

LAMPS Working Group
Internet-Draft
Updates: [4210](#) (if approved)
Intended status: Standards Track
Expires: August 19, 2020

H. Brockhaus
Siemens
February 16, 2020

CMP Updates
draft-ietf-lamps-cmp-updates-00

Abstract

This document contains a set of updates to the base syntax of Certificate Management Protocol (CMP) version 2. This document updates [RFC 4210](#).

Specifically, the CMP services updated in this document comprise the enabling of using EnvelopedData instead of EncryptedValue and the definition of extended key usages to identify certificates of CMP endpoints on certification and registration authorities.

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 19, 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	2
2.	Introduction	3
2.1.	Convention and Terminology	3
3.	Updates to RFC 4210 - Certificate Management Protocol (CMP) .	4
3.1.	New Section 1.1. - Changes since RFC 4210	4
3.2.	New Section 4.5 - Extended Key Usage	5
3.3.	Replace Section 5.1.3.4 - Multiple Protection	6
3.4.	Replace Section 5.2.2. - Encrypted Values	7
3.5.	Update Section 5.3.4. - Certification Response	9
3.6.	Replace Section 5.3.19.9. - Revocation Passphrase	9
3.7.	Update Section 5.3.22 - Polling Request and Response . . .	10
3.8.	Update Appendix B - The Use of Revocation Passphrase . . .	11
3.9.	Update Appendix C - Request Message Behavioral Clarifications	12
3.10.	Update Appendix D.4. - Initial Registration/Certification (Basic Authenticated Scheme)	12
4.	IANA Considerations	12
5.	Security Considerations	13
6.	Acknowledgements	13
7.	References	13
7.1.	Normative References	13
7.2.	Informative References	14
Appendix A.	ASN.1 Modules	14
	Author's Address	15

1. History of changes

From [draft-brockhaus-lamps-cmp-updates-03](#) -> [draft-brockhaus-lamps-cmp-updates-03](#):

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

From version 02 -> 03:

- o Added some clarification in [Section 3.1](#)

From version 01 -> 02:

- o Added clarification to section on multiple protection

- o Added clarification on new EKUs after some exchange with Tomas Gustavsson
- o Reused OIDs from [RFC 6402](#) [[RFC6402](#)] as suggested by Sean Turner at IETF 106
- o Added clarification on the field containing the key identifier for a revocation passphrase
- o Minor changes in wording

From version 00 -> 01:

- o Added a section describing the new extended key usages
- o Completed the section on changes to the specification of encrypted values
- o Added a section on clarification to [Appendix D.4](#)
- o Minor generalization in [RFC 4210](#) [[RFC4210](#)] Sections [5.1.3.4](#) and 5.3.22
- o Minor changes in wording

2. Introduction

While using CMP [[RFC4210](#)] in industrial and IoT environments and developing the Lightweight CMP Profile [[I-D.brockhaus-lamps-lightweight-cmp-profile](#)] some limitations were identified in the original CMP specification. This document updates [RFC 4210](#) [[RFC4210](#)] to overcome these limitations.

In general, this document aims to improve the crypto agility of CMP to be flexible to react on future advances in cryptography.

This document also introduces new extended key usages to identify CMP endpoints on registration and certification authorities.

2.1. 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](#)], and [RFC 5280](#) [[RFC5280](#)]. 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.

KGA: Key generation authority, which generates key pairs on behalf of an EE. The KGA could be co-located with an RA or a CA.

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

3. Updates to [RFC 4210](#) - Certificate Management Protocol (CMP)

3.1. New [Section 1.1](#). - Changes since [RFC 4210](#)

The following subsections describe feature updates to [RFC 4210](#) [[RFC4210](#)]. They are always related to the base specification. Hence references to the original sections in [RFC 4210](#) [[RFC4210](#)] are used whenever possible.

Insert this section at the end of the current [Section 1](#).

