LAMPS Working Group
Internet-Draft
Obsoletes: 5274, 6402 (if approved)
Intended status: Standards Track
Expires: 29 November 2025

J. Mandel, Ed
AKAYLA, Inc.
S. Turner, Ed
sn3rd

28 May 2025

Certificate Management Messages over CMS (CMC): Compliance Requirements
draft-ietf-lamps-rfc5274bis-06

Abstract

This document provides a set of compliance statements about the CMC (Certificate Management over CMS) enrollment protocol. The ASN.1 structures and the transport mechanisms for the CMC enrollment protocol are covered in other documents. This document provides the information needed to make a compliant version of CMC.

This document obsoletes RFC 5274 and RFC 6402.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-lamps-rfc5274bis/.

Discussion of this document takes place on the WG LAMPS mailing list (mailto:spasm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/spasm/. Subscribe at https://www.ietf.org/mailman/listinfo/spasm/.

Source for this draft and an issue tracker can be found at https://github.com/TBD.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

This Internet-Draft will expire on 29 November 2025.

Copyright Notice

Table of Contents

1. Introduction

The CMC (Certificate Management over CMS) protocol is designed in terms of a client/server relationship. In the simplest case, the client is the requestor of the certificate (i.e., the End Entity (EE)) and the server is the issuer of the certificate (i.e., the Certification Authority (CA)). The introduction of a Registration Authority (RA) into the set of agents complicates the picture only slightly. The RA becomes the server with respect to the certificate requestor, and it becomes the client with respect to the certificate issuer. Any number of RAs can be inserted into the picture in this manner.

The RAs may serve specialized purposes that are not currently covered by this document. One such purpose would be a Key Escrow agent. As such, all certificate requests for encryption keys would be directed through this RA and it would take appropriate action to do the key archival. Key recovery requests could be defined in the CMC methodology allowing for the Key Escrow agent to perform that operation acting as the final server in the chain of agents.

If there are multiple RAs in the system, it is considered normal that not all RAs will see all certificate requests. The routing between the RAs may be dependent on the content of the certificate requests involved.

This document is divided into six sections, each section specifying the requirements that are specific to a class of agents in the CMC model. These are 1) all Entities, 2) all Servers, 3) all Clients, 4) all End-Entities, 5) all Registration Authorities, 6) all Certification Authorities.

This document obsoletes [CMC-COMPv1] and [CMC-Updates].

2. Terminology

There are several different terms, abbreviations, and acronyms used in this document that we define here for convenience and consistency of usage:

End-Entity (EE): Refers to the entity that owns a key pair and for whom a certificate is issued.

Registration Authority (RA) or Local RA (LRA): Refers to an entity that acts as an intermediary between the EE and the CA. Multiple RAs can exist between the End-Entity and the Certification Authority. RAs may perform additional services such as key generation or key archival. This document uses the term RA for both RA and LRA.

Certification Authority (CA): Refers to the entity that issues certificates.

Client: Refers to an entity that creates a PKI Request. In this document, both RAs and EEs can be clients.

Server: Refers to the entities that process PKI Requests and create PKI Responses. In this document both CAs and RAs can be servers.

PKCS #10: Refers to the Public Key Cryptography Standard #10 [PKCS10], which defines a certification request syntax.

CRMF: Refers to the Certificate Request Message Format RFC [CRMF]. CMC uses this certification request syntax defined in this document as part of the protocol.

CMS: Refers to the Cryptographic Message Syntax RFC [CMS]. This document provides for basic cryptographic services including encryption and signing with and without key management.

PKI Request/Response: Refers to the requests/responses described in this document. PKI Requests include certification requests, revocation requests, etc. PKI Responses include certs-only messages, failure messages, etc.

Proof-of-Identity: Refers to the client proving they are who they say that they are to the server.

Proof-of-Possession (POP): Refers to a value that can be used to prove that the private key corresponding to a public key is in the possession and can be used by an end-entity.

Transport wrapper: Refers to the outermost CMS wrapping layer.

3. 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.

4. Changes since RFC 5274 and 6402

Merged [CMC-Updates] text.

Added RA Identity Proof Witness and Response Body Controls to CMC Controls Attribute table.

