NAME

SM2 – Chinese SM2 signature and encryption algorithm support

DESCRIPTION

The **SM2** algorithm was first defined by the Chinese national standard GM/T 0003–2012 and was later standardized by ISO as ISO/IEC 14888. **SM2** is actually an elliptic curve based algorithm. The current implementation in OpenSSL supports both signature and encryption schemes via the EVP interface.

When doing the **SM2** signature algorithm, it requires a distinguishing identifier to form the message prefix which is hashed before the real message is hashed.

NOTES

SM2 signatures can be generated by using the 'DigestSign' series of APIs, for instance, **EVP_DigestSignInit()**, **EVP_DigestSignUpdate()** and **EVP_DigestSignFinal()**. Ditto for the verification process by calling the 'DigestVerify' series of APIs.

There are several special steps that need to be done before computing an **SM2** signature.

The **EVP_PKEY** structure will default to using ECDSA for signatures when it is created. It should be set to **EVP_PKEY_SM2** by calling:

```
EVP_PKEY_set_alias_type(pkey, EVP_PKEY_SM2);
```

Then an ID should be set by calling:

```
EVP_PKEY_CTX_set1_id(pctx, id, id_len);
```

When calling the **EVP_DigestSignInit()** or **EVP_DigestVerifyInit()** functions, a pre-allocated **EVP_PKEY_CTX** should be assigned to the **EVP_MD_CTX**. This is done by calling:

```
EVP_MD_CTX_set_pkey_ctx(mctx, pctx);
```

And normally there is no need to pass a **pctx** parameter to **EVP_DigestSignInit()** or **EVP_DigestVerifyInit()** in such a scenario.

EXAMPLE

This example demonstrates the calling sequence for using an **EVP_PKEY** to verify a message with the SM2 signature algorithm and the SM3 hash algorithm:

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

/* obtain an EVP_PKEY using whatever methods... */
EVP_PKEY_set_alias_type(pkey, EVP_PKEY_SM2);
mctx = EVP_MD_CTX_new();
pctx = EVP_PKEY_CTX_new(pkey, NULL);
EVP_PKEY_CTX_set1_id(pctx, id, id_len);
EVP_MD_CTX_set_pkey_ctx(mctx, pctx);;
EVP_DigestVerifyInit(mctx, NULL, EVP_sm3(), NULL, pkey);
EVP_DigestVerifyUpdate(mctx, msg, msg_len);
EVP_DigestVerifyFinal(mctx, sig, sig_len)
```

SEE ALSO

EVP_PKEY_CTX_new (3), **EVP_PKEY_set_alias_type** (3), **EVP_DigestSignInit** (3), **EVP_DigestVerifyInit** (3), **EVP_PKEY_CTX_set1_id** (3), **EVP_MD_CTX_set_pkey_ctx** (3)

COPYRIGHT