S/MIME WG
Internet Draft
Intended Status: Information
Updates: 3278 (once approved)
Expires: September 31, 2008

Sean Turner, IECA
March 31, 2008

Update to Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS) draft-ietf-smime-rfc3278-update-01.txt

Status of this Memo

Copyright Notice

Abstract

RFC 3278 describes how to use Elliptic Curve Cryptography (ECC) public-key algorithms in the Cryptographic Message Syntax (CMS). This document updates RFC 3278 to add support for the SHA2 family of hash algorithms.

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 [MUST].

Discussion

This draft is being discussed on the 'ietf-smime' mailing list. To subscribe, send a message to ietf-smime-request@imc.org with the single word subscribe in the body of the message. There is a Web site for the mailing list at <http://www.imc.org/ietf-smime/>.

Table of Contents

1. Introduction

RFC 3278 describes how to use Elliptic Curve Cryptography (ECC) public-key algorithms in the Cryptographic Message Syntax (CMS). This document updates RFC 3278 to add support for the SHA2 family of hash algorithms.

The following summarizes the changes:

- Paragraph 2.1.1 limited the digest algorithm to SHA-1. This document expands the allowed algorithms to SHA-224, SHA-256, SHA-384, and SHA-512.

- Paragraph 5 added requirements for hash algorithms and recommendations for matching curves and hash algorithms.

- Paragraph 7 added S/MIME capabilities for ECDSA with SHA-224, SHA-256, SHA-384, and SHA-512.

- Paragraph 8.1 listed the algorithm identifiers for SHA-1 and SHA-1 with ECDSA. This document adds algorithms for SHA-224, SHA-256, SHA-384, and SHA-512 and SHA-224, SHA-256, SHA-384, and SHA-512 with ECDSA.

- Paragraph 9 references need to be updated.

- Security considerations paragraph referring to definitions of SHA-224, SHA-256, SHA-384, and SHA-512 needs to be deleted.

2. Updates to Paragraph 2.1.1

Old:

> digestAlgorithm MUST contain the algorithm identifier sha-1 (see Section 8.1) which identifies the SHA-1 hash algorithm.
>
> signatureAlgorithm contains the algorithm identifier ecdsa-with-SHA1 (see Section 8.1) which identifies the ECDSA signature algorithm.

New:

> digestAlgorithm MUST contain the algorithm identifier of the hash algorithm (see Section 8.1): id-sha1 identifies the SHA-1 hash algorithm, id-sha224 identifies the SHA-224 hash algorithm, id-sha256 identifies the SHA-256 hash algorithm, id-sha384 identifies the SHA-384 algorithm, and id-sha512 identifies the SHA-512 algorithm.
>
> signatureAlgorithm contains the signature algorithm identifier (see Section 8.1): ecdsa-with-SHA1, ecdsa-with-SHA224, ecdsa-with-SHA256, ecdsa-with-SHA384, ecdsa-with-SHA512.

3. Updates to Paragraph 5

Add the following to the end of the section:

Implementations of this specification MUST implement the SHA-256 hash algorithm. The SHA-1, SHA-224, SHA-384, SHA-512 hash algorithms MAY be supported.

When ECDSA is used, it is RECOMMENDED that the P-256 curve be used with SHA-256, the P-384 curve be used with SHA-384, and the P-521 curve be used with SHA-512.

4. Updates to Paragraph 7

Old:

The SMIMECapability value to indicate support for the ECDSA signature algorithm is the SEQUENCE with the capabilityID field containing the object identifier ecdsa-with-SHA1 with NULL parameters. The DER encoding is:

```
30 0b 06 07 2a 86 48 ce 3d 04 01 05 00
```

New:

The SMIMECapability value to indicate support for the ECDSA signature algorithm is the SEQUENCE with the capabilityID field containing the object identifiers ecdsa-with-SHA1, ecdsa-with-withSHA224, ecdsa-with-SHA256, ecdsa-with-SHA384, and ecdsa-with-SHA512 all with NULL parameters. The DER encodings are:

```
ecdsa-with-SHA1:   30 0b 06 07 2a 86 48 ce 3d 04 01 05 00

ecdsa-with-SHA224: 30 0c 06 08 2a 86 48 ce 3d 04 03 01 05 00

ecdsa-with-SHA256: 30 0c 06 08 2a 86 48 ce 3d 04 03 02 05 00

ecdsa-with-SHA384: 30 0c 06 08 2a 86 48 ce 3d 04 03 03 05 00

ecdsa-with-SHA512: 30 0c 06 08 2a 86 48 ce 3d 04 03 04 05 00
```

5. Updates to Paragraph 8.1

Old:

The algorithm identifiers used in this document are taken from [X9.62], [SEC1] and [SEC2].

The following object identifier indicates the hash algorithm used in this document:

```
sha-1 OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
oiw(14) secsig(3) algorithm(2) 26 }
```

New:

The algorithm identifiers used in this document are taken from [SMIME-SHA2]

The following object identifier indicates the hash algorithm used in this document:

```
id-sha1 OBJECT IDENTIFIER ::= { iso(1) identified-
organization(3) oiw(14) secsig(3) algorithm(2) 26 }

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 }
```

Old:

The following object identifier indicates the digital signature algorithm used in this document:

```
ecdsa-with-SHA1 OBJECT IDENTIFIER ::= { ansi-x9-62
```

```
      signatures(4) 1 }
```

New:

The following object identifier indicates the digital signature algorithm used in this document:

```
      ecdsa-with-SHA1 OBJECT IDENTIFIER ::= { ansi-x9-62
      signatures(4) 1 }

      ecdsa-with-SHA224 OBJECT IDENTIFIER ::= { ansi-x9-62
      signatures(4) ecdsa-with-SHA2(3) 1 }

      ecdsa-with-SHA256 OBJECT IDENTIFIER ::= { ansi-x9-62
      signatures(4) ecdsa-with-SHA2(3) 2 }

      ecdsa-with-SHA384 OBJECT IDENTIFIER ::= { ansi-x9-62
      signatures(4) ecdsa-with-SHA2(3) 3 }
```



```
      ecdsa-with-SHA512 OBJECT IDENTIFIER ::= { ansi-x9-62
      signatures(4) ecdsa-with-SHA2(3) 4 }
```

6. Updates to Paragraph 9

Add the following reference:

[SMIME-SHA2] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", work-in-progress.

Update the following references:

Old:

[PKI-ALG] Bassham, L., Housley R. and W. Polk, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 3279, April 2002.

[FIPS-180] FIPS 180-1, "Secure Hash Standard", National Institute of Standards and Technology, April 17, 1995.

New:

[PKI-ALG] Turner, S., Brown, D., Yiu, K., Housley, R., and W. Polk," Elliptic Curve Cryptography Subject Public Key Information", work-in-progress.

[FIPS] FIPS 180-2, "Secure Hash Standard", National Institute of Standards and Technology, August 1, 2002.

7. Changes to Security Considerations

Delete the following:

> When 256, 384, and 512 bit hash functions succeed SHA-1 in future revisions of [FIPS], [FIPS-186-2], [X9.62] and [SEC1], then they can similarly succeed SHA-1 in a future revision of this document.

8. Security Considerations

No new security considerations to those already specified in [RFC3278], [SMIME-SHA2], and [PKI-ALG].

9. IANA Considerations

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

10. References

10.1. Normative References

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

[PKI-ALG] Turner, S., Brown, D., Yiu, K., Housley, R., and W. Polk, "Elliptic Curve Cryptography Subject Public Key Information", work-in-progress.

[SMIME-SHA2] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", work-in-progress.

[RFC3278] Blake-Wilson, S., Brown, D., and P. Lambert, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 3278, April 2002.

10.2. Informative References

None.

Author's Addresses

Sean Turner

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

Email: turners@ieca.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).