Updated the Cryptographic Algorithm Requirements, and added section to maintain backwards compatability.

Replaced SHA-1 for SHA-256 Replaced HMAC-SHA-1 for HMAC-SHA-256

Updated the Introduction section, changed "all agents" to "all entities" in the overview to maintain consistency throughout the document, and re-numbered the section headers.

5. Requirements for All Entities

All [CMC-STRUCT] and [CMC-TRANS] compliance statements MUST be adhered to unless specifically stated otherwise in this document.

All entities MUST support Full PKI Requests, Simple PKI Responses, and Full PKI Responses. Servers SHOULD support Simple PKI Requests.

All entities MUST support the use of the CRMF syntax for certification requests. Support for the PKCS #10 syntax for certification requests SHOULD be implemented by servers.

The extendedFailInfo field SHOULD NOT be populated in the CMCStatusInfoV2 object; the failInfo field SHOULD be used to relay this information. If the extendedFailInfo field is used, it is suggested that an additional CMCStatusInfoV2 item exist for the same body part with a failInfo field.

All entities MUST implement the HTTP transport mechanism as defined in [CMC-TRANS]. Other transport mechanisms MAY be implemented.

5.1. Cryptographic Algorithm Requirements

All entities MUST verify RSA-SHA256 signatures in SignedData; see [CMS-ALG2]. Entities MAY verify other signature algorithms.

All entities MUST generate RSA-SHA256 signatures for SignedData; see [CMS-ALG2]. Other signatures algorithms MAY be used for generation.

All entities MUST support Advanced Encryption Standard (AES) as the content encryption algorithm for EnvelopedData; see [CMS-AES]. Other content encryption algorithms MAY be implemented.

All entities MUST support RSA as a key transport algorithm for EnvelopedData; see [CMS-ALG2]. Other key transport algorithms MAY be implemented.

If an entity supports key agreement for EnvelopedData, it MUST support Diffie-Hellman; see [CMS-DH].

If an entity supports PasswordRecipientInfo for EnvelopedData or AuthenticatedData, it MUST support PBKDF2 [PBKDF2] for key derivation algorithms. It MUST support AES key wrap see [AES-WRAP] as the key encryption algorithm.

If AuthenticatedData is supported, PasswordRecipientInfo MUST be supported.

Algorithm requirements for the Identity Proof Version 2 control Section 6.2.1 of [CMC-STRUCT] are: SHA-256 MUST be implemented for hashAlgId. HMAC-SHA256 MUST be implemented for macAlgId.

Algorithm requirements for the Pop Link Witness Version 2 control Section 6.3.1 of [CMC-STRUCT] are: SHA-256 MUST be implemented for keyGenAlgorithm. PBKDF2 [PBKDF2] MAY be implemented for keyGenAlgorithm. HMAC-SHA256 MUST be implemented for macAlgorithm.

Algorithm requirements for the Encrypted POP and Decrypted POP controls Section 6.7 of [CMC-STRUCT] are: SHA-256 MUST be implemented for witnessAlgID. HMAC-SHA256 MUST be implemented for thePOPAlgID.

Algorithm requirements for Publish Trust Anchors control Section 6.15 of [CMC-STRUCT] are: SHA-256 MUST be implemented for hashAlgorithm.

If an EE generates DH keys for certification, it MUST support Section 4 of [DH-POP]]. EEs MAY support Section 3 of [DH-POP]. CAs and RAs that do POP verification MUST support Section 4 of [DH-POP] and SHOULD support Section 3 of [DH-POP].

EEs that need to use a signature algorithm for keys that cannot produce a signature MUST support Appendix C of [CMC-STRUCT] and MUST support the Encrypted/Decrypted POP controls. CAs and RAs that do POP verification MUST support this signature algorithm and MUST support the Encrypted/Decrypted POP controls.

For backwards compatibility with the previous version of CMC, servers MAY offer the algorithms specified therein, but SHOULD use the CMC requests to identify which certificates should be transitioned to more secure algorithms, if possible.

5.2. Controls

The following table lists the name and level of support required for each control.

