NAME

openssl−cmp − Certificate Management Protocol (CMP, RFC 4210) application

SYNOPSIS

openssl cmp [**−help**] [**−config** *filename*] [**−section** *names*] [**−verbosity** *level*]

Generic message options:

[**−cmd** *ir|cr|kur|p10cr|rr|genm*] [**−infotype** *name*] [**−geninfo** *OID:int:N*]

Certificate enrollment options:

[**−newkey** *filename|uri*] [**−newkeypass** *arg*] [**−subject** *name*] [**−issuer** *name*] [**−days** *number*] [**−reqexts** *name*] [**−sans** *spec*] [**−san_nodefault**] [**−policies** *name*] [**−policy_oids** *names*] [**−policy_oids_critical**] [**−popo** *number*] [**−csr** *filename*] [**−out_trusted** *filenames|uris*] [**−implicit_confirm**] [**−disable_confirm**] [**−certout** *filename*] [**−chainout** *filename*]

Certificate enrollment and revocation options:

[**−oldcert** *filename|uri*] [**−revreason** *number*]

Message transfer options:

[**−server** *[http[s]://][userinfo@]host[:port][/path][?query][#fragment]*] [**−proxy** *[http[s]://][userinfo@]host[:port][/path][?query][#fragment]*] [**−no_proxy** *addresses*] [**−recipient** *name*] [**−path** *remote_path*] [**−keep_alive** *value*] [**−msg_timeout** *seconds*] [**−total_timeout** *seconds*]

Server authentication options:

[**−trusted** *filenames|uris*] [**−untrusted** *filenames|uris*] [**−srvcert** *filename|uri*] [**−expect_sender** *name*] [**−ignore_keyusage**] [**−unprotected_errors**] [**−extracertsout** *filename*] [**−cacertsout** *filename*]

Client authentication and protection options:

[**−ref** *value*] [**−secret** *arg*] [**−cert** *filename|uri*] [**−own_trusted** *filenames|uris*] [**−key** *filename|uri*] [**−keypass** *arg*] [**−digest** *name*] [**−mac** *name*] [**−extracerts** *filenames|uris*] [**−unprotected_requests**]

Credentials format options:

[**−certform** *PEM|DER*] [**−keyform** *PEM|DER|P12|ENGINE*] [**−otherpass** *arg*] [**−engine** *id*] [**−provider** *name*] [**−provider−path** *path*] [**−propquery** *propq*]

Random state options:

[**−rand** *files*] [**−writerand** *file*]

TLS connection options:

[**−tls_used**] [**−tls_cert** *filename|uri*] [**−tls_key** *filename|uri*] [**−tls_keypass** *arg*] [**−tls_extra** *filenames|uris*] [**−tls_trusted** *filenames|uris*] [**−tls_host** *name*]

Client-side debugging options:

[**−batch**] [**−repeat** *number*] [**−reqin** *filenames*] [**−reqin_new_tid**] [**−reqout** *filenames*] [**−rspin** *filenames*] [**−rspout** *filenames*] [**−use_mock_srv**]

Mock server options:

[**−port** *number*] [**−max_msgs** *number*] [**−srv_ref** *value*] [**−srv_secret** *arg*] [**−srv_cert** *filename|uri*] [**−srv_key** *filename|uri*] [**−srv_keypass** *arg*] [**−srv_trusted** *filenames|uris*] [**−srv_untrusted** *filenames|uris*] [**−rsp_cert** *filename|uri*] [**−rsp_extracerts** *filenames|uris*] [**−rsp_capubs** *filenames|uris*] [**−poll_count** *number*] [**−check_after** *number*] [**−grant_implicitconf**] [**−pkistatus** *number*] [**−failure** *number*] [**−failurebits** *number*] [**−statusstring** *arg*] [**−send_error**] [**−send_unprotected**] [**−send_unprot_err**] [**−accept_unprotected**] [**−accept_unprot_err**] [**−accept_raverified**]

Certificate verification options, for both CMP and TLS:

[**−allow_proxy_certs**] [**−attime** *timestamp*] [**−no_check_time**] [**−check_ss_sig**] [**−crl_check**] [**−crl_check_all**] [**−explicit_policy**] [**−extended_crl**] [**−ignore_critical**] [**−inhibit_any**] [**−inhibit_map**]

[**−partial_chain**] [**−policy** *arg*] [**−policy_check**] [**−policy_print**] [**−purpose** *purpose*] [**−suiteB_128**] [**−suiteB_128_only**] [**−suiteB_192**] [**−trusted_first**] [**−no_alt_chains**] [**−use_deltas**] [**−auth_level** *num*] [**−verify_depth** *num*] [**−verify_email** *email*] [**−verify_hostname** *hostname*] [**−verify_ip** *ip*] [**−verify_name** *name*] [**−x509_strict**] [**−issuer_checks**]

DESCRIPTION

The **cmp** command is a client implementation for the Certificate Management Protocol (CMP) as defined in RFC4210. It can be used to request certificates from a CA server, update their certificates, request certificates to be revoked, and perform other types of CMP requests.

