LAMPS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2021

H. Brockhaus
Siemens

October 24, 2020

CMP Algorithms
draft-ietf-lamps-cmp-algorithms-00

Abstract

This document describes the conventions for using several cryptographic algorithms with the Certificate Management Protocol (CMP). CMP is used to enroll and further manage the lifecycle of X.509 certificates.

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 April 27, 2021.

Table of Contents

1. Introduction

1.1. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Message Digest Algorithms

This section specifies the conventions employed by CMP implementations that support SHA-1 or SHA2 algorithm family.

Digest algorithm identifiers are located in the hashAlg field of OOBCertHash, the owf field of Challenge, PBMParameter, and DHBMParameter, and the digestAlgorithms field of SignedData and the digestAlgorithm field of SignerInfo.

Digest values are located in the hashVal field of OOBCertHash, the witness field of Challenge, and the certHash field of CertStatus. In addition, digest values are input to signature algorithms.

2.1. SHA2

The SHA2 message digest algorithm family is defined in FIPS Pub 180-4 [FIPS180-4].

The message digest algorithms SHA-224, SHA-256, SHA-384, and SHA-512 produce a 224-bit are identified by the following object identifiers (OIDs):

```
id-sha224 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
   us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
   hashalgs(2) 4 }
id-sha256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
   us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
   hashalgs(2) 1 }
id-sha384 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
   us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
   hashalgs(2) 2 }
id-sha512 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
   us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
   hashalgs(2) 3 }
```

Further conventions to be considered are specified in RFC 5754 Section 2 [RFC5754].

3. Signature Algorithms

This section specifies the conventions employed by CMP implementations that support DSA, RSA, or ECDSA.

The signature algorithm is referred to as MSG_SIG_ALG in RFC 4210 Appendix D and E [RFC4210] and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Signature algorithm identifiers are located in the protectionAlg field of PKIHeader, the algorithmIdentifier field of POPOSigningKey, signatureAlgorithm field of p10cr, SignKeyPairTypes, and the SignerInfo signatureAlgorithm field of SignedData.

Signature values are located in the protection field of PKIMessage, signature field of POPOSigningKey, signature field of p10cr, and SignerInfo signature field of SignedData.

3.1. DSA

The DSA signature algorithm is defined in FIPS Pub 186-5 [FIPS186-5] and MAY be used with SHA-224 and SHA-256 as specified in RFC 5754 [RFC5754].

The algorithm identifiers for DSA with SHA2 signature values are:

```
id-dsa-with-sha224 OBJECT IDENTIFIER ::=  { joint-iso-ccitt(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   algorithms(4) id-dsa-with-sha2(3) 1 }
id-dsa-with-sha256 OBJECT IDENTIFIER ::=  { joint-iso-ccitt(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   algorithms(4) id-dsa-with-sha2(3) 2 }
```

Further conventions to be considered are specified in RFC 5754 Section 3.1 [RFC5754].

3.2. RSA

The RSA (RSASSA-PSS and RSASSA-PKCS1-v1_5) signature algorithm is defined in RFC 8017 [RFC8017]. RSASSA-PKCS1-v1_5 MAY be used with SHA-224, SHA-256, SHA-384, or SHA-512 as specified in RFC 5754 [RFC5754].

The algorithm identifiers for RSASAA-PSS signatures as specified in RFC 4055 [RFC4055] is:

```
id-RSASSA-PSS OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 10 }
```

Further conventions to be considered are specified in RFC 4056 [RFC4056].

The algorithm identifiers for RSASSA-PKCS1-v1_5 signatures as specified in RFC 4055 [RFC4055] are:

```
sha224WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)
   member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 14 }
sha256WithRSAEncryption  OBJECT IDENTIFIER  ::=  { iso(1)
   member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 11 }
sha384WithRSAEncryption  OBJECT IDENTIFIER  ::=  { iso(1)
   member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 12 }
sha512WithRSAEncryption  OBJECT IDENTIFIER  ::=  { iso(1)
   member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 13 }
```

Further conventions to be considered are specified in RFC 5754 Section 3.2 [RFC5754].

3.3. ECDSA

The ECDSA signature algorithm is defined in FIPS Pub 186-5 [FIPS186-5] and MAY be used with SHA-224, SHA-256, SHA-384, or SHA-512 as specified in RFC 5754 [RFC5754].

The algorithm identifiers for ECDSA with SHA2 signature values are:

```
ecdsa-with-SHA224 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 1 }
ecdsa-with-SHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 2 }
ecdsa-with-SHA384 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 3 }
ecdsa-with-SHA512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 4 }
```

Further conventions to be considered are specified in RFC 5754 Section 3.3 [RFC5754].

4. Key Management Algorithms

CMP accommodates the following general key management techniques: key agreement, key transport, and passwords.

CRMF [RFC4211] and CMP Updates [I-D.ietf-lamps-cmp-updates] facilitate the use of CMS [RFC5652] EnvelopedData by deprecating the use of EncryptedValue.

