Rocky Enterprise Linux 9.2 Manual Pages on command 'EVP_KEYMGMT-DSA.7ossl'

$ man EVP_KEYMGMT-DSA.7ossl

EVP_PKEY-DSA(7ossl) OpenSSL EVP_PKEY-DSA(7ossl)

NAME

EVP_PKEY-DSA, EVP_KEYMGMT-DSA - EVP_PKEY DSA keytype and algorithm support

DESCRIPTION

For DSA the FIPS186-4 standard specifies that the values used for FFC parameter generation are also required for parameter validation. This means that optional FFC domain parameter values for seed, pcounter and gindex may need to be stored for validation purposes. For DSA these fields are not stored in the ASN1 data so they need to be stored externally if validation is required.

DSA parameters

The DSA key type supports the FFC parameters (see "FFC parameters" in EVP_PKEY-FFC(7)).

DSA key generation parameters

The DSA key type supports the FFC key generation parameters (see "FFC key generation parameters" in EVP_PKEY-FFC(7)

The following restrictions apply to the "pbits" field:

For "fips186_4" this must be either 2048 or 3072. For "fips186_2" this must be 1024. For "group" this can be any one of 2048, 3072, 4096, 6144 or 8192.

DSA key validation

For DSA keys, EVP_PKEY_param_check(3) behaves in the following way: The OpenSSL FIPS provider conforms to the rules within the FIPS186-4 standard for FFC parameter validation. For backwards compatibility the OpenSSL default provider uses a much simpler check (see below) for parameter validation, unless the seed parameter is set.

For DSA keys, EVP_PKEY_param_check_quick(3) behaves in the following way: A simple check of L and N and partial g is performed. The default provider also supports validation of legacy "fips186_2" keys.

For DSA keys, EVP_PKEY_public_check(3), EVP_PKEY_private_check(3) and EVP_PKEY_pairwise_check(3) the OpenSSL default and FIPS providers conform to the rules within SP800-56Ar3 for public, private and pairwise tests respectively.

EXAMPLES

An EVP_PKEY context can be obtained by calling:

```
EVP_PKEY_CTX *pctx = EVP_PKEY_CTX_new_from_name(NULL, "DSA", NULL);
```

The DSA domain parameters can be generated by calling:

```
    unsigned int pbits = 2048;
    unsigned int qbits = 256;
    int gindex = 1;
    OSSL_PARAM params[5];
    EVP_PKEY *param_key = NULL;
    EVP_PKEY_CTX *pctx = NULL;

    pctx = EVP_PKEY_CTX_new_from_name(NULL, "DSA", NULL);
    EVP_PKEY_paramgen_init(pctx);

    params[0] = OSSL_PARAM_construct_uint("pbits", &pbits);
    params[1] = OSSL_PARAM_construct_uint("qbits", &qbits);
    params[2] = OSSL_PARAM_construct_int("gindex", &gindex);
    params[3] = OSSL_PARAM_construct_utf8_string("digest", "SHA384", 0);
    params[4] = OSSL_PARAM_construct_end();
    EVP_PKEY_CTX_set_params(pctx, params);

    EVP_PKEY_generate(pctx, &param_key);
    EVP_PKEY_CTX_free(pctx);

    EVP_PKEY_print_params(bio_out, param_key, 0, NULL);
```

A DSA key can be generated using domain parameters by calling:

```
    EVP_PKEY *key = NULL;
    EVP_PKEY_CTX *gctx = NULL;

    gctx = EVP_PKEY_CTX_new_from_pkey(NULL, param_key, NULL);
    EVP_PKEY_keygen_init(gctx);
    EVP_PKEY_generate(gctx, &key);
    EVP_PKEY_CTX_free(gctx);
    EVP_PKEY_print_private(bio_out, key, 0, NULL);
```

CONFORMING TO

The following sections of FIPS186-4:

A.1.1.2 Generation of Probable Primes p and q Using an Approved Hash Function.

A.2.3 Generation of canonical generator g.

A.2.1 Unverifiable Generation of the Generator g.

SEE ALSO

EVP_PKEY-FFC(7), EVP_SIGNATURE-DSA(7) EVP_PKEY(3), provider-keymgmt(7), EVP_KEYMGMT(3), OSSL_PROVIDER-default(7), OSSL_PROVIDER-FIPS(7)

COPYRIGHT