

**NAME**

CMS\_SignerInfo\_set1\_signer\_cert, CMS\_get0\_SignerInfos, CMS\_SignerInfo\_get0\_signer\_id, CMS\_SignerInfo\_get0\_signature, CMS\_SignerInfo\_cert\_cmp - CMS signedData signer functions

**SYNOPSIS**

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

```
STACK_OF(CMS_SignerInfo) *CMS_get0_SignerInfos(CMS_ContentInfo *cms);
```

```
int CMS_SignerInfo_get0_signer_id(CMS_SignerInfo *si, ASN1_OCTET_STRING **keyid,  
                                X509_NAME **issuer, ASN1_INTEGER **sno);
```

```
ASN1_OCTET_STRING *CMS_SignerInfo_get0_signature(CMS_SignerInfo *si);
```

```
int CMS_SignerInfo_cert_cmp(CMS_SignerInfo *si, X509 *cert);
```

```
void CMS_SignerInfo_set1_signer_cert(CMS_SignerInfo *si, X509 *signer);
```

**DESCRIPTION**

The function **CMS\_get0\_SignerInfos()** returns all the CMS\_SignerInfo structures associated with a CMS signedData structure.

**CMS\_SignerInfo\_get0\_signer\_id()** retrieves the certificate signer identifier associated with a specific CMS\_SignerInfo structure **si**. Either the keyidentifier will be set in **keyid** or **both** issuer name and serial number in **issuer** and **sno**.

**CMS\_SignerInfo\_get0\_signature()** retrieves the signature associated with **si** in a pointer to an ASN1\_OCTET\_STRING structure. This pointer returned corresponds to the internal signature value if **si** so it may be read or modified.

**CMS\_SignerInfo\_cert\_cmp()** compares the certificate **cert** against the signer identifier **si**. It returns zero if the comparison is successful and non zero if not.

**CMS\_SignerInfo\_set1\_signer\_cert()** sets the signers certificate of **si** to **signer**.

**NOTES**

The main purpose of these functions is to enable an application to lookup signers certificates using any appropriate technique when the simpler method of **CMS\_verify()** is not appropriate.

In typical usage and application will retrieve all CMS\_SignerInfo structures using **CMS\_get0\_SignerInfo()** and retrieve the identifier information using CMS. It will then obtain the signer certificate by some unspecified means (or return an error if it cannot be found) and set it using **CMS\_SignerInfo\_set1\_signer\_cert()**.

Once all signer certificates have been set **CMS\_verify()** can be used.

Although **CMS\_get0\_SignerInfos()** can return NULL if an error occurs **or** if there are no signers this is not a problem in practice because the only error which can occur is if the **cms** structure is not of type signedData due to application error.

## RETURN VALUES

**CMS\_get0\_SignerInfos()** returns all CMS\_SignerInfo structures, or NULL there are no signers or an error occurs.

**CMS\_SignerInfo\_get0\_signer\_id()** returns 1 for success and 0 for failure.

**CMS\_SignerInfo\_cert\_cmp()** returns 0 for a successful comparison and non zero otherwise.

**CMS\_SignerInfo\_set1\_signer\_cert()** does not return a value.

Any error can be obtained from **ERR\_get\_error(3)**

## SEE ALSO

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

## COPYRIGHT

Copyright 2008-2018 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>.