OPTIONS

−help

Display a summary of all options

−config *filename*

Configuration file to use. An empty string `""` means none. Default filename is from the environment variable `OPENSSL_CONF`.

−section *names*

Section(s) to use within config file defining CMP options. An empty string `""` means no specific section. Default is `cmp`.

Multiple section names may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "..."). Contents of sections named later may override contents of sections named before. In any case, as usual, the `[default]` section and finally the unnamed section (as far as present) can provide per-option fallback values.

−verbosity *level*

Level of verbosity for logging, error output, etc. 0 = EMERG, 1 = ALERT, 2 = CRIT, 3 = ERR, 4 = WARN, 5 = NOTE, 6 = INFO, 7 = DEBUG, 8 = TRACE. Defaults to 6 = INFO.

Generic message options

−cmd *ir|cr|kur|p10cr|rr|genm*

CMP command to execute. Currently implemented commands are:

```
ir    − Initialization Request
cr    − Certificate Request
p10cr − PKCS#10 Certification Request (for legacy support)
kur   − Key Update Request
rr    − Revocation Request
genm  − General Message
```

ir requests initialization of an end entity into a PKI hierarchy by issuing a first certificate.

cr requests issuing an additional certificate for an end entity already initialized to the PKI hierarchy.

p10cr requests issuing an additional certificate similarly to **cr** but using legacy PKCS#10 CSR format.

kur requests a (key) update for an existing certificate.

rr requests revocation of an existing certificate.

genm requests information using a General Message, where optionally included **InfoTypeAndValue**s may be used to state which info is of interest. Upon receipt of the General Response, information about all received ITAV **infoType**s is printed to stdout.

−infotype *name*

Set InfoType name to use for requesting specific info in **genm**, e.g., `signKeyPairTypes`.

−geninfo *OID:int:N*

generalInfo integer values to place in request PKIHeader with given OID, e.g., `1.2.3.4:int:56789`.

Certificate enrollment options

−newkey *filename|uri*

The source of the private or public key for the certificate requested in Initialization Request (IR), Certification Request(CR), or Key Update Request (KUR). Defaults to the public key in the PKCS#10 CSR given with the **−csr** option, the public key of the reference certificate, or the current client key.

−newkeypass *arg*

Pass phrase source for the key given with the **−newkey** option. If not given here, the password will be prompted for if needed.

For more information about the format of *arg* see **openssl−passphrase−options** (1).

−subject *name*

X509 Distinguished Name (DN) of subject to use in the requested certificate template. For KUR, it defaults to the public key in the PKCS#10 CSR given with the **−csr** option, if provided, or of the reference certificate (see **−oldcert**) if provided. This default is used for IR and CR only if no SANs are set. If the NULL-DN (`"/"`) is given then no subject is placed in the template.

If provided and neither **−cert** nor **−oldcert** is given, the subject DN is used as fallback sender of outgoing CMP messages.

The argument must be formatted as */type0=value0/type1=value1/type2=...*. Special characters may be escaped by `\` (backslash); whitespace is retained. Empty values are permitted, but the corresponding type will not be included. Giving a single `/` will lead to an empty sequence of RDNs (a NULL-DN). Multi-valued RDNs can be formed by placing a `+` character instead of a `/` between the AttributeValueAssertions (AVAs) that specify the members of the set. Example:

```
/DC=org/DC=OpenSSL/DC=users/UID=123456+CN=John Doe
```

−issuer *name*

X509 issuer Distinguished Name (DN) of the CA server to place in the requested certificate template in IR/CR/KUR. If the NULL-DN (`"/"`) is given then no issuer is placed in the template.

If provided and neither **−recipient** nor **−srvcert** is given, the issuer DN is used as fallback recipient of outgoing CMP messages.

The argument must be formatted as */type0=value0/type1=value1/type2=...*. For details see the description of the **−subject** option.

