NAME

PKCS7_sign_add_signer, PKCS7_add_certificate, PKCS7_add_crl - add information to PKCS7 structure

SYNOPSIS

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

 PKCS7_SIGNER_INFO *PKCS7_sign_add_signer(PKCS7 *p7, X509 *signcert,
                                          EVP_PKEY *pkey, const EVP_MD *md, int flags);
 int PKCS7_add_certificate(PKCS7 *p7, X509 *cert);
 int PKCS7_add_crl(PKCS7 *p7, X509_CRL *crl);
```

DESCRIPTION

PKCS7_sign_add_signer() adds a signer with certificate *signcert* and private key *pkey* using message digest *md* to a PKCS7 signed data structure *p7*.

The **PKCS7** structure should be obtained from an initial call to **PKCS7_sign()** with the flag **PKCS7_PARTIAL** set or in the case or re-signing a valid PKCS#7 signed data structure.

If the *md* parameter is NULL then the default digest for the public key algorithm will be used.

Unless the **PKCS7_REUSE_DIGEST** flag is set the returned **PKCS7** structure is not complete and must be finalized either by streaming (if applicable) or a call to **PKCS7_final()**.

NOTES

The main purpose of this function is to provide finer control over a PKCS#7 signed data structure where the simpler **PKCS7_sign()** function defaults are not appropriate. For example if multiple signers or non default digest algorithms are needed.

Any of the following flags (ored together) can be passed in the *flags* parameter.

If **PKCS7_REUSE_DIGEST** is set then an attempt is made to copy the content digest value from the **PKCS7** structure: to add a signer to an existing structure. An error occurs if a matching digest value cannot be found to copy. The returned **PKCS7** structure will be valid and finalized when this flag is set.

If **PKCS7_PARTIAL** is set in addition to **PKCS7_REUSE_DIGEST** then the **PKCS7_SIGNER_INO** structure will not be finalized so additional attributes can be added. In this case an explicit call to **PKCS7_SIGNER_INFO_sign()** is needed to finalize it.

If **PKCS7_NOCERTS** is set the signer's certificate will not be included in the **PKCS7** structure, the signer's certificate must still be supplied in the *signcert* parameter though. This can reduce the size of the signature if the signers certificate can be obtained by other means: for example a previously signed message.

The signedData structure includes several PKCS#7 authenticatedAttributes including the signing time, the PKCS#7 content type and the supported list of ciphers in an SMIMECapabilities attribute. If **PKCS7_NOATTR** is set then no authenticatedAttributes will be used. If **PKCS7_NOSMIMECAP** is set then just the SMIMECapabilities are omitted.

If present the SMIMECapabilities attribute indicates support for the following algorithms: triple DES, 128 bit RC2, 64 bit RC2, DES and 40 bit RC2. If any of these algorithms is disabled then it will not be included.

PKCS7_sign_add_signers() returns an internal pointer to the **PKCS7_SIGNER_INFO** structure just added, which can be used to set additional attributes before it is finalized.

PKCS7_add_certificate() adds to the **PKCS7** structure *p7* the certificate *cert*, which may be an end-entity (signer) certificate or a CA certificate useful for chain building. This is done internally by **PKCS7_sign_ex**(3) and similar signing functions. It may have to be used before calling **PKCS7_verify**(3) in order to provide any missing certificate(s) needed for verification.

PKCS7_add_crl() adds the CRL *crl* to the **PKCS7** structure *p7*. This may be called to provide certificate status information to be included when signing or to use when verifying the **PKCS7** structure.

RETURN VALUES

PKCS7_sign_add_signers() returns an internal pointer to the **PKCS7_SIGNER_INFO** structure just added or NULL if an error occurs.

PKCS7_add_certificate() and **PKCS7_add_crl()** return 1 on success, 0 on error.

SEE ALSO

ERR_get_error(3), **PKCS7_sign_ex**(3), **PKCS7_final**(3), **PKCS7_verify**(3)

HISTORY

The **PPKCS7_sign_add_signer()** function was added in OpenSSL 1.0.0.

COPYRIGHT