The following updates are made in [draft-brockhaus-lamps-cmp-updates](#):

- o Add new extended key usages for different CMP server types, e.g. registration authority and certification authority, to express the authorization of the entity identified in the certificate containing the respective extended key usage extension to act as the indicated PKI management component.
- o Extend the description of multiple protection to cover additional use cases, e.g., batch processing of messages.
- o Offering EnvelopedData as another choice next to EncryptedValue to extend crypto agility in CMP. 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. [RFC 4211](#) [[RFC4211](#)] offers the EncryptedKey structure, a choice of EncryptedValue and EnvelopedData for migration to EnvelopedData. For reasons of completeness and consistency the exchange of EncryptedValue is performed for all usages in [RFC 4210](#) [[RFC4210](#)].

This includes the protection of centrally generated private keys, encryption of certificates, and revocation passphrases.

- o Extend the usage of polling also to p10cr messages.

3.2. New [Section 4.5](#) - Extended Key Usage

Insert this section.

The Extended Key Usage (EKU) extension indicates the purposes for which the certified public key may be used. It therefore restricts the use of a certificate to specific applications.

A CA may want to delegate parts of their duties to other PKI management entities. The mechanism to prove this delegation explained in this section offers zero-touch means to check the authorization of such delegation. Such delegation could also be expressed by other means, e.g., explicit configuration.

To offer automatic validation means for the delegation of a role by a CA, the certificates used by PKI management entities for CMP message protection or signed data for central key generation **MUST** be issued by the delegating CA and **MUST** contain the respective EKUs. This proves the authorization of this entity by the delegating CA to act as the PKI management entity as described below.

The ASN.1 to define these EKUs is:

```
id-kp-cmpCA OBJECT IDENTIFIER ::= { id-kp 27 }
id-kp-cmpRA OBJECT IDENTIFIER ::= { id-kp 28 }
id-kp-cmpKGA OBJECT IDENTIFIER ::= { id-kp ... }
```

< TBD: id-kp-cmpKGA to be defined. >

Note: [RFC 6402 section 2.10](#) [[RFC6402](#)] specifies OIDs for a CMC CA and a CMC RA. As the functionality of a CA and RA is not specific to whether use CMC or CMP as certificate management protocol, the same OIDs **SHALL** be used for a cmpCA and a cmpRA.

< TBD: It needs to be clarified, if the Name and Description of the OIDs can be adapted or extended to avoid confusion as they currently only refer to CMC endpoints. >

The description of the PKI management entity for each of the EKUs is as follows:

CMP Certification Authorities are CMP endpoints on CA equipment as described in [section 3.1.1.2](#). The key used in the context of CMP

management operations, especially CMP message protection, need not be the same key that signs the certificates. It is necessary, however, to ensure that the entity acting as cmpCA is authorized to do so. Therefore, the cmpCA MUST do one of the following,

- o use the CA private key on the CMP endpoint, or
- o explicitly designate this authority to another entity.

For automatic validation of such delegation it MUST be indicated by the id-kp-cmpCA extended key usage. This extended key usage MUST be placed into the certificate used on the CA equipment and the CA that delegates this role MUST issue the cmpCA certificate.

Note: Using a separate key pair for protecting CMP management operations at the CA decreases the number of operations of the private key used to sign certificates.

CMP Registration Authorities are CMP endpoints on RA equipment as described in [section 3.1.1.3](#). A cmpRA is identified by the id-kp-cmpRA extended key usage. This extended key usage is placed into RA certificates. The CA that delegated this role is identified by the CA that issued the cmpRA certificate.

CMP Key Generation Authorities are identified by the id-kp-cmpKGA extended key usage. Though the cmpKGA knows the private key it generated on behalf of the end entity, this is a very sensible service and needs specific authorization. This authorization is either with the CA certificate itself, or indicated by placing the id-kp-cmpKGA extended key usage into the cmpRA or cmpCA certificate used to authenticate the origin of the private key to express the authorization to offer this service.