−days *number*

Number of days the new certificate is requested to be valid for, counting from the current time of the host. Also triggers the explicit request that the validity period starts from the current time (as seen by the host).

−reqexts *name*

Name of section in OpenSSL config file defining certificate request extensions. If the**−csr** option is present, these extensions augment the extensions contained the given PKCS#10 CSR, overriding any extensions with same OIDs.

−sans *spec*

One or more IP addresses, DNS names, or URIs separated by commas or whitespace (where in the latter case the whole argument must be enclosed in ‘‘...’’) to add as Subject Alternative Name(s) (SAN) certificate request extension. If the special element ‘‘critical’’ is given the SANs are flagged as critical. Cannot be used if any Subject Alternative Name extension is set via **−reqexts**.

−san_nodefault

When Subject Alternative Names are not given via **−sans** nor defined via **−reqexts**, they are copied by default from the reference certificate (see **−oldcert**). This can be disabled by giving the **−san_nodefault** option.

−policies *name*
Name of section in OpenSSL config file defining policies to be set as certificate request extension. This option cannot be used together with **−policy_oids**.

−policy_oids *names*
One or more OID(s), separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "...") to add as certificate policies request extension. This option cannot be used together with **−policies**.

−policy_oids_critical
Flag the policies given with **−policy_oids** as critical.

−popo *number*
Proof-of-Possession (POPO) method to use for IR/CR/KUR; values: `-1`..<2> where `-1` = NONE, `0` = RAVERIFIED, `1` = SIGNATURE (default), `2` = KEYENC.

Note that a signature-based POPO can only be produced if a private key is provided via the **−newkey** or **−key** options.

−csr *filename*
PKCS#10 CSR in PEM or DER format containing a certificate request. With **−cmd** *p10cr* it is used directly in a legacy P10CR message. When used with **−cmd** *ir*, *cr*, or *kur*, it is transformed into the respective regular CMP request. It may also be used with **−cmd** *rr* to specify the certificate to be revoked via the included subject name and public key.

−out_trusted *filenames|uris*
Trusted certificate(s) to use for validating the newly enrolled certificate.

Multiple sources may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "..."). Each source may contain multiple certificates.

The certificate verification options **−verify_hostname**, **−verify_ip**, and **−verify_email** only affect the certificate verification enabled via this option.

−implicit_confirm
Request implicit confirmation of newly enrolled certificates.

−disable_confirm
Do not send certificate confirmation message for newly enrolled certificate without requesting implicit confirmation to cope with broken servers not supporting implicit confirmation correctly. **WARNING:** This leads to behavior violating RFC 4210.

−certout *filename*
The file where the newly enrolled certificate should be saved.

−chainout *filename*
The file where the chain of the newly enrolled certificate should be saved.

Certificate enrollment and revocation options

−oldcert *filename|uri*
The certificate to be updated (i.e., renewed or re-keyed) in Key Update Request (KUR) messages or to be revoked in Revocation Request (RR) messages. For KUR the certificate to be updated defaults to **−cert**, and the resulting certificate is called *reference certificate*. For RR the certificate to be revoked can also be specified using **−csr**.

The reference certificate, if any, is also used for deriving default subject DN and Subject Alternative Names and the default issuer entry in the requested certificate template of an IR/CR/KUR. Its subject is used as sender of outgoing messages if **−cert** is not given. Its issuer is used as default recipient in CMP message headers if neither **−recipient**, **−srvcert**, nor **−issuer** is given.

−revreason *number*
Set CRLReason to be included in revocation request (RR); values: `0`..`10` or `-1` for none (which is the default).

Reason numbers defined in RFC 5280 are:

```
   CRLReason ::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        cACompromise            (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        certificateHold         (6),
        -- value 7 is not used
        removeFromCRL           (8),
        privilegeWithdrawn      (9),
        aACompromise           (10)
    }
```

Message transfer options

−server *[http[s]://][userinfo@]host[:port][/path][?query][#fragment]*

The DNS hostname or IP address and optionally port of the CMP server to connect to using HTTP(S). This excludes *−port* and *−use_mock_srv* and is ignored with *−rspin*.

The scheme `https` may be given only if the **−tls_used** option is used. In this case the default port is 443, else 80. The optional userinfo and fragment components are ignored. Any given query component is handled as part of the path component. If a path is included it provides the default value for the **−path** option.

