Internet Engineering Task Force
Internet-Draft
Updates: 4210 (if approved)
Intended status: Standards Track
Expires: January 8, 2020

H. Brockhaus
S. Fries
D. von Oheimb
Siemens

July 7, 2019

Lightweight CMP Profile
draft-brockhaus-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. 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 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 January 8, 2020.

Table of Contents

1. History of changes

From version 00 -> 01:

- Change focus from industrial to more multi-purpose use cases and lightweight CMP profile.

- Incorporate the omitted confirmation into the header specified in section Section 4.1 and described in the standard enrollment use case in section Section 5.1.1 due to discussion with Tomas Gustavsson.

- Change from OPTIONAL to RECOMMENDED for use case 'Revoke another's entities certificate' in section Section 6.2 and , because it is regarded as important functionality in many environments to enable the management station to revoke EE certificates.

- Complete the specification of the revocation message flow in section Section 5.2 and Section 6.2.

- 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 certificate management transactions implementing machine-to-machine and IoT use cases. The focus lies on maximum automation and interoperable implementation of all involved components 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], and HTTP transfer for CMP [RFC6712]. 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 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 UNISG 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. 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 not address the specific needs for certificate management automation for unattended machine or application-oriented end entities.

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:

- signature-based protection is the default protection; initial transactions may also use HMAC,
- certification of a second key pair within the same transaction is not supported,
- proof-of-possession (POPO) with self-signature of the certTemplate according to [RFC4211] section 4.1 clause 3 is the only supported POPO method,
- confirmation of newly enrolled certificates may be omitted, and
- 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:

- protection of initial transactions may be HMAC-based,

- the subject name is mandatory in certificate templates, and

- 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:

- as of RFC 4210 [RFC4210] the messageTime is required to be Greenwich Mean Time coded as generalizedTime (Note: While UNISIG explicetely states that the messageTime in 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. Therfore UNISG may be in conflict with RFC 4210 [RFC4210]. Both time formates are described in RFC 5280 [RFC5280] section 4.1.2.5.), and

- 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 constrained devices. On the side of the central PKI components the profile accepts higher resource needs.

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 chapter 3, 4, and 5 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 SHOULD ignore it.

2.6. Structure of this document

Chapter 2 introduces the general PKI architecture and approach to certificate management using CMP that is assumed in this document. Then it enlists the certificate management use cases 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.

Chapter 3 profiles the CMP message header, protection, and extraCerts section as they are general elements of CMP messages.

Chapter 4 profiles the exchange of CMP messages between an EE and the first PKI component. There are various flavors of certificate enrollment requests optionally with polling, revocation, error handling, and general support transactions.

Chapter 5 profiles the exchange between further PKI components. 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.

Chapter 6 outlines different mechanisms for CMP message transfer, namely http-based transfer as already specified in [RFC6712], using an additional TLS layer, offline file-based transport, CoAP [RFC7252], or piggybacking CMP messages on other protocols.

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 end entities.

EE: End entity, a user or device or service that holds a PKI certificate. An identifier for the EE is given as the subject of the 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 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, as asynchronous communication is not performed in a timely consistent manner, e.g., because of a delayed message delivery.

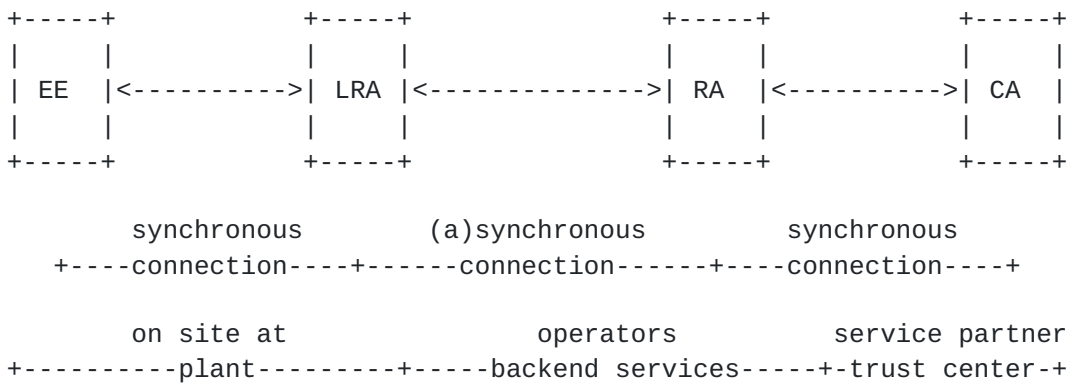

Figure 1: Certificate management on site

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 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.

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.

- Header of a CMP message; see Section 4.1.
- Protection of a CMP message; see Section 4.2.
- 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, if a 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 constrained devices to the most crucial types of transactions.

Section 5 - End Entity focused certificate management use cases

- Request a certificate from a new PKI with signature protection; see Section 5.1.1.
- Request to update an existing certificate with signature protection; see Section 5.1.2.
- Error reporting; see Section 5.3.

Section 6 - LRA and RA focused certificate management use cases

- Forward messages without changes; see Section 6.1.1.
- Forward messages with replaced protection and raVerified as proof-of-possession; see Section 6.1.2.2.
- 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

- Request a certificate from a PKI with MAC protection; see Section 5.1.3.
- Handle delayed enrollment due to asynchronous message delivery.

< Motivation see Section 5.1.6, specification TBD >

- Revoke an own certificate.

Section 6 - LRA and RA focused certificate management use cases

- 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

- Request a certificate from a legacy PKI using a PKCS#10 [RFC2986] request.

< Motivation see Section 5.1.4, specification TBD >

- Add central generation of a key pair to a certificate request.

< Motivation see Section 5.1.5, specification TBD >

- Additional support messages, e.g., to update a Root CA certificate or to request an RFC 8366 [RFC8366] voucher.

< Motivation see Section 5.4, specification TBD >

Section 6 - LRA and RA focused certificate management use cases

- Initiate delayed enrollment due to asynchronous message delivery.

< Motivation see Section 6.1.3, specification TBD >

3.4. CMP message transport

Recommended transport

- Transfer CMP messages using HTTP; see Section 7.1.

Optional transport

- Transfer CMP messages using HTTPS with certificate-based authentication; see Section 7.2.
- Transfer CMP messages using HTTPS with shared-secret based protection; see Section 7.3.
- File-based CMP message transport.

< Motivation see Section 7.4, specification TBD >

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:

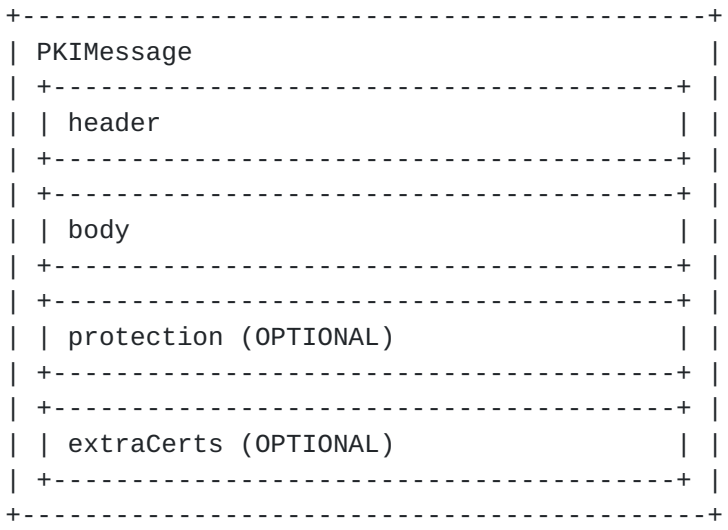

Figure 2: CMP message structure

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 signing certificate used for
    -- protection of this message
  recipient                 REQUIRED
    -- MUST be the name of the intended 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 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
    parameters              PROHIBITED
    -- MUST be absent
  senderKID                 RECOMMENDED
    -- MUST be the SubjectKeyIdentifier, if available, of the
    -- certificate used for protecting this message
  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.

```
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. OCSP or CRLs SHOULD be used for status checking of 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 signing certificate together with its
    -- chain, if needed
    -- 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. 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:

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

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:

- Using a certificate from a trusted PKI and the corresponding private key, e.g., a manufacturer certificate
- Using the certificate to be updated and the corresponding private key
- 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. (< The functional extension for central key generation is TBD if needed. >) 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 a voucher [RFC8366] for enrolment, or 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 [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 sequenceis 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 3.1

body
    -- The request of the EE for a new certificate
  ir                           REQUIRED
    -- MUST be exactly one CertReqMsg
    -- If more certificates are required, further requests MUST be
    -- packaged in separate PKI Messages
    certReq                    REQUIRED
      certReqId                REQUIRED
    -- MUST be set to 0
      certTemplate             REQUIRED
        version                OPTIONAL
    -- MUST be 2 if supplied.
        subject                REQUIRED
    -- MUST contain the suggested subject name of the EE
    -- certificate
        publicKey              REQUIRED
    -- 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           REQUIRED
        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 3.2
```

```
   extraCerts                    REQUIRED
       -- As described in section 3.3


   Certification Response -- ip

   Field                         Value

   header
       -- As described in section 3.1

   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

   protection                    REQUIRED
       -- As described in section 3.2
```

```
   extraCerts                    REQUIRED
       -- As described in section 3.3
       -- MUST contain the chain of the issued certificate
       -- Duplicate certificates MAY be omitted


   Certificate Confirmation -- certConf

   Field                         Value

   header
       -- As described in section 3.1

   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 3.2
       -- 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 3.1	
body	
pkiConf	REQUIRED
-- The content of this field MUST be NULL	
protection	REQUIRED
-- As described in section 3.2	
-- 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. 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.

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
```

5.1.3. 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 too.

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 PBMParameter sequence. The fields of PBMParameter SHOULD remain constant throughout this certificate management transaction to reduce the computational overhead.

```
      PBMParameter              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
```

5.1.4. 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 5.1.3.

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 can extract the PKCS#10 certificate request from the p10cr and provide it separately to the CA.

< Details need to be defined later >

5.1.5. Generate the key pair centrally at the (L)RA/CA

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 has the private key.

There are some rare 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 anyway 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.

As the protection of centrally generated keys in the response message is being extended from EncryptedValue to EncryptedKey by CMP Updates [brockhaus-lamps-cmp-updates] also the alternative EnvelopedData can be used. As EncryptedValue offers only key transport, e.g. using RSA or symmetic encryption, EnvelopedData offers further key management techniques, e.g. key agreement, and therefore more crypto agility.

Note that according to RFC 4211 [RFC4211] section 2.1.9 the use of the EncryptedValue structure has been deprecated in favor of the EnvelopedData structure.

< Details need to be defined later >

5.1.6. Delayed enrollment

This functional extension can be applied in combination with certificate enrollment as described in Section 5.1.1 to Section 5.1.4. 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 registration officer interaction. Depending on the PKI architecture, it is not necessarily the PKI component directly communicating with the EE that initiates the delayed enrollment. In this case this PKI component MUST include the status waiting in the response and this response MUST not contain a newly issued certificate. When receiving a response with status waiting the EE MUST send a poll request to the (L)RA/CA. The (L)RA/CA 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.

< Details need to be defined later >

5.1.7. Omitted confirmation

This section will be removed though the functionality was incorporated into the header specified in section Section 4.1 and described in the standard enrollment use case in section Section 5.1.1 due to discussion with Tomas Gustavsson.

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 3.1

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
    -- [RFC 5280] 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 3.2 and the private key related to the
    -- certificate to be revoked

extraCerts                     REQUIRED
    -- As described in section 3.3


Revocation Response -- rp

Field                          Value

header
    -- As described in section 3.1
```

```
 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 3.2

 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:

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

- statusString: Here any human-readable valid value for logging or to display in a GUI SHOULD be added.

- 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 3.1

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 3.2

extraCerts                    OPTIONAL
    -- As described in section 3.3
```

5.4. Support messages

The following support messages offer on demand in-band transport of content that may be relevant to the EE. The general request messages and general response messages are used for this purpose.

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 certificate management transactions, or general-purpose messages as needed in a specific environment.

Possible content described here address:

- Update of Root CA certificates
- Parameters needed for a planned certificate request message <TBD>
- Request an enrollment voucher

< Details need to be defined later >

5.4.1. Root CA certificate update

This message sequence can be used by an EE to request an update of a Root CA Certificate by the EE. It utilizes the root CA key update announcement message as described in [RFC4210] Appendix E.4 as response to a respective general request message.

An EE requests a root CA certificate update from the (L)RA/CA by sending a general message with OID id-it-caKeyUpdateInfo. 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 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].

< Details need to be defined later >

5.4.2. Get enrollment voucher

This message sequence can be used by an EE to request an enrollment voucher containing the root certificate of a new 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 contains the voucher.

< Details need to be defined later >

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.

< In CMP Updates [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.:

- 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.

- An (L)RA may need to add fields to certificate request messages.

- An (L)RA may need to store data from a message in a database for later usage or documentation purposes.

- An (L)RA may provide traversal of a network boundary.

- An (L)RA may need to double-check if the messages transferred back and forth are properly protected and well formed.

- An RA can collect messages from different LRAs and forward them to the CA.

- An (L)RA may provide a proof that it has performed all required checks.

- An (L)RA may initiate a delayed enrollment due to offline upstream communication or registration officer interaction.

- An (L)RA may grant the request of an EE to omit sending a confirmation message.

Therefore, the decision if a message should be forwarded

- unchanged with the original protection,

- unchanged with a new protection, or

- changed with a new protection

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

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

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

- if the (L)RA performs changes to the header or the body of the message,

- if the (L)RA needs to prove checks or validations performed on the message to one of the next (upstream) PKI components,

- if the (L)RA needs to protect the message using a key and certificate from a different PKI, or

- 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. 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 5.1.2 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 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. 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 5.1.6.

6.1.4. Granting omitted confirmation

This section will be removed though the functionality was incorporated into the standard enrollment use case in section Section 5.1.1 due to discussion with Tomas Gustavsson.

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.

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:

- 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).

- If no TLS certificate is available at the EE, server-only authenticated TLS SHOULD be used.

- The EE MUST validate the TLS server certificate of its communication partner.

(L)RA:

- Each (L)RA SHOULD use a TLS client certificate on its upstream (client) interface.

- Each (L)RA SHOULD use a TLS server certificate on its downstream (server) interface.

- 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. OCSP or CRLs SHOULD be used for status checking of 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:

- 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.

< 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

[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>.

[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>.

[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

[brockhaus-lamps-cmp-updates] Brockhaus, H., "CMP Updates (work in progress)", July 2019, <https://datatracker.ietf.org/doc/draft-brockhaus-lamps-cmp-updates/>.

[ETSI-3GPP]
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/>.

[IEC62443-3-3]
International Electrotechnical Commission, "IEC 62443 Part 3-3 - System security requirements and security levels", IEC 62443-3-3, August 2013, <Informative References>.

[IEEE802.1AR]
IEEE, "IEEE 802.1AR Secure Device Identifier", 06 2018, <http://standards.ieee.org/findstds/standard/802.1AR-2009.html>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[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>.

[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>.

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

[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>.

[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, "UNISIG subset-137; ERTMS/ETCS On-line Key Management FFFIS; V1.0.0", December 2015, <https://www.era.europa.eu/filebrowser/download/542_en>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Hendrik Brockhaus
Siemens AG
Otto-Hahn-Rin 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/