Note: In device PKIs, especially those issuing IDevID certificates, CA may have very long validity (including the GeneralizedTime value 99991231235959Z to indicate a not well-defined expiration date as specified in IEEE 802.1AR [section 8.5](#) [IEEE802.1AR] and [RFC 5280 Section 4.1.2.5](#) [RFC5280]). Such validity periods SHOULD NOT be used for protection of CMP messages. Certificates for delegated CMP message protection (cmpCA, cmpRA, cmpKGA) MUST NOT use indefinite expiration date.

3.3. Replace [Section 5.1.3.4](#) - Multiple Protection

[Section 5.1.3.4 of RFC 4210](#) [RFC4210] describes the nested message. This document opens the usage of nested messages also for batch transport of PKI messages between different PKI management entities.

Replace the text of the section with the following text.

In cases where an end entity sends a protected PKI message to an RA, the RA MAY forward that message to a CA, adding its own protection (which MAY be a MAC or a signature, depending on the information and certificates shared between the RA and the CA). There are different use cases for such multi protected messages.

- o The RA confirms the validation and authorization of a message and forwards the original message unchanged.
- o The RA collects several messages and forwards them in a batch. This can for instance be used to bridge an off-line connection between two PKI management entities. In communication to the CA request messages and in communication from the CA response or announcement messages will be collected in the batch.
- o The RA modifies the message(s) in some way (e.g., add or modify particular field values or add new extensions) before forwarding them, then it MAY create its own desired PKIBody. In case the changes made by the RA to PKIMessage breaks the POP, the RA MUST either set the POP RAVerified or include the original PKIMessage from the EE in the generalInfo field of PKIHeader of the nested message (to force the CA to check POP on the original message). The infoType to be used in this situation is {id-it 15} (see [Section 5.3.19](#) for the value of id-it) and the infoValue is PKIMessages (contents MUST be in the same order as the requests in PKIBody). For simplicity reasons, if batching is used in combination with inclusion of the original PKIMessage in the generalInfo field, all messages in the batch MUST be of the same type (e.g., ir).

These use cases are accomplished by nesting the messages sent by the end entity within a new PKI message. The structure used is as follows.

NestedMessageContent ::= PKIMessages

(The use of PKIMessages, a SEQUENCE OF PKIMessage, lets the RA batch the requests of several EEs in a single new message.)

3.4. Replace [Section 5.2.2](#). - Encrypted Values

[Section 5.2.2 of RFC 4210](#) [[RFC4210](#)] describes the usage of EncryptedValue to transport encrypted data. This document extends the encryption of data to also use EnvelopedData.

Replace the text of the section with the following text.

Where encrypted data (restricted, in this specification, to be either private keys, certificates, or passwords) are sent in PKI messages, the EncryptedKey data structure is used.

```
EncryptedKey ::= CHOICE {  
    encryptedValue      EncryptedValue, -- deprecated  
    envelopedData      [0] EnvelopedData }
```

See CRMF [[RFC4211](#)] for EncryptedKey and EncryptedValue syntax and for EnvelopedData syntax see CMS [[RFC5652](#)]. Using the EncryptedKey data structure, the choice to either use EncryptedValue (for backward compatibility only) or EnvelopedData is offered. The use of the EncryptedValue structure has been deprecated in favor of the EnvelopedData structure. Therefore, it is recommended to use EnvelopedData.

The EncryptedKey data structure is used in CMP to either transport a private key, certificate or revocation passphrase in encrypted form.

EnvelopedData is used as follows:

- o Contains only one recipientInfo structure because the content is encrypted only for one recipient.
- o Contains private key in a SignedData structure as specified in CMS [section 5 \[RFC5652\]](#) signed by the Key Generation Authority.
- o Contains certificate or revocation passphrase directly in the encryptedContent field.

