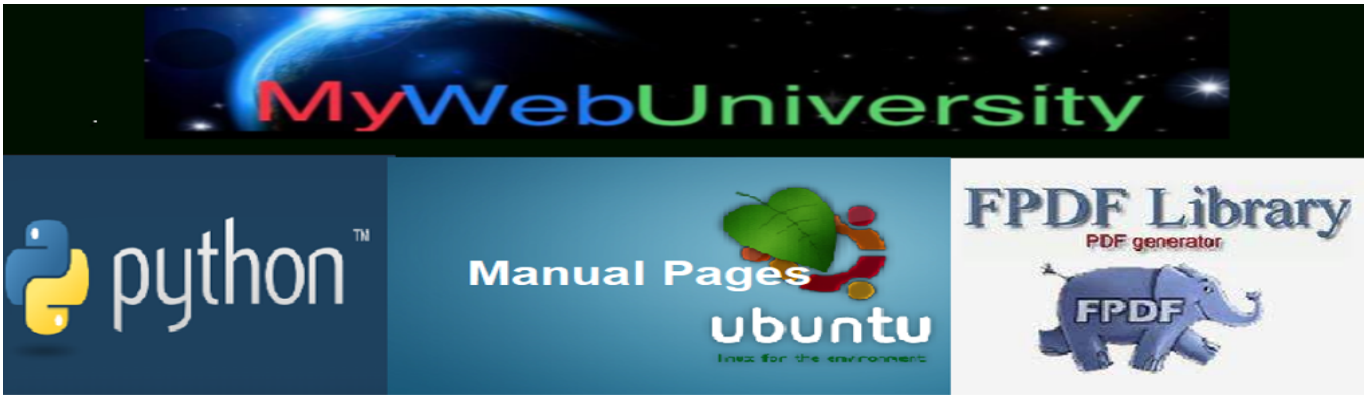

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::PK::DH.3pm'

$ man Crypt::PK::DH.3pm

Crypt::PK::DH(3pm) User Contributed Perl Documentation Crypt::PK::DH(3pm)

NAME

Crypt::PK::DH - Public key cryptography based on Diffie-Hellman

SYNOPSIS

```
### OO interface
#Shared secret
my $priv = Crypt::PK::DH->new('Alice_priv_dh1.key');
my $pub = Crypt::PK::DH->new('Bob_pub_dh1.key');
my $shared_secret = $priv->shared_secret($pub);
#Key generation
my $pk = Crypt::PK::DH->new();
$pk->generate_key(128);
my $private = $pk->export_key('private');
my $public = $pk->export_key('public');
or
my $pk = Crypt::PK::DH->new();
$pk->generate_key('ike2048');
my $private = $pk->export_key('private');
my $public = $pk->export_key('public');
or
my $pk = Crypt::PK::DH->new();
$pk->generate_key({ p => $p, g => $g });
my $private = $pk->export_key('private');
```

```
    my $public = $pk->export_key('public');
    ### Functional interface
    #Shared secret
    my $shared_secret = dh_shared_secret('Alice_priv_dh1.key', 'Bob_pub_dh1.key');
```

METHODS

new

```
     my $pk = Crypt::PK::DH->new();
     #or
     my $pk = Crypt::PK::DH->new($priv_or_pub_key_filename);
     #or
     my $pk = Crypt::PK::DH->new(\$buffer_containing_priv_or_pub_key);
```

generate_key

Uses Yarrow-based cryptographically strong random number generator seeded with random data taken from "/dev/random" (UNIX) or "CryptGenRandom" (Win32).

```
     $pk->generate_key($groupsize);
     ### $groupsize (in bytes) corresponds to DH parameters (p, g) predefined by libtomcrypt
     # 96   =>  DH-768
     # 128  =>  DH-1024
     # 192  =>  DH-1536
     # 256  =>  DH-2048
     # 384  =>  DH-3072
     # 512  =>  DH-4096
     # 768  =>  DH-6144
     # 1024 =>  DH-8192
```

