NAME

EVP_DigestSignInit, EVP_DigestSignUpdate, EVP_DigestSignFinal - EVP signing functions

SYNOPSIS

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

int EVP_DigestSignInit(EVP_MD_CTX *ctx, EVP_PKEY_CTX **pctx,
const EVP_MD *type, ENGINE *e, EVP_PKEY *pkey);
int EVP_DigestSignUpdate(EVP_MD_CTX *ctx, const void *d, unsigned int cnt);
int EVP_DigestSignFinal(EVP_MD_CTX *ctx, unsigned char *sig, size_t *siglen);
```

DESCRIPTION

The EVP signature routines are a high level interface to digital signatures.

EVP_DigestSignInit() sets up signing context **ctx** to use digest **type** from ENGINE **impl** and private key **pkey**. **ctx** must be initialized with *EVP_MD_CTX_init()* before calling this function. If **pctx** is not NULL the EVP_PKEY_CTX of the signing operation will be written to ***pctx**: this can be used to set alternative signing options.

EVP_DigestSignUpdate() hashes **cnt** bytes of data at **d** into the signature context **ctx**. This function can be called several times on the same **ctx** to include additional data. This function is currently implemented usig a macro.

EVP_DigestSignFinal() signs the data in **ctx** places the signature in **sig**. If**sig** is **NULL** then the maximum size of the output buffer is written to the **siglen** parameter. If **sig** is not **NULL** then before the call the **siglen** parameter should contain the length of the **sig** buffer, if the call is successful the signature is written to **sig** and the amount of data written to **siglen**.

RETURN VALUES

EVP_DigestSignInit() EVP_DigestSignUpdate() and *EVP_DigestSignaFinal()* return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

The error codes can be obtained from *ERR_get_error(3)*.

NOTES

The **EVP** interface to digital signatures should almost always be used in preference to the low level interfaces. This is because the code then becomes transparent to the algorithm used and much more flexible.

In previous versions of OpenSSL there was a link between message digest types and public key algorithms. This meant that ''clone'' digests such as *EVP_dss1()* needed to be used to sign using SHA1 and DSA. This is no longer necessary and the use of clone digest is now discouraged.

For some key types and parameters the random number generator must be seeded or the operation will fail.

The call to *EVP_DigestSignFinal()* internally finalizes a copy of the digest context. This means that calls to *EVP_DigestSignUpdate()* and *EVP_DigestSignFinal()* can be called later to digest and sign additional data.

Since only a copy of the digest context is ever finalized the context must be cleaned up after use by calling *EVP_MD_CTX_cleanup()* or a memory leak will occur.

The use of *EVP_PKEY_size()* with these functions is discouraged because some signature operations may have a signature length which depends on the parameters set. As a result *EVP_PKEY_size()* would have to return a value which indicates the maximum possible signature for any set of parameters.

SEE ALSO

EVP_DigestVerifyInit(3), *EVP_DigestInit(3)*, *err(3)*, *evp(3)*, *hmac(3)*, *md2(3)*, *md5(3)*, *mdc2(3)*, *ripemd(3)*, *sha(3)*, *dgst(1)*

HISTORY

EVP_DigestSignInit(), *EVP_DigestSignUpdate()* and *EVP_DigestSignFinal()* were first added to OpenSSL 1.0.0.