S/MIME WG
Internet Draft
Intended Status: Standard Track
Updates: 3370 (once approved)
Expires: July 16, 2009

Sean Turner, IECA
January 16, 2009

Using SHA2 Algorithms with Cryptographic Message Syntax
draft-ietf-smime-sha2-11.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the conventions for using the Secure Hash Algorithm (SHA) message digest algorithms (SHA-224, SHA-256, SHA-384, SHA-512) with the Cryptographic Message Syntax (CMS). It also describes the conventions for using these algorithms with CMS and the Digital Signature Algorithm (DSA), Rivest Shamir Adleman (RSA), and Elliptic Curve DSA (ECDSA) signature algorithms. Further, it provides SMIMECapabilities attribute values for each algorithm.

Conventions used in this document

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 [RFC2119].

Table of Contents

1. Introduction

This document specifies the algorithm identifiers and specifies parameters for the message digest algorithms SHA-224, SHA-256, SHA-384, and SHA-512 for use with the Cryptographic Message Syntax (CMS) [RFC3852]. The message digest algorithms are defined in [SHS] and reference code is provided in [RFC4634].

This document also specifies the algorithm identifiers and parameters for use of SHA-224, SHA-256, SHA-384, and SHA-512 with DSA [DSS], RSA (RSASSA-PKCS1-v1_5) [RFC3447], and ECDSA [DSS].

This document does not define new identifiers; they are taken from [RFC3874], [RFC4055], and [ECCADD]. Additionally, the parameters follow the conventions specified therein. Therefore, there is no Abstract Syntax Notation One (ASN.1) module included in this document.

Note that [RFC4231] specifies the conventions for the message authentication code (MAC) algorithms: HMAC with SHA-224, HMAC with SHA-256, HMAC with SHA-384, and HMAC with SHA-512.

In CMS, the various algorithm identifiers use the AlgorithmIdentifier syntax, which is included here for convenience:

```
AlgorithmIdentifier  ::=  SEQUENCE  {
  algorithm   OBJECT IDENTIFIER,
  parameters  ANY DEFINED BY algorithm OPTIONAL  }
```

This document also specifies the SMIMECapabilities attribute values [RFCTBD1] for each algorithm. The values provided are for the SMIMECapability field, which is included here for convenience:

```
SMIMECapability ::= SEQUENCE {
  capabilityID  OBJECT IDENTIFIER,
  parameters    ANY DEFINED BY capabilityID OPTIONAL }
```

2. Message Digest Algorithms

Digest algorithm identifiers are located in the SignedData digestAlgorithms field, the SignerInfo digestAlgorithm field, the DigestedData digestAlgorithm field, and the AuthenticatedData digestAlgorithm field. The object identifiers are taken from [RFC4055].

Digest values are located in the DigestedData digest field and the Message Digest authenticated attribute. In addition, digest values are input to signature algorithms.

The digest algorithm identifiers use the AlgorithmIdentifier syntax elaborated upon in Section 1.

The algorithm field and SMIMECapabilities attribute are discussed in Sections 2.1-2.4 for each message digest algorithm. Section 3 provides some signatures that use SHA2 algorithms. Consult the signature algorithm definitions for the procedures to compute the digest values (i.e., DigestInfo).

The AlgorithmIdentifier parameters field is OPTIONAL. Implementations MUST accept SHA2 AlgorithmIdentifiers with absent parameters. Implementations MUST accept SHA2 AlgorithmIdentifiers with NULL parameters. Implementations MUST generate SHA2 AlgorithmIdentifiers with absent parameters.

NOTE: There are two possible encodings for the AlgorithmIdentifier parameters field associated with these object identifiers. The two alternatives arise from the loss of the OPTIONAL associated with the algorithm identifier parameters when the 1988 syntax for AlgorithmIdentifier was translated into the 1997 syntax. Later the OPTIONAL was recovered via a defect report, but by then many people thought that algorithm parameters were mandatory. Because of this history some implementations encode parameters as a NULL element while others omit them entirely. The correct encoding is to omit the parameters field; however, when some uses of these algorithms were defined, it was done using the NULL parameters rather than absent parameters. For example, PKCS#1 [RFC3447] requires that the padding used for RSA signatures (EMSA-PKCS1-v1_5) MUST use SHA2 AlgorithmIdentifiers with NULL parameters (to clarify, the requirement "MUST generate SHA2 AlgorithmIdentifiers with absent parameters" in the previous paragraph does not apply to this padding).

2.1. SHA-224

The SHA-224 message digest algorithm is defined in [SHS]. The algorithm identifier for SHA-224 is:

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

The parameters are as specified in Section 2.

