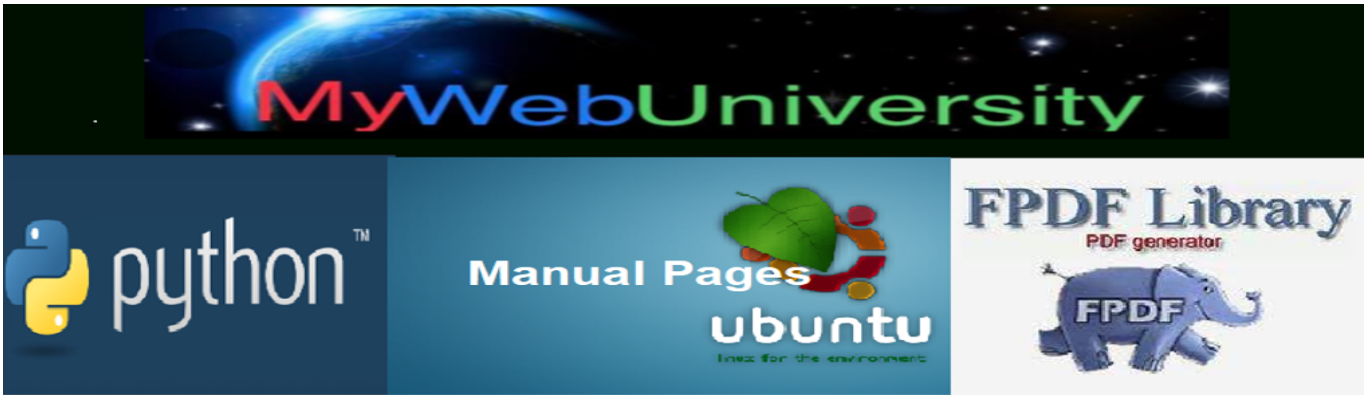

Rocky Enterprise Linux 9.2 Manual Pages on command 'EVP_KEYEXCH-DH.7ssl'

$ man EVP_KEYEXCH-DH.7ssl

EVP_KEYEXCH-DH(7SSL) OpenSSL EVP_KEYEXCH-DH(7SSL)

NAME

EVP_KEYEXCH-DH - DH Key Exchange algorithm support

DESCRIPTION

Key exchange support for the DH key type.

DH key exchange parameters

"pad" (OSSL_EXCHANGE_PARAM_PAD) <unsigned integer>

Sets the padding mode for the associated key exchange ctx. Setting a value of 1 will turn padding on. Setting a value of 0 will turn padding off. If padding is off then the derived shared secret may be smaller than the largest possible secret size. If padding is on then the derived shared secret will have its first bytes filled with zeros where necessary to make the shared secret the same size as the largest possible secret size. The padding mode parameter is ignored (and padding implicitly enabled) when the KDF type is set to "X942KDF-ASN1" (OSSL_KDF_NAME_X942KDF_ASN1).

"kdf-type" (OSSL_EXCHANGE_PARAM_KDF_TYPE) <UTF8 string>

See "Common Key Exchange parameters" in provider-keyexch(7).

"kdf-digest" (OSSL_EXCHANGE_PARAM_KDF_DIGEST) <UTF8 string>

See "Common Key Exchange parameters" in provider-keyexch(7).

"kdf-digest-props" (OSSL_EXCHANGE_PARAM_KDF_DIGEST_PROPS) <UTF8 string>

See "Common Key Exchange parameters" in provider-keyexch(7).

"kdf-outlen" (OSSL_EXCHANGE_PARAM_KDF_OUTLEN) <unsigned integer>

See "Common Key Exchange parameters" in provider-keyexch(7).

"kdf-ukm" (OSSL_EXCHANGE_PARAM_KDF_UKM) <octet string>

See "Common Key Exchange parameters" in provider-keyexch(7).

"cekalg" (OSSL_KDF_PARAM_CEK_ALG) <octet string ptr>

See "KDF Parameters" in provider-kdf(7).

EXAMPLES

The examples assume a host and peer both generate keys using the same named group (or domain parameters). See "Examples" in EVP_PKEY-DH(7). Both the host and peer transfer their public key to each other.

To convert the peer's generated key pair to a public key in DER format in order to transfer to the host:

```
EVP_PKEY *peer_key; /* It is assumed this contains the peers generated key */
unsigned char *peer_pub_der = NULL;
int peer_pub_der_len;
peer_pub_der_len = i2d_PUBKEY(peer_key, &peer_pub_der);
...
OPENSSL_free(peer_pub_der);
```

To convert the received peer's public key from DER format on the host:

```
const unsigned char *pd = peer_pub_der;
EVP_PKEY *peer_pub_key = d2i_PUBKEY(NULL, &pd, peer_pub_der_len);
...
EVP_PKEY_free(peer_pub_key);
```

To derive a shared secret on the host using the host's key and the peer's public key:

```
/* It is assumed that the host_key and peer_pub_key are set up */
void derive_secret(EVP_KEY *host_key, EVP_PKEY *peer_pub_key)
{
    unsigned int pad = 1;
    OSSL_PARAM params[2];
    unsigned char *secret = NULL;
    size_t secret_len = 0;
    EVP_PKEY_CTX *dctx = EVP_PKEY_CTX_new_from_pkey(NULL, host_key, NULL);
    EVP_PKEY_derive_init(dctx);
    /* Optionally set the padding */
    params[0] = OSSL_PARAM_construct_uint(OSSL_EXCHANGE_PARAM_PAD, &pad);
    params[1] = OSSL_PARAM_construct_end();
```

```
    EVP_PKEY_CTX_set_params(dctx, params);
    EVP_PKEY_derive_set_peer(dctx, peer_pub_key);
    /* Get the size by passing NULL as the buffer */
    EVP_PKEY_derive(dctx, NULL, &secret_len);
    secret = OPENSSL_zalloc(secret_len);
    EVP_PKEY_derive(dctx, secret, &secret_len);
    ...
    OPENSSL_clear_free(secret, secret_len);
    EVP_PKEY_CTX_free(dctx);
}
```

Very similar code can be used by the peer to derive the same shared secret using the host's public key and the peer's generated key pair.

SEE ALSO

EVP_PKEY-DH(7), EVP_PKEY-FFC(7), EVP_PKEY(3), provider-keyexch(7), provider-keymgmt(7), OSSL_PROVIDER-default(7), OSSL_PROVIDER-FIPS(7),

COPYRIGHT