rfc6712bis and rfc4210bis

draft-ietf-lamps-rfc6712bis-07
Hendrik Brockhaus, David von Oheimb , Mike Ounsworth, John Gray

draft-ietf-lamps-rfc4210bis-14
Hendrik Brockhaus, David von Oheimb , Mike Ounsworth, John Gray

Hendrik Brockhaus

IETF 121 – LAMPS Working Group

Activities since IETF 120 on rfc4210bis

Changes since IETF 120:
- Addressed OPSDIR and TSVART expert review comments
- Updated the definition of "NULL-DN" in Section 5.1.1 and Appendix D.1
- Added specification of how RA/CA shall generate rid content with direct POP in Section 5.2.8.3.3
- Implemented some editorial changes throughout the document
- Updated reference from draft-ietf-lamps-cms-kemri to RFC 9629

Next Steps:
- OPSDIR: Has nits, TSVART: Ready w/issues, GENART and SECDIR: Ready, ARTART: Incomplete, IANA: OK- Actions needed
- The current version of the draft is available on https://github.com/lamps-wg/cmp-updates addressing all expert review commnets
- The draft-ietf-lamps-attestation-freshness will contain an ASN.1 module containing nonceRequest /nonceResponse syntax

Activities since IETF 120 on rfc6712bis

Changes since IETF 120:

- Addressed HTTPDIR expert review comment
  - Removing HTTP/1.0 requiment
  - Removing redundant text in Section 3.4 on the “Contnet-Length” header field
  - Removed Section 3.8 due to redundancy with current HTTP specifications
- Addressed SECDIR, OPSDIR and ARTART expert review comments
- Use ‘PKIMessage sequences’ as plural of PKIMessage to prevent confusion with ASN.1 type PKIMessages
- Added normative language in Sections 3.3 and 3.7 for clarity
- Aligned Section 3.6 and Section 5 with RFC 9483 and draft-ietf-anima-brski-ae
- Updates IANA considerations addressing IANA early review

Next Steps:

- HTTPDIR: Not ready, OPSDIR: Has issued, GENART and ARTART: Almost ready, SECDIR: Ready, TSVART: Ready w/issues, IANA: OK- Actions needed
- The current version of the draft is available on https://github.com/lamps-wg/cmp-updates addressing all expert review comments

HTTP error status codes vs. CMP error reporting

In cases where the CMP request is not successful and an CMP error message or the CMP response contains a PKIStatusInfo indicating an error. The authors realized that different applications handle this situation differently.

There are two questions:

- If the CMP application reports an error, e.g., rejects the request, and provides application-level content for the HTTP response, shall the server provide this content together with a 200 (OK) or an HTTP error status code? What would be the right HTTP status code for this situation?
- If there is an HTTP error status code and a content in an HTTP response, shall the client provide the received content to the application layer, or drop it?

BAK

New ASN.1 structures for KEM-based message protection

```
id-KemBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 16}

KemBMParameter ::= SEQUENCE {
  kdf               AlgorithmIdentifier{KEY-DERIVATION, {...}},
  kemContext    [0] OCTET STRING     OPTIONAL,
  len               INTEGER (1..MAX),
  mac               AlgorithmIdentifier{MAC-ALGORITHM, {...}}
}
```

Algorithm identifier to be used in PKIHeader.protectionAlg when KEM-based MAC is used.
Entrust registered the OID in the same branch as PBMParameter.
Optional kemContext if needed with the used KEM algorithm like ukm in cms-kemri.