−proxy *[http[s]://][userinfo@]host[:port][/path][?query][#fragment]*

The HTTP(S) proxy server to use for reaching the CMP server unless **−no_proxy** applies, see below. The proxy port defaults to 80 or 443 if the scheme is `https`; apart from that the optional `http://` or `https://` prefix is ignored (note that TLS may be selected by **−tls_used**), as well as any path, userinfo, and query, and fragment components. Defaults to the environment variable `http_proxy` if set, else `HTTP_PROXY` in case no TLS is used, otherwise `https_proxy` if set, else `HTTPS_PROXY`. This option is ignored if *−server* is not given.

−no_proxy *addresses*

List of IP addresses and/or DNS names of servers not to use an HTTP(S) proxy for, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "..."). Default is from the environment variable `no_proxy` if set, else `NO_PROXY`. This option is ignored if *−server* is not given.

−recipient *name*

Distinguished Name (DN) to use in the recipient field of CMP request message headers, i.e., the CMP server (usually the addressed CA).

The recipient field in the header of a CMP message is mandatory. If not given explicitly the recipient is determined in the following order: the subject of the CMP server certificate given with the **−srvcert** option, the **−issuer** option, the issuer of the certificate given with the **−oldcert** option, the issuer of the CMP client certificate (**−cert** option), as far as any of those is present, else the NULL-DN as last resort.

The argument must be formatted as */type0=value0/type1=value1/type2=...*. For details see the description of the **−subject** option.

−path *remote_path*

HTTP path at the CMP server (aka CMP alias) to use for POST requests. Defaults to any path given with **−server**, else `"/"`.

−keep_alive *value*

If the given value is 0 then HTTP connections are not kept open after receiving a response, which is the default behavior for HTTP 1.0. If the value is 1 or 2 then persistent connections are requested. If the value is 2 then persistent connections are required, i.e., in case the server does not grant them an error occurs. The default value is 1, which means preferring to keep the connection open.

−msg_timeout *seconds*
Number of seconds (or 0 for infinite) a CMP request-response message round trip is allowed to take before a timeout error is returned. Default is to use the **−total_timeout** setting.

−total_timeout *seconds*
Maximum number seconds an overall enrollment transaction may take, including attempts polling for certificates on `waiting` PKIStatus. Default is 0 (infinite).

Server authentication options

−trusted *filenames*|*uris*
When validating signature-based protection of CMP response messages, these are the CA certificate(s) to trust while checking certificate chains during CMP server authentication. This option gives more flexibility than the **−srvcert** option because the server-side CMP signer certificate is not pinned but may be any certificate for which a chain to one of the given trusted certificates can be constructed.

If no **−trusted**, **−srvcert**, and **−secret** option is given then protected response messages from the server are not authenticated.

Multiple sources may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "..."). Each source may contain multiple certificates.

The certificate verification options **−verify_hostname**, **−verify_ip**, and **−verify_email** have no effect on the certificate verification enabled via this option.

−untrusted *filenames*|*uris*
Non-trusted intermediate CA certificate(s). Any extra certificates given with the **−cert** option are appended to it. All these certificates may be useful for cert path construction for the CMP client certificate (to include in the extraCerts field of outgoing messages) and for the TLS client certificate (if TLS is enabled) as well as for chain building when validating the CMP server certificate (checking signature-based CMP message protection) and when validating newly enrolled certificates.

Multiple sources may be given, separated by commas and/or whitespace. Each file may contain multiple certificates.

−srvcert *filename*|*uri*
The specific CMP server certificate to expect and directly trust (even if it is expired) when validating signature-based protection of CMP response messages. May be set alternatively to the **−trusted** option to pin the accepted server.

If set, the subject of the certificate is also used as default value for the recipient of CMP requests and as default value for the expected sender of incoming CMP messages.

−expect_sender *name*
Distinguished Name (DN) expected in the sender field of incoming CMP messages. Defaults to the subject DN of the pinned **−srvcert**, if any.

This can be used to make sure that only a particular entity is accepted as CMP message signer, and attackers are not able to use arbitrary certificates of a trusted PKI hierarchy to fraudulently pose as a CMP server. Note that this option gives slightly more freedom than setting the **−srvcert**, which pins the server to the holder of a particular certificate, while the expected sender name will continue to match after updates of the server cert.

The argument must be formatted as */type0=value0/type1=value1/type2=...*. For details see the description of the **−subject** option.

−ignore_keyusage
Ignore key usage restrictions in CMP signer certificates when validating signature-based protection of incoming CMP messages, else `digitalSignature` must be allowed for signer certificate.