Note: When transferring a centrally generated private key in a certificate response message to the EE, the algorithm identifier and the associated public key will anyhow be transported in this response message. Therefore, the private key will not be delivered in a key package structure as specified in [[RFC5958](#)] and [[RFC6032](#)]. But the wrapping of the private key in a SignedData structure that is wrapped in the EnvelopedData structure as specified in [[RFC6032](#)] is applied.

The content of the EnvelopedData structure, as specified in CMS [section 3 \[RFC5652\]](#), MUST be encrypted using a newly generated symmetric content-encryption key. This content-encryption key MUST be securely provided to the recipient using one of three key management techniques.

The choice of the key management technique to be used by the sender depends on the credential available for the recipient:

- o Jointly shared secret: The content-encryption key will be protected using the symmetric key-encryption key management technique, as specified in CMS [section 5.2.3 \[RFC5652\]](#).
- o Recipient's certificate that contains a key usage extension asserting keyAgreement: The content-encryption key will be protected using the key agreement key management technique, as specified in CMS [section 5.2.2 \[RFC5652\]](#).
- o Recipient's certificate that contains a key usage extension asserting keyEncipherment: The content-encryption key will be protected using the key transport key management technique, as specified in CMS [section 5.2.1 \[RFC5652\]](#).

3.5. Update [Section 5.3.4](#). - Certification Response

[Section 5.3.4 of RFC 4210 \[RFC4210\]](#) describes the Certification Response. This document updates the syntax by using the parent structure EncryptedKey instead of EncryptedValue as described in [Section 3.1](#) above.

Replace the ASN.1 syntax of CertifiedKeyPair and CertOrEncCert with the following text.

```
CertifiedKeyPair ::= SEQUENCE {  
    certOrEncCert      CertOrEncCert,  
    privateKey         [0] EncryptedKey      OPTIONAL,  
    -- see [CRMF] for comment on encoding  
    publicationInfo [1] PKIPublicationInfo  OPTIONAL  
}  
  
CertOrEncCert ::= CHOICE {  
    certificate      [0] Certificate,  
    encryptedCert   [1] EncryptedKey  
}
```

Add the following paragraphs to the end of the section.

The use of EncryptedKey is described in [section 5.2.2](#).

3.6. Replace [Section 5.3.19.9](#). - Revocation Passphrase

[Section 5.3.19.9 of RFC 4210 \[RFC4210\]](#) describes the provisioning of a revocation passphrase for authenticating a later revocation request. This document updates the handling by using the parent structure EncryptedKey instead of EncryptedValue to transport this information as described in [Section 3.1](#) above.

Replace the text of the section with the following text.

This MAY be used by the EE to send a passphrase to a CA/RA for the purpose of authenticating a later revocation request (in the case that the appropriate signing private key is no longer available to authenticate the request). See [Appendix B](#) for further details on the use of this mechanism.

GenMsg: {id-it 12}, EncryptedKey
GenRep: {id-it 12}, < absent >

The use of EncryptedKey is described in [section 5.2.2](#).

3.7. Update [Section 5.3.22](#) - Polling Request and Response

[Section 5.3.22 of RFC 4210](#) [RFC4210] describes when and how polling messages are used. This document adds the polling mechanism also to outstanding p10cr transactions.

Replace all paragraphs in front of the state machine diagram with the following text.

This pair of messages is intended to handle scenarios in which the client needs to poll the server in order to determine the status of an outstanding ir, cr, p10cr, or kur transaction (i.e., when the "waiting" PKIStatus has been received).

```
PollReqContent ::= SEQUENCE OF SEQUENCE {  
    certReqId    INTEGER }  
  
PollRepContent ::= SEQUENCE OF SEQUENCE {  
    certReqId    INTEGER,  
    checkAfter   INTEGER, -- time in seconds  
    reason       PKIFreeText OPTIONAL }
```

The following clauses describe when polling messages are used, and how they are used. It is assumed that multiple certConf messages can be sent during transactions. There will be one sent in response to each ip, cp, or kup that contains a CertStatus for an issued certificate.

