NAME

dsa - Digital Signature Algorithm

SYNOPSIS

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

DSA * DSA_new(void);
void DSA_free(DSA *dsa);

int DSA_size(const DSA *dsa);

DSA * DSA_generate_parameters(int bits, unsigned char *seed,
int seed_len, int *counter_ret, unsigned long *h_ret,
void (*callback)(int, int, void *), void *cb_arg);

DH * DSA_dup_DH(const DSA *r);

int DSA_generate_key(DSA *dsa);

int DSA_sign(int dummy, 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 dummy, const unsigned char *dgst, int len,
const unsigned char *sigbuf, int siglen, DSA *dsa);

void DSA_set_default_method(const DSA_METHOD *meth);
const DSA_METHOD *DSA_get_default_method(void);
int DSA_set_method(DSA *dsa, const DSA_METHOD *meth);
DSA *DSA_new_method(ENGINE *engine);
const DSA_METHOD *DSA_OpenSSL(void);

int DSA_get_ex_new_index(long argl, char *argp, int (*new_func)(),
int (*dup_func)(), void (*free_func)());
int DSA_set_ex_data(DSA *d, int idx, char *arg);
char *DSA_get_ex_data(DSA *d, int idx);

DSA_SIG *DSA_SIG_new(void);
void DSA_SIG_free(DSA_SIG *a);
int i2d_DSA_SIG(const DSA_SIG *a, unsigned char **pp);
DSA_SIG *d2i_DSA_SIG(DSA_SIG **v, unsigned char **pp, long length);

DSA_SIG *DSA_do_sign(const unsigned char *dgst, int dlen, DSA *dsa);
int DSA_do_verify(const unsigned char *dgst, int dgst_len,
DSA_SIG *sig, DSA *dsa);

DSA * d2i_DSAPublicKey(DSA **a, unsigned char **pp, long length);
DSA * d2i_DSAPrivateKey(DSA **a, unsigned char **pp, long length);
DSA * d2i_DSAparams(DSA **a, unsigned char **pp, long length);
int i2d_DSAPublicKey(const DSA *a, unsigned char **pp);
int i2d_DSAPrivateKey(const DSA *a, unsigned char **pp);
int i2d_DSAparams(const DSA *a,unsigned char **pp);

int DSAparams_print(BIO *bp, const DSA *x);
```

```
int DSAparams_print_fp(FILE *fp, const DSA *x);
int DSA_print(BIO *bp, const DSA *x, int off);
int DSA_print_fp(FILE *bp, const DSA *x, int off);
```

DESCRIPTION

These functions implement the Digital Signature Algorithm (DSA). The generation of shared DSA parameters is described in *DSA_generate_parameters(3)*; *DSA_generate_key(3)* describes how to generate a signature key. Signature generation and verification are described in *DSA_sign(3)*.

The **DSA** structure consists of several BIGNUM components.

```
struct
{
BIGNUM *p; // prime number (public)
BIGNUM *q; // 160-bit subprime, q | p-1 (public)
BIGNUM *g; // generator of subgroup (public)
BIGNUM *priv_key; // private key x
BIGNUM *pub_key; // public key y = gx
// ...
}
DSA;
```

In public keys, **priv_key** is NULL.

Note that DSA keys may use non-standard **DSA_METHOD** implementations, either directly or by the use of **ENGINE** modules. In some cases (eg. an ENGINE providing support for hardware-embedded keys), these BIGNUM values will not be used by the implementation or may be used for alternative data storage. For this reason, applications should generally avoid using DSA structure elements directly and instead use API functions to query or modify keys.

CONFORMING TO

US Federal Information Processing Standard FIPS 186 (Digital Signature Standard, DSS), ANSI X9.30

SEE ALSO

bn(3), *dh(3)*, *err(3)*, *rand(3)*, *rsa(3)*, *sha(3)*, *engine(3)*, *DSA_new(3)*, *DSA_size(3)*, *DSA_generate_parameters(3)*, *DSA_dup_DH(3)*, *DSA_generate_key(3)*, *DSA_sign(3)*, *DSA_set_method(3)*, *DSA_get_ex_new_index(3)*, *RSA_print(3)*