PKIX Working Group
Internet Draft
Intended Category:Standards Track
Expires: December 2008

Q. Dang (NIST)
S. Santesson (Microsoft)
K. Moriarty (RSA)
D. Brown (Certicom Corp.)
T. Polk (NIST)

June 19, 2008

Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA

<draft-ietf-pkix-sha2-dsa-ecdsa-04.txt>

Status of this Memo

Copyright Notice

Abstract

This document supplements RFC 3279. It specifies algorithm identifiers and ASN.1 encoding rules for the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) digital signatures when using SHA-224, SHA-256, SHA-384 or SHA-512 as hashing algorithm. This specification applies to the Internet X.509 Public Key Infrastructure (PKI) when digital signatures are used to sign certificates and certificate revocation lists (CRLs).

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 [RFC 2119].

Table of Contents

1. Introduction

This specification supplements [RFC 3279], "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" and extends the list of algorithms defined for use in the Internet PKI. This document specifies algorithm identifiers and ASN.1 [X.660] encoding rules for DSA and ECDSA digital signatures in certificates and CRLs when using one of the SHA-2 hash algorithms (SHA-224, SHA-256, SHA-384, and SHA-512) as the hash algorithm.

This specification defines the contents of the signatureAlgorithm, signatureValue and signature fields within Internet X.509 certificates and CRLs when these objects are signed using DSA or ECDSA with a SHA-2 hash algorithm. These fields are more fully described in [RFC 5280].

This document profiles material presented in the "Secure Hash Standard" [FIPS 180-3], "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Standard (ECDSA)" [X9.62], and the "Digital Signature Standard" [FIPS 186-3].

Algorithm identifiers and encoding rules for RSA, DSA and ECDSA when used with SHA-1 are specified in [RFC 3279]. Algorithm identifiers and encoding rules for RSA when used with SHA-2 are specified in [RFC 4055].

2. One-way Hash Functions

This section identifies four additional hash algorithms for use with DSA and ECDSA in the Internet X.509 certificate and CRL profile [RFC 5280].

SHA-224, SHA-256, SHA-384, and SHA-512 produce a 224-bit, 256-bit, 384-bit and 512-bit "hash" of the input respectively and are fully described in the Federal Information Processing Standard 180-3 [FIPS 180-3].

The listed one-way hash functions 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 }
```

When one of these OIDs appears in an AlgorithmIdentifier, all implementations MUST accept both NULL and absent parameters as legal and equivalent encodings.

3. Signature Algorithm

Certificates and CRLs conforming to [RFC 5280] may be signed with any public key signature algorithm. The certificate or CRL indicates the algorithm through an identifier, which appears in the signatureAlgorithm field within the Certificate or CertificateList. This algorithm identifier is an OID and has optionally associated parameters. This section denotes algorithm identifiers and parameters that MUST be used in the signatureAlgorithm field in a Certificate or CertificateList.

Signature algorithms are always used in conjunction with a one-way hash function. This section identifies OIDs for DSA and ECDSA with SHA-224, SHA-256, SHA-384, and SHA-512. The contents of the parameters component for each signature algorithm vary; details are provided for each algorithm.

The data to be signed (e.g., the one-way hash function output value) is formatted for the signature algorithm to

be used. Then, a private key operation (e.g., DSA encryption) is performed to generate the signature value. This signature value is then ASN.1 encoded as a BIT STRING and included in the Certificate or CertificateList in the signature field. More detail on how digital signatures are generated can be found in [FIPS 186-3].

Entities that validate DSA signatures MUST support SHA-224 and SHA-256. Entities that validate ECDSA signatures MUST support SHA-224 and SHA-256 and should support SHA-384 and SHA-512.

3.1. DSA Signature Algorithm

The DSA is defined in the Digital Signature Standard (DSS) [FIPS 186-3]. DSA was developed by the U.S. Government, and can be used in conjunction with a SHA2 one-way hash function such as SHA-224 or SHA-256. DSA is fully described in [FIPS 186-3].

[FIPS 186-3] specifies four size-choices for a DSA key pair of the form (public key size, private key size) in bits. The four choices are (1024, 160), (2048, 224), (2048, 256), and (3072, 256). More information can be found in [FIPS 186-3].

DSA key pairs of the sizes (1024, 160) and (2048, 224) may be used with SHA-224. DSA key pairs of all the four sizes may use SHA-256. The following are the OIDs of the DSA digital signature algorithm when used with SHA-224 or SHA-256.

When SHA-224 is used, the OID is:

```
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 }
```

When SHA-256 is used, the OID is:

```
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 }
```

The(3072, 256) DSA key pair provides 128 bits of security and provides the most security among all the four sizes of DSA key pairs. More information on security strength

assessments of DSA and other cryptographic algorithms can be found in [SP 800-57]. A digital signature algorithm has the same security strength as its asymmetric key algorithm like DSA or ECDSA only if its hashing algorithm has at least the same security strength as the asymmetric key algorithm. Therefore, a 128-bit security strength hashing algorithm which is SHA-256 will be sufficient to build a 128-bit security strength DSA digital signature algorithm when a DSA key pair of the size (3072, 256) is used. Therefore, it is only needed to specify DSA with SHA-224 and SHA-256 because SHA-256 provides sufficient security for using with any DSA key pair of any of the four size choices. More information on security strengths of the hash functions SHAs specified in [FIPS 180-3] and the digital signature algorithms specified in [FIPS 186-3] can be found in [SP 800-107] and [SP 800-57].

When the id-dsa-with-sha224 or id-dsa-with-sha256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding SHALL omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component, the OID id-dsa-with-sha224 or id-dsa-with-sha256.

Encoding rules for DSA signature values are specified in [RFC 3279]. For completeness, this information is repeated below:

When signing, the DSA algorithm generates two values commonly referred to as r and s. To easily transfer these two values as one signature, they SHALL be ASN.1 encoded using the following ASN.1 structure:

```
Dss-Sig-Value  ::=  SEQUENCE  {
        r       INTEGER,
        s       INTEGER  }
```

The DSA parameters in the subjectPublicKeyInfo field of the certificate of the issuer SHALL apply to the verification of the signature.

3.2. ECDSA Signature Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) is defined in, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Standard (ECDSA)" [X9.62]. [X9.62] provides alternative mechanisms for specifying the hash algorithm used in the

signature generation process. Three methods are specified in that document.

1) The signature OID may explicitly identify the hash algorithm, as specified in Section 3.2.1 below.

2) The signature OID may specify that the signer used the recommended hash algorithm for a given key size, as described in Section 3.2.2. A verifier infers from the size of the public key which hash algorithm was used.

3) The signature OID may indicate that the hash algorithm is specified as a parameter of the signature OID. The verifier identifies the appropriate hash algorithm according to the hash algorithm OID in the parameters field. The hash algorithm must be one of the SHA-2 hash algorithms.

Conforming CA implementations MUST specify the hash algorithm explicitly, using the OIDs specified in Section 3.2.1, when encoