

LAMPS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 17, 2020

H. Brockhaus
S. Fries
D. von Oheimb
Siemens
February 14, 2020

Lightweight CMP Profile
draft-ietf-lamps-lightweight-cmp-profile-00

Abstract

The goal of this document is to facilitate interoperability and automation by profiling the Certificate Management Protocol (CMP) version 2 and the related Certificate Request Message Format (CRMF) version 2 and the HTTP Transfer for the Certificate Management Protocol. It specifies a subset of CMP and CRMF focusing on typical uses cases relevant for managing certificates of devices in many industrial and IoT scenarios. To limit the overhead of certificate management for more constrained devices only the most crucial types of transactions are specified as mandatory. To foster interoperability also in more complex scenarios, other types of transactions are specified as recommended or optional.

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/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 17, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	History of changes	3
2.	Introduction	5
2.1.	Motivation for profiling CMP	5
2.2.	Motivation for a lightweight profile for CMP	6
2.3.	Existing CMP profiles	7
2.4.	Compatibility with existing CMP profiles	9
2.5.	Scope of this document	10
2.6.	Structure of this document	11
2.7.	Convention and Terminology	11
3.	Architecture and use cases	12
3.1.	Solution architecture	12
3.2.	Basic generic CMP message content	13
3.3.	Supported use cases	14
3.3.1.	Mandatory use cases	14
3.3.2.	Recommended Use Cases	14
3.3.3.	Optional use cases	15
3.4.	CMP message transport	15
4.	Generic parts of the PKI message	16
4.1.	General description of the CMP message header	17
4.2.	General description of the CMP message protection	18
4.3.	General description of CMP message extraCerts	19
5.	End Entity focused certificate management use cases	19
5.1.	Requesting a new certificate from a PKI	20
5.1.1.	A certificate from a new PKI with signature protection	21
5.1.2.	A certificate from a trusted PKI with signature protection	27
5.1.3.	Update an existing certificate with signature protection	27
5.1.4.	A certificate from a PKI with MAC protection	28
5.1.5.	A certificate from a legacy PKI using PKCS#10 request	30
5.1.6.	Generate the key pair centrally at the (L)RA/CA	32
5.1.6.1.	Using symmetric key-encryption key management technique	37
5.1.6.2.	Using key agreement key management technique	38
5.1.6.3.	Using key transport key management technique	39
5.1.7.	Delayed enrollment	40
5.2.	Revoking a certificate	45

5.3.	Error reporting	47
5.4.	Support messages	49
5.4.1.	General message and response	49
5.4.2.	Get CA certificates	51
5.4.3.	Get root CA certificate update	51
5.4.4.	Get certificate request parameters	53
5.4.5.	Get certificate management configuration	54
5.4.6.	Get enrollment voucher	56
6.	LRA and RA focused certificate management use cases	57
6.1.	Forwarding of messages	57
6.1.1.	Not changing protection	59
6.1.2.	Replacing protection	60
6.1.2.1.	Keeping proof-of-possession	60
6.1.2.2.	Breaking proof-of-possession	61
6.1.3.	Adding Protection	61
6.1.4.	Initiating delayed enrollment	61
6.2.	Revoking certificates on behalf of another's entities	61
6.3.	Error reporting	62
7.	CMP message transport variants	63
7.1.	HTTP transport	63
7.2.	HTTPS transport using certificates	65
7.3.	HTTPS transport using shared secrets	65
7.4.	File-based transport	66
7.5.	CoAP transport	66
7.6.	Piggybacking on other reliable transport	66
8.	IANA Considerations	66
9.	Security Considerations	66
10.	Acknowledgements	66
11.	References	67
11.1.	Normative References	67
11.2.	Informative References	68
Appendix A.	Additional Stuff	70
	Authors' Addresses	70

[1.](#) History of changes

Note: This section will be deleted in the final version of the document.

From [draft-brockhaus-lamps-lightweight-cmp-profile-03](#) -> [draft-ietf-lamps-lightweight-cmp-profile-00](#):

- o Changes required to reflect WG adoption
- o Minor changes in wording

From version 02 -> 03:

- o Added a short summary of [\[RFC4210\] Appendix D](#) and E in [Section 2.3](#).
- o Clarified some references to different sections and added some clarification in response to feedback from Michael Richardson and Tomas Gustavsson.
- o Added an additional label to the operational path to address multiple CAs or certificate profiles in [Section 7.1](#).

From version 01 -> 02:

- o Added some clarification on the key management techniques for protection of centrally generated keys in [Section 5.1.6](#).
- o Added some clarifications on the certificates for root CA certificate update in [Section 5.4.3](#).
- o Added a section to specify the usage of nested messages for RAs to add an additional protection for further discussion, see [Section 6.1.3](#).
- o Added a table containing endpoints for HTTP transport in [Section 7.1](#) to simplify addressing PKI management entities.
- o Added some Todos resulting from discussion with Tomas Gustavsson.
- o Minor clarifications and changes in wording.

From version 00 -> 01:

- o Added a section to specify the enrollment with a already trusted PKI for further discussion, see [Section 5.1.2](#).
- o Complete specification of requesting a certificate from a legacy PKI using a PKCS#10 [\[RFC2986\]](#) request in [Section 5.1.5](#).
- o Complete specification of adding central generation of a key pair on behalf of an end entity in [Section 5.1.6](#).
- o Complete specification of handling delayed enrollment due to asynchronous message delivery in [Section 5.1.7](#).
- o Complete specification of additional support messages, e.g., to update a Root CA certificate or to request an [RFC 8366](#) [\[RFC8366\]](#) voucher, in [Section 5.4](#).
- o Minor changes in wording.

From [draft-brockhaus-lamps-industrial-cmp-profile-00](#) -> [draft-brockhaus-lamps-lightweight-cmp-profile-00](#):

- o Change focus from industrial to more multi-purpose use cases and lightweight CMP profile.
- o Incorporate the omitted confirmation into the header specified in [Section 4.1](#) and described in the standard enrollment use case in [Section 5.1.1](#) due to discussion with Tomas Gustavsson.
- o Change from OPTIONAL to RECOMMENDED for use case 'Revoke another's entities certificate' in [Section 6.2](#), because it is regarded as important functionality in many environments to enable the management station to revoke EE certificates.
- o Complete the specification of the revocation message flow in [Section 5.2](#) and [Section 6.2](#).
- o The CoAP based transport mechanism and piggybacking of CMP messages on top of other reliable transport protocols is out of scope of this document and would need to be specified in another document.
- o Further minor changes in wording.

2. Introduction

This document specifies PKI management operations supporting machine-to-machine and IoT use cases. The focus lies on maximum automation and interoperable implementation of all involved PKI entities from end entities (EE) through an optional Local Registration Authority (LRA) and the RA up to the CA. The profile makes use of the concepts and syntax specified in CMP [[RFC4210](#)], CRMF [[RFC4211](#)], HTTP transfer for CMP [[RFC6712](#)], and CMP Updates [[I-D.brockhaus-lamps-cmp-updates](#)]. Especially CMP and CRMF are very feature-rich standards, while only a limited subset of the specified functionality is needed in many environments. Additionally, the standards are not always precise enough on how to interpret and implement the described concepts. Therefore, we aim at tailoring and specifying in more detail how to use these concepts to implement lightweight automated certificate management.

2.1. Motivation for profiling CMP

CMP was standardized in 1999 and is implemented in several CA products. In 2005 a completely reworked and enhanced version 2 of CMP [[RFC4210](#)] and CRMF [[RFC4211](#)] has been published followed by a

document specifying a transfer mechanism for CMP messages using http [[RFC6712](#)] in 2012.

Though CMP is a very solid and capable protocol it could be used more widely. The most important reason for not more intense application of CMP appears to be that the protocol is offering a large set of features and options but being not always precise enough and leaving room for interpretation. On the one hand, this makes CMP applicable to a very wide range of scenarios, but on the other hand a full implementation of all options is unrealistic because this would take enormous effort.

Moreover, many details of the CMP protocol have been left open or have not been specified in full preciseness. The profiles specified in [Appendix D](#) and E of [[RFC4210](#)] offer some more detailed certificate use cases. But the specific needs of highly automated scenarios for a machine-to-machine communication are not covered sufficiently.

As also 3GPP and UNISIG already put across, profiling is a way of coping with the challenges mentioned above. To profile means to take advantage of the strengths of the given protocol, while explicitly narrowing down the options it provides to exactly those needed for the purpose(s) at hand and eliminating all identified ambiguities. In this way all the general and applicable aspects of the protocol can be taken over and only the peculiarities of the target scenario need to be dealt with specifically.

Doing such a profiling for a new target environment can be a high effort because the range of available options needs to be well understood and the selected options need to be consistent with each other and with the intended usage scenario. Since most industrial use cases typically have much in common it is worth sharing this effort, which is the aim of this document. Other standardization bodies can then reference the profile from this document and do not need to come up with individual profiles.

[2.2.](#) Motivation for a lightweight profile for CMP

The profiles specified in [Appendix D](#) and E of CMP have been developed in particular to manage certificates of human end entities. With the evolution of distributed systems and client-server architectures, certificates for machines and applications on them have become widely used. This trend has strengthened even more in emerging industrial and IoT scenarios. CMP is sufficiently flexible to support these very well.

Today's IT security architectures for industrial solutions typically use certificates for endpoint authentication within protocols like

IPSec, TLS, or SSH. Therefore, the security of these architectures highly relies upon the security and availability of the implemented certificate management procedures.

Due to increasing security in operational networks as well as availability requirements, especially on critical infrastructures and systems with a high volume of certificates, a state-of-the-art certificate management must be constantly available and cost-efficient, which calls for high automation and reliability. The NIST Cyber Security Framework [[NIST-CSFW](#)] also refers to proper processes for issuance, management, verification, revocation, and audit for authorized devices, users and processes involving identity and credential management. Such PKI operation according to commonly accepted best practices is also required in IEC 62443-3-3 [[IEC62443-3-3](#)] for security level 2 up to security level 4.

Further challenges in many industrial systems are network segmentation and asynchronous communication, where PKI operation is often not deployed on-site but in a more protected environment of a data center or trust center. Certificate management must be able to cope with such network architectures. CMP offers the required flexibility and functionality, namely self-contained messages, efficient polling, and support for asynchronous message transfer with end-to-end security.

[2.3.](#) Existing CMP profiles

As already stated, CMP contains profiles with mandatory and optional transactions in the Appendixes D and E of [[RFC4210](#)]. Those profiles focus on management of human user certificates and do only partly address the specific needs for certificate management automation for unattended machine or application-oriented end entities.

[RFC4210] specifies in [Appendix D](#) the following mandatory PKI management operations (all require support of, in the meantime outdated, algorithms, e.g., SHA-1 and 3-DES; all operations may enroll up to two certificates, one for a locally generated and another optional one for a centrally generated key pair; all require use of certConf/PKIConf messages for confirmation):

- o Initial registration/certification; an (uninitialized) end entity requests a (first) certificate from a CA using shared secret based message authentication. The content is similar to PKI management operation specified in [Section 5.1.4](#) of this document.
- o Certificate request; an (initialized) end entity requests a certificate from a CA (for any reason) using signature or shared secret based message authentication. The content is similar to

PKI management operation specified in [Section 5.1.2](#) of this document.

- o Key update; an (initialized) end entity requests a certificate from a CA (to update the key pair and/or corresponding certificate that it already possesses) using signature or shared secret based message authentication. The content is similar to PKI management operation specified in [Section 5.1.3](#) of this document.

Due to the two certificates that may be enrolled and the shared secret based authentication, these PKI management operations focus more on the enrollment of human users at a PKI.

[RFC4210] specifies in [Appendix E](#) the following optional transactions (all require support of, in the meantime outdated, algorithms, e.g., SHA-1 and 3-DES):

- o Root CA key update; a root CA updates its key pair and produces a CA key update announcement message that can be made available (via some transport mechanism) to the relevant end entities. This operation only supports a push and no pull model. The content is similar to PKI management operation specified in [Section 5.4.3](#) of this document.
- o Information request/response; an end entity sends a general message to the PKI requesting details that will be required for later PKI management operations. The content is similar to PKI management operation specified in [Section 5.4.4](#) and [Section 5.4.5](#) of this document.
- o Cross-certification request/response (1-way); creation of a single cross-certificate (i.e., not two at once). The requesting CA MAY choose who is responsible for publication of the cross-certificate created by the responding CA through use of the PKIPublicationInfo control.
- o In-band initialization using external identity certificate (this PKI management operation may also enroll up to two certificates and requires use of certConf/PKIconf messages for confirmation as specified in [Appendix D of \[RFC4210\]](#)). An (uninitialized) end entity wishes to initialize into the PKI with a CA, CA-1. It uses, for authentication purposes, a pre-existing identity certificate issued by another (external) CA, CA-X. A trust relationship must already have been established between CA-1 and CA-X so that CA-1 can validate the EE identity certificate signed by CA-X. Furthermore, some mechanism must already have been established within the Personal Security Environment (PSE) of the EE that would allow it to authenticate and verify PKIMessages

signed by CA-1. The content is similar to PKI management operation specified in [Section 5.1.1](#) of this document. The trust establishment of the EE in CA-1 and of the CA/RA in CA-X can be automated using, e.g., the exchange of a certificate management configuration as specified in [Section 5.4.5](#) or an enrollment voucher as specified in [Section 5.4.6](#) of this document.

Both Appendixes focus on EE to CA/RA PKI management operations and do not address further profiling of RA to CA communication as typically used for full backend automation.

3GPP makes use of CMP [[RFC4210](#)] in its Technical Specification 133 310 [[ETSI-3GPP](#)] for automatic management of IPsec certificates in UMTS, LTE, and 5G backbone networks. Since 2010 a dedicated CMP profile for initial certificate enrollment and update transactions between end entities and the RA/CA is specified in the document.

UNISIG has included a CMP profile for certificate enrollment in the subset 137 specifying the ETRAM/ECTS on-line key management for train control systems [[UNISIG](#)] in 2015.

Both standardization bodies use CMP [[RFC4210](#)], CRMF [[RFC4211](#)], and HTTP transfer for CMP [[RFC6712](#)] to add tailored means for automated certificate management for unattended machine or application-oriented end entities.

2.4. Compatibility with existing CMP profiles

The profile specified in this document is compatible with CMP [[RFC4210](#)] Appendixes D and E (PKI Management Message Profiles), with the following exceptions:

- o signature-based protection is the default protection; initial transactions may also use HMAC,
- o certification of a second key pair within the same transaction is not supported,
- o proof-of-possession (POPO) with self-signature of the certTemplate according to [\[RFC4211\] section 4.1](#) clause 3 is the recommended default POPO method (deviations are possible by EEs when requesting central key generation and by (L)RAs when using raVerified),
- o confirmation of newly enrolled certificates may be omitted, and
- o all transactions consist of request-response message pairs originating at the EE, i.e., announcement messages are omitted.

The profile specified in this document is compatible with the CMP profile for UMTS, LTE, and 5G network domain security and authentication framework [[ETSI-3GPP](#)], except that:

- o protection of initial transactions may be HMAC-based,
- o the subject name is mandatory in certificate templates, and
- o confirmation of newly enrolled certificates may be omitted.

The profile specified in this document is compatible with the CMP profile for on-line key management in rail networks as specified in UNISIG subset-137 [[UNISIG](#)], except that:

- o as of [RFC 4210](#) [[RFC4210](#)] the messageTime is required to be Greenwich Mean Time coded as generalizedTime (Note: While UNISIG explicitly states that the messageTime is required to be 'UTC time', it is not clear if this means a coding as UTCTime or generalizedTime and if other time zones than Greenwich Mean Time shall be allowed. Therefore UNISIG may be in conflict with [RFC 4210](#) [[RFC4210](#)]. Both time formats are described in [RFC 5280](#) [[RFC5280](#) section 4.1.2.5.], and
- o in case the request message is MAC protected, also the response, certConf, and PKIConf messages have a MAC-based protection (Note: if changing to signature protection of the response the caPubs field cannot be used securely anymore.).

[2.5.](#) Scope of this document

This document specifies requirements on generating messages on the sender side. It does not specify strictness of verification on the receiving side and how in detail to handle error cases.

Especially on the EE side this profile aims at a lightweight protocol that can be implemented on more constrained devices. On the side of the central PKI management entities the profile accepts higher resource needed.

For the sake of robustness and preservation of security properties implementations should, as far as security is not affected, adhere to Postel's law: "Be conservative in what you do, be liberal in what you accept from others" (often reworded as: "Be conservative in what you send, be liberal in what you accept").

When in [Section 4](#), [Section 5](#), and [Section 6](#) a field of the ASN.1 syntax as defined in [RFC 4210](#) [[RFC4210](#)] and [RFC 4211](#) [[RFC4211](#)] is not explicitly specified, it SHOULD not be used by the sending entity.

The receiving entity MUST NOT require its absence and if present MUST gracefully handle its presence.

2.6. Structure of this document

[Section 3](#) introduces the general PKI architecture and approach to certificate management using CMP that is assumed in this document. Then it enlists the PKI management operations specified in this document and describes them in general words. The list of supported certificate management use cases is divided into mandatory, recommended, and optional ones.

[Section 4](#) profiles the CMP message header, protection, and extraCerts section as they are general elements of CMP messages.

[Section 5](#) profiles the exchange of CMP messages between an EE and the first PKI management entities. There are various flavors of certificate enrollment requests optionally with polling, revocation, error handling, and general support transactions.

[Section 6](#) profiles the exchange between PKI management entities. These are in the first place the forwarding of messages coming from or going to an EE. This includes also initiating delayed delivery of messages, which involves polling. Additionally, it specifies transactions where the PKI component manages certificates on behalf of an EE or for itself.

[Section 7](#) outlines different mechanisms for CMP message transfer, namely http-based transfer as already specified in [\[RFC6712\]](#), using an additional TLS layer, or offline file-based transport. CoAP [\[RFC7252\]](#) and piggybacking CMP messages on other protocols is out of scope and left for further documents.

2.7. Convention and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [\[RFC2119\]](#).

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in [RFC 2119](#).

Technical terminology is used in conformance with [RFC 4210](#) [\[RFC4210\]](#), [RFC 4211](#) [\[RFC4211\]](#), [RFC 5280](#) [\[RFC5280\]](#), and IEEE 802.1AR [\[IEEE802.1AR\]](#). The following key words are used:

CA: Certification authority, which issues certificates.

RA: Registration authority, an optional system component to which a CA delegates certificate management functions such as authorization checks.

LRA: Local registration authority, an optional RA system component with proximity to the end entities.

KGA: Key generation authority, an optional system component, typically co-located with an LRA, RA, or CA, that offers key generation services to end entities.

EE: End entity, a user, device, or service that holds a PKI certificate. An identifier for the EE is given as the subject of its certificate.

3. Architecture and use cases

3.1. Solution architecture

Typically, a machine EE will be equipped with a manufacturer issued certificate during production. Such a manufacturer issued certificate is installed during production to identify the device throughout its lifetime. This manufacturer certificate can be used to protect the initial enrollment of operational certificates after installation of the EE in a plant or industrial network. An operational certificate is issued by the owner or operator of the device to identify the device during operation, e.g., within a security protocol like IPsec, TLS, or SSH. In IEEE 802.1AR [IEEE802.1AR] a manufacturer certificate is called IDevID certificate and an operational certificate is called LDevID certificate.

All certificate management transactions specified in this document are initiated by the EE. The EE creates a CMP request message, protects it using its manufacturer or operational certificate, if available, and sends it to its locally reachable PKI component. This PKI component may be an LRA, RA, or the CA, which checks the request, responds to it itself, or forwards the request upstream to the next PKI component. In case an (L)RA changes the CMP request message header or body or wants to prove a successful verification or authorization, it can apply a protection of its own. Especially the communication between an LRA and RA can be performed synchronously or asynchronously. Synchronous communication describes a timely uninterrupted communication between two communication partners, while asynchronous communication is not performed in a timely consistent manner, e.g., because of a delayed message delivery.

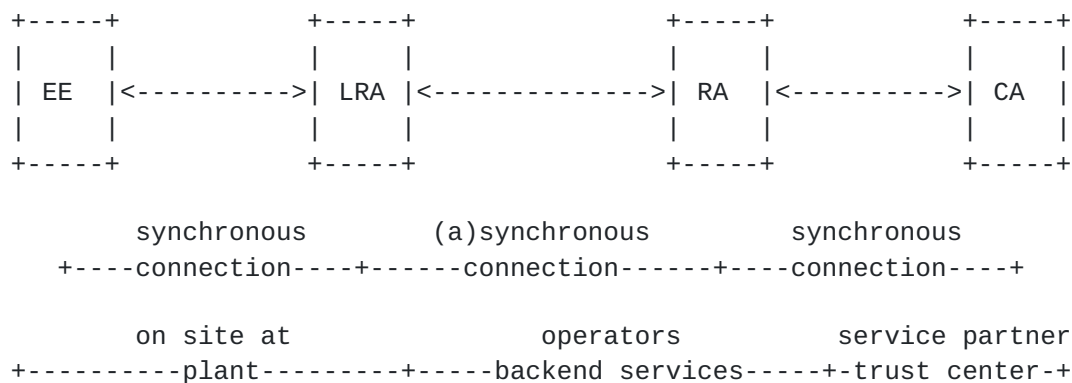


Figure 1: Certificate management on site

In operation environments a layered LRA-RA-CA architecture can be deployed, e.g., with LRAs bundling requests from multiple EEs at dedicated locations and one (or more than one) central RA aggregating the requests from multiple LRAs. Every (L)RA in this scenario will have its own dedicated certificate containing an extended key usage as specified in CMP Updates [[I-D.brockhaus-lamps-cmp-updates](#)] and private key allowing it to protect CMP messages it processes (CMP signing key/certificate). The figure above shows an architecture using one LRA and one RA. It is also possible to have only an RA or multiple LRAs and/or RAs. Depending on the network infrastructure, the communication between different PKI components may be synchronous online-communication, delayed asynchronous communication, or even offline file transfer.

This profile focusses on specifying the pull model, where the EE always requests a specific PKI management operation. CMP response messages, especially in case of central key generation, as described in [Section 5.1.6](#), can also be used proactively to implement the push model towards the EE.

Third-party CAs typically implement different variants of CMP or even use proprietary interfaces for certificate management. Therefore, the LRA or the RA may need to adapt the exchanged CMP messages to the flavor of communication required by the CA.

3.2. Basic generic CMP message content

[Section 4](#) specifies the generic parts of the CMP messages as used later in [Section 5](#) and [Section 6](#).

- o Header of a CMP message; see [Section 4.1](#).
- o Protection of a CMP message; see [Section 4.2](#).

- o ExtraCerts field of a CMP message; see [Section 4.3](#).

3.3. Supported use cases

Following the outlined scope from [Section 2.5](#), this section gives a brief overview of the certificate management use cases specified in [Section 5](#) and [Section 6](#) and points out, whether an implementation by compliant EE or PKI component is mandatory, recommended or optional.

3.3.1. Mandatory use cases

The mandatory uses case in this document shall limit the overhead of certificate management for more constrained devices to the most crucial types of transactions.

[Section 5](#) - End Entity focused certificate management use cases

- o Request a certificate from a new PKI with signature protection; see [Section 5.1.1](#).
- o Request to update an existing certificate with signature protection; see [Section 5.1.3](#).
- o Error reporting; see [Section 5.3](#).

[Section 6](#) - LRA and RA focused certificate management use cases

- o Forward messages without changes; see [Section 6.1.1](#).
- o Forward messages with replaced protection and raVerified as proof-of-possession; see [Section 6.1.2.2](#).
- o Error reporting; see [Section 6.3](#).

3.3.2. Recommended Use Cases

Additional recommended use cases shall support some more complex scenarios, that are considered as beneficial for environments with more specific boundary conditions.

[Section 5](#) - End Entity focused certificate management use cases

- o Request a certificate from a PKI with MAC protection; see [Section 5.1.4](#).
- o Handle delayed enrollment due to asynchronous message delivery; see [Section 5.1.7](#).

< TBD: There still some discussion ongoing if this should be recommended or optional. >

- o Revoke an own certificate.

[Section 6](#) - LRA and RA focused certificate management use cases

- o Revoke another's entities certificate.

[3.3.3.](#) Optional use cases

The optional use cases support specific requirements seen only in a subset of environments.

[Section 5](#) - End Entity focused certificate management use cases

- o Request a certificate from a legacy PKI using a PKCS#10 [[RFC2986](#)] request; see [Section 5.1.5](#).
- o Add central generation of a key pair to a certificate request; see [Section 5.1.6](#). If central key generation is supported, the key agreement key management technique is REQUIRED to be supported, and the key transport and symmetric key-encryption key management techniques are OPTIONAL.
- o Additional support messages, e.g., to update a Root CA certificate or to request an [RFC 8366](#) [[RFC8366](#)] voucher; see [Section 5.4](#).

[Section 6](#) - LRA and RA focused certificate management use cases

- o Initiate delayed enrollment due to asynchronous message delivery; see [Section 6.1.4](#).

[3.4.](#) CMP message transport

On different links between PKI entities, e.g., EE<->RA and RA<->CA, different transport MAY be used. As CMP has only very limited requirement regarding the mechanisms used for message transport and in different environments different transport mechanisms are supported, e.g. HTTP, CoAP, or even offline files based, this document requires no specific transport protocol to be supported by all conforming implementations.

HTTP transfer is RECOMMENDED to use for all PKI entities, but there is no transport specified as mandatory to be flexible for devices with special constraints to choose whatever transport is suitable.

Recommended transport

- o Transfer CMP messages using HTTP; see [Section 7.1](#).

Optional transport

- o Transfer CMP messages using HTTPS with certificate-based authentication; see [Section 7.2](#).
- o Transfer CMP messages using HTTPS with shared-secret based protection; see [Section 7.3](#).
- o File-based CMP message transport.

< TBD: Motivation see [Section 7.4](#) >

< TBD: Michael Richardson proposed to also specify a CoAP based message transport profile. If there is further support for this profile and someone volunteering to provide the necessary input for this section, I would add it to the document. >

4. Generic parts of the PKI message

To reduce redundancy in the text and to ease implementation, the contents of the header, protection, and extraCerts fields of the CMP messages used in the transactions specified in [Section 5](#) and [Section 6](#) are standardized to the maximum extent possible. Therefore, the generic parts of a CMP message are described centrally in this section.

As described in [section 5.1 of \[RFC4210\]](#), all CMP messages have the following general structure:

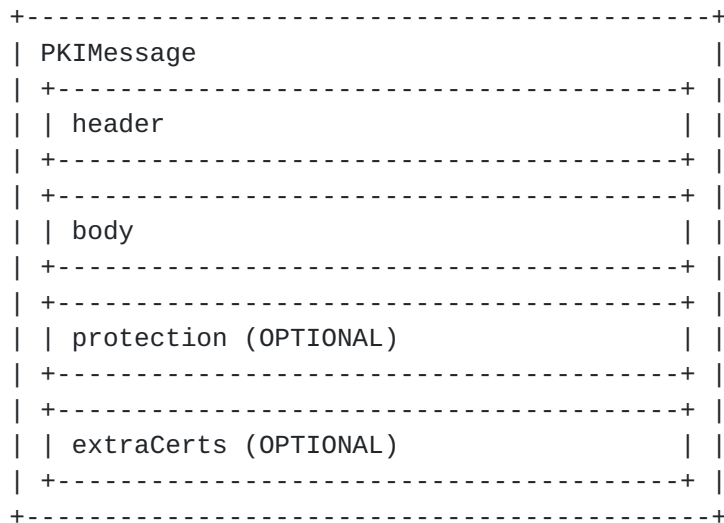


Figure 2: CMP message structure

The general contents of the message header, protection, and extraCerts fields are specified in the [Section 4.1](#) to [Section 4.3](#).

In case a specific CMP message needs different contents in the header, protection, or extraCerts fields, the differences are described in the respective message.

The CMP message body contains the message-specific information. It is described in the context of [Section 5](#) and [Section 6](#).

The behavior in case an error occurs while handling a CMP message is described in [Section 6.3](#).

[4.1](#). General description of the CMP message header

This section describes the generic header field of all CMP messages with signature-based protection. The only variations described here are in the fields recipient, transactionID, and recipNonce of the first message of a transaction.

In case a message has MAC-based protection the changes are described in the respective section. The variations will affect the fields sender, protectionAlg, and senderKID.

For requirements about proper random number generation please refer to [\[RFC4086\]](#). Any message-specific fields or variations are described in the respective sections of this chapter.

header

pvno	REQUIRED
-- MUST be set to 2 to indicate CMP V2	
sender	REQUIRED
-- MUST be the subject of the protection certificate used for,	
-- the certificate for the private key used to sign the message	
recipient	REQUIRED
-- SHOULD be the name of the intended recipient and	
-- MAY be a NULL_DN if the sender does not know the DN of	
-- the recipient	
-- If this is the first message of a transaction: SHOULD be the	
-- subject of the issuing CA certificate	
-- In all other messages: SHOULD be the same name as in the	
-- sender field of the previous message in this transaction	
messageTime	RECOMMENDED
-- MUST be the time at which the message was produced, if	
-- present	
protectionAlg	REQUIRED
-- MUST be the algorithm identifier of the signature or algorithm	
-- id-PasswordBasedMac algorithm used for calculation of the	


```
-- protection bits
-- The signature algorithm MUST be consistent with the
-- SubjectPublicKeyInfo field of the signer's certificate
-- The hash algorithm used SHOULD be SHA-256
algorithm                REQUIRED
-- MUST be the OID of the signature algorithm, like
-- sha256WithRSAEncryption or ecdsa-with-SHA256, or
-- id-PasswordBasedMac
senderKID                RECOMMENDED
-- MUST be the SubjectKeyIdentifier, if available, of the
-- protection certificate
transactionID            REQUIRED
-- If this is the first message of a transaction:
-- MUST be 128 bits of random data for the start of a
-- transaction to reduce the probability of having the
-- transactionID already in use at the server
-- In all other messages:
-- MUST be the value from the previous message in the same
-- transaction
senderNonce              REQUIRED
-- MUST be fresh 128 random bits
recipNonce               RECOMMENDED
-- If this is the first message of a transaction: SHOULD be
-- absent
-- In all other messages: MUST be present and contain the value
-- from senderNonce of the previous message in the same
-- transaction
generalInfo              OPTIONAL
implicitConfirm          OPTIONAL
  ImplicitConfirmValue    REQUIRED
-- The field is optional though it only applies to
-- ir/cr/kur/p10cr requests and ip/cp/kup responses
-- ImplicitConfirmValue of the request message MUST be NULL if
-- the EE wants to request not to send a confirmation message
-- ImplicitConfirmValue MUST be set to NULL if the (L)RA/CA wants
-- to grant not sending a confirmation message
```

4.2. General description of the CMP message protection

This section describes the generic protection field of all CMP messages with signature-based protection. The certificate for the private key used to sign a CMP message is called 'protection certificate'.

```
protection              REQUIRED
-- MUST contain the signature calculated using the signature
-- algorithm specified in protectionAlg
```


Only for MAC-based protection major differences apply as described in the respective message.

The CMP message protection provides, if available, message origin authentication and integrity protection for the CMP message header and body. The CMP message extraCerts is not covered by this protection.

NOTE: The requirements for checking certificates given in [[RFC5280](#)] MUST be followed for the CMP message protection. In case the CMP signer certificates is not the CA certificate that signed the newly issued certificate, certificate status checking SHOULD be used for the CMP signer certificates of communication partners.

4.3. General description of CMP message extraCerts

This section describes the generic extraCerts field of all CMP messages with signature-based protection.

extraCerts	RECOMMENDED
--	SHOULD contain the protection certificate together with its
--	chain, if needed
--	If present, the first certificate in this field MUST
--	be the protection certificate
--	Self-signed certificates SHOULD NOT be included in
--	extraCerts and MUST NOT be trusted based on the listing in
--	extraCerts in any case

5. End Entity focused certificate management use cases

This chapter focuses on the communication of the EE and the first PKI component it talks to. Depending on the network and PKI solution, this will either be the LRA, the RA or the CA.

Profiles of the Certificate Management Protocol (CMP) [[RFC4210](#)] handled in this chapter cover the following certificate management use cases:

- o Requesting a certificate from a PKI with variations like initial requests and updating, central key generation and different protection means
- o Revocation of a certificate
- o General messages for further support functions

The use cases mainly specify the message body of the CMP messages and utilize the specification of the message header, protection and extraCerts as specified in [Section 5](#).

The behavior in case an error occurs is described in [Section 5.3](#).

This chapter is aligned to [Appendix D](#) and [Appendix E of \[RFC4210\]](#). The general rules for interpretation stated in [Appendix D.1 in \[RFC4210\]](#) need to be applied here, too.

This document does not mandate any specific supported algorithms like [Appendix D.2 of \[RFC4210\]](#), [\[ETSI-3GPP\]](#), and [\[UNISIG\]](#) do. Using the message sequences described here require agreement upon the algorithms to support and thus the algorithm identifiers for the specific target environment.

5.1. Requesting a new certificate from a PKI

There are different approaches to request a certificate from a PKI.

These approaches differ on the one hand in the way the EE can authenticate itself to the PKI it wishes to get a new certificate from and on the other hand in its capabilities to generate a proper new key pair. The authentication means may be as follows:

- o Using a certificate from a trusted PKI and the corresponding private key, e.g., a manufacturer certificate
- o Using the certificate to be updated and the corresponding private key
- o Using a shared secret known to the EE and the PKI

Typically, such EE requests a certificate from a CA. When the (L)RA/CA responds with a message containing a certificate, the EE MUST reply with a confirmation message. The (L)RA/CA then MUST send confirmation back, closing the transaction.

The message sequences in this section allow the EE to request certification of a locally generated public-private key pair. For requirements about proper random number and key generation please refer to [\[RFC4086\]](#). The EE MUST provide a signature-based proof-of-possession of the private key associated with the public key contained in the certificate request as defined by [\[RFC4211\] section 4.1](#) case 3. To this end it is assumed that the private key can technically be used as signing key. The most commonly used algorithms are RSA and ECDSA, which can technically be used for

signature calculation regardless of potentially intended restrictions of the key usage.

The requesting EE provides the binding of the proof-of-possession to its identity by signature-based or MAC-based protection of the CMP request message containing that POPO. The (L)RA/CA needs to verify whether this EE is authorized to obtain a certificate with the requested subject and other attributes and extensions. Especially when removing the protection provided by the EE and applying a new protection the (L)RA MUST verify in particular the included proof-of-possession self-signature of the certTemplate using the public key of the requested certificate and MUST check that the EE, as authenticated by the message protection, is authorized to request a certificate with the subject as specified in the certTemplate (see [Section 6.1.2](#)).

There are several ways to install the Root CA certificate of a new PKI on an EE. The installation can be performed in an out-of-band manner, using general messages, a voucher [[RFC8366](#)], or other formats for enrollment, or in-band of CMP by the caPubs field in the certificate response message. In case the installation of the new Root CA certificate is performed using the caPubs field, the certificate response message MUST be properly authenticated, and the sender of this message MUST be authorized to install new Root CA certificates on the EE. This authorization MUST be indicated by the extended key usage in the (L)RA/CA certificate as specified in CMP Updates [[I-D.brockhaus-lamps-cmp-updates](#)].

[5.1.1](#). A certificate from a new PKI with signature protection

This message sequence should be used by an EE to request a certificate of a new PKI using an existing certificate from an external PKI, e.g., a manufacturer certificate, to prove its identity to the new PKI. The EE already has established trust in this new PKI it is about to enroll to, e.g., by configuration means. The initialization request message is signature-protected using the existing certificate.

Preconditions:

- 1 The EE MUST have a certificate enrolled by an external PKI in advance to this transaction to authenticate itself to the (L)RA/CA using signature-based protection, e.g., using a manufacturer certificate.
- 2 The EE SHOULD know the subject name of the new CA it requests a certificate from; this name MAY be established using an enrollment voucher or other configuration means. If the EE does not know the

name of the CA, the (L)RA/CA MUST know where to route this request to.

- 3 The EE MUST authenticate responses from the (L)RA/CA; trust MAY be established using an enrollment voucher or other configuration means
- 4 The (L)RA/CA MUST trust the external PKI the EE uses to authenticate itself; trust MAY be established using some configuration means

This message sequence is like that given in [\[RFC4210\] Appendix E.7](#).

Message flow:

Step#	EE		(L)RA/CA
1	format ir		
2		-> ir	->
3			handle, re-protect or forward ir
4			format or receive ip
5			possibly grant implicit confirm
6		<- ip	<-
7	handle ip		
8			In case of status "rejection" in the ip message, no certConf and pkiConf are sent
9	format certConf (optional)		
10		-> certConf	->
11			handle, re-protect or forward certConf
12			format or receive PKIConf
13		<- pkiConf	<-
14	handle pkiConf (optional)		

For this message sequence the EE MUST include exactly one single CertReqMsg in the ir. If more certificates are required, further requests MUST be sent using separate CMP Messages. If the EE wants to omit sending a certificate confirmation message after receiving the ip to reduce the number of protocol messages exchanged in a transaction, it MUST request this by setting the implicitControlValue in the ir to NULL.

If the CA accepts the request it MUST return the new certificate in the certifiedKeyPair field of the ip message. If the EE requested to omit sending a certConf message after receiving the ip, the (L)RA/CA

MAY confirm this by also setting the `implicitControlValue` in the `ip` to `NULL`.

If the EE did not request implicit confirmation or the request was not granted by the (L)RA/CA the confirmation as follows MUST be performed. If the EE successfully receives the certificate and accepts it, the EE MUST send a `certConf` message, which MUST be answered by the (L)RA/CA with a `pkiConf` message. If the (L)RA/CA does not receive the expected `certConf` message in time it MUST handle this like a rejection by the EE.

If the certificate request was refused by the CA, the (L)RA/CA must return an `ip` message containing the status code "rejection" and no `certifiedKeyPair` field. Such an `ip` message MUST NOT be followed by the `certConf` and `pkiConf` messages.

Detailed message description:

Certification Request -- `ir`

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

-- The request of the EE for a new certificate

<code>ir</code>	REQUIRED
-----------------	----------

-- MUST be exactly one `CertReqMsg`

-- If more certificates are required, further requests MUST be
 -- packaged in separate PKI Messages

<code>certReq</code>	REQUIRED
----------------------	----------

<code>certReqId</code>	REQUIRED
------------------------	----------

-- MUST be set to 0

<code>certTemplate</code>	REQUIRED
---------------------------	----------

<code>version</code>	OPTIONAL
----------------------	----------

-- MUST be 2 if supplied.

<code>subject</code>	REQUIRED
----------------------	----------

-- MUST contain the suggested subject name of the EE

-- certificate

<code>publicKey</code>	REQUIRED
------------------------	----------

<code>algorithm</code>	REQUIRED
------------------------	----------

-- MUST include the subject public key algorithm ID and value

-- In case a central key generation is requested, this field

-- contains the algorithm and parameter preferences of the

-- requesting entity regarding the to-be-generated key pair

<code>subjectPublicKey</code>	REQUIRED
-------------------------------	----------

-- MUST contain the public key to be included into the requested


```

-- certificate in case of local key-generation
-- MUST contain a zero-length BIT STRING in case a central key
-- generation is requested
-- MUST include the subject public key algorithm ID and value
    extensions                OPTIONAL
-- MAY include end-entity-specific X.509 extensions of the
-- requested certificate like subject alternative name,
-- key usage, and extended key usage
Popo                          REQUIRED
    POPOSigningKey            OPTIONAL
-- MUST be used in case subjectPublicKey contains a public key
-- MUST be absent in case subjectPublicKey contains a
-- zero-length BIT STRING
    poposkInput               PROHIBITED
-- MUST NOT be used because subject and publicKey are both
-- present in the certTemplate
    algorithmIdentifier        REQUIRED
-- The signature algorithm MUST be consistent with the
-- publicKey field of the certTemplate
-- The hash algorithm used SHOULD be SHA-256
    signature                  REQUIRED
-- MUST be the signature computed over the DER-encoded
-- certTemplate

protection                    REQUIRED
-- As described in section 4.2

extraCerts                    REQUIRED
-- As described in section 4.3

Certification Response -- ip

Field                          Value

header
-- As described in section 4.1

body
-- The response of the CA to the request as appropriate
ip                              REQUIRED
    caPubs                      OPTIONAL
-- MAY be used
-- If used it MUST contain only the root certificate of the
-- certificate contained in certOrEncCert
response                        REQUIRED
-- MUST be exactly one CertResponse
    certReqId                   REQUIRED

```



```

-- MUST be set to 0
    status                REQUIRED
-- PKIStatusInfo structure MUST be present
    status                REQUIRED
-- positive values allowed: "accepted", "grantedWithMods"
-- negative values allowed: "rejection"
-- In case of rejection no certConf and pkiConf messages will
-- be sent
    statusString          OPTIONAL
-- MAY be any human-readable text for debugging, logging or to
-- display in a GUI
    failInfo              OPTIONAL
-- MUST be present if status is "rejection" and in this case
-- the transaction MUST be terminated
-- MUST be absent if the status is "accepted" or
-- "grantedWithMods"
    certifiedKeyPair      OPTIONAL
-- MUST be present if status is "accepted" or "grantedWithMods"
-- MUST be absent if status is "rejection"
    certOrEncCert         REQUIRED
-- MUST be present when certifiedKeyPair is present
    certificate            REQUIRED
-- MUST be present when certifiedKeyPair is present
-- MUST contain the newly enrolled X.509 certificate
    privateKey            OPTIONAL
-- MUST be absent in case of local key-generation
-- MUST contain the encrypted private key in an EnvelopedData
-- structure as specified in section 5.1.5 in case the private
-- key was generated centrally

protection                REQUIRED
-- As described in section 4.2

extraCerts                REQUIRED
-- As described in section 4.3
-- MUST contain the chain of the issued certificate
-- Duplicate certificates MAY be omitted

Certificate Confirmation -- certConf

Field                      Value

header
-- As described in section 4.1

body
-- The message of the EE sends confirmation to the (L)RA/CA

```



```

    -- to accept or reject the issued certificates
certConf                                REQUIRED
    -- MUST be exactly one CertStatus
CertStatus                             REQUIRED
    certHash                            REQUIRED
    -- MUST be the hash of the certificate, using the same hash
    -- algorithm as used to create the certificate signature
    certReqId                           REQUIRED
    -- MUST be set to 0
    status                               RECOMMENDED
    -- PKIStatusInfo structure SHOULD be present
    -- Omission indicates acceptance of the indicated certificate
    status                               REQUIRED
    -- positive values allowed: "accepted"
    -- negative values allowed: "rejection"
    statusString                         OPTIONAL
    -- MAY be any human-readable text for debugging or logging
    failInfo                            OPTIONAL
    -- MUST be present if status is "rejection"
    -- MUST be absent if the status is "accepted"

protection                             REQUIRED
    -- As described in section 4.2
    -- MUST use the same certificate as for protection of the ir

extraCerts                             RECOMMENDED
    -- SHOULD contain the protection certificate together with its
    -- chain
    -- If present, the first certificate in this field MUST be the
    -- certificate used for signing this message
    -- Self-signed certificates SHOULD NOT be included in
    -- extraCerts and
    -- MUST NOT be trusted based on the listing in extraCerts in
    -- any case

PKI Confirmation -- pkiConf

Field                                  Value

header
    -- As described in section 4.1

body
    pkiConf                            REQUIRED
    -- The content of this field MUST be NULL

protection                             REQUIRED

```


- As described in [section 4.2](#)
- SHOULD use the same certificate as for protection of the ip

extraCerts

RECOMMENDED

- SHOULD contain the protection certificate together with its
- chain
- If present, the first certificate in this field MUST be the
- certificate used for signing this message
- Self-signed certificates SHOULD NOT be included in extraCerts
- and
- MUST NOT be trusted based on the listing in extraCerts in
- any case

[5.1.2.](#) A certificate from a trusted PKI with signature protection

< TBD: In case the PKI is already trusted the cr/cp messages could be used instead of ir/ip. It needs to be decided, whether an additional section should be added here, or the previous section should be extended to also cover this use case. >

[5.1.3.](#) Update an existing certificate with signature protection

This message sequence should be used by an EE to request an update of one of the certificates it already has and that is still valid. The EE uses the certificate it wishes to update to prove its identity and possession of the private key for the certificate to be updated to the PKI. Therefore, the key update request message is signed using the certificate that is to be updated.

The general message flow for this message sequence is the same as given in [Section 5.1.1](#).

Preconditions:

- 1 The certificate the EE wishes to update MUST NOT be expired or revoked.
- 2 A new public-private key pair SHOULD be used.

The message sequence for this exchange is like that given in [\[RFC4210\] Appendix D.6](#).

The message sequence for this exchange is identical to that given in [Section 5.1.1](#), with the following changes:

- 1 The body of the first request and response MUST be kur and kup, respectively.

- 2 Protection of the kur MUST be performed using the certificate to be updated.
- 3 The subject field of the CertTemplate MUST contain the subject name of the existing certificate to be updated, without modifications.
- 4 The CertTemplate MUST contain the subject, issuer and publicKey fields only.
- 5 The regCtrl OldCertId SHOULD be used to make clear, even in case an (L)RA changes the message protection, which certificate is to be used.
- 6 The caPubs field in the kup message MUST be absent.

As part of the certReq structure of the kur the control is added right after the certTemplate.

```
controls
  type                RECOMMENDED
  -- MUST be the value id-regCtrl-oldCertID, if present
  value
    issuer             REQUIRED
    serialNumber       REQUIRED
  -- MUST contain the issuer and serialNumber of the certificate
  -- to be updated
```

5.1.4. A certificate from a PKI with MAC protection

This message sequence should be used by an EE to request a certificate of a new PKI without having a certificate to prove its identity to the target PKI, but there is a shared secret established between the EE and the PKI. Therefore, the initialization request is MAC-protected using this shared secret. The (L)RA checking the MAC-protection SHOULD replace this protection according to [Section 6.1.2](#) in case the next hop does not know the shared secret.

For requirements with regard to proper random number and key generation please refer to [\[RFC4086\]](#).

The general message flow for this message sequence is the same as given in [Section 5.1.1](#).

Preconditions:

- 1 The EE and the (L)RA/CA MUST share a symmetric key, this MAY be established by a service technician during initial local configuration.
- 2 The EE SHOULD know the subject name of the new CA it requests a certificate from; this name MAY be established using an enrollment voucher or other configuration means. If the EE does not know the name of the CA, the (L)RA/CA MUST know where to route this request to.
- 3 The EE MUST authenticate responses from the (L)RA/CA; trust MAY be established using the shared symmetric key.

The message sequence for this exchange is like that given in [\[RFC4210\] Appendix D.4](#).

The message sequence for this exchange is identical to that given in [Section 5.1.1](#), with the following changes:

- 1 The protection of all messages MUST be calculated using Message Authentication Code (MAC); the protectionAlg field MUST be id-PasswordBasedMac as described in [section 5.1.3.1 of \[RFC4210\]](#).
- 2 The sender MUST contain a name representing the originator of the message. The senderKID MUST contain a reference all participating entities can use to identify the symmetric key used for the protection.
- 3 The extraCerts of the ir, certConf, and PKIConf messages MUST be absent.
- 4 The extraCerts of the ip message MUST contain the chain of the issued certificate and root certificates SHOULD not be included and MUST NOT be trusted in any case.

Part of the protectionAlg structure, where the algorithm identifier MUST be id-PasswordBasedMac, is a PBMPParameter sequence. The fields of PBMPParameter SHOULD remain constant for message protection throughout this certificate management transaction to reduce the computational overhead.

PBMPParameter REQUIRED
salt REQUIRED
-- MUST be the random value to salt the secret key
owf REQUIRED
-- MUST be the algorithm identifier for the one-way function
-- used
-- The one-way function SHA-1 MUST be supported due to
-- [\[RFC4211\]](#) requirements, but SHOULD NOT be used any more
-- SHA-256 SHOULD be used instead
iterationCount REQUIRED
-- MUST be a limited number of times the OWF is applied
-- To prevent brute force and dictionary attacks a reasonable
-- high number SHOULD be used
mac REQUIRED
-- MUST be the algorithm identifier of the MAC algorithm used
-- The MAC function HMAC-SHA1 MUST be supported due to
-- [\[RFC4211\]](#) requirements, but SHOULD NOT be used any more
-- HMAC-SHA-256 SHOULD be used instead

< TBD: SHA-1 is no collision resistant hash algorithm. Due to this fact the usage of SHA-1 has significantly decreased. Currently HMAC-SHA-1 seems relatively secure, it is currently recommended by cryptographers to also depreciate the uses of SHA-1 in the context of HMAC calculation. Should we depreciate the support of SHA-1 here completely? >

5.1.5. A certificate from a legacy PKI using PKCS#10 request

This message sequence should be used by an EE to request a certificate of a legacy PKI only capable to process PKCS#10 [\[RFC2986\]](#) certification requests. The EE can prove its identity to the target PKI by using various protection means as described in [Section 5.1.1](#) or [Section 5.1.4](#).

In contrast to the other transactions described in [Section 5.1](#), this transaction uses PKCS#10 [\[RFC2986\]](#) instead of CRMF [\[RFC4211\]](#) for the certificate request for compatibility reasons with legacy CA systems that require a PKCS#10 certificate request and cannot process CMP [\[RFC4210\]](#) or CRMF [\[RFC4211\]](#) messages. In such case the (L)RA must extract the PKCS#10 certificate request from the p10cr and provides it separately to the CA.

The general message flow for this message sequence is the same as given in [Section 5.1.1](#), but the public key is contained in the subjectPKInfo of the PKCS#10 certificate request.

Preconditions:

- 1 The EE MUST either have a certificate enrolled from this or any other accepted PKI, or a shared secret known to the PKI and the EE to authenticate itself to the (L)RA/CA.
- 2 The EE SHOULD know the subject name of the CA it requests a certificate from; this name MAY be established using an enrollment voucher or other configuration means. If the EE does not know the name of the CA, the (L)RA/CA MUST know where to route this request to.
- 3 The EE MUST authenticate responses from the (L)RA/CA; trust MAY be established by an available root certificate, using an enrollment voucher, or other configuration means.
- 4 The (L)RA/CA MUST trust the current or the PKI the EE uses to authenticate itself; trust MAY be established by a corresponding available root certificate or using some configuration means.

The profile for this exchange is identical to that given in [Section 5.1.1](#), with the following changes:

- 1 The body of the first request and response MUST be p10cr and cp, respectively.
- 2 The subject name of the CA MUST be in the recipient field of the p10cr message header.
- 3 The certReqId in the cp message MUST be 0.
- 4 The caPubs field in the cp message SHOULD be absent.

Detailed description of the p10cr message:

Certification Request -- p10cr

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

-- The request of the EE for a new certificate using a PKCS#10

-- certificate request

p10cr	REQUIRED
-------	----------

CertificationRequestInfo	REQUIRED
--------------------------	----------

version	REQUIRED
---------	----------

-- MUST be set to 0 to indicate PKCS#10 V1.7

subject	REQUIRED
---------	----------

-- MUST contain the suggested subject name of the EE

subjectPKInfo	REQUIRED
---------------	----------

-- MUST include the subject public key algorithm ID and value

attributes	OPTIONAL
------------	----------

-- MAY contain a set of end-entity-specific attributes or X.509

-- extensions to be included in the requested certificate or used

-- otherwise

signatureAlgorithm	REQUIRED
--------------------	----------

-- The signature algorithm MUST be consistent with the

-- subjectPKInfo field. The hash algorithm used SHOULD be SHA-256

signature	REQUIRED
-----------	----------

-- MUST containing the self-signature for proof-of-possession

protection	REQUIRED
------------	----------

-- As described in [section 4.2](#)

extraCerts	REQUIRED
------------	----------

-- As described in [section 4.3](#)

[5.1.6](#). Generate the key pair centrally at the (L)RA/CA

This functional extension can be applied in combination with certificate enrollment as described in [Section 5.1.1](#) and [Section 5.1.4](#). The functional extension can be used in case an EE is not able or is not willing to generate its new public-private key pair itself. It is a matter of the local implementation which central PKI components will perform the key generation. This component must have a proper (L)RA/CA certificate containing the additional extended key usage id-kp-cmckGA to be identified by the EE as a legitimate key-generation instance. In case the (L)RA generated

the new key pair for the EE, it can use [Section 5.1.1](#) to [Section 5.1.4](#) to request the certificate for this key pair as usual.

Generally speaking, in a machine-to-machine scenario it is strongly preferable to generate public-private key pairs locally at the EE. Together with proof-of-possession of the private key in the certification request, this is to make sure that only the entity identified in the newly issued certificate is the only entity who ever hold the private key.

There are some cases where an EE is not able or not willing to locally generate the new key pair. Reasons for this may be the following:

- o Lack of sufficient initial entropy.

Note: Good random numbers are not only needed for key generation, but also for session keys and nonces in any security protocol. Therefore, we believe that a decent security architecture should anyways support good random number generation on the EE side or provide enough entropy for the RNG seed during manufacturing to guarantee good initial pseudo-random number generation.

- o Due to lack of computational resources, e.g., in case of RSA keys.

Note: As key generation can be performed in advance to the certificate enrollment communication, it is typical not time critical.

Note: Besides the initial enrollment right after the very first bootup of the device, where entropy available on the device may be insufficient, we do not see any good reason for central key generation.

Note: As mentioned in [Section 3.1](#) central key generation may be required in a push model, where the certificate response message is transferred by the (L)RA/CA to the EE without receiving a previous request message.

If the EE wishes to request central key generation, it MUST fill the subjectPublicKey field in the certTemplate structure of the request message with a zero-length BIT STRING. This indicates to the (L)RA/CA that a new key pair shall be generated centrally on behalf of the EE.

Note: As the protection of centrally generated keys in the response message is being extended from EncryptedValue to EncryptedKey by CMP Updates [[I-D.brockhaus-lamps-cmp-updates](#)] also the alternative

EnvelopedData can be used. In CRMF [Section 2.1.9 \[RFC4211\]](#) the use of EncryptedValue has been deprecated in favor of the EnvelopedData structure. Therefore, this profile specifies using EnvelopedData as specified in CMS [Section 6 \[RFC5652\]](#) to offer more crypto agility.

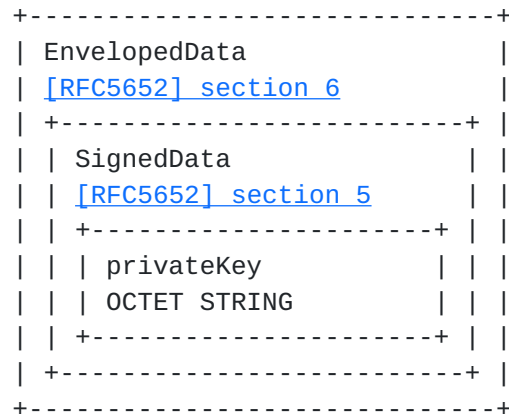


Figure 3: Encrypted private key container

The (L)RA/CA delivers the private key in the privateKey field in the certifiedKeyPair structure of the response message also containing the newly issued certificate.

The private key MUST be wrapped in a SignedData structure, as specified in CMS [Section 5 \[RFC5652\]](#), signed by the KGA generating the key pair. The signature MUST be performed using a CMP signer certificate asserting the extended key usage kp-id-cmpKGA as described in CMP Updates [\[I-D.brockhaus-lamps-cmp-updates\]](#) to show the authorization to generate key pairs on behalf of an EE.

This SignedData structure MUST be wrapped in an EnvelopedData structure, as specified in CMS [Section 6 \[RFC5652\]](#), encrypting it using a newly generated symmetric content-encryption key.

Note: Instead of the specification in CMP [Appendix D 4.4 \[RFC4210\]](#) this content-encryption key is not generated on the EE side. As we just mentioned, central key generation should only be used in this profile in case of lack of randomness on the EE.

As part of the EnvelopedData structure this content-encryption key MUST be securely provided to the EE using one of three key management techniques. The choice of the key management technique to be used by the (L)RA/CA depends on the authentication mechanism the EE choose to protect the request message, see CMP Updates [section 3.4 \[I-D.brockhaus-lamps-cmp-updates\]](#) for more details on which key management technique to use.

- o MAC protected request message: The content-encryption key SHALL be protected using the symmetric key-encryption key management technique, see [Section 5.1.6.1](#), only if the EE used MAC protection for the respected request message.
- o Signature protected request message using a certificate that contains a key usage extension asserting keyAgreement: The content-encryption key SHALL be protected using the key agreement key management technique, see [Section 5.1.6.2](#), if the certificate used by the EE for signing the respective request message contains the key usage keyAgreement. If the certificate also contains the key usage keyEncipherment, the key transport key management technique SHALL NOT be used.
- o Signature protected request message using a certificate that contains a key usage extension asserting keyEncipherment: The content-encryption key SHALL be protected using the key transport key management technique, see [Section 5.1.6.3](#), if the certificate used by the EE for signing the respective request message contains the key usage keyEncipherment and not keyAgreement.

The key agreement key management technique can be supported by most signature algorithms, as key transport key management technique can only be supported by a very limited number of algorithms. The symmetric key-encryption key management technique shall only be used in combination with MAC protection, which is a side-line in this profile. Therefore, this profile REQUIRES support of the key agreement key management technique and the key transport and symmetric key-encryption key management techniques are OPTIONAL.

For encrypting the SignedData structure containing the private key a fresh content-encryption key MUST be generated with enough entropy with regard to the used symmetric encryption algorithm.

Note: Depending on the lifetime of the certificate and the criticality of the generated private key, it is advisable to use the strongest possible symmetric encryption algorithm. Therefore, this specification recommends using at least AES-256.

The detailed description of the privateKey field looks like this:

```
privateKey          OPTIONAL
-- MUST be an envelopedData structure as specified in
-- CMS \[RFC5652\] section 6
version            REQUIRED
-- MUST be set to 2
recipientInfos      REQUIRED
-- MUST be exactly one RecipientInfo
```



```
        recipientInfo      REQUIRED
-- MUST be either KEKRecipientInfo (see section 5.1.5.1),
-- KeyAgreeRecipientInfo (see section 5.1.5.2), or
-- KeyTransRecipientInfo (see section 5.1.5.3) is used
        encryptedContentInfo
                                REQUIRED
        contentType        REQUIRED
-- MUST be id-signedData
        contentEncryptionAlgorithm
                                REQUIRED
-- MUST be the algorithm identifier of the symmetric
-- content-encryption algorithm used
-- As private keys need long-term protection, the use of AES-256
-- or a stronger symmetric algorithm is RECOMMENDED
        encryptedContent    REQUIRED
-- MUST be the encrypted signedData structure as specified in
-- CMS \[RFC5652\] section 5
        version            REQUIRED
-- MUST be set to 3
        digestAlgorithms
                                REQUIRED
-- MUST be exactly one digestAlgorithm identifier
        digestAlgorithmIdentifier
                                REQUIRED
-- MUST be the OID of the digest algorithm used for generating
-- the signature
-- The hash algorithm used SHOULD be SHA-256
        encapContentInfo
                                REQUIRED
-- MUST be the content that is to be signed
        contentType        REQUIRED
-- MUST be id-data
        content            REQUIRED
-- MUST be the privateKey as OCTET STRING
        certificates        REQUIRED
-- SHOULD contain the certificate, for the private key used
-- to sign the content, together with its chain
-- If present, the first certificate in this field MUST
-- be the certificate used for signing this content
-- Self-signed certificates SHOULD NOT be included
-- and MUST NOT be trusted based on the listing in any case
        crls                OPTIONAL
-- MAY be present to provide status information on the signer or
-- its CA certificates
        signerInfos         REQUIRED
-- MUST be exactly one signerInfo
        version            REQUIRED
-- MUST be set to 3
```



```
        sid                REQUIRED
        subjectKeyIdentifier
                           REQUIRED
-- MUST be the subjectKeyIdentifier of the signer's certificate
   digest algorithm
                           REQUIRED
-- MUST be the same OID as in digest algorithm
   signatureAlgorithm
                           REQUIRED
-- MUST be the algorithm identifier of the signature algorithm
-- used for calculation of the signature bits,
-- like sha256WithRSAEncryption or ecdsa-with-SHA256
-- The signature algorithm MUST be consistent with the
-- SubjectPublicKeyInfo field of the signer's certificate
   signature              REQUIRED
-- MUST be the result of the digital signature generation
```

5.1.6.1. Using symmetric key-encryption key management technique

This key management technique can be applied in combination with the message flow specified in [Section 5.1.4](#) using MAC protected CMP messages. The shared secret used for the MAC protection MUST also be used for the encryption of the content-encryption key but with a different seed in the PBMPParameter sequence. To use this key management technique the KEKRecipientInfo structure MUST be used in the contentInfo field.

The KEKRecipientInfo structure included into the envelopedData structure is specified in CMS [Section 6.2.3 \[RFC5652\]](#).

The detailed description of the KEKRecipientInfo structure looks like this:

```

    recipientInfo      REQUIRED
-- MUST be KEKRecipientInfo as specified in
-- CMS section 6.2.3 \[RFC5652\]
    version            REQUIRED
-- MUST be set to 4
    kekid              REQUIRED
    keyIdentifier      REQUIRED
-- MUST contain the same value as the senderKID in the respective
-- request messages
    keyEncryptionAlgorithm
                        REQUIRED
-- MUST be id-PasswordBasedMac
    PBMPParameter      REQUIRED
    salt               REQUIRED
-- MUST be the random value to salt the secret key
-- MUST be a different value than used in the PBMPParameter
-- data structure of the CMP message protection in the
-- header of this message
    owf                REQUIRED
-- MUST be the same value than used in the PBMPParameter
-- data structure in the header of this message
    iterationCount
                        REQUIRED
-- MUST be a limited number of times the OWF is applied
-- To prevent brute force and dictionary attacks a reasonable
-- high number SHOULD be used
    mac                REQUIRED
-- MUST be the same as in the contentEncryptionAlgorithm field
    encryptedKey       REQUIRED
-- MUST be the encrypted content-encryption key

```

< TBD: To make use of a different symmetric keys for encrypting the private key and for MAC-protection of the CMP message, we derive another key using the same PBMPParameter structure from CMP, even though from the perspective of field names, it is not intended to be used for deriving encryption keys. Does anyone sees a better solution here? >

5.1.6.2. Using key agreement key management technique

This key management technique can be applied in combination with the message flow specified in [Section 5.1.1](#) using signature-based protected CMP messages. The public key of the EE certificate used for the signature-based protection of the request message MUST also be used for the Ephemeral-Static Diffie-Hellmann key establishment of

the content-encryption key. To use this key management technique the KeyAgreeRecipientInfo structure MUST be used in the contentInfo field.

The KeyAgreeRecipientInfo structure included into the envelopedData structure is specified in CMS [Section 6.2.2 \[RFC5652\]](#).

The detailed description of the KeyAgreeRecipientInfo structure looks like this:

```

    recipientInfo      REQUIRED
-- MUST be KeyAgreeRecipientInfo as specified in
    version           REQUIRED
-- MUST be set to 3
    originator        REQUIRED
-- MUST contain the originatorKey sequence
    algorithm          REQUIRED
-- MUST be the algorithm identifier of the
-- static-ephemeral Diffie-Hellmann algorithm
    publicKey          REQUIRED
-- MUST be the ephemeral public key of the sending party
    ukm                OPTIONAL
-- MUST be used when 1-pass ECMQV is used
    keyEncryptionAlgorithm
                        REQUIRED
-- MUST be the same as in the contentEncryptionAlgorithm field
    recipientEncryptedKeys
                        REQUIRED
-- MUST be exactly one recipientEncryptedKey sequence
    recipientEncryptedKey
                        REQUIRED
    rid                REQUIRED
    rKeyId              REQUIRED
    subjectKeyID        REQUIRED
-- MUST contain the same value as the senderKID in the respective
-- request messages
    encryptedKey        REQUIRED
-- MUST be the encrypted content-encryption key

```

[5.1.6.3](#). Using key transport key management technique

This key management technique can be applied in combination with the message flow specified in [Section 5.1.1](#) using signature-based protected CMP messages. The public key of the EE certificate used for the signature-based protection of the request message MUST also be used for key encipherment of the content-encryption key. To use

this key management technique the KeyTransRecipientInfo structure MUST be used in the contentInfo field.

The KeyTransRecipientInfo structure included into the envelopedData structure is specified in CMS [Section 6.2.1 \[RFC5652\]](#).

The detailed description of the KeyTransRecipientInfo structure looks like this:

```
        recipientInfo      REQUIRED
-- MUST be KeyTransRecipientInfo as specified in
-- CMS section 6.2.1 \[RFC5652\]
        version            REQUIRED
-- MUST be set to 2
        rid                REQUIRED
        subjectKeyIdentifier
                        REQUIRED
-- MUST contain the same value as the senderKID in the respective
-- request messages
        keyEncryptionAlgorithm
                        REQUIRED
-- MUST contain the key encryption algorithm identifier used for
-- public key encryption
        encryptedKey      REQUIRED
-- MUST be the encrypted content-encryption key
```

[5.1.7.](#) Delayed enrollment

This functional extension can be applied in combination with certificate enrollment as described in [Section 5.1.1](#) to [Section 5.1.5](#). The functional extension can be used in case a (L)RA/CA cannot respond to the certificate request in a timely manner, e.g., due to offline upstream communication or required registration officer interaction. Depending on the PKI architecture, it is not necessary that the PKI component directly communicating with the EE initiates the delayed enrollment.

The PKI component initiating the delayed enrollment MUST include the status "waiting" in the response and this response MUST not contain the newly issued certificate. When receiving a response with status "waiting" the EE MUST send a poll request to the (L)RA/CA. The PKI component that initiated the delayed enrollment MUST answers with a poll response containing a checkAfter time. This value indicates the minimum number of seconds that must elapse before the EE sends another poll request. As soon as the (L)RA/CA can provide the final response message for the initial request of the EE, it MUST provide this in response to a poll request. After receiving this response,

the EE can continue the original message sequence as described in the respective section of this document, e.g., send a certConf message.

Typically, intermediate PKI entities SHOULD NOT change the sender and recipient nonce even in case an intermediate (L)RA modifies a request or a response message. In the special case of polling between EE and LRA with offline transport between an LRA and RA, see [Section 6.1.4](#), an exception occurs. The EE and LRA exchange pollReq and pollRep messages handle the nonce words as described. When, after pollRep, the final response from the CA arrives at the LRA, the next response will contain the recipientNonce set to the value of the senderNonce in the original request message (copied by the CA). The LRA needs to replace the recipientNonce in this case with the senderNonce of the last pollReq because the EE will validate it in this way.

Message flow:

Step#	EE	(L)RA/CA
1	format ir/cr/p10cr/kur As described in the respective section in this document	
2		->ir/cr/p10cr/kur->
3		handle request as described in the respective section in this document
4		in case no immediate final response is possible, receive or format ip, cp or kup message containing status "waiting"
5		<- ip/cp/kup <-
6	handle ip/cp/kup	
7	format pollReq	
8		-> pollReq ->
9		handle, re-protect or forward pollReq
10		in case the requested certificate or a corresponding response message is available, receive or format ip, cp, or kup containing the issued certificate, or format or receive pollRep with appropriate checkAfter value
11		<- pollRep <-
12	handle pollRep	
13	let checkAfter time elapse	
14	continue with line 7	

Detailed description of the first ip/cp/kup:

Response with status 'waiting' -- ip/cp/kup

Field	Value
-------	-------

header

- MUST contain a header as described for the first response
- message of the respective scheme

body

- The response of the (L)RA/CA to the request in case no
- immediate appropriate response can be sent
- ip/cp/kup REQUIRED
- response REQUIRED
- MUST be exactly one CertResponse
- certReqId REQUIRED
- MUST be set to 0
- status REQUIRED
- PKIStatusInfo structure MUST be present
- status REQUIRED
- MUST be set to "waiting"
- statusString OPTIONAL
- MAY be any human-readable text for debugging, logging or to
- display in a GUI
- failInfo PROHIBITED
- certifiedKeyPair PROHIBITED

protection REQUIRED

- MUST contain protection as described for the first response
- message of the respective profile, but
- MUST use the protection key of the (L)RA/CA initiating the
- delayed enrollment and creating this response message

extraCerts REQUIRED

- MUST contain certificates as described for the first response
- message of the respective profile.
- As no new certificate is issued yet, no respective certificate
- chain is included.

Polling Request -- pollReq

Field	Value
-------	-------

header

- MUST contain a header as described for the certConf message
- of the respective scheme

body

- The message of the EE asks for the final response or for a
- time to check again
- pollReq REQUIRED
- certReqId REQUIRED
- MUST be exactly one value
- MUST be set to 0

protection REQUIRED


```
-- MUST contain protection as described for the certConf message
-- of the respective profile
```

extraCerts OPTIONAL

```
-- If present, it MUST contain certificates as described for the
-- certConf message of the respective profile.
```

Polling Response -- pollRep

Field	Value
-------	-------

header

```
-- MUST contain a header as described for the pkiConf message
-- of the respective scheme
```

body	pollRep
------	---------

```
-- The message indicated the time to after which the EE may
-- send another pollReq messaged for this transaction
```

pollRep REQUIRED

```
-- MUST be exactly one set of the following values
```

certReqId	REQUIRED
-----------	----------

- MUST be set to 0

checkAfter	REQUIRED
------------	----------

```
-- time in seconds to elapse before a new pollReq may be sent by
-- the EE
```

protection REQUIRED

```
-- MUST contain protection as described for the pkiConf message
-- of the respective profile, but
-- MUST use the protection key of the (L)RA/CA that initiated the
-- delayed enrollment and is creating this response message
```

extraCerts OPTIONAL

```
-- If present, it MUST contain certificates as described for the
-- pkiConf message of the respective profile.
```

Final response -- ip/cp/kup

Field	Value
-------	-------

header

```
-- MUST contain a header as described for the first
-- response message of the respective scheme
-- but the recipientNonce MUST be the senderNonce of the last
-- pollReq message
```


body

- The response of the (L)RA/CA to the initial request as
- described in the respective profile

protection

REQUIRED

- MUST contain protection as described for the first response
- message of the respective profile, but
- MUST use the protection key of the (L)RA/CA that initiated the
- delayed enrollment and forwarding the response message

extraCerts

REQUIRED

- MUST contain certificates as described for the first
- response message of the respective profile

5.2. Revoking a certificate

This message sequence should be used by an entity to request the revocation of a certificate. Here the revocation request is used by an EE to revoke one of its own certificates. A (L)RA could also act as an EE to revoke one of its own certificates.

The revocation request message MUST be signed using the certificate that is to be revoked to prove the authorization to revoke to the PKI. The revocation request message is signature-protected using this certificate.

An EE requests the revocation of an own certificate at the CA that issued this certificate. The (L)RA/CA responds with a message that contains the status of the revocation from the CA.

Preconditions:

- 1 The certificate the EE wishes to revoke is not yet expired or revoked.

Message flow:

Step#	EE		(L)RA/CA
1	format rr		
2		-> rr	->
3			handle, re-protect or forward rr
4			receive rp
5		<- rp	<-
6	handle rp		

For this profile, the EE MUST include exactly one RevDetails structure in the rr. In case no error occurred the response to the rr MUST be an rp message. The (L)RA/CA MUST produce a rp containing a status field with a single set of values.

Detailed message description:

Revocation Request -- rr

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

-- The request of the EE to revoke its certificate

rr REQUIRED

-- MUST contain exactly one element of type RevDetails

-- If more revocations are desired, further requests MUST be

-- packaged in separate PKI Messages

certDetails REQUIRED

-- MUST be present and is of type CertTemplate

serialNumber REQUIRED

-- MUST contain the certificate serialNumber attribute of the

-- X.509 certificate to be revoked

issuer REQUIRED

-- MUST contain the issuer attribute of the X.509 certificate to

-- be revoked

crlEntryDetails REQUIRED

-- MUST contain exactly one reasonCode of type CRLReason (see

-- [\[RFC5280\] section 5.3.1](#))

-- If the reason for this revocation is not known or shall not be

-- published the reasonCode MUST be 0 = unspecified

protection REQUIRED

-- As described in [section 4.2](#) and the private key related to the

-- certificate to be revoked

extraCerts REQUIRED

-- As described in [section 4.3](#)

Revocation Response -- rp

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

- The responds of the (L)RA/CA to the request as appropriate

rp REQUIRED

status REQUIRED

- MUST contain exactly one element of type PKIStatusInfo

status REQUIRED

- positive value allowed: "accepted"
- negative value allowed: "rejection"

statusString OPTIONAL

- MAY be any human-readable text for debugging, logging or to
- display in a GUI

failInfo OPTIONAL

- MAY be present if and only if status is "rejection"

protection REQUIRED

- As described in [section 4.2](#)

extraCerts REQUIRED

5.3. Error reporting

This functionality should be used by an EE to report any error conditions upstream to the (L)RA/CA. Error reporting by the (L)RA downstream to the EE is described in [Section 6.3](#).

In case the error condition is related to specific details of an ip, cp, or kup response message and a confirmation is expected the error condition MUST be reported in the respective certConf message with negative contents.

General error conditions, e.g., problems with the message header, protection, or extraCerts, and negative feedback on rp, pollRep, or pkiConf messages MAY be reported in the form of an error message.

In both situations the error is reported in the PKIStatusInfo structure of the respective message.

The (L)RA/CA MUST respond to an error message with a pkiConf message, or with another error message if any part of the header is not valid. Both sides MUST treat this message as the end of the current transaction.

The PKIStatusInfo structure is used to report errors. The PKIStatusInfo structure SHOULD consist of the following fields:

- o status: Here the PKIStatus value rejection is the only one allowed.

- o statusString: Here any human-readable valid value for logging or to display in a GUI SHOULD be added.
- o failInfo: Here the PKIFailureInfo values MAY be used in the following way. For explanation of the reason behind a specific value, please refer to [\[RFC4210\] Appendix F](#).
- * transactionIdInUse: This is sent in case the received request contains a transaction ID that is already in use for another transaction. An EE receiving such error message SHOULD resend the request in a new transaction using a different transaction ID.
- * systemUnavail or systemFailure: This is sent in case a back-end system is not available or currently not functioning correctly. An EE receiving such error message SHOULD resend the request in a new transaction after some time.

Detailed error message description:

Error Message -- error

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

-- The message sent by the EE or the (L)RA/CA to indicate an
 -- error that occurred

error	REQUIRED
-------	----------

pKIStatusInfo	REQUIRED
---------------	----------

status	REQUIRED
--------	----------

-- MUST have the value "rejection"

statusString	RECOMMENDED
--------------	-------------

-- SHOULD be any human-readable text for debugging, logging
 -- or to display in a GUI

failInfo	OPTIONAL
----------	----------

-- MAY be present

protection	REQUIRED
------------	----------

-- As described in [section 4.2](#)

extraCerts	OPTIONAL
------------	----------

-- As described in [section 4.3](#)

5.4. Support messages

The following support messages offer on demand in-band transport of content that may be provided by the (L)RA/CA and relevant to the EE. The general messages and general response are used for this purpose. Depending on the environment, these requests are answered by the LRA, RA, or CA.

The general message and general response transport InfoTypeAndValue structures. In addition to those infoType values defined in CMP [\[RFC4210\]](#) further OIDs MAY be defined to define new PKI management operations, or general-purpose messages as needed in a specific environment.

Possible content described here address:

- o Request of CA certificates
- o Update of Root CA certificates
- o Parameters needed for a planned certificate request message
- o Voucher request and enrollment voucher exchange

5.4.1. General message and response

The general message transaction is similar to that given in CMP [Appendix E.5 \[RFC4210\]](#). In this section the general message (genm) and general response (genp) are described. The specific InfoTypeAndValue structures are described in the following sections.

The behavior in case an error occurs is described in [Section 5.3](#).

Message flow:

Step#	EE		(L)RA/CA
1	format genm		
2		-> genm	->
3			handle, re-protect or forward genm
4			format or receive genp
5		<- genp	<-
6	handle genp		

Detailed message description:

General Message -- genm

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

-- The request of the EE to receive information

genm REQUIRED

-- MUST contain exactly one element of type

-- InfoTypeAndValue

infoType REQUIRED

-- MUST be the OID identifying the specific scheme

-- described below

infoValue OPTIONAL

-- MUST be as described in the specific scheme described

-- below

protection REQUIRED

-- As described in [section 4.2](#)

extraCerts REQUIRED

-- As described in [section 4.3](#)

General Response -- genp

Field	Value
-------	-------

header

-- As described in [section 4.1](#)

body

-- The response of the (L)RA/CA to the information request

genp REQUIRED

-- MUST contain exactly one element of type

-- InfoTypeAndValue

infoType REQUIRED

-- MUST be the OID identifying the specific scheme

-- described below

infoValue OPTIONAL

-- MUST be as described in the specific scheme described

-- below

protection REQUIRED

-- As described in [section 4.2](#)

extraCerts REQUIRED

-- As described in [section 4.3](#)

5.4.2. Get CA certificates

This scheme can be used by an EE to request CA certificates from the (L)RA/CA.

An EE requests CA certificates from the (L)RA/CA by sending a general message with OID id-it-getCaCerts. The (L)RA/CA responds with a general response with the same OID that either contains a SEQUENCE of certificates populated with the available CA intermediate and issuing CA certificates or with no content in case no CA certificate is available.

< NOTE: The OID id-it-getCaCerts is not yet defined. It should be registered in the tree 1.3.6.1.5.5.7.4 (id-it) like other infoType OIDs, see CMP [Appendix F \[RFC4210\]](#) on page 92. >

The profile for this exchange is as given in [Section 5.4.1](#), with the following specific content:

- 1 the body MUST contain as infoType the OID id-it-getCaCerts
- 2 the infoValue of the request MUST be absent
- 3 if present, the infoValue of the response MUST be caCerts field

The infoValue field of the general response containing the id-it-getCaCerts OID looks like this:

infoValue	OPTIONAL
-- MUST be absent if no CA certificate is available	
-- MUST be present if CA certificates are available	
caCerts	REQUIRED
-- MUST be present if infoValue is present	
-- MUST be a sequence of CMPCertificate	

5.4.3. Get root CA certificate update

This scheme can be used by an EE to request an update of an existing root CA Certificate by the EE. It utilizes the CAKeyUpdAnnContent structure as described in CMP [Appendix E.4 \[RFC4210\]](#) as response to a respective general message.

An EE requests a root CA certificate update from the (L)RA/CA by sending a general message with OID id-it-caKeyUpdateInfo as infoType and no infoValue. The (L)RA/CA responds with a general response with the same OID that either contains the update of the root CA certificate consisting of up to three certificates, or with no content in case no update is available.

These three certificates are described in more detail in [section 4.4.1](#), [section 6.2](#), and [Appendix E.3 of \[RFC4210\]](#). The newWithNew certificate is the new root CA certificates and is REQUIRED to be present in the response message. The newWithOld certificate RECOMMENDED to be present in the response message though it is required for those cases where the receiving entity trusts the old root CA certificate and wishes to gain trust in the new root CA certificate. The oldWithNew certificate is OPTIONAL though it is only needed in a scenario where the requesting entity already trusts the new root CA certificate and wants to gain trust in the old root certificate.

