

**NAME**

PKCS7\_sign\_ex, PKCS7\_sign - create a PKCS#7 signedData structure

**SYNOPSIS**

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

```
PKCS7 *PKCS7_sign_ex(X509 *signcert, EVP_PKEY *pkey, STACK_OF(X509) *certs,  
                    BIO *data, int flags, OSSL_LIB_CTX *libctx,  
                    const char *propq);
```

```
PKCS7 *PKCS7_sign(X509 *signcert, EVP_PKEY *pkey, STACK_OF(X509) *certs,  
                 BIO *data, int flags);
```

**DESCRIPTION**

**PKCS7\_sign\_ex()** creates and returns a PKCS#7 signedData structure. *signcert* is the certificate to sign with, *pkey* is the corresponding private key. *certs* is an optional set of extra certificates to include in the PKCS#7 structure (for example any intermediate CAs in the chain). The library context *libctx* and property query *propq* are used when retrieving algorithms from providers.

The data to be signed is read from BIO *data*.

*flags* is an optional set of flags.

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

Many S/MIME clients expect the signed content to include valid MIME headers. If the **PKCS7\_TEXT** flag is set MIME headers for type "text/plain" are prepended to the data.

If **PKCS7\_NOCERTS** is set the signer's certificate and the extra *certs* 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 signatures if the signer's certificates can be obtained by other means: for example a previously signed message.

The data being signed is included in the PKCS7 structure, unless **PKCS7\_DETACHED** is set in which case it is omitted. This is used for PKCS7 detached signatures which are used in S/MIME plaintext signed messages for example.

Normally the supplied content is translated into MIME canonical format (as required by the S/MIME specifications) if **PKCS7\_BINARY** is set no translation occurs. This option should be used if the supplied data is in binary format otherwise the translation will corrupt it.

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.

If the flag **PKCS7\_STREAM** is set then the returned **PKCS7** structure is just initialized ready to perform the signing operation. The signing is however **not** performed and the data to be signed is not read from the `data` parameter. Signing is deferred until after the data has been written. In this way data can be signed in a single pass.

If the **PKCS7\_PARTIAL** flag is set a partial **PKCS7** structure is output to which additional signers and capabilities can be added before finalization.

If the flag **PKCS7\_STREAM** is set the returned **PKCS7** structure is **not** complete and outputting its contents via a function that does not properly finalize the **PKCS7** structure will give unpredictable results.

Several functions including `SMIME_write_PKCS7()`, `i2d_PKCS7_bio_stream()`, `PEM_write_bio_PKCS7_stream()` finalize the structure. Alternatively finalization can be performed by obtaining the streaming ASN1 **BIO** directly using `BIO_new_PKCS7()`.

If a signer is specified it will use the default digest for the signing algorithm. This is **SHA1** for both RSA and DSA keys.

The `certs`, `signcert` and `pkey` parameters can all be NULL if the **PKCS7\_PARTIAL** flag is set. One or more signers can be added using the function `PKCS7_sign_add_signer()`. `PKCS7_final()` must also be called to finalize the structure if streaming is not enabled. Alternative signing digests can also be specified using this method.

If `signcert` and `pkey` are NULL then a certificates only PKCS#7 structure is output.

In versions of OpenSSL before 1.0.0 the `signcert` and `pkey` parameters must not be NULL.

`PKCS7_sign()` is like `PKCS7_sign_ex()` except that it uses default values of NULL for the library context `libctx` and the property query `propq`. This is retained for API backward compatibility.

**BUGS**

Some advanced attributes such as counter signatures are not supported.

**RETURN VALUES**

**PKCS7\_sign\_ex()** and **PKCS7\_sign()** return either a valid PKCS7 structure or NULL if an error occurred. The error can be obtained from **ERR\_get\_error(3)**.

**SEE ALSO**

**ERR\_get\_error(3)**, **PKCS7\_verify(3)**

**HISTORY**

The function **PKCS7\_sign\_ex()** was added in OpenSSL 3.0.

The **PKCS7\_PARTIAL** flag, and the ability for *certs*, *signcert*, and *pkey* parameters to be NULL were added in OpenSSL 1.0.0.

The **PKCS7\_STREAM** flag was added in OpenSSL 1.0.0.

**COPYRIGHT**

Copyright 2002-2023 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <<https://www.openssl.org/source/license.html>>.