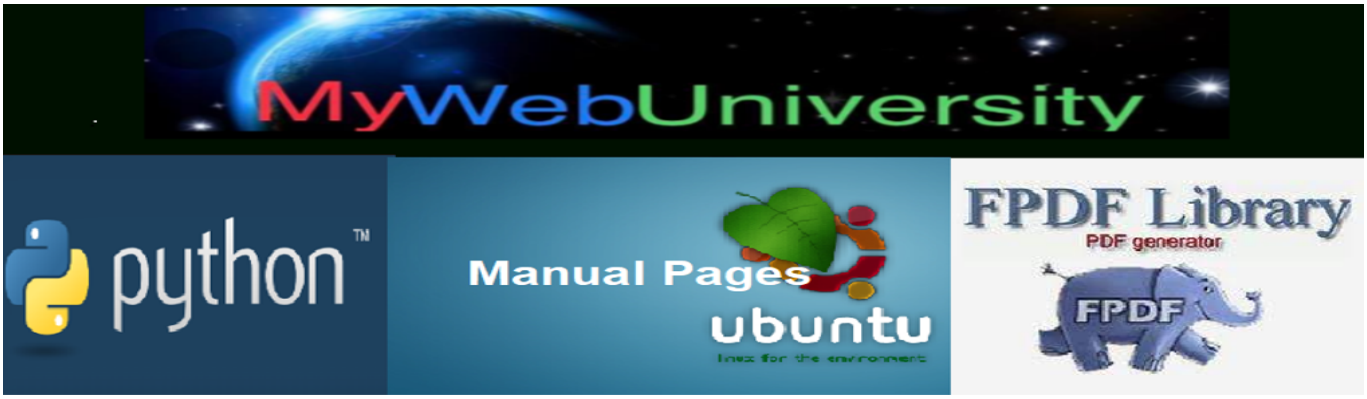

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Mode::ECB.3pm'

$ man Crypt::Mode::ECB.3pm

Crypt::Mode::ECB(3pm) User Contributed Perl Documentation Crypt::Mode::ECB(3pm)

NAME

Crypt::Mode::ECB - Block cipher mode ECB [Electronic codebook]

SYNOPSIS

```
use Crypt::Mode::ECB;
my $m = Crypt::Mode::ECB->new('AES');

#(en|de)crypt at once
my $ciphertext = $m->encrypt($plaintext, $key);
my $plaintext = $m->decrypt($ciphertext, $key);

#encrypt more chunks
$m->start_encrypt($key);
my $ciphertext = $m->add('some data');
$ciphertext .= $m->add('more data');
$ciphertext .= $m->finish;

#decrypt more chunks
$m->start_decrypt($key);
my $plaintext = $m->add($some_ciphertext);
$plaintext .= $m->add($more_ciphertext);
```

```
    $plaintext .= $m->finish;
```

DESCRIPTION

This module implements ECB cipher mode. NOTE: it works only with ciphers from CryptX (Crypt::Cipher::NNNN). BEWARE: ECB is inherently insecure, if you are not sure go for Crypt::Mode::CBC!

METHODS

new

```
    my $m = Crypt::Mode::ECB->new($name);
    #or
    my $m = Crypt::Mode::ECB->new($name, $padding);
    #or
    my $m = Crypt::Mode::ECB->new($name, $padding, $cipher_rounds);

    # $name ..... one of 'AES', 'Anubis', 'Blowfish', 'CAST5', 'Camellia', 'DES', 'DES_EDE',
    #             'KASUMI', 'Khazad', 'MULTI2', 'Noekeon', 'RC2', 'RC5', 'RC6',
    #             'SAFERP', 'SAFER_K128', 'SAFER_K64', 'SAFER_SK128', 'SAFER_SK64',
    #             'SEED', 'Skipjack', 'Twofish', 'XTEA', 'IDEA', 'Serpent'
    #             simply any <NAME> for which there exists Crypt::Cipher::<NAME>
    # $padding .... 0 no padding (plaintext size has to be multiple of block length)
    #             1 PKCS5 padding, Crypt::CBC's "standard" - DEFAULT
    #             2 Crypt::CBC's "oneandzeroes"
    #             3 ANSI X.923 padding
    #             4 zero padding
    #             5 zero padding (+a block of zeros if the output length is divisible by the blocksize)
    # $cipher_rounds ... optional num of rounds for given cipher
```

encrypt

```
    my $ciphertext = $m->encrypt($plaintext, $key);
```

decrypt

```
    my $plaintext = $m->decrypt($ciphertext, $key);
```

start_encrypt

```
$m->start_encrypt($key);
```

start_decrypt

```
$m->start_decrypt($key);
```

add

```
# in encrypt mode
my $plaintext = $m->add($ciphertext);

# in decrypt mode
my $ciphertext = $m->add($plaintext);
```

finish

```
#encrypt more chunks
$m->start_encrypt($key);
my $ciphertext = '';
$ciphertext .= $m->add('some data');
$ciphertext .= $m->add('more data');
$ciphertext .= $m->finish;

#decrypt more chunks
$m->start_decrypt($key);
my $plaintext = '';
$plaintext .= $m->add($some_ciphertext);
$plaintext .= $m->add($more_ciphertext);
$plaintext .= $m->finish;
```

SEE ALSO

- CryptX, Crypt::Cipher
- Crypt::Cipher::AES, Crypt::Cipher::Blowfish, ...

? <https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Electronic_codebook_.28ECB.29>