The SMIMECapabilities attribute value indicates support for SHA-224 in a SEQUENCE with the capabilityID field containing the object identifier id-sha224 with absent parameters. The DER encoding for this SMIMECapability is:

```
   id-sha224: 30 0b 06 09 60 86 48 01 65 03 04 02 04
```

2.2. SHA-256

The SHA-256 message digest algorithm is defined in [SHS]. The algorithm identifier for SHA-256 is:

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

The parameters are as specified in Section 2.

The SMIMECapabilities attribute value indicates support for SHA-256 in a SEQUENCE with the capabilityID field containing the object identifier id-sha256 with absent parameters. The DER encoding for this SMIMECapability value is:

```
id-sha256: 30 0b 06 09 60 86 48 01 65 03 04 02 01
```

2.3. SHA-384

The SHA-384 message digest algorithm is defined in [SHS]. The algorithm identifier for SHA-384 is:

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

The parameters are as specified in Section 2.

The SMIMECapabilities attribute value indicates support for SHA-384 in a SEQUENCE with the capabilityID field containing the object identifier id-sha384 with absent parameters. The DER encoding for this SMIMECapability value is:

```
id-sha384: 30 0b 06 09 60 86 48 01 65 03 04 02 02
```

2.4. SHA-512

The SHA-512 message digest algorithm is defined in [SHS]. The algorithm identifier for SHA-512 is:

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

The parameters are as specified in Section 2.

The SMIMECapabilities attribute value indicates support for SHA-384 in a SEQUENCE with the capabilityID field containing the object identifier id-sha384 with absent parameters. The DER encoding for this SMIMECapability value is:

```
id-sha512: 30 0b 06 09 60 86 48 01 65 03 04 02 03
```

3. Signature Algorithms

This section specifies the conventions employed by CMS implementations that support DSA, RSA, and ECDSA with SHA2 algorithms.

Signature algorithm identifiers are located in the SignerInfo signatureAlgorithm field of SignedData. Also, signature algorithm identifiers are located in the SignerInfo signatureAlgorithm field of countersignature attributes.

Signature values are located in the SignerInfo signature field of SignedData. Also, signature values are located in the SignerInfo signature field of countersignature attributes.

3.1. DSA

[RFC3370] section 3.1 specifies the conventions for DSA with SHA-1 public key algorithm identifiers, parameters, public keys, and signature values. DSA with SHA2 algorithms uses the same conventions for these public key algorithm identifiers, parameters, public keys, and signature values. DSA MAY be used with SHA-224 and SHA-256. The object identifiers are taken from [ECCADD].

DSA has not been specified with SHA-384 and SHA-512. SHA-384 and SHA-512 are not supported because the maximum bit length of p (specified as L) is 3072 for DSA. For consistent cryptographic strength, SHA-384 would be used with DSA where L is 7680, and SHA-512 would be used with DSA where L is 15360.

The algorithm identifier for DSA with SHA-224 signature values is:

```
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 }
```

The algorithm identifier for DSA with SHA-256 signature values is:

```
  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 }
```

When either of these algorithm identifiers is used, the AlgorithmIdentifier parameters field MUST be absent.

The SMIMECapabilities attribute value indicates support for one of the DSA signature algorithms in a SEQUENCE with the capabilityID field containing the object identifier id-dsa-with-sha* (where * is 224 or 256) with absent parameters. The DER encoding for these SMIMECapability values are:

```
   id-dsa-with-sha224: 30 0b 06 09 60 86 48 01 65 03 04 03 01

   id-dsa-with-sha256: 30 0b 06 09 60 86 48 01 65 03 04 03 02
```

3.2. RSA

[RFC3370] section 3.2 specifies the conventions for RSA with SHA-1 (RSASSA-PKCS1-v1_5) public key algorithm identifiers, parameters, public keys, and signature values. RSA with SHA2 algorithms uses the same conventions for these public key algorithm identifiers, parameters, public keys, and signature values. RSA (RSASSA-PKCS1-v1_5) [RFC3447] MAY be used with SHA-224, SHA-256, SHA-384, or SHA-512. The object identifiers are taken from [RFC4055].

The object identifier for RSA with SHA-224 signature values is:

```
  sha224WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 14 }
```

The object identifier for RSA with SHA-256 signature values is:

```
  sha256WithRSAEncryption  OBJECT IDENTIFIER  ::=  { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 11 }
```

The object identifier for RSA with SHA-384 signature values is:

```
  sha384WithRSAEncryption  OBJECT IDENTIFIER  ::=  { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 12 }
```

The object identifier for RSA with SHA-512 signature values is:

```
  sha512WithRSAEncryption  OBJECT IDENTIFIER  ::=  { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 13 }
```

When any of these four object identifiers appears within an AlgorithmIdentifier, the parameters MUST be NULL. Implementations MUST accept the parameters being absent as well as present.

