Lightweight CMP Profile and CMP Updates

draft-brockhaus-lamps-cmp-profile-00
draft-brockhaus-lamps-cmp-updates-00

Hendrik Brockhaus, Steffen Fries, David von Oheimb

IETF 105 – LAMPS Working Group

Results of IETF 104

- Changes to CMP V2 needs a standard track RFC and therefore must be adopted by a working group.
- The CMP Profile could either be an individual contribution or a working group item as it would normatively reference the CMP RFC for profiling.
- Steffen discussed with Jim, if ACE would be interested in this work. The conclusion was, that the topic is not specific for constrained environments and that LAMPS seems to be the better home for this work (also as LAMPS is the successor of PKIX, which CMP originated from).
- Hendrik generated a separate draft on the updates for CMP and updated the profile draft.
- Hendrik proposed a text for the LAMPS WG charter to add the work on CMP.

Lightweight CMP Profile

draft-brockhaus-lamps-lightweight-cmp-profile-00

Replaces draft-brockhaus-lamps-industral-cmp-profile-00

Change history:

- Widen focus from industrial to more **multi-purpose use cases** and **lightweight** CMP profile.
- Incorporate the **omitted confirmation** into the header specified and described in the standard enrollment use case due to discussion with Tomas.
- Change from OPTIONAL to RECOMMENDED for use case '**Revoke another's entities certificate**', because it is regarded as important functionality in many environments to enable a management station to revoke EE certificates.
- Complete the specification of the **revocation message flow**.
- Remove the CoAP-based transport mechanism and piggybacking of CMP messages on top of other reliable transport protocols as they are **out of scope** of this document and would need to be specified in another document.

CMP Updates

draft-brockhaus-lamps-cmp-updates-00

Initial version

Topics covered in the document:

- **Offering EnvelopedData** as another choice next to EncryptedValue to extend crypto agility in CMP.
  As discussed with Jim Schaad we exchanged EncryptedValue with EncryptedKey as EncryptedKey also offers EnvelopedData as a choice and provides backward compatibility.

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

  Note that according to RFC 4211 section 2.1 the use of EncryptedValue has already been deprecated in favor of EnvelopedData.
- Add new **extended key usages for CMP server**, e.g., Registration Authority and Certification Authority.

Request to add the work on CMP to the WG charter

We propose the following addition to the charter of LAMPS to adopt the work on CMP:

As certificate management gets increasingly important in many environments, it needs to be tailored to the specific needs. CMP as existing protocol offers a vast range of options. As it is already being applied in different industrial environments it needs to be enhanced to more efficiently support of these use cases, crypto agility and specific communication relations on the one hand and profiled to the necessary functionality on the other hand to ease application and to better facilitate interoperable implementation.

As Russ counted, there were **six people in support of work on a CMP profile, four people willing to review, three people plan to implement, and one person is opposed** to this being added to the charter.
So quite a positive feedback :-)

Backup

Looking for guidance on the best way to implement EnvelopedData

- Using EncryptedKey instead of EncryptedVale as proposed by Jim Schaad

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

}

CertifiedKeyPair ::= SEQUENCE {
   certOrEncCert           CertOrEncCert,
   privateKey        [0]   EncryptedKey          OPTIONAL,
   publicationIn     [1]   PKIPublicationInfo    OPTIONAL
}
```

- Or adding a new value for privateKey coded as EnvelopedData as proposes by others from within my company

```
CertifiedKeyPair ::= SEQUENCE {
   certOrEncCert           CertOrEncCert,
   privateKey        [0]   EncryptedValue        OPTIONAL,
   publicationInfo   [1]   PKIPublicationInfo    OPTIONAL,
   privateKey        [2]   EnvelopedData         OPTIONAL
}
```

Looking for the opinion of other ASN.1 experts what the better way is with regard to backward compatibility.