4.1. Key Agreement Algorithms

The key agreement algorithm is referred to as PROT_ENC_ALG in RFC 4210 Appendix D and E [RFC4210] and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Key agreement algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData together with the key agreement key management technique. When a key agreement algorithm is used, a key-encryption algorithm (Section 4.3) is needed next to the content-encryption algorithm (Section 5).

Key agreement algorithm identifiers are located in the EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm fields.

Key encryption algorithm identifiers are located in the EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm field.

Wrapped content-encryption keys are located in the EnvelopedData RecipientInfos KeyAgreeRecipientInfo RecipientEncryptedKeys encryptedKey field.

4.1.1. Diffie-Hellman

Diffie-Hellman key agreement is defined in RFC 2631 [RFC2631] and MAY be used in the ephemeral-static or a static-static variant as specified in RFC 3370 [RFC3370].

The Diffie-Hellman algorithm identifiers are:

```
id-alg-ESDH OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 5 }
id-alg-SSDH OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 10 }
```

Further conventions to be considered are specified in RFC 3370 Section 4.1 [RFC3370].

4.1.2. ECDH

Elliptic Curve Diffie-Hellman (ECDH) key agreement is defined in RFC 5753 [RFC5753] and MAY be used on the ephemeral-static variant in RFC 5753 [RFC5753], the 1-Pass ECMQV variant as specified in RFC 5753 [RFC5753] or the static-static variant as specified in RFC RFC 6278 [RFC6278].

Algorithm Identifiers and further conventions to be considered are specified in RFC RFC 5753 [RFC5753] and RFC 6278 [RFC6278].

4.2. Key Transport Algorithms

The key transport algorithm is also referred to as PROT_ENC_ALG in RFC 4210 Appendix D and E [RFC4210] and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Key transport algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData together with the key transport key management technique.

Key transport algorithm identifiers are located in the EnvelopedData RecipientInfos KeyTransRecipientInfo keyEncryptionAlgorithm field.

Key transport encrypted content-encryption keys are located in the EnvelopedData RecipientInfos KeyTransRecipientInfo encryptedKey field.

4.2.1. RSA

The RSA key transport algorithm is the RSA encryption scheme defined in RFC 8017 [RFC8017].

The algorithm identifier for RSA (PKCS #1 v1.5) is:

```
rsaEncryption OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 1 }
```

The algorithm identifier for RSAES-OAEP is:

```
id-RSAES-OAEP  OBJECT IDENTIFIER  ::=  { iso(1) member-body(2)
   us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 7 }
```

Further conventions to be considered for PKCS #1 v1.5 are specified in RFC 3370 Section 4.2.1 [RFC3370] and for RSAES-OAEP in RFC 3560 [RFC3560].

4.3. Symmetric Key-Encryption Algorithms

The symmetric key-encryption algorithm is also referred to as PROT_SYM_ALG in RFC 4210 Appendix D and E [RFC4210] and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

As symmetric key-encryption key management technique is not used by CMP, the symmetric key-encryption algorithm is only needed when using the key agreement or password-based key management technique with CMS [RFC5652] EnvelopedData.

Key-encryption algorithm identifiers are located in the EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm and EnvelopedData RecipientInfos PassworRecipientInfo keyEncryptionAlgorithm fields.

Wrapped content-encryption keys are located in the EnvelopedData RecipientInfos KeyAgreeRecipientInfo RecipientEncryptedKeys encryptedKey and EnvelopedData RecipientInfos PassworRecipientInfo encryptedKey fields.

4.3.1. AES Key Wrap with Padding

The AES key encryption algorithm is defined in RFC 3394 [RFC3394] and the respective padding is defined in RFC 5649 [RFC5649].

AES key encryption has the algorithm identifier:

```
id-aes256-wrap-pad OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
   country(16) us(840) organization(1) gov(101) csor(3)
   nistAlgorithm(4) aes(1) 48 }
```

Further conventions to be considered for AES key wrap with padding are specified in RFC 5649 Section 4 [RFC5649].

4.4. Key Derivation Algorithms

Key derivation algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData together with password-based key management technique.

Key derivation algorithm identifiers are located in the EnvelopedData RecipientInfos PassworRecipientInfo keyDerivationAlgorithm field.

4.4.1. Password-based Key Derivation Function 2

The password-based key derivation function 2 (PBKDF2) is defined in RFC 8018 [RFC8018].

Password-based key derivation function 2 has the algorithm identifier:

```
id-PBKDF2 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
   rsadsi(113549) pkcs(1) pkcs-5(5) 12 }
```

Further conventions to be considered for PBKDF2 are specified in RFC 3370 Section 4.4.1 [RFC3370] and RFC 8018 Section 5.2 [RFC8018].

5. Content Encryption Algorithms

The content encryption algorithm is also referred to as PROT_SYM_ALG in RFC 4210 Appendix D and E [RFC4210] and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Content encryption algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData to transport a signed private key package in case of central key generation or key archiving, a certificate to facilitate implicit prove-of-possession, or a revocation passphrase in encrypted form.