The SMIMECapabilities attribute value indicates support for one of the DSA signature algorithms in a SEQUENCE with the capabilityID field containing the object identifier sha*WithRSAEncryption (where * is 224, 256, 384, or 512) with NULL parameters. The DER encoding for these SMIMECapability values are:

```
   sha224WithRSAEncryption: 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0e
                            05 00

   sha256WithRSAEncryption: 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b
                            05 00

   sha384WithRSAEncryption: 30 0d 06 09 2a 86 48 86 f7 0d 01 01 Oc
                            05 00

   sha512WithRSAEncryption: 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0d
                            05 00
```

3.3. ECDSA

[RFCTBD2] section 2.1 specifies the conventions for ECDSA with SHA-1 public key algorithm identifiers, parameters, public keys, and signature values. ECDSA with SHA2 algorithms uses the same conventions for these public key algorithm identifiers, parameters, public keys, and signature values, except that the digestAlgorithm MUST include the corresponding message digest algorithm identifier, and not the SHA-1 object identifier. ECDSA MAY be used with SHA-224, SHA-256, SHA-384, or SHA-512. The object identifiers are taken from [ECCADD].

The algorithm identifier for ECDSA with SHA-224 signature values is:

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

The algorithm identifier for ECDSA with SHA-256 signature values is:

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

The algorithm identifier for ECDSA with SHA-384 signature values is:

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

The algorithm identifier for ECDSA with SHA-512 signature values is:

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

When any of these four object identifiers appears within an AlgorithmIdentifier, the parameters field MUST be absent. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component: the OID ecdsa-with-SHA224, ecdsa-with-SHA256, ecdsa-with-SHA384 or ecdsa-with-SHA512.

The SMIMECapabilities attribute value indicates support for one of the ECDSA signature algorithms in a SEQUENCE with the capabilityID field containing the object identifier ecdsa-with-SHA1* (where * is 224, 256, 384, or 512) with absent parameters. The DER encoding for these SMIMECapability values are:

```
   ecdsa-with-SHA224: 30 0a 06 08 2a 86 48 ce 3d 04 03 01

   ecdsa-with-SHA256: 30 0a 06 08 2a 86 48 ce 3d 04 03 02

   ecdsa-with-SHA384: 30 0a 06 08 2a 86 48 ce 3d 04 03 03

   ecdsa-with-SHA512: 30 0a 06 08 2a 86 48 ce 3d 04 03 04
```

4. Security Considerations

The security considerations in [RFC3370], [RFC3874], [RFC4055], [RFCTBD2], and [ECCADD] apply. No new security considerations are introduced as a result of this specification.

5. IANA Considerations

None: All identifiers are already registered. Please remove this section prior to publication as an RFC.

6. References

6.1. Normative References

[ECCADD] Dang, S., Santesson, S., Moriarty, K., and Brown, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", draft-ietf-pkix-sha2-dsa-ecdsa-05.txt (work-in-progress).

[DSS] National Institute of Standards and Technology (NIST), FIPS Publication 186-3: Digital Signature Standard, (draft) November 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119. March 1997.

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.

[RFC3447] Kaliski, B. and J. Jonsson "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[RFC3852] Housley, R., "The Cryptographic Message Syntax (CMS)", RFC 3852. July 2004.

[RFC3874] Housley, R., "A 224-bit One Way Hash Function: SHA-224", RFC 3874. September 2004.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055. June 2005.

[RFCTBD1] Ramsdell, B., and S. Turner, "S/MIME Version 3.2 Message Specification", draft-ietf-smime-3851bis-08.txt, work-in-progress.

//* RFC EDITOR: Note replace the above TBD1 with the RFC # for draft-ietf-smime-3851bis-08.txt. *//

[RFCTBD2] Turner, S., and D. Brown, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", draft-ietf-smime-3278bis-05, work-in-progress.

//* RFC EDITOR: Note replace the above TBD2 with the RFC # for draft-ietf-smime-3278bis-05.txt. *//

[SHS] National Institute of Standards and Technology (NIST), FIPS Publication 180-3: Secure Hash Standard, October 2008.

6.2. Informative References

[RFC4231] Nystrom, A. "Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512", RFC4231. December 2005.

[RFC4634] Eastlake, D., and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006.

Author's Addresses

Sean Turner

IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

EMail: turners@ieca.com