Control	EE	RA	CA
Extended CMC Status Info	MUST	MUST	MUST
CMC Status Info	SHOULD	SHOULD	SHOULD
Identity Proof Version 2	MUST	MUST	MUST
Identity Proof	SHOULD	SHOULD	SHOULD
Identification	MUST	MUST	MUST
POP Link Random	MUST	MUST	MUST
POP Link Witness Version 2	MUST	MUST	MUST
POP Link Witness	SHOULD	MUST	MUST
Data Return	MUST	MUST	MUST
Modify Cert Request	N/A	MUST	(2)
Add Extensions	N/A	MAY	(1)
Transaction ID	MUST	MUST	MUST
Sender Nonce	MUST	MUST	MUST
Recipient Nonce	MUST	MUST	MUST
Encrypted POP	(4)	(5)	SHOULD

Decrypted POP	(4)	(5)	SHOULD
RA POP Witness	N/A	SHOULD	(1)
Get Certificate	optional	optional	optional
Get CRL	optional	optional	optional
Revocation Request	SHOULD	SHOULD	MUST
Registration Info	SHOULD	SHOULD	SHOULD
Response Information	SHOULD	SHOULD	SHOULD
Query Pending	MUST	MUST	MUST
Confirm Cert. Acceptance	MUST	MUST	MUST
Publish Trust Anchors	(3)	(3)	(3)
Authenticate Data	(3)	(3)	(3)
Batch Request	N/A	MUST	(2)
Batch Responses	N/A	MUST	(2)
Publication Information	optional	optional	optional
Control Processed	N/A	MUST	(2)
RA Identity Proof Witness	N/A	MUST	(2)
Response Body	(6)	(6)	N/A.

Table 1: CMC Control Attributes

Notes:

1. CAs SHOULD implement this control if designed to work with RAs.

2. CAs MUST implement this control if designed to work with RAs.

3. Implementation is optional for these controls. We strongly suggest that they be implemented in order to populate client trust anchors.

4. EEs only need to implement this if (a) they support key agreement algorithms or (b) they need to operate in environments where the hardware keys cannot provide POP.

5. RAs SHOULD implement this if they implement RA POP Witness.

6. EE’s SHOULD implement if designed to work with RAs and MUST implement if intended to be used in environments where RAs are used for identity validation or key generation. RAs SHOULD implement and validate responses for consistency.

Strong consideration should be given to implementing the Authenticate Data and Publish Trust Anchors controls as this gives a simple method for distributing trust anchors into clients without user intervention.

5.3. CRMF Feature Requirements

The following additional restrictions are placed on CRMF features:

The registration control tokens id-regCtrl-regToken and id-regCtrl-authToken MUST NOT be used. No specific CMC feature is used to replace these items, but generally the CMC controls identification and identityProof will perform the same service and are more specifically defined.

The control token id-regCtrl-pkiArchiveOptions SHOULD NOT be supported. An alternative method is under development to provide this functionality.

The behavior of id-regCtrl-oldCertID is not presently used. It is replaced by issuing the new certificate and using the id-cmc-publishCert to remove the old certificate from publication. This operation would not normally be accompanied by an immediate revocation of the old certificate; however, that can be accomplished by the id-cmc-revokeRequest control.

The id-regCtrl-protocolEncrKey is not used.

6. Requirements for Clients

There are no additional requirements.

7. Requirements for Servers

There are no additional requirements.

8. Requirements for EEs

If an entity implements Diffie-Hellman, it MUST implement either the DH-POP Proof-of-Possession as defined in Section 4 of [DH-POP] or the challenge-response POP controls id-cmc-encryptedPOP and id-cmc-decryptedPOP.

9. Requirements for RAs

RAs SHOULD be able to do delegated POP. RAs implementing this feature MUST implement the id-cmc-lraPOPWitness control.

All RAs MUST implement the promotion of the id-aa-cmc-unsignedData as covered in Section 3.2.3 of [CMC-STRUCT].

10. Requirements for CAs

Providing for CAs to work in an environment with RAs is strongly suggested. Implementation of such support is strongly suggested as this permits the delegation of substantial administrative interaction onto an RA rather than at the CA.

CAs MUST perform at least minimal checks on all public keys before issuing a certificate. At a minimum, a check for syntax would occur with the POP operation. Additionally, CAs SHOULD perform simple checks for known bad keys such as small subgroups for DSA-SHA1 and DH keys [SMALL-SUB-GROUP] or known bad exponents for RSA keys.