−unprotected_errors
Accept missing or invalid protection of negative responses from the server. This applies to the following message types and contents:

- error messages
- negative certificate responses (IP/CP/KUP)
- negative revocation responses (RP)
- negative PKIConf messages

WARNING: This setting leads to unspecified behavior and it is meant exclusively to allow interoperability with server implementations violating RFC 4210, e.g.:

- section 5.1.3.1 allows exceptions from protecting only for special cases: ''There MAY be cases in which the PKIProtection BIT STRING is deliberately not used to protect a message [...] because other protection, external to PKIX, will be applied instead.''
- section 5.3.21 is clear on ErrMsgContent: ''The CA MUST always sign it with a signature key.''
- appendix D.4 shows PKIConf message having protection

−extracertsout *filename*
The file where to save all certificates contained in the extraCerts field of the last received response message (except for pollRep and PKIConf).

−cacertsout *filename*
The file where to save any CA certificates contained in the caPubs field of the last received certificate response (i.e., IP, CP, or KUP) message.

Client authentication options

−ref *value*
Reference number/string/value to use as fallback senderKID; this is required if no sender name can be determined from the **−cert** or <−subject> options and is typically used when authenticating with pre-shared key (password-based MAC).

−secret *arg*
Prefer PBM-based message protection with given source of a secret value. The secret is used for creating PBM-based protection of outgoing messages and (as far as needed) for validating PBM-based protection of incoming messages. PBM stands for Password-Based Message Authentication Code. This takes precedence over the **−cert** and **−key** options.

For more information about the format of *arg* see **openssl−passphrase−options** (1).

−cert *filename|uri*
The client's current CMP signer certificate. Requires the corresponding key to be given with **−key**. The subject of this certificate will be used as sender of outgoing CMP messages, while the subject of **−oldcert** or **−subjectName** may provide fallback values. The issuer of this certificate is used as one of the recipient fallback values and as fallback issuer entry in the certificate template of IR/CR/KUR. When using signature-based message protection, this ''protection certificate'' will be included first in the extraCerts field of outgoing messages and the signature is done with the corresponding key. In Initialization Request (IR) messages this can be used for authenticating using an external entity certificate as defined in appendix E.7 of RFC 4210. For Key Update Request (KUR) messages this is also used as the certificate to be updated if the **−oldcert** option is not given. If the file includes further certs, they are appended to the untrusted certs because they typically constitute the chain of the client certificate, which is included in the extraCerts field in signature-protected request messages.

−own_trusted *filenames|uris*
If this list of certificates is provided then the chain built for the client-side CMP signer certificate given with the **−cert** option is verified using the given certificates as trust anchors.

Multiple sources may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in ''...''). Each source may contain multiple certificates.

The certificate verification options **−verify_hostname**, **−verify_ip**, and **−verify_email** have no effect on the certificate verification enabled via this option.

−key *filename|uri*
The corresponding private key file for the client's current certificate given in the **−cert** option. This will be used for signature-based message protection unless the **−secret** option indicating PBM or **−unprotected_requests** is given.

−keypass *arg*
Pass phrase source for the private key given with the **−key** option. Also used for **−cert** and **−oldcert** in case it is an encrypted PKCS#12 file. If not given here, the password will be prompted for if needed.

For more information about the format of *arg* see **openssl−passphrase−options** (1).

−digest *name*
Specifies name of supported digest to use in RFC 4210's MSG_SIG_ALG and as the one-way function (OWF) in MSG_MAC_ALG. If applicable, this is used for message protection and Proof-of-Possession (POPO) signatures. To see the list of supported digests, use `openssl list −digest−commands`. Defaults to `sha256`.

−mac *name*
Specifies the name of the MAC algorithm in MSG_MAC_ALG. To get the names of supported MAC algorithms use `openssl list −mac−algorithms` and possibly combine such a name with the name of a supported digest algorithm, e.g., hmacWithSHA256. Defaults to `hmac−sha1` as per RFC 4210.

−extracerts *filenames|uris*
Certificates to append in the extraCerts field when sending messages. They can be used as the default CMP signer certificate chain to include.

Multiple sources may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "..."). Each source may contain multiple certificates.

−unprotected_requests
Send messages without CMP-level protection.

Credentials format options

−certform *PEM|DER*
File format to use when saving a certificate to a file. Default value is PEM.

−keyform *PEM|DER|P12|ENGINE*
The format of the key input; unspecified by default. See "Format Options" in **openssl** (1) for details.