Content encryption algorithm identifiers are located in the EnvelopedData EncryptedContentInfo contentEncryptionAlgorithmrithm field.

Encrypted content is located in the EnvelopedData EncryptedContentInfo encryptedContent field.

5.1. AES

Since the using CMP, the content encrypted is a cryptographic key and its attributes, a certificate or a password, the same algorithms as specified in Section 4.3.1 are used for content encryption.

6. Message Authentication Code Algorithms

The message authentication code algorithm is also referred to as MSG_MAC_ALG in RFC 4210 Appendix D and E [RFC4210] and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Message authentication code algorithm identifiers are located in the mac field of PBMParameter and DHBMParameter, the PBKDF2-params prf field.

Message authentication code values are located in the EnvelopedData EncryptedContentInfo encryptedContent field.

6.1. Password-based MAC

The password-based MAC is defined in RFC 4210 [RFC4210].

The algorithm identifiers for password-based MAC is:

```
id-PasswordBasedMac OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) nt(113533) nsn(7) algorithms(66) 13 }
```

Further conventions to be considered for password-based MAC are specified in RFC 4210 Section 5.1.3.1 [RFC4210].

6.2. Diffie-Hellman-based MAC

The Diffie-Hellman-based MAC is defined in RFC 4210 [RFC4210].

The algorithm identifiers for Diffie-Hellman-based MAC is:

```
id-DHBasedMac OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) nt(113533) nsn(7) algorithms(66) 30 }
```

Further conventions to be considered for Diffie-Hellman-based MAC are specified in RFC 4210 Section 5.1.3.2 [RFC4210].

6.3. HMAC SHA2

The HMAC is defined in RFC 2104 [RFC2104].

The algorithm identifiers for HMAC with SHA2 as specified in RFC 4231 [RFC4231] are:

```
id-hmacWithSHA224 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) digestAlgorithm(2) 8 }
id-hmacWithSHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) digestAlgorithm(2) 9 }
id-hmacWithSHA384 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) digestAlgorithm(2) 10 }
id-hmacWithSHA512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
   us(840) rsadsi(113549) digestAlgorithm(2) 11 }
```

Further conventions to be considered for HMAC with SHA2 are specified in RFC 4231 Section 3.1 [RFC4231].

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

TBD

10. References

10.1. Normative References

[FIPS180-4]
NIST, "FIPS Pub 180-4: Secure Hash Standard (SHA)", August 2015 , <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

[FIPS186-5]
NIST, "FIPS Pub 186-5: Digital Signature Standard (DSS)", October 2019, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf>.

[I-D.ietf-lamps-cmp-updates]
Brockhaus, H., "CMP Updates", draft-ietf-lamps-cmp-updates-05 (work in progress), September 2020.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[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>.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, DOI 10.17487/RFC2631, June 1999, <https://www.rfc-editor.org/info/rfc2631>.

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, DOI 10.17487/RFC3370, August 2002, <https://www.rfc-editor.org/info/rfc3370>.

[RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, DOI 10.17487/RFC3394, September 2002, <https://www.rfc-editor.org/info/rfc3394>.

[RFC3560] Housley, R., "Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS)", RFC 3560, DOI 10.17487/RFC3560, July 2003, <https://www.rfc-editor.org/info/rfc3560>.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, DOI 10.17487/RFC4055, June 2005, <https://www.rfc-editor.org/info/rfc4055>.

[RFC4056] Schaad, J., "Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS)", RFC 4056, DOI 10.17487/RFC4056, June 2005, <https://www.rfc-editor.org/info/rfc4056>.

[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>.

[RFC4231] Nystrom, M., "Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512", RFC 4231, DOI 10.17487/RFC4231, December 2005, <https://www.rfc-editor.org/info/rfc4231>.

[RFC5649] Housley, R. and M. Dworkin, "Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm", RFC 5649, DOI 10.17487/RFC5649, September 2009, <https://www.rfc-editor.org/info/rfc5649>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC5753] Turner, S. and D. Brown, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 5753, DOI 10.17487/RFC5753, January 2010, <https://www.rfc-editor.org/info/rfc5753>.

[RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, DOI 10.17487/RFC5754, January 2010, <https://www.rfc-editor.org/info/rfc5754>.

[RFC6278] Herzog, J. and R. Khazan, "Use of Static-Static Elliptic Curve Diffie-Hellman Key Agreement in Cryptographic Message Syntax", RFC 6278, DOI 10.17487/RFC6278, June 2011, <https://www.rfc-editor.org/info/rfc6278>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[RFC8018] Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, <https://www.rfc-editor.org/info/rfc8018>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[I-D.ietf-lamps-lightweight-cmp-profile] Brockhaus, H., Fries, S., and D. Oheimb, "Lightweight CMP Profile", draft-ietf-lamps-lightweight-cmp-profile-03 (work in progress), October 2020.

Appendix A. History of changes

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

Author's Address

Hendrik Brockhaus
Siemens AG

Email: hendrik.brockhaus@siemens.com