CAs MUST enforce POP checking before issuing any certificate. CAs MAY delegate the POP operation to an RA for those cases where 1) a challenge/response message pair must be used, 2) an RA performs escrow of a key and checks for POP in that manner, or 3) an unusual algorithm is used and that validation is done at the RA.

CAs SHOULD implement both the DH-POP Proof-of-Possession as defined in Section 4 of [DH-POP] and the challenge-response POP controls id-cmc-encryptedPOP and id-cmc-decryptedPOP.

11. Security Considerations

This document uses [CMC-STRUCT] and [CMC-TRANS] as building blocks to this document. The security sections of those two documents are included by reference.

Knowledge of how an entity is expected to operate is vital in determining which sections of requirements are applicable to that entity. Care needs to be taken in determining which sections apply and fully implementing the necessary code.

Cryptographic algorithms have and will be broken or weakened. Implementers and users need to check that the cryptographic algorithms listed in this document make sense from a security level. The IETF from time to time may issue documents dealing with the current state of the art. Two examples of such documents are [SMALL-SUB-GROUP] and [HASH-ATTACKS].

12. IANA Considerations

This document does not require action from IANA.

13. References

13.1. Normative References

[AES-WRAP] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, DOI 10.17487/RFC3394, September 2002, <https://www.rfc-editor.org/rfc/rfc3394>.

[CMC-STRUCT] Mandel, J. and S. Turner, "Certificate Management over CMS (CMC)", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc5272bis-06, 28 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc5272bis-06>.

[CMC-TRANS] Mandel, J. and S. Turner, "Certificate Management over CMS (CMC): Transport Protocols", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc5273bis-05, 1 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc5273bis-05>.

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/rfc/rfc5652>.

[CMS-AES] Schaad, J., "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)", RFC 3565, DOI 10.17487/RFC3565, July 2003, <https://www.rfc-editor.org/rfc/rfc3565>.

[CMS-ALG2] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, DOI 10.17487/RFC5754, January 2010, <https://www.rfc-editor.org/rfc/rfc5754>.

[CMS-DH] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, DOI 10.17487/RFC2631, June 1999, <https://www.rfc-editor.org/rfc/rfc2631>.

[CRMF] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <https://www.rfc-editor.org/rfc/rfc4211>.

[DH-POP] Schaad, J. and H. Prafullchandra, "Diffie-Hellman Proof-of-Possession Algorithms", RFC 6955, DOI 10.17487/RFC6955, May 2013, <https://www.rfc-editor.org/rfc/rfc6955>.

[PBKDF2] Kario, H., "Use of Password-Based Message Authentication Code 1 (PBMAC1) in PKCS #12 Syntax", RFC 9579, DOI 10.17487/RFC9579, May 2024, <https://www.rfc-editor.org/rfc/rfc9579>.

[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/rfc/rfc2119>.

[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/rfc/rfc8174>.

13.2. Informative References

[CMC-COMPv1] Schaad, J. and M. Myers, "Certificate Management Messages over CMS (CMC): Compliance Requirements", RFC 5274, DOI 10.17487/RFC5274, June 2008, <https://www.rfc-editor.org/rfc/rfc5274>.

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

[HASH-ATTACKS] Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashes in Internet Protocols", RFC 4270, DOI 10.17487/RFC4270, November 2005, <https://www.rfc-editor.org/rfc/rfc4270>.

[PKCS10] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/rfc/rfc2986>.

[SMALL-SUB-GROUP] Zuccherato, R., "Methods for Avoiding the "Small-Subgroup" Attacks on the Diffie-Hellman Key Agreement Method for S/MIME", RFC 2785, DOI 10.17487/RFC2785, March 2000, <https://www.rfc-editor.org/rfc/rfc2785>.

Acknowledgements

Obviously, the authors of this version of the document would like to thank Jim Schaad and Michael Myers for their work on the previous version of this document.

The acknowledgment from the previous version 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
AKAYLA, Inc.
Email: joe@akayla.com

Sean Turner
sn3rd

Email: sean@sn3rd.com