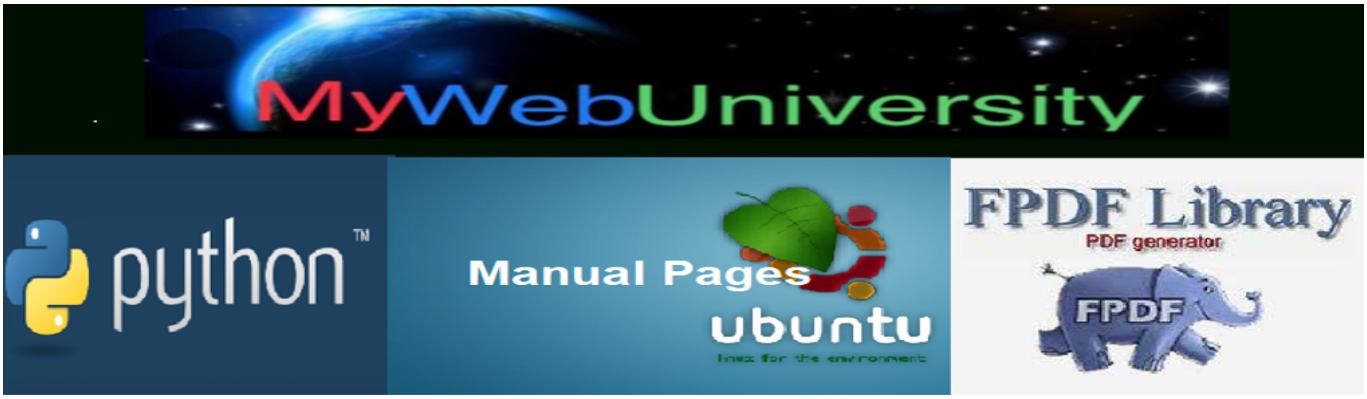

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::CAST5.3pm'

$ man Crypt::CAST5.3pm

CAST5(3pm) User Contributed Perl Documentation CAST5(3pm)

NAME

Crypt::CAST5 - CAST5 block cipher

SYNOPSIS

```
use Crypt::CBC;

my $crypt = Crypt::CBC->new({
   key    => "secret key",
   cipher => "CAST5",
});

my $message = "All mimsy were the borogoves";
my $ciphertext = $crypt->encrypt($message);
print unpack("H*", $ciphertext), "\n";

my $plaintext = $crypt->decrypt($ciphertext);
print $plaintext, "\n";
```

DESCRIPTION

This module provides an implementation of the CAST5 block cipher using compiled C code for increased speed. CAST5 is also known as CAST-128. It is a product of the CAST design

procedure developed by C. Adams and S. Tavares.

The CAST5 cipher is available royalty-free.

FUNCTIONS

blocksize

Returns the CAST5 block size, which is 8 bytes. This function exists so that Crypt::CAST5 can work with Crypt::CBC.

keysize

Returns the maximum CAST5 key size, 16 bytes.

new

```
$cast5 = Crypt::CAST5->new($key);
```

Create a new encryption object. If the optional key parameter is given, it will be passed to the init() function.

init

```
$cast5->init($key);
```

Set or change the encryption key to be used. The key must be from 40 bits (5 bytes) to 128 bits (16 bytes) in length. Note that if the key used is 80 bits or less, encryption and decryption will be somewhat faster.

It is best for the key to be random binary data, not something printable like a password. A message digest function may be useful for converting a password to an encryption key; see Digest::SHA1 or Digest::MD5. Note that Crypt::CBC runs the given "key" through MD5 to get the actual encryption key.

encrypt

```
$ciphertext = $cast5->encrypt($plaintext);
```

Encrypt a block of plaintext using the current encryption key, and return the corresponding ciphertext. The input must be 8 bytes long, and the output has the same length. Note that the encryption is in ECB mode, which means that it encrypts each block independently. That can leave you vulnerable to dictionary attacks, so it is generally best to use some form of chaining between blocks; see Crypt::CBC.

decrypt

```
$plaintext = $cast5->decrypt($ciphertext);
```

Decrypt the ciphertext and return the corresponding plaintext.

SEE ALSO

RFC 2144, "The CAST-128 Encryption Algorithm", C. Adams, May 1997

Crypt::CBC

AUTHOR

Bob Mathews, <bobmathews@alumni.calpoly.edu>

COPYRIGHT AND LICENSE