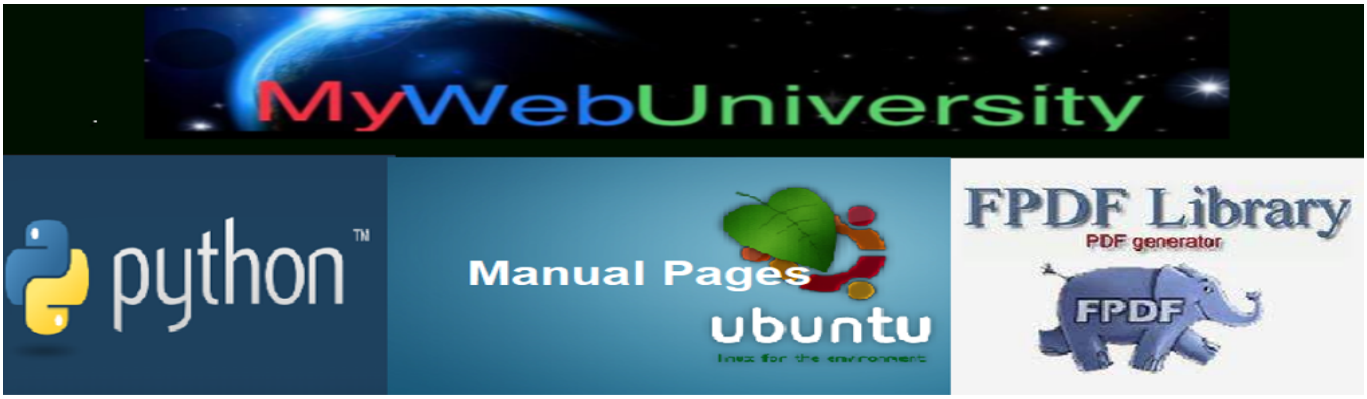

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Cipher::SEED.3pm'

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

NAME

Crypt::Cipher::SEED - Symmetric cipher SEED, key size: 128 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('SEED');
my $ciphertext = $cbc->encrypt("secret data", $key, $iv);

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

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

DESCRIPTION

This module implements the SEED 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

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

encrypt

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

decrypt

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

keysize

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

blocksize

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

max_keysize

```
$c->max_keysize;
#or
Crypt::Cipher::SEED->max_keysize;
#or
Crypt::Cipher::SEED::max_keysize;
```

min_keysize

```
$c->min_keysize;
#or
Crypt::Cipher::SEED->min_keysize;
#or
Crypt::Cipher::SEED::min_keysize;
```

default_rounds

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

SEE ALSO

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