Stream:	Internet Engineering Task Force (IETF)	
RFC:	10003	
Obsoletes:	5273, 6402	
Category:	Standards Track	
Published:	July 2026	
ISSN:	2070-1721	
Authors:	J. Mandel, Ed.	S. Turner, Ed.
	AKAYLA, Inc.	*sn3rd*

RFC 10003
Certificate Management over CMS (CMC): Transport Protocols

Abstract

This document defines a number of transport mechanisms that are used to move Certificate Management over CMS (CMC) messages. The transport mechanisms described in this document are HTTP, file, mail, and TCP.

This document obsoletes RFCs 5273 and 6402.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc10003.

Copyright Notice

Table of Contents

1. Introduction

This document defines a number of transport methods that are used to move CMC messages (defined in [CMC-STRUCT]). The transport mechanisms described in this document are HTTP, file, mail, and TCP.

This document obsoletes RFCs 5273 [CMC-TRANSv1] and 6402 [CMC-Updates]. This document also incorporates [Err3593].

2. Requirements Terminology

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Changes Since RFCs 5273 and 6402

Merged [CMC-Updates] text.

IANA assigned TCP port 5318 for the use of CMC.

Clarified the file extensions for Full Public Key Infrastructure (PKI) Requests and Responses.

Added examples of encoding types for mail-based Requests and Responses.

Replaced TLS 1.0 with TLS 1.2 or later and added that implementations are required to follow the recommendations in [BCP195].

Addressed [Err3593].

Added a reference to [HTTP-IMP] for HTTP guidance.

Restricted early data (0-RTT) if using TLS 1.3 or QUIC.

Restricted the use of TCP-Pipelining.

Clarified the limitations of SMTP-over-TLS and the use of authenticated TLS for message delivery.

4. File-Based Protocol

Enrollment messages and responses may be transferred between clients and servers using file-system-based mechanisms, such as when enrollment is performed for an offline client. When files are used to transport Full PKI Request or Full PKI Response messages, there **MUST** be only

one instance of a request or response message in a single file, and the file **MUST** be binary encoded. The abbreviations crq and crp stand for Full PKI Request/Response, respectively; for clarity, we define file extensions for them. The following file type extensions **SHOULD** be used:

Message Type	File Extension
Simple PKI Request	.p10
Full PKI Request	.crq
Simple PKI Response	.p7c
Full PKI Response	.crp

Table 1: File PKI Request/Response Identification

5. Mail-Based Protocol

MIME wrapping is defined for those environments that support MIME. The basic mime wrapping in this section is taken from [SMIMEV4]. When using a mail-based protocol, MIME wrapping between the layers of Cryptographic Message Syntax (CMS) wrapping is optional. Note that this is different from the standard S/MIME (Secure MIME) message.

What follows is a set of Simple PKI Request and Response messages and a set of Full PKI Request and Response messages. The headers discussed below appear in the top-level content of the messages, and the messages' contents are the entire messages' bodies.

> WARNING: The examples that follow are purposely truncated for brevity.

Simple enrollment requests are encoded using the "application/pkcs10" content type [RFC5967]. A file name **MUST** be included either in a Content-Type or a Content-Disposition header in the name or filename parameter, respectively. The extension for the file **MUST** be ".p10". An example similar to that from [RFC5967] follows:

```
From: cmc-client@example.com
Message-Id: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7A@example.com>
To: cmc-server@example.com
Subject: Simple Enrollment Request
Date: Tue, 3 Feb 2026 16:08:28 -0500
MIME-Version: 1.0
Content-Type: application/pkcs10; name=smime.p10
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename=smime.p10

< message contents >
```

Figure 1: Simple PKI Request Message Example

Simple PKI Response messages **MUST** be encoded as content type "application/pkcs7-mime". A smime-type parameter **MUST** be on the Content-Type header with a value of "certs-only". A file name with the ".p7c" extension **MUST** be specified as part of the Content-Type or Content-Disposition header in the name or filename parameter, respectively. An example similar to that from [SMIMEV4] follows:

```
From: cmc-server@example.com
Message-Id: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7B@example.com>
To: cmc-client@example.com
Subject: Re: Simple Enrollment Request
Date: Tue, 3 Feb 2026 16:09:28 -0500
MIME-Version: 1.0
References: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7A@example.com>
In-Reply-To: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7A@example.com>
Content-Type: application/pkcs7-mime; smime-type=certs-only;
  name=smime.p7c
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename=smime.p7c

< message contents >
```