- 1 In response to an ip, cp, or kup message, an EE will send a certConf for all issued certificates and, following the ack, a pollReq for all pending certificates.
- 2 In response to a pollReq, a CA/RA will return an ip, cp, or kup if one or more of the pending certificates is ready; otherwise, it will return a pollRep.

- 3 If the EE receives a pollRep, it will wait for at least as long as the checkAfter value before sending another pollReq.
- 4 If an ip, cp, or kup is received in response to a pollReq, then it will be treated in the same way as the initial response.

Note: A p10cr message contains exactly one CertificationRequestInfo data structure as specified in PKCS#10 [RFC2986] but not certificate request number. Therefore, the certReqId MUST be set to 0 in all following messages of this transaction.

3.8. Update [Appendix B](#) - The Use of Revocation Passphrase

[Appendix B of RFC 4210](#) [RFC4210] describes the usage of the revocation passphrase. As this document updates [RFC 4210](#) [RFC4210] to utilize the parent structure EncryptedKey instead of EncryptedValue as described in [Section 3.1](#) above, the description is updated accordingly.

Replace the first bullet point of this section with the following text.

- o The OID and value specified in [Section 5.3.19.9 of RFC 4210](#) [RFC4210] MAY be sent in a GenMsg message at any time, or MAY be sent in the generalInfo field of the PKIHeader of any PKIMessage at any time. (In particular, the EncryptedKey as described in [section 5.2.2](#) may be sent in the header of the certConf message that confirms acceptance of certificates requested in an initialization request or certificate request message.) This conveys a revocation passphrase chosen by the entity (i.e., for use of EnvelopedData this is in the decrypted bytes of encryptedContent of the EnvelopedData structure and for use of EncryptedValue this is in the decrypted bytes of the encValue field) to the relevant CA/RA; furthermore, the transfer is accomplished with appropriate confidentiality characteristics.

Replace the third bullet point of this section with the following text.

- o When using EnvelopedData the unprotectedAttrs and when using EncryptedValue the valueHint field MAY contain a key identifier (chosen by the entity, along with the passphrase itself) to assist in later retrieval of the correct passphrase (e.g., when the revocation request is constructed by the entity and received by the CA/RA).

< TBD: The attribute structure containing the key identifier in the unprotectedAttr field could either be pkcs-9-at-friendlyName or pkcs-

9-at-localKeyId as specified in [RFC 2985 section 5.5](#) [[RFC2985](#)]. Are there preferences for either one? >

3.9. Update [Appendix C](#) - Request Message Behavioral Clarifications

[Appendix C of RFC 4210](#) [[RFC4210](#)] provides clarifications to the request message behavior. As this document updates [RFC 4210](#) [[RFC4210](#)] to utilize the parent structure EncryptedKey instead of EncryptedValue as described in [Section 3.1](#) above, the description is updated accordingly.

Replace the note coming after the ASN.1 syntax of POPOPrivKey of this section with the following text.

```
-- *****
-- *   the type of "thisMessage" is given as BIT STRING in RFC 4211
-- *   [RFC4211]; it should be "EncryptedKey" (in accordance with
-- *   Section 5.2.2, "Encrypted Values", of this specification).
-- *   Therefore, this document makes the behavioral clarification of
-- *   specifying that the contents of "thisMessage" MUST be encoded
-- *   either as EnvelopedData or EncryptedValue (only for backward
-- *   compatibility) and then wrapped in a BIT STRING. This allows
-- *   the necessary conveyance and protection of the private key
-- *   while maintaining bits-on-the-wire compatibility with RFC 4211
-- *   [RFC4211].
-- *****
```

3.10. Update [Appendix D.4](#). - Initial Registration/Certification (Basic Authenticated Scheme)

[Appendix D.4 of RFC 4210](#) [[RFC4210](#)] provides the initial registration/certification scheme. This scheme shall continue to use EncryptedValue for backward compatibility reasons.

