NAME

DSA_sign, DSA_sign_setup, DSA_verify - DSA signatures

SYNOPSIS

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

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

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() may be used to precompute part of the signing operation in case signature generation is time-critical. It expects **dsa** to contain DSA parameters. It places the precomputed values in newly allocated **BIGNUM**s at ***kinvp** and ***rp**, after freeing the old ones unless ***kinvp** and ***rp** are NULL. These values may be passed to *DSA_sign()* in **dsa->kinv** and **dsa->r**. **ctx** is a pre-allocated **BN_CTX** or NULL.

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 PRNG must be seeded before *DSA_sign()* (or *DSA_sign_setup()*) is called.

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 FIPS 186 (Digital Signature Standard, DSS), ANSI X9.30

SEE ALSO

dsa(3), *ERR_get_error(3)*, *rand(3)*, *DSA_do_sign(3)*

HISTORY

DSA_sign() and *DSA_verify()* are available in all versions of SSLeay. *DSA_sign_setup()* was added in SSLeay 0.8.