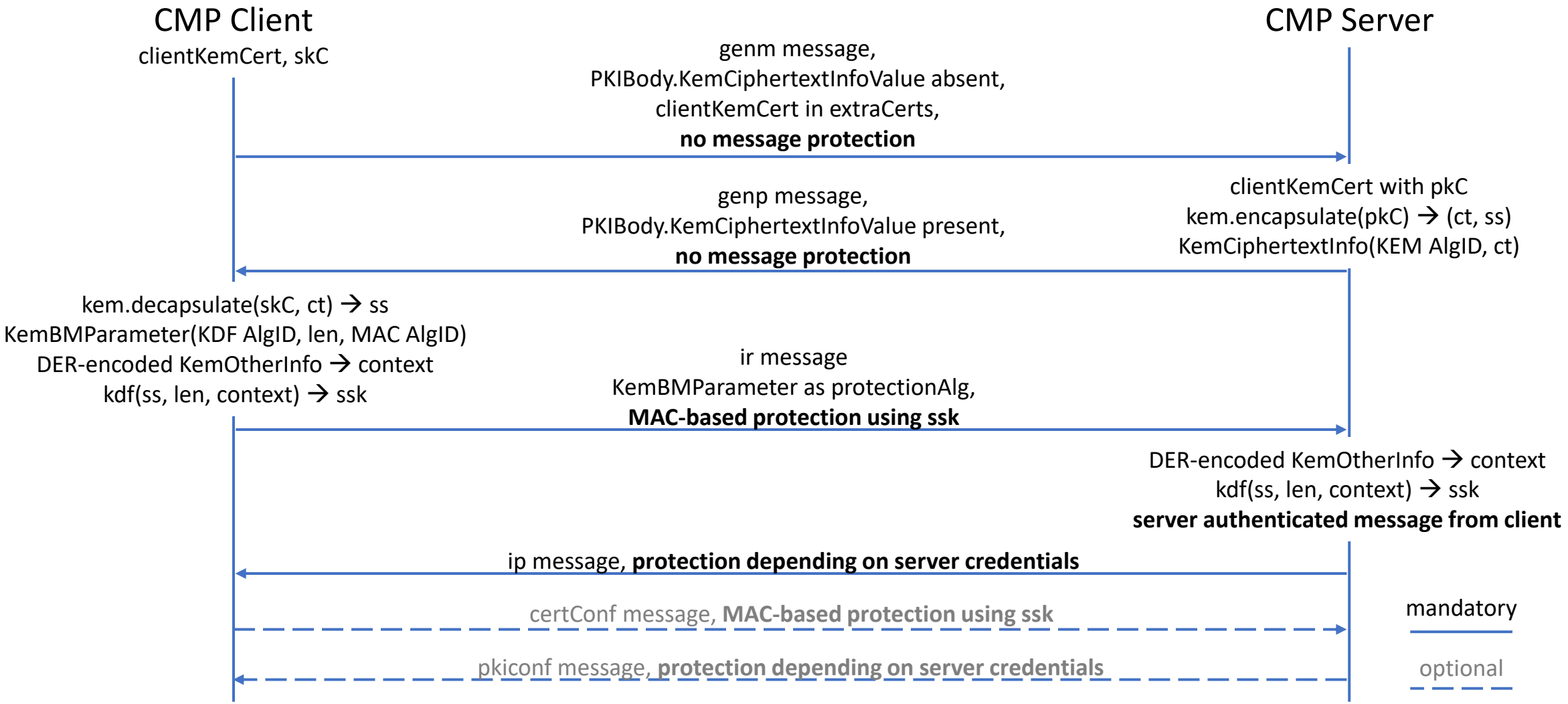
```
id-it-KemCiphertextInfo OBJECT IDENTIFIER ::= { id-it TBD1 }
KemCiphertextInfoValue ::= KemCiphertextInfo

KemCiphertextInfo ::= SEQUENCE {
  kem               AlgorithmIdentifier{KEM-ALGORITHM, {...}},
  ct                OCTET STRING
}
```

InfoTypeAndValue to deliver the KEM ciphertext in body of general message or in generalInfo field of message header.

```
KemOtherInfo ::= SEQUENCE {
  staticString      PKIFreeText,
  transactionID     OCTET STRING,
  kemContext    [0] OCTET STRING     OPTIONAL
}
```

Context information as input to the KDF for domain separation and for ensuring uniqueness of MAC-keys.
Uses transactionID from the message containing the KemCiphertextInfoValue.ct.
Optional kemContext if needed with the used KEM algorithm like ukm in cms-kemri.

Client owns KEM key pair



Server owns KEM key pair