NAME

PKCS7_verify, PKCS7_get0_signers - verify a PKCS#7 signedData structure

SYNOPSIS

```
#include <openssl/pkcs7.h>

int PKCS7_verify(PKCS7 *p7, STACK_OF(X509) *certs, X509_STORE *store, BIO *indat

STACK_OF(X509) *PKCS7_get0_signers(PKCS7 *p7, STACK_OF(X509) *certs, int flags);
```

DESCRIPTION

PKCS7_verify() verifies a PKCS#7 signedData structure. **p7** is the PKCS7 structure to verify. **certs** is a set of certificates in which to search for the signer's certificate. **store** is a trusted certificate store (used for chain verification). **indata** is the signed data if the content is not present in **p7** (that is it is detached). The content is written to **out** if it is not NULL.

flags is an optional set of flags, which can be used to modify the verify operation.

PKCS7_get0_signers() retrieves the signer's certificates from **p7**, it does **not** check their validity or whether any signatures are valid. The **certs** and **flags** parameters have the same meanings as in *PKCS7_verify()*.

VERIFY PROCESS

Normally the verify process proceeds as follows.

Initially some sanity checks are performed on **p7**. The type of **p7** must be signedData. There must be at least one signature on the data and if the content is detached **indata** cannot be **NULL**. If the content is not detached and **indata** is not **NULL**, then the structure has both embedded and external content. To treat this as an error, use the flag **PKCS7_NO_DUAL_CONTENT**. The default behavior allows this, for compatibility with older versions of OpenSSL.

An attempt is made to locate all the signer's certificates, first looking in the **certs** parameter (if it is not **NULL**) and then looking in any certificates contained in the **p7** structure itself. If any signer's certificates cannot be located the operation fails.

Each signer's certificate is chain verified using the **smimesign** purpose and the supplied trusted certificate store. Any internal certificates in the message are used as untrusted CAs. If any chain verify fails an error code is returned.

Finally the signed content is read (and written to **out** is it is not NULL) and the signature's checked.

If all signature's verify correctly then the function is successful.

Any of the following flags (ored together) can be passed in the **flags** parameter to change the default verify behaviour. Only the flag **PKCS7_NOINTERN** is meaningful to *PKCS7_get0_signers()*.

If **PKCS7_NOINTERN** is set the certificates in the message itself are not searched when locating the signer's certificate. This means that all the signers certificates must be in the **certs** parameter.

If the **PKCS7_TEXT** flag is set MIME headers for type **text/plain** are deleted from the content. If the content is not of type **text/plain** then an error is returned.

If **PKCS7_NOVERIFY** is set the signer's certificates are not chain verified.

If **PKCS7_NOCHAIN** is set then the certificates contained in the message are not used as untrusted CAs. This means that the whole verify chain (apart from the signer's certificate) must be contained in the trusted store.

If **PKCS7_NOSIGS** is set then the signatures on the data are not checked.

NOTES

One application of **PKCS7_NOINTERN** is to only accept messages signed by a small number of certificates. The acceptable certificates would be passed in the **certs** parameter. In this case if the signer is not one of the certificates supplied in **certs** then the verify will fail because the signer cannot be found.

Care should be taken when modifying the default verify behaviour, for example setting

PKCS7_NOVERIFY|PKCS7_NOSIGS will totally disable all verification and any signed message will be considered valid. This combination is however useful if one merely wishes to write the content to **out** and its validity is not considered important.

Chain verification should arguably be performed using the signing time rather than the current time. However since the signing time is supplied by the signer it cannot be trusted without additional evidence (such as a trusted timestamp).

RETURN VALUES

PKCS7_verify() returns one for a successful verification and zero if an error occurs.

PKCS7_get0_signers() returns all signers or **NULL** if an error occurred.

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

BUGS

The trusted certificate store is not searched for the signers certificate, this is primarily due to the inadequacies of the current **X509_STORE** functionality.

The lack of single pass processing and need to hold all data in memory as mentioned in *PKCS7_sign()* also applies to *PKCS7_verify()*.

SEE ALSO

ERR_get_error(3) , *PKCS7_sign(3)*

COPYRIGHT