Figure 2: Simple PKI Response Message Example

Full PKI Request messages **MUST** be encoded as content type "application/pkcs7-mime". The smime-type parameter **MUST** be included with a value of "CMC-Request". A file name with the ".p7m" extension **MUST** be specified as part of the Content-Type or Content-Disposition header in the name or filename parameter, respectively. An example similar to that from [SMIMEV4] follows:

```
From: cmc-client@example.com
Message-Id: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7C@example.com>
To: cmc-server@example.com
Subject: Full Enrollment Request
Date: Tue, 3 Feb 2026 16:10:28 -0500
MIME-Version: 1.0
Content-Type: application/pkcs7-mime; smime-type=CMC-Request;
  name=smime.p7c
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename=smime.p7m

< message contents >
```

Figure 3: Full PKI Request Message Example

Full PKI Response messages **MUST** be encoded as content type "application/pkcs7-mime". The smime-type parameter **MUST** be included with a value of "CMC-Response". A file name with the ".p7m" extension **MUST** be specified as part of the Content-Type or Content-Disposition statement. An example similar to that from [SMIMEV4] follows:

```
From: cmc-server@example.com
Message-Id: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7D@example.com>
To: cmc-client@example.com
Subject: Re: Full Enrollment Request
Date: Tue, 3 Feb 2026 16:11:28 -0500
MIME-Version: 1.0
References: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7C@example.com>
In-Reply-To: <E06C3FA6-FF15-4851-AC7F-DB9F3B1C2C7C@example.com>
Content-Type: application/pkcs7-mime; smime-type=CMC-Response;
  name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename=smime.p7m

< message contents >
```

Figure 4: Full PKI Response Message Example

For file names present in the name or filename parameters, non-ASCII text is prohibited.

Item	MIME Type	File Extension	SMIME Type
Simple PKI Request	application/pkcs10	.p10	N/A
Full PKI Request	application/pkcs7-mime	.p7m	CMC-Request
Simple PKI Response	application/pkcs7-mime	.p7c	certs-only

Item	MIME Type	File Extension	SMIME Type
Full PKI Response	application/pkcs7-mime	.p7m	CMC-Response

Table 2: MIME PKI Request/Response Identification

6. HTTP-Based Protocol

This section describes the conventions for use of HTTP [HTTP] as a data transfer protocol. Consult [HTTP-IMP] for additional information. The use of HTTPS [HTTP] provides any necessary content protection from eavesdroppers.

In order for CMC clients and servers using HTTP to interoperate, the following rules apply:

- Clients are configured with sufficient information to form the server URI [RFC3986].
- Client requests are submitted by use of the POST method.
- Servers **MUST** use the 2XX response codes for successful responses.
- Clients **MAY** attempt to send certification requests using HTTPS [HTTP]. Although servers are not required to support TLS/QUIC, a secure channel might be available regardless depending on the HTTP version implemented [HTTP_1.0], [HTTP_1.1], [HTTP_2], [HTTP_3], or later. If TLS is used by the HTTP version, then the implementation **MUST** follow the recommendations in [BCP195]. CMC implementations that support TLS 1.3 or QUIC **MUST NOT** use early data (i.e., 0-RTT) because POST is not idempotent.
- Clients are not required to support any type of HTTP authentication (see Section 11 of [HTTP]) nor Cookies [COOKIES]. Thus, servers cannot rely on these features to be available.
- Clients and servers are expected to follow other rules and restrictions in [HTTP]. Note that some of those rules are for HTTP methods other than POST; clearly, only the rules that apply to POST are relevant for this specification.

6.1. PKI Request

A PKI Request using the POST method is constructed as follows.

The Content-Type field **MUST** have the appropriate value from Table 2.

