Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'DSA_verify.3ossl' command

$ man DSA_verify.3ossl

DSA_SIGN(3ossl) OpenSSL DSA_SIGN(3ossl)

NAME

DSA_sign, DSA_sign_setup, DSA_verify - DSA signatures

SYNOPSIS

```
#include <openssl/dsa.h>
```

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining OPENSSL_API_COMPAT with a suitable version value, see openssl_user_macros(7):

```
int DSA_sign(int type, const unsigned char *dgst, int len,
             unsigned char *sigret, unsigned int *siglen, DSA *dsa);

int DSA_sign_setup(DSA *dsa, BN_CTX *ctx, BIGNUM **kinvp, BIGNUM **rp);

int DSA_verify(int type, const unsigned char *dgst, int len,
               unsigned char *sigbuf, int siglen, DSA *dsa);
```

DESCRIPTION

All of the functions described on this page are deprecated.

Applications should instead use EVP_PKEY_sign_init(3),

EVP_PKEY_sign(3), EVP_PKEY_verify_init(3) and EVP_PKEY_verify(3).

DSA_sign() computes a digital signature on the len byte message digest dgst using the private key dsa and places its ASN.1 DER encoding at sigret. The length of the signature is places in *siglen. sigret must point to DSA_size(dsa) bytes of memory.

DSA_sign_setup() is defined only for backward binary compatibility and should not be used. Since OpenSSL 1.1.0 the DSA type is opaque and the output of DSA_sign_setup() cannot be used anyway: calling this function will only cause overhead, and does not affect the actual signature (pre-)computation.

DSA_verify() verifies that the signature sigbuf of size siglen matches a given message digest dgst of size len. dsa is the signer's public key.

The type parameter is ignored.

The random generator must be seeded when DSA_sign() (or DSA_sign_setup()) is called. If the automatic seeding or reseeding of the OpenSSL CSPRNG fails due to external circumstances (see RAND(7)), the operation will fail.

RETURN VALUES

DSA_sign() and DSA_sign_setup() return 1 on success, 0 on error. DSA_verify() returns 1 for a valid signature, 0 for an incorrect signature and -1 on error. The error codes can be obtained by ERR_get_error(3).

CONFORMING TO

US Federal Information Processing Standard FIPS186-4 (Digital Signature Standard, DSS), ANSI X9.30

SEE ALSO

DSA_new(3), ERR_get_error(3), RAND_bytes(3), DSA_do_sign(3), RAND(7)

HISTORY

All of these functions were deprecated in OpenSSL 3.0.

COPYRIGHT