Replace the comment after the privateKey field of crc[1].certifiedKeyPair in the syntax of the Initialization Response message with the following text.

```
-- see Appendix C, Request Message Behavioral Clarifications
-- for backward compatibility reasons, use EncryptedValue
```

4. IANA Considerations

< TBD: Add any IANA considerations >

5. Security Considerations

No changes are made to the existing security considerations of [RFC 4210](#) [[RFC4210](#)].

6. Acknowledgements

Special thank goes to Jim Schaad for his guidance and the inspiration on structuring and writing this document I got from [[RFC6402](#)] that updates CMC.

I also like to thank all reviewers of this document for their valuable feedback.

7. References

7.1. 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>>.
- [RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", [RFC 2985](#), DOI 10.17487/RFC2985, November 2000, <<https://www.rfc-editor.org/info/rfc2985>>.
- [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>>.
- [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>>.
- [RFC6402] Schaad, J., "Certificate Management over CMS (CMC) Updates", [RFC 6402](#), DOI 10.17487/RFC6402, November 2011, <<https://www.rfc-editor.org/info/rfc6402>>.

[7.2. Informative References](#)

- [I-D.brockhaus-lamps-lightweight-cmp-profile] Brockhaus, H., Fries, S., and D. Oheimb, "Lightweight CMP Profile", [draft-brockhaus-lamps-lightweight-cmp-profile-03](#) (work in progress), January 2020.
- [RFC5958] Turner, S., "Asymmetric Key Packages", [RFC 5958](#), DOI 10.17487/RFC5958, August 2010, <<https://www.rfc-editor.org/info/rfc5958>>.
- [RFC6032] Turner, S. and R. Housley, "Cryptographic Message Syntax (CMS) Encrypted Key Package Content Type", [RFC 6032](#), DOI 10.17487/RFC6032, December 2010, <<https://www.rfc-editor.org/info/rfc6032>>.

[Appendix A. ASN.1 Modules](#)

Changes to the following parts are needed

- o Import from PKCS-9

friendlyName, localKeyId

```
FROM PKCS-9 {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs-9(9) modules(0) pkcs-9(1)}
```

< TBD: Either friendlyName or localKeyId need to be imported here. >

- o Import from PKIXCRMF-2005


```
CertTemplate, PKIPublicationInfo, EncryptedKey, CertId,
CertReqMessages
  FROM PKIXCRMF-2005 {iso(1) identified-organization(3)
    dod(6) internet(1) security(5) mechanisms(5) pkix(7)
    id-mod(0) id-mod-crmf2005(36)}
```

o In CertifiedKeyPair, CertOrEncCert and id-it-revPassphrase

```
CertifiedKeyPair ::= SEQUENCE {
  certOrEncCert      CertOrEncCert,
  privateKey         [0] EncryptedKey      OPTIONAL,
  -- see [CRMF] for comment on encoding
  publicationInfo    [1] PKIPublicationInfo OPTIONAL
}
```

```
CertOrEncCert ::= CHOICE {
  certificate      [0] CMPCertificate,
  encryptedCert    [1] EncryptedKey
}
```

```
-- id-it-revPassphrase    OBJECT IDENTIFIER ::= {id-it 12}
--   RevPassphraseValue    ::= EncryptedKey
```

```
--
-- Extended Key Usage extension for PKI entities used in
-- CMP operations
--
```

```
id-kp-cmpCA OBJECT IDENTIFIER ::= { id-kp 27 }
id-kp-cmpRA OBJECT IDENTIFIER ::= { id-kp 28 }
id-kp-cmpKGA OBJECT IDENTIFIER ::= { id-kp ... }
```

< TBD: id-kp-cmpKGA to be defined. >

< TBD: If needed the complete ASN.1 Module from [RFC 4210](#) section needs to be copied here. >

Author's Address

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

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