A Content-Type field for a request:

```
Content-Type: application/pkcs7-mime; smime-type=CMC-Request; name=request.p7m
```

The content of the message is the binary value of the encoding of the PKI Request.

6.2. PKI Response

The content of an HTTP-based PKI Response is the binary value of the BER (Basic Encoding Rules) encoding [X690] of either a Simple or Full PKI Response.

The Content-Type field **MUST** have the appropriate value from Table 2.

A Content-Type field for a response:

```
Content-Type: application/pkcs7-mime; smime-type=CMC-Response;
name=response.p7m
```

7. TCP-Based Protocol

When CMC messages are sent over a TCP-based connection, no wrapping is required of the message. Messages are sent in their binary encoded form.

The client closes a connection after receiving a response, or it issues another request to the server using the same connection. Reusing one connection for multiple successive requests, instead of opening multiple connections that are only used for a single request, is **RECOMMENDED** for performance and resource conservation reasons. The client **MUST** wait for the full response before making another request on the same connection. A server **MAY** close a connection after it has been idle for some period of time; this timeout would typically be several minutes long.

CMC requires a registered port number to send and receive CMC messages over TCP. The Service Name is "pkix-cmc". The TCP port number is 5318.

Prior to [CMC-Updates], CMC did not have a registered port number and used an externally configured port from the Private Port range. Client implementations **MAY** continue to use a port chosen from the Private Port range. A TCP Server **SHOULD** use the port assigned to the CMC service: 5318. It is expected that HTTP will continue to be the primary transport method used by CMC installations.

8. IANA Considerations

IANA has assigned a TCP port number in the "Dynamic and/or Private Ports Range" of the "Service Name and Transport Protocol Port Number Registry" for the use of CMC.

Service Name: pkix-cmc

Port Number: 5318

Transport Protocol: tcp

Description: PKIX Certificate Management using CMS (CMC)

Assignee: iesg@ietf.org

Contact: chair@ietf.org

Reference: RFC 10003

IANA has updated the references to [CMC-TRANSv1] in the "Parameter Values for the smime-type Parameter" registry in the "Media Type Sub-Parameter Registries" registry group for CMC-Request and CMC-Response to instead point to this document.

9. Security Considerations

Mechanisms for thwarting replay attacks may be required in particular implementations of this protocol depending on the operational environment. In cases where the Certification Authority (CA) maintains significant state information, replay attacks may be detectable without the inclusion of the (optional) CMC nonce mechanisms. Implementers and designers of this protocol need to carefully consider environmental conditions before choosing whether or not to implement or use the senderNonce and recipientNonce attributes described in Section 6.6 of [CMC-STRUCT]. Developers of state-constrained PKI clients are strongly encouraged to incorporate the use of these attributes.

Initiation of a secure communications channel between an End-Entity (EE) and a CA or Registration Authority (RA) -- and, similarly, between an RA and another RA or CA -- necessarily requires an out-of-band trust initiation mechanism. For example, a secure channel may be constructed between the EE and the CA via IPsec [IPsec] or TLS [TLS]. Many such schemes exist, and the choice of any particular scheme for trust initiation is outside the scope of this document. Implementers of this protocol are strongly encouraged to consider generally accepted principles of secure key management when integrating this capability within an overall security architecture.

In some instances, no out-of-band trust will have been initiated prior to use of this protocol. This can occur when the protocol itself is being used to download onto the system the set of trust anchors to be used for these protocols. In these instances, the `EnvelopedData` content type (Section 3.2.1.3.3 of [CMC-STRUCT]) or `AuthEnvelopedData` content type Section 3.2.1.3.5 of [CMC-STRUCT] provides the same shrouding that TLS would have provided.

For the mail-based protocol, the `EnvelopedData` or `AuthEnvelopedData` content types can also be used to apply confidentiality protection (content shrouding) to the conveyed messages. Note that, even if the application uses SMTP-over-TLS [RFC3207] with its preferred Message Submission Agent (MSA) for initial submission of the message for delivery, SMTP in subsequent relay hops may not be either authenticated or encrypted. For some combinations of initial MSA and destination domains, it may be possible to request use of authenticated TLS at every relay "hop" of message delivery via the mechanism specified in [RFC8689]. This **MAY** be used, when supported, and expected to work, but risks non-delivery if some of the SMTP servers along the relay chain do not support the REQUIRETLS ESMTP extension.