The profile for this exchange is as given in [Section 5.4.1](#), with the following specific content:

- 1 the body MUST contain as infoType the OID id-it-caKeyUpdateInfo
- 2 the infoValue of the request MUST be absent
- 3 if present, the infoValue of the response MUST be a CAKeyUpdAnnContent structure

The infoValue field of the general response containing the id-it-caKeyUpdateInfo extension looks like this:

infoValue	OPTIONAL
-- MUST be absent if no update of the root CA certificate is available	
-- MUST be present if an update of the root CA certificate is available	
caKeyUpdateInfo	REQUIRED
-- MUST be present and be of type CAKeyUpdAnnContent	
oldWithNew	OPTIONAL
-- MUST be present if infoValue is present	
-- MUST contain an X.509 certificate containing the old public root CA key signed with the new private root CA key	
newWithOld	RECOMMENDED
-- MUST be present if infoValue is present	
-- MUST contain an X.509 certificate containing the new public root CA key signed with the old private root CA key	
newWithNew	REQUIRED
-- MUST be present if infoValue is present	
-- MUST contain the new root CA certificate	

5.4.4. Get certificate request parameters

This scheme can be used by an EE to request configuration parameters for a planned certificate request transaction.

An EE requests certificate request parameters from the (L)RA/CA by sending a general message with OID id-it-getCSRParam. The (L)RA/CA responds with a general response with the same OID that either contains the required fields, e.g., algorithm identifier for key pair generation or other attributes and extensions or with no content in case no specific requirements are made by the (L)RA/CA.

< NOTE: The OID id-it-getCSRParam is not yet defined. It should be registered in the tree 1.3.6.1.5.5.7.4 (id-it) like other infoType OIDs, see CMP [Appendix F \[RFC4210\]](#) on page 92. >

The EE SHOULD follow the requirements from the received CertTemplate and the optional RSA key length. In case a field is present but the value is absent, it means that this field is required but its content has to be provided by the EE.

< TBD: There is some more explanation needed to explain how to prefill the certTemplate structure. Possibly an example will help to clarify this. >

The profile for this exchange is as given in [Section 5.4.1](#), with the following specific content:

- 1 the body MUST contain as infoType the OID id-it-getCSRParam
- 2 the infoValue of the request MUST be absent
- 3 if present, the infoValue of the response MUST be a SEQUENCE of a certTemplate structure and an rsaKeyLen field of type INTEGER

The infoValue field of the general response containing the id-it-getCSRParam OID looks like this:

```
infoValue          OPTIONAL
-- MUST be absent if no requirements are available
-- MUST be present if the (L)RA/CA has any requirements on the
-- content of the certificates to be requested.
  certTemplate      REQUIRED
-- MUST be present if infoValue is present
-- MUST contain the prefilled certTemplate structure
    rsaKeyLen       OPTIONAL
-- This field is of type INTEGER. Any reasonable RSA key length
-- SHOULD be specified if the algorithm in the
-- subjectPublicKeyInfo field of the certTemplate is of type
-- rsaEncryption.
```

5.4.5. Get certificate management configuration

This scheme can be used by an EE to request the current certificate management configuration information by the EE in advance to a planned certificate management transaction, e.g., in case no out-of-band transport is available. Such certificate management configuration can consist of all information the EE needs to know to generate and deliver a proper certificate request, such as

- o algorithm, curve, and key length for key generation
- o various certificate attributes and extensions to be used for the certificate request
- o specific host name, port and path on the RA/LRA to send this CMP request to
- o Infrastructure Root CA Certificate, e.g., the root of the (L)RA TLS and CMP signer certificates.

There is an overlap with [Section 5.4.2](#) with regard to transport of CA certificates and with [Section 5.4.4](#) with regard to key generation parameter and certificate request attributes and extensions. This profile offers to request a proprietary configuration file containing all information needed in one exchange.

< TBD: Especially with [section 5.4.4](#) there is some overlap regarding algorithms, attributes and, extensions of the certificate that will be requested. It needs to be decided if both variants have a right to exist next to the other or if one option should be removed from this document. >

An EE requests certificate management configuration from the (L)RA/CA by sending a general message with the OID id-it-getCertMgtConfig. The (L)RA/CA responds with a general response with the same OID that either contains a certMgtConfig field containing the configuration file encoded as OCTET STRING or with no content in case no certificate management configuration is available.

< NOTE: The OID id-it-getCertMgtConfig is not yet defined. It should be registered in the tree 1.3.6.1.5.5.7.4 (id-it) like other infoType OIDs, see CMP [Appendix F \[RFC4210\]](#) on page 92. >

The EE SHOULD use the contents of this certMgtConfig to format and deliver the certificate request. The certificate management configuration may contain contact details, e.g., like an URI and issuing CA distinguished name, where to address the request messages to and may also contain certificate request parameters as described in [Section 5.4.4](#).

The certMgtConfig field may be of any format suitable for the EE, e.g., CMS [[RFC5652](#)], JWT [[RFC7519](#)] or, XML [[W3C XML](#)]. The certMgtConfig contents MAY be signed, e.g., like CMS SignedData [[RFC5652](#)], JWS [[RFC7515](#)] or, XML-DSig [[W3C XML-Dsig](#)]. For interoperability the format of the certMgtConfig field should be specified in detail if needed.

The profile for this exchange is as given in [Section 5.4.1](#), with the following specific content:

- 1 the body MUST contain as infoType the OID id-it-getCertMgtConfig
- 2 the infoValue of the request MUST be absent
- 3 if present, the infoValue of the response MUST be a certMgtConfig structure

The infoValue field of the general response containing the id-it-getCertMgtConfig extension looks like this:

infoValue	OPTIONAL
-- MUST be absent if no certificate management configuration	
-- is available	
-- MUST be present if the (L)RA/CA provides any certificate	
-- management configuration	
certMgtConfig	REQUIRED
-- MUST be present if infoValue is present	
-- MUST contain the certificate management configuration as OCTET	
-- OCTET STRING	

5.4.6. Get enrollment voucher

This scheme can be used by an EE to request an enrollment voucher containing the root certificate of a new, additional, or alternative PKI to establish trust in this PKI, e.g., in case no out-of-band transport is available. Such an enrollment voucher can be used in advance to an enrollment to this new environment. It may contain further information depending on the use case.

An EE requests an enrollment voucher from the (L)RA/CA by sending a general message. The (L)RA/CA responds with a general response with the same OID that either contains the voucher or with no content in case no voucher is available.

The (L)RA MAY use the content of the voucherRequest to get an enrollment voucher from other backend components, e.g., as described in BRSKI [[I-D.ietf-anima-bootstrapping-keyinfra](#)]. The EE SHOULD use the contents of the received enrollmentVoucher to authenticate the (L)RA/CA it is about to enroll to. The enrollment voucher may for example contain the Root CA certificate of the new PKI or the CMP signer certificate of the (L)RA. The general response message MUST be properly authenticated and the EE MUST verify the authorization of the sender to install new root certificates. One example for an enrollment voucher is specified in [RFC8366](#) [[RFC8366](#)].

The voucherRequest and enrollmentVoucher fields may be of any format suitable for the EE, e.g., CMS [[RFC5652](#)], JWT [[RFC7519](#)] or, XML [[W3C XML](#)]. The voucherRequest and enrollmentVoucher contents MAY contain a signature, e.g., CMS SignedData [[RFC5652](#)], JWS [[RFC7515](#)] or, XML-DSig [[W3C XML-Dsig](#)]. For interoperability the format of the voucherRequest and enrollmentVoucher field should be specified in detail if needed, e.g., as defined in BRSKI [[I-D.ietf-anima-bootstrapping-keyinfra](#)] and [RFC8366](#) [[RFC8366](#)].

< TBD: The content of the voucherRequest and enrollmentVoucher fields can also be limited to the specifications in BRSKI [[I-D.ietf-anima-bootstrapping-keyinfra](#)] and [RFC8366](#) [[RFC8366](#)]. >

The profile for this exchange is as given in [Section 5.4.1](#), with the following specific content:

- 1 the body MUST contain as infoType the OID id-it-getEnrollmentVoucher
- 2 if present, the infoValue of the request MUST be a voucherRequest structure

- 3 if present, the infoValue of the response MUST be an enrollmentVoucher structure

The infoValue field of the general message containing the id-it-getEnrollmentVoucher extension looks like this:

```
infoValue          OPTIONAL
-- MUST be absent if no voucher request is available
-- MUST be present if the EE provides the voucher request
  voucherRequest    REQUIRED
-- MUST be present if infoValue is present
-- MUST contain the voucher request as OCTET STRING
```

The infoValue field of the general response containing the id-it-getEnrollmentVoucher extension looks like this:

```
infoValue          OPTIONAL
-- MUST be absent if no enrollment voucher is available
-- MUST be present if the (L)RA/CA provides the enrollment
  voucher
  enrollmentVoucher  REQUIRED
-- MUST be present if infoValue is present
-- MUST contain the enrollment voucher as OCTET STRING
```

6. LRA and RA focused certificate management use cases

This chapter focuses on the communication of PKI backend components with each other. Depending on the network and PKI solution design, these will either be an LRA, RA or CA.

Typically, an (L)RA forwards messages from downstream, but it may also reply to them itself. Besides forwarding of received messages an (L)RA could also need to revoke certificates of EEs, report errors, or may need to manage its own certificates.

< TBD: In CMP Updates [[I-D.brockhaus-lamps-cmp-updates](#)] additional extended key usages like id-kp-cmpRA will be defined to indicate that a key pair is entitled to be used for signature-based protection of a CMP message by an (L)RA/CA. >

6.1. Forwarding of messages

Each CMP request message (i.e., ir, cr, p10cr, kur, pollReq, or certConf) or error message coming from an EE or the previous (downstream) PKI component MUST be sent to the next (upstream) PKI component. This PKI component MUST forward response messages to the next (downstream) PKI component or EE.

The (L)RA SHOULD verify the protection, the syntax, the required message fields, the message type, and if applicable the authorization and the proof-of-possession of the message. Additional checks or actions MAY be applied depending on the PKI solution requirements and concept. If one of these verification procedures fails, the (L)RA SHOULD respond with a negative response message and SHOULD not forward the message further upstream. General error conditions should be handled as described in [Section 5.3](#) and [Section 6.3](#).

An (L)RA SHOULD not change the received message if not necessary. The (L)RA SHOULD only update the message protection if it is technically necessary. Concrete PKI system specifications may define in more detail if and when to do so.

This is particularly relevant in the upstream communication of a request message.

Each hop in a chain of PKI components has one or more functionalities, e.g.,

- o An (L)RA may need to verify the identities of EEs or base authorization decisions for certification request processing on specific knowledge of the local setup, e.g., by consulting an inventory or asset management system.
- o An (L)RA may need to add fields to certificate request messages.
- o An (L)RA may need to store data from a message in a database for later usage or documentation purposes.
- o An (L)RA may provide traversal of a network boundary.
- o An (L)RA may need to double-check if the messages transferred back and forth are properly protected and well formed.
- o An (L)RA may provide a proof that it has performed all required checks.
- o An (L)RA may initiate a delayed enrollment due to offline upstream communication or registration officer interaction.
- o An (L)RA may grant the request of an EE to omit sending a confirmation message.
- o An RA can collect messages from different LRAs and forward them to the CA.

Therefore, the decision if a message should be forwarded

- o unchanged with the original protection,
- o unchanged with a new protection, or
- o changed with a new protection

depends on the PKI solution design and the associated security policy (CP/CPS [[RFC3647](#)]).

< TBD: In [CMP Updates] different circumstances that require adding of an additional protection by an (L)RA or batching CMP messages at an (L)RA by using the nested messages is described. It needs to be decided which of these variants should be specified here. Finally, I guess they will all be OPTIONAL. >

This section specifies the different options an (L)RA may implement and use.

An (L)RA MAY update the protection of a message

- o if the (L)RA performs changes to the header or the body of the message,
- o if the (L)RA needs to prove checks or validations performed on the message to one of the next (upstream) PKI components,
- o if the (L)RA needs to protect the message using a key and certificate from a different PKI, or
- o if the (L)RA needs to replace a MAC based-protection.

This is particularly relevant in the upstream communication of certificate request messages.

The message protection covers only the header and the body and not the extraCerts. The (L)RA MAY change the extraCerts in any of the following message adaptations, e.g., to sort or add needed or to delete needless certificates to support the next hop. This may be particularly helpful to extend upstream messages with additional certificates or to reduce the number of certificates in downstream messages when forwarding to constrained devices.

6.1.1.1. Not changing protection

This message adaptation can be used by any (L)RA to forward an original CMP message without changing the header, body or protection. In any of these cases the (L)RA acts more like a proxy, e.g., on a

network boundary, implementing no specific RA-like security functionality to the PKI.

This message adaptation MUST be used for forwarding kur messages that must not be approved by the respective (L)RA.

6.1.2. Replacing protection

The following two message adaptations can be used by any (L)RA to forward a CMP message with or without changes, but providing its own protection using its CMP signer key providing approval of this message. In this case the (L)RA acts as an actual Registration Authority (RA), which implements important security functionality of the PKI.

Before replacing the existing protection by a new protection, the (L)RA MUST verify the protection provided by the EE or by the previous PKI component and approve its content including any own modifications. For certificate requests the (L)RA MUST verify in particular the included proof-of-possession self-signature of the certTemplate using the public key of the requested certificate and MUST check that the EE, as authenticated by the message protection, is authorized to request a certificate with the subject as specified in the certTemplate.