The following variants are available since CryptX-0.032

```
     $pk->generate_key($groupname)
     ### $groupname corresponds to values defined in RFC7296 and RFC3526
     # 'ike768'  =>  768-bit MODP (Group 1)
     # 'ike1024' => 1024-bit MODP (Group 2)
     # 'ike1536' => 1536-bit MODP (Group 5)
     # 'ike2048' => 2048-bit MODP (Group 14)
     # 'ike3072' => 3072-bit MODP (Group 15)
     # 'ike4096' => 4096-bit MODP (Group 16)
```

```
    # 'ike6144' => 6144-bit MODP (Group 17)
    # 'ike8192' => 8192-bit MODP (Group 18)
    $pk->generate_key($param_hash)
    # $param_hash is { g => $g, p => $p }
    # where $g is the generator (base) in a hex string and $p is the prime in a hex string
    $pk->generate_key(\$dh_param)
    # $dh_param is the content of DER or PEM file with DH parameters
    # e.g. openssl dhparam 2048
import_key
   Loads private or public key (exported by "export_key").
     $pk->import_key($filename);
     #or
     $pk->import_key(\$buffer_containing_key);
import_key_raw
   Since: CryptX-0.032
     $pk->import_key_raw($raw_bytes, $type, $params)
     ### $raw_bytes is a binary string containing the key
     ### $type is either 'private' or 'public'
     ### $param is either a name ('ike2038') or hash containing the p,g values { g=>$g, p=>$p }
     ### in hex strings
export_key
   BEWARE: DH key format change - since v0.049 it is compatible with libtomcrypt 1.18.
    my $private = $pk->export_key('private');
    #or
    my $public = $pk->export_key('public');
export_key_raw
   Since: CryptX-0.032
    $raw_bytes = $dh->export_key_raw('public')
    #or
    $raw_bytes = $dh->export_key_raw('private')
shared_secret
    # Alice having her priv key $pk and Bob's public key $pkb
    my $pk  = Crypt::PK::DH->new($priv_key_filename);
```

```
    my $pkb = Crypt::PK::DH->new($pub_key_filename);
    my $shared_secret = $pk->shared_secret($pkb);
    # Bob having his priv key $pk and Alice's public key $pka
    my $pk = Crypt::PK::DH->new($priv_key_filename);
    my $pka = Crypt::PK::DH->new($pub_key_filename);
    my $shared_secret = $pk->shared_secret($pka);  # same value as computed by Alice
is_private
    my $rv = $pk->is_private;
    # 1 .. private key loaded
    # 0 .. public key loaded
    # undef .. no key loaded
size
    my $size = $pk->size;
    # returns key size in bytes or undef if no key loaded
key2hash
    my $hash = $pk->key2hash;
    # returns hash like this (or undef if no key loaded):
    {
      type => 0,   # integer: 1 .. private, 0 .. public
      size => 256, # integer: key size in bytes
      x => "FBC1062F73B9A17BB8473A2F5A074911FA7F20D28FB...", #private key
      y => "AB9AAA40774D3CD476B52F82E7EE2D8A8D40CD88BF4...", #public key
      g => "2", # generator/base
      p => "FFFFFFFFFFFFFFFFC90FDAA22168C234C4C6628B80D...", # prime
    }
params2hash
   Since: CryptX-0.032
    my $params = $pk->params2hash;
    # returns hash like this (or undef if no key loaded):
    {
      g => "2", # generator/base
      p => "FFFFFFFFFFFFFFFFC90FDAA22168C234C4C6628B80D...", # prime
    }
```

FUNCTIONS

dh_shared_secret

DH based shared secret generation. See method "shared_secret" below.

```
#on Alice side
my $shared_secret = dh_shared_secret('Alice_priv_dh1.key', 'Bob_pub_dh1.key');
#on Bob side
my $shared_secret = dh_shared_secret('Bob_priv_dh1.key', 'Alice_pub_dh1.key');
```

DEPRECATED INTERFACE

The following functions/methods were removed in removed in v0.049:

encrypt

decrypt

sign_message

verify_message

sign_hash

verify_hash

dh_encrypt

dh_decrypt

dh_sign_message

dh_verify_message

dh_sign_hash

dh_verify_hash

SEE ALSO

? <https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>