For the file-based protocol, an additional method of applying confidentiality protection (content shrouding) to the conveyed messages is usually available in the form of filesystem permissions. The local system may allow for read access to be limited to just a single user or group that corresponds to the entity authorized to read the request or response, respectively, and diligent use of these filesystem permissions can be a useful mechanism in multi-user environments.

10. References

10.1. Normative References

[BCP195] Best Current Practice 195, <https://www.rfc-editor.org/info/bcp195>.
At the time of writing, this BCP comprises the following:

Moriarty, K. and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1", BCP 195, RFC 8996, DOI 10.17487/RFC8996, March 2021, <https://www.rfc-editor.org/info/rfc8996>.

Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[CMC-STRUCT] Mandel, J., Ed. and S. Turner, Ed., "Certificate Management over CMS (CMC)", RFC 10002, DOI 10.17487/RFC10002, July 2026, <https://www.rfc-editor.org/info/rfc10002>.

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[HTTP-IMP] Nottingham, M., "Building Protocols with HTTP", BCP 56, RFC 9205, DOI 10.17487/RFC9205, June 2022, <https://www.rfc-editor.org/info/rfc9205>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5967] Turner, S., "The application/pkcs10 Media Type", RFC 5967, DOI 10.17487/RFC5967, August 2010, <https://www.rfc-editor.org/info/rfc5967>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[SMIMEV4] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[X690] ITU-T, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.690>.

10.2. Informative References

[CMC-TRANSv1] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC): Transport Protocols", RFC 5273, DOI 10.17487/RFC5273, June 2008, <https://www.rfc-editor.org/info/rfc5273>.

[CMC-Updates] Schaad, J., "Certificate Management over CMS (CMC) Updates", RFC 6402, DOI 10.17487/RFC6402, November 2011, <https://www.rfc-editor.org/info/rfc6402>.

[COOKIES] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <https://www.rfc-editor.org/info/rfc6265>.

[Err3593] RFC Errata, Erratum ID 3593, RFC 5273, <https://www.rfc-editor.org/errata/eid3593>.

[HTTP_1.0] Berners-Lee, T., Fielding, R., and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, DOI 10.17487/RFC1945, May 1996, <https://www.rfc-editor.org/info/rfc1945>.

[HTTP_1.1] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[HTTP_2] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/info/rfc9113>.

[HTTP_3] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/info/rfc9114>.

[IPsec] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <https://www.rfc-editor.org/info/rfc3207>.

[RFC8689] Fenton, J., "SMTP Require TLS Option", RFC 8689, DOI 10.17487/RFC8689, November 2019, <https://www.rfc-editor.org/info/rfc8689>.

[TLS] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 9846, DOI 10.17487/RFC9846, July 2026, <https://www.rfc-editor.org/info/rfc9846>.

Acknowledgements

Obviously, the authors would like to thank Jim Schaad and Michael Myers for their work on RFC 5273.

Thank you to Julian Reschke, Benjamin Kaduk, Vidhi Goel, Thomas Fossati, Gorry Fairhurst, Éric Vyncke, Gunter Van de Velde, Mahesh Jethanandani, Mike Bishop, Mohamed Boucadair, Viktor Dukhovni, and Eliot Lear for reviewing the document and providing comments.

The Acknowledgements section from RFC 5273 of this document follows:

> The authors and the PKIX Working Group are grateful for the participation of Xiaoyi Liu and Jeff Weinstein in helping to author the original versions of this document.
>
> The authors would like to thank Brian LaMacchia for his work in developing and writing up many of the concepts presented in this document. The authors would also like to thank Alex Deacon and Barb Fox for their contributions.

Contributors

Jim Schaad
August Cellars

Michael Myers
TraceRoute Security, Inc.

Authors' Addresses

Joseph Mandel (EDITOR)
AKAYLA, Inc.
Email: joe@akayla.com

Sean Turner (EDITOR)
sn3rd
Email: sean@sn3rd.com