In case the received message has been protected by a CA or another (L)RA, the current (L)RA MUST verify its protection and approve its content including any own modifications. For certificate requests the (L)RA MUST check that the other (L)RA, as authenticated by the message protection, is authorized to issue or forward the request.

These message adaptations MUST NOT be applied to kur request messages as described in [Section 5.1.3](#) since their original protection using the key and certificate to be updated needs to be preserved, unless the regCtrl OldCertId is used to clearly identify the certificate to be updated.

6.1.2.1. Keeping proof-of-possession

This message adaptation can be used by any (L)RA to forward a CMP message with or without modifying the message header or body while preserving any included proof-of-possession.

By replacing the existing protection using its own CMP signer key the (L)RA provides a proof of verifying and approving of the message as described above.

In case the (L)RA modifies the certTemplate of an ir or cr message, the message adaptation in [Section 6.1.2.2](#) needs to be applied instead.

[6.1.2.2](#). Breaking proof-of-possession

This message adaptation can be used by any (L)RA to forward an ir or cr message with modifications of the certTemplate i.e., modification, addition, or removal of fields. Such changes will break the proof-of-possession provided by the EE in the original message.

By replacing the existing or applying an initial protection using its own CMP signer key the (L)RA provides a proof of verifying and approving the new message as described above.

In addition to the above the (L)RA MUST verify in particular the proof-of-possession contained in the original message as described above. If these checks were successfully performed the (L)RA MUST change the popo to raVerified.

The popo field MUST contain the raVerified choice in the certReq structure of the modified message as follows:

```
popo
  raVerified          REQUIRED
  -- MUST have the value NULL and indicates that the (L)RA
  -- verified the popo of the original message.
```

[6.1.3](#). Adding Protection

< TBD: In [CMP Updates] different circumstances that require adding of an additional protection by an (L)RA or batching CMP messages at an (L)RA by using the nested messages is described. It needs to be decided which of these variants should be specified here. Finally, I guess they will all be OPTIONAL. >

[6.1.4](#). Initiating delayed enrollment

This message adaptation can be used by an (L)RA to initiate delayed enrollment. In this case a (L)RA/CA MUST add the status waiting in the response message. The (L)RA/CA MUST then reply to the pollReq messages as described in [Section 5.1.7](#).

[6.2](#). Revoking certificates on behalf of another's entities

This message sequence can be used by an (L)RA to revoke a certificate of any other entity. This revocation request message MUST be signed

by the (L)RA using its own CMP signer key to prove to the PKI authorization to revoke the certificate on behalf of the EE.

The general message flow for this profile is the same as given in section [Section 5.2](#).

Preconditions:

- 1 the certificate to be revoked MUST be known to the (L)RA
- 2 the (L)RA MUST have the authorization to revoke the certificates of other entities issued by the corresponding CA

The profile for this exchange is identical to that given in section [Section 5.2](#), with the following changes:

- 1 it is not required that the certificate to be revoked is not yet expired or revoked
- 2 the (L)RA acts as EE for this message exchange
- 3 the rr messages MUST be signed using the CMP signer key of the (L)RA.

[6.3](#). Error reporting

This functionality should be used by the (L)RA to report any error conditions downstream to the EE. Potential error reporting by the EE upstream to the (L)RA/CA is described in [Section 5.3](#).

In case the error condition is related to specific details of an ir, cr, p10cr, or kur request message it MUST be reported in the specific response message, i.e., an ip, cp, or kup with negative contents.

General error conditions, e.g., problems with the message header, protection, or extraCerts, and negative feedback on rr, pollReq, certConf, or error messages MUST be reported in the form of an error message.

In both situations the (L)RA reports the errors in the PKIStatusInfo structure of the respective message as described in [Section 5.3](#).

An EE receiving any such negative feedback SHOULD log the error appropriately and MUST terminate the current transaction.

7. CMP message transport variants

The CMP messages are designed to be self-contained, such that in principle any transport can be used. HTTP SHOULD be used for online transport while file-based transport MAY be used in case offline transport is required. In case HTTP transport is not desired or possible, CMP messages MAY also be piggybacked on any other reliable transport protocol, e.g., CoAP [[RFC7252](#)].

Independently of the means of transport it could happen that messages are lost, or a communication partner does not respond. In order to prevent waiting indefinitely, each CMP client component SHOULD use a configurable per-request timeout, and each CMP server component SHOULD use a configurable per-response timeout in case a further message is to be expected from the client side. In this way a hanging transaction can be closed cleanly with an error and related resources (for instance, any cached extraCerts) can be freed.

7.1. HTTP transport

This transport mechanism can be used by an EE and (L)RA/CA to transfer CMP messages over HTTP. If HTTP transport is used the specifications as described in [[RFC6712](#)] MUST be followed.

Each PKI management entity supporting HTTP(S) transport MUST support the use of the path-prefix of '/.well-known/' as defined in [[RFC5785](#)] and the registered name of 'cmp' to ease interworking in a multi-vendor environment.

The CMP client MUST be configured with sufficient information to form the CMP server URI. This MUST be at least the authority portion of the URI, e.g., 'www.example.com:80', or the full operational path of the CA/RA. An additional arbitrary label, e.g., 'arbitraryLabel1', MAY be configured as a separate component or as part of the full operational path to provide further information to address multiple CAs or certificate profiles. A valid full operational path can look like this:

- 1 http://www.example.com/.well-known/cmp
- 2 http://www.example.com/.well-known/cmp/keyupdate
- 3 http://www.example.com/.well-known/cmp/arbitraryLabel1
- 4 http://www.example.com/.well-known/cmp/arbitraryLabel1/keyupdate

PKI management operations SHOULD use the following URI path:

PKI management operation	Path	Details
Enroll client to new PKI (REQUIRED)	/initialization	Section 5.1.1
Enroll client to existing PKI (OPTIONAL)	/certification	Section 5.1.2
Update client certificate (REQUIRED)	/keyupdate	Section 5.1.3
Enroll client using PKCS#10 (OPTIONAL)	/p10	Section 5.1.5
Enroll client using central key generation (OPTIONAL)	/serverkeygen	Section 5.1.6
Revoke client certificate (RECOMMENDED)	/revocation	Section 5.2
Get CA certificates (OPTIONAL)	/getCAcert	Section 5.4.2
Get root CA certificate update (OPTIONAL)	/getRootCAcertUpdate	Section 5.4.3
Get certificate request parameters (OPTIONAL)	/getCSRparam	Section 5.4.4
Get certificate management configuration (OPTIONAL)	/getCertMgtConfig	Section 5.4.5
Get enrollment voucher (OPTIONAL)	/getVoucher	Section 5.4.6

Table 1: HTTP endpoints

Subsequent certConf, error, and pollReq messages are sent to the URI of the respective PKI management operation.

< TBD: It needs to be defined if specific path values for communication between PKI management entities as specified in [section 6](#) are needed, e.g., 'forward' or 'nested'.>

7.2. HTTPS transport using certificates

This transport mechanism can be used by an EE and (L)RA/CA to further protect the HTTP transport as described in [Section 7.1](#) using TLS 1.2 [[RFC5246](#)] or TLS 1.3 [[RFC8446](#)] as described in [[RFC2818](#)] with certificate-based authentication. Using this transport mechanism, the CMP transport via HTTPS MUST use TLS server authentication and SHOULD use TLS client authentication.

EE:

- o The EE SHOULD use a TLS client certificate as far as available. If no dedicated TLS certificate is available the EE SHOULD use an already existing certificate identifying the EE (e.g., a manufacturer certificate).
- o If no TLS certificate is available at the EE, server-only authenticated TLS SHOULD be used.
- o The EE MUST validate the TLS server certificate of its communication partner.

(L)RA:

- o Each (L)RA SHOULD use a TLS client certificate on its upstream (client) interface.
- o Each (L)RA SHOULD use a TLS server certificate on its downstream (server) interface.
- o Each (L)RA MUST validate the TLS certificate of its communication partner.

NOTE: The requirements for checking certificates given in [[RFC5280](#)], [[RFC5246](#)] and [[RFC8446](#)] MUST be followed for the TLS layer. Certificate status checking SHOULD be used for the TLS certificates of communication partners.

7.3. HTTPS transport using shared secrets

This transport mechanism can be used by an EE and (L)RA/CA to further protect the HTTP transport as described in [Section 7.1](#) using TLS 1.2 [[RFC5246](#)] or TLS 1.3 [[RFC8446](#)] as described in [[RFC2818](#)] with mutual authentication based on shared secrets as described in [[RFC5054](#)].

EE:

- o The EE MUST use the shared symmetric key for authentication.

(L)RA:

- o The (L)RA MUST use the shared symmetric key for authentication.

7.4. File-based transport

For offline transfer file-based transport MAY be used. Offline transport is typically used between LRA and RA nodes.

Connection and error handling mechanisms like those specified for HTTP in [[RFC6712](#)] need to be implemented.

< TBD: Details need to be defined later >

7.5. CoAP transport

In constrained environments where no HTTP transport is desired or possible, CoAP [[RFC7252](#)] MAY be used instead. Connection and error handling mechanisms like those specified for HTTP in [[RFC6712](#)] may need to be implemented.

Such specification is out of scope of this document and would need to be specifies in a separate document.

7.6. Piggybacking on other reliable transport

For online transfer where no HTTP transport is desired or possible CMP messages MAY also be transported on some other reliable protocol. Connection and error handling mechanisms like those specified for HTTP in [[RFC6712](#)] need to be implemented.

Such specification is out of scope of this document and would need to be specifies in a separate document, e.g., in the scope of the respective transport protocol used.

8. IANA Considerations

<Add any IANA considerations>

9. Security Considerations

<Add any security considerations>

10. Acknowledgements

We would like to thank the various reviewers of this CMP profile.

11. References

11.1. Normative References

- [I-D.brockhaus-lamps-cmp-updates]
Brockhaus, H., "CMP Updates", [draft-brockhaus-lamps-cmp-updates-03](#) (work in progress), January 2020.
- [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>>.
- [RFC2986] 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/info/rfc2986>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", [BCP 106](#), [RFC 4086](#), DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", [RFC 4210](#), DOI 10.17487/RFC4210, September 2005, <<https://www.rfc-editor.org/info/rfc4210>>.
- [RFC4211] 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/info/rfc4211>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, [RFC 5652](#), DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", [RFC 5785](#), DOI 10.17487/RFC5785, April 2010, <<https://www.rfc-editor.org/info/rfc5785>>.

- [RFC6712] Kause, T. and M. Peylo, "Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)", [RFC 6712](#), DOI 10.17487/RFC6712, September 2012, <<https://www.rfc-editor.org/info/rfc6712>>.

11.2. Informative References

- [ETSI-3GPP]
3GPP, "TS33.310; Network Domain Security (NDS); Authentication Framework (AF); Release 16; V16.1.0", December 2018, <http://www.3gpp.org/ftp/Specs/archive/33_series/33.310/>.
- [I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", [draft-ietf-anima-bootstrapping-keyinfra-35](#) (work in progress), February 2020.
- [IEC62443-3-3]
IEC, "Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels", IEC 62443-3-3, August 2013, <<https://webstore.iec.ch/publication/7033>>.
- [IEEE802.1AR]
IEEE, "802.1AR Secure Device Identifier", June 2018, <<http://standards.ieee.org/findstds/standard/802.1AR-2009.html>>.
- [NIST-CSFW]
NIST, "Framework for Improving Critical Infrastructure Cybersecurity Version 1.1", April 2018, <<https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), DOI 10.17487/RFC2818, May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", [RFC 3647](#), DOI 10.17487/RFC3647, November 2003, <<https://www.rfc-editor.org/info/rfc3647>>.

- [RFC5054] Taylor, D., Wu, T., Mavrogiannopoulos, N., and T. Perrin, "Using the Secure Remote Password (SRP) Protocol for TLS Authentication", [RFC 5054](#), DOI 10.17487/RFC5054, November 2007, <<https://www.rfc-editor.org/info/rfc5054>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.
- [RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", [RFC 7515](#), DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/info/rfc7515>>.
- [RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", [RFC 7519](#), DOI 10.17487/RFC7519, May 2015, <<https://www.rfc-editor.org/info/rfc7519>>.
- [RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", [RFC 8366](#), DOI 10.17487/RFC8366, May 2018, <<https://www.rfc-editor.org/info/rfc8366>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [UNISIG] UNISIG, "Subset-137; ERTMS/ETCS On-line Key Management FFFIS; V1.0.0", December 2015, <https://www.era.europa.eu/filebrowser/download/542_en>.
- [W3C_XML] W3C, "Extensible Markup Language (XML) 1.0", W3C XML, November 2008, <<https://www.w3.org/TR/xml/>>.
- [W3C_XML-Dsig] W3C, "XML Signature Syntax and Processing Version 2.0", W3C XML-DSIG, July 2015, <<https://www.w3.org/TR/xmlldsig-core2/>>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Hendrik Brockhaus
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

Email: hendrik.brockhaus@siemens.com
URI: <http://www.siemens.com/>

Steffen Fries
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

Email: steffen.fries@siemens.com
URI: <http://www.siemens.com/>

David von Oheimb
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

Email: david.von.oheimb@siemens.com
URI: <http://www.siemens.com/>