−otherpass *arg*
Pass phrase source for certificate given with the **−trusted**, **−untrusted**, **−own_trusted**, **−srvcert**, **−out_trusted**, **−extracerts**, **−srv_trusted**, **−srv_untrusted**, **−rsp_extracerts**, **−rsp_capubs**, **−tls_extra**, and **−tls_trusted** options. If not given here, the password will be prompted for if needed.

For more information about the format of *arg* see **openssl−passphrase−options** (1).

−engine *id*
See "Engine Options" in **openssl** (1). This option is deprecated.

As an alternative to using this combination:

```
    −engine {engineid} −key {keyid} −keyform ENGINE
```

... it's also possible to just give the key ID in URI form to **−key**, like this:

```
    −key org.openssl.engine:{engineid}:{keyid}
```

This applies to all options specifying keys: **−key**, **−newkey**, and **−tls_key**.

Provider options

−provider *name*
−provider−path *path*
−propquery *propq*
: See ''Provider Options'' in **openssl** (1), **provider** (7), and **property** (7).

Random state options

−rand *files*, **−writerand** *file*
: See ''Random State Options'' in **openssl** (1) for details.

TLS connection options

−tls_used
: Enable using TLS (even when other TLS_related options are not set) when connecting to CMP server via HTTP. This option is not supported with the*−port* option and is ignored with the *−use_moc k_srv* and *−rspin* options or if the *−server* option is not given.

−tls_cert *filename|uri*
: Client's TLS certificate. If the source includes further certs they are used (along with **−untrusted** certs) for constructing the client cert chain provided to the TLS server.

−tls_key *filename|uri*
: Private key for the client's TLS certificate.

−tls_keypass *arg*
: Pass phrase source for client's private TLS key **−tls_key**. Also used for**−tls_cert** in case it is an encrypted PKCS#12 file. If not given here, the password will be prompted for if needed.

  For more information about the format of *arg* see **openssl−passphrase−options** (1).

−tls_extra *filenames|uris*
: Extra certificates to provide to TLS server during TLS handshake

−tls_trusted *filenames|uris*
: Trusted certificate(s) to use for validating the TLS server certificate. This implies hostname validation.

  Multiple sources may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in ''...''). Each source may contain multiple certificates.

  The certificate verification options **−verify_hostname**, **−verify_ip**, and **−verify_email** have no effect on the certificate verification enabled via this option.

−tls_host *name*
: Address to be checked during hostname validation. This may be a DNS name or an IP address. If not given it defaults to the **−server** address.

Client-side debugging options

−batch
: Do not interactively prompt for input, for instance when a password is needed. This can be useful for batch processing and testing.

−repeat *number*
: Invoke the command the given positive number of times with the same parameters. Default is one invocation.

−reqin *filenames*
: Take sequence of CMP requests from file(s).

  Multiple filenames may be given, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in ''...''). As many files are read as needed for a complete transaction.

−reqin_new_tid
: Use a fresh transactionID for CMP request messages read using **−reqin**, which requires re-protecting them as far as they were protected before. This may be needed in case the sequence of requests is

reused and the CMP server complains that the transaction ID has already been used.

−reqout *filenames*
Save sequence of CMP requests to file(s).

Multiple filenames may be given, separated by commas and/or whitespace. As many files are written as needed to store the complete transaction.

−rspin *filenames*
Process sequence of CMP responses provided in file(s), skipping server. This excludes *−server*, *−port*, and *−use_mock_srv*.

Multiple filenames may be given, separated by commas and/or whitespace. As many files are read as needed for the complete transaction.

−rspout *filenames*
Save sequence of CMP responses to file(s).

Multiple filenames may be given, separated by commas and/or whitespace. As many files are written as needed to store the complete transaction.

−use_mock_srv
Test the client using the internal CMP server mock-up at API level, bypassing socket-based transfer via HTTP. This excludes *−server*, *−port*, and *−rspin*.

Mock server options

−port *number*
Act as HTTP-based CMP server mock-up listening on the given port. This excludes *−server*, *−rspin*, and *−use_mock_srv*.

−max_msgs *number*
Maximum number of CMP (request) messages the CMP HTTP server mock-up should handle, which must be nonnegative. The default value is 0, which means that no limit is imposed. In any case the server terminates on internal errors, but not when it detects a CMP-level error that it can successfully answer with an error message.

−srv_ref *value*
Reference value to use as senderKID of server in case no **−srv_cert** is given.

−srv_secret *arg*
Password source for server authentication with a pre-shared key (secret).

−srv_cert *filename|uri*
Certificate of the server.

−srv_key *filename|uri*
Private key used by the server for signing messages.

−srv_keypass *arg*
Server private key (and cert) file pass phrase source.

−srv_trusted *filenames|uris*
Trusted certificates for client authentication.

The certificate verification options **−verify_hostname**, **−verify_ip**, and **−verify_email** have no effect on the certificate verification enabled via this option.

−srv_untrusted *filenames|uris*
Intermediate CA certs that may be useful when validating client certificates.

−rsp_cert *filename|uri*
Certificate to be returned as mock enrollment result.

−rsp_extracerts *filenames|uris*
Extra certificates to be included in mock certification responses.

−rsp_capubs *filenames|uris*
CA certificates to be included in mock Initialization Response (IP) message.

−poll_count *number*
Number of times the client must poll before receiving a certificate.

−check_after *number*
The checkAfter value (number of seconds to wait) to include in poll response.

−grant_implicitconf
Grant implicit confirmation of newly enrolled certificate.

−pkistatus *number*
PKIStatus to be included in server response. Valid range is 0 (accepted) .. 6 (keyUpdateWarning).

−failure *number*
A single failure info bit number to be included in server response. Valid range is 0 (badAlg) .. 26 (duplicateCertReq).

−failurebits *number* Number representing failure bits to be included in server response. Valid range is 0 .. 2ˆ27 − 1.
−statusstring *arg*
Text to be included as status string in server response.

−send_error
Force server to reply with error message.

−send_unprotected
Send response messages without CMP-level protection.

−send_unprot_err
In case of negative responses, server shall send unprotected error messages, certificate responses (IP/CP/KUP), and revocation responses (RP). WARNING: This setting leads to behavior violating RFC 4210.

−accept_unprotected
Accept missing or invalid protection of requests.

−accept_unprot_err
Accept unprotected error messages from client.

−accept_raverified
Accept RAVERIFED as proof-of-possession (POPO).

Certificate verification options, for both CMP and TLS

−allow_proxy_certs, **−attime**, **−no_check_time**, **−check_ss_sig**, **−crl_check**, **−crl_check_all**, **−explicit_policy**, **−extended_crl**, **−ignore_critical**, **−inhibit_any**, **−inhibit_map**, **−no_alt_chains**, **−partial_chain**, **−policy**, **−policy_check**, **−policy_print**, **−purpose**, **−suiteB_128**, **−suiteB_128_only**, **−suiteB_192**, **−trusted_first**, **−use_deltas**, **−auth_level**, **−verify_depth**, **−verify_email**, **−verify_hostname**, **−verify_ip**, **−verify_name**, **−x509_strict −issuer_checks**
Set various options of certificate chain verification. See ‘‘Verification Options’’ in **openssl−verification−options** (1) for details.

The certificate verification options **−verify_hostname**, **−verify_ip**, and **−verify_email** only affect the certificate verification enabled via the **−out_trusted** option.

NOTES

When setting up CMP configurations and experimenting with enrollment options typically various errors occur until the configuration is correct and complete. When the CMP server reports an error the client will by default check the protection of the CMP response message. Yet some CMP services tend not to protect negative responses. In this case the client will reject them, and thus their contents are not shown although they usually contain hints that would be helpful for diagnostics. For assisting in such cases the CMP client offers a workaround via the **−unprotected_errors** option, which allows accepting such negative messages.

EXAMPLES

Simple examples using the default OpenSSL configuration file

This CMP client implementation comes with demonstrative CMP sections in the example configuration file *openssl/apps/openssl.cnf*, which can be used to interact conveniently with the Insta Demo CA.

In order to enroll an initial certificate from that CA it is sufficient to issue the following shell commands.

```
export OPENSSL_CONF=/path/to/openssl/apps/openssl.cnf

openssl genrsa −out insta.priv.pem
openssl cmp −section insta
```

This should produce the file *insta.cert.pem* containing a new certificate for the private key held in *insta.priv.pem*. It can be viewed using, e.g.,

```
openssl x509 −noout −text −in insta.cert.pem
```

In case the network setup requires using an HTTP proxy it may be given as usual via the environment variable **http_proxy** or via the **−proxy** option in the configuration file or the CMP command-line argument **−proxy**, for example

```
−proxy http://192.168.1.1:8080
```

In the Insta Demo CA scenario both clients and the server may use the pre-shared secret *insta* and the reference value *3078* to authenticate to each other.

Alternatively, CMP messages may be protected in signature-based manner, where the trust anchor in this case is *insta.ca.crt* and the client may use any certificate already obtained from that CA, as specified in the **[signature]** section of the example configuration. This can be used in combination with the **[insta]** section simply by

```
openssl cmp −section insta,signature
```

By default the CMP IR message type is used, yet CR works equally here. This may be specified directly at the command line:

```
openssl cmp −section insta −cmd cr
```

or by referencing in addition the **[cr]** section of the example configuration:

```
openssl cmp −section insta,cr
```

In order to update the enrolled certificate one may call

```
openssl cmp −section insta,kur
```

using with PBM-based protection or

```
openssl cmp −section insta,kur,signature
```

using signature-based protection.

In a similar way any previously enrolled certificate may be revoked by

```
openssl cmp −section insta,rr −trusted insta.ca.crt
```

or

```
openssl cmp −section insta,rr,signature
```

Many more options can be given in the configuration file and/or on the command line. For instance, the **−reqexts** CLI option may refer to a section in the configuration file defining X.509 extensions to use in certificate requests, such as `v3_req` in *openssl/apps/openssl.cnf*:

```
openssl cmp −section insta,cr −reqexts v3_req
```

Certificate enrollment

The following examples do not make use of a configuration file at first. They assume that a CMP server can be contacted on the local TCP port 80 and accepts requests under the alias */pkix/*.

For enrolling its very first certificate the client generates a client key and sends an initial request message to the local CMP server using a pre-shared secret key for mutual authentication. In this example the client does not have the CA certificate yet, so we specify the name of the CA with the **−recipient** option and save any CA certificates that we may receive in the `capubs.pem` file.

In below command line usage examples the \ at line ends is used just for formatting; each of the command invocations should be on a single line.

```
openssl genrsa −out cl_key.pem
openssl cmp −cmd ir −server 127.0.0.1:80/pkix/ −recipient "/CN=CMPserver" \
  −ref 1234 −secret pass:1234−5678 \
  −newkey cl_key.pem −subject "/CN=MyName" \
  −cacertsout capubs.pem −certout cl_cert.pem
```

Certificate update

Then, when the client certificate and its related key pair needs to be updated, the client can send a key update request taking the certs in `capubs.pem` as trusted for authenticating the server and using the previous cert and key for its own authentication. Then it can start using the new cert and key.

```
openssl genrsa −out cl_key_new.pem
openssl cmp −cmd kur −server 127.0.0.1:80/pkix/ \
  −trusted capubs.pem \
  −cert cl_cert.pem −key cl_key.pem \
  −newkey cl_key_new.pem −certout cl_cert.pem
cp cl_key_new.pem cl_key.pem
```

This command sequence can be repated as often as needed.

Requesting information from CMP server

Requesting “all relevant information” with an empty General Message. This prints information about all received ITAV **infoType**s to stdout.

```
openssl cmp −cmd genm −server 127.0.0.1/pkix/ −recipient "/CN=CMPserver" \
  −ref 1234 −secret pass:1234−5678
```

Using a custom configuration file

For CMP client invocations, in particular for certificate enrollment, usually many parameters need to be set, which is tedious and error-prone to do on the command line. Therefore, the client offers the possibility to read options from sections of the OpenSSL config file, usually called *openssl.cnf*. The values found there can still be extended and even overridden by any subsequently loaded sections and on the command line.

After including in the configuration file the following sections:

```
[cmp]
server = 127.0.0.1
path = pkix/
trusted = capubs.pem
cert = cl_cert.pem
key = cl_key.pem
newkey = cl_key.pem
certout = cl_cert.pem

[init]
recipient = "/CN=CMPserver"
trusted =
cert =
key =
ref = 1234
secret = pass:1234−5678−1234−567
subject = "/CN=MyName"
cacertsout = capubs.pem
```

the above enrollment transactions reduce to

```
openssl cmp −section cmp,init
openssl cmp −cmd kur −newkey cl_key_new.pem
```

and the above transaction using a general message reduces to

```
openssl cmp −section cmp,init −cmd genm
```

SEE ALSO

openssl−genrsa(1), **openssl−ecparam**(1), **openssl−list**(1), **openssl−req**(1), **openssl−x509**(1), **x509v3_config**(5)

HISTORY

The **cmp** application was added in OpenSSL 3.0.

The **−engine option** was deprecated in OpenSSL 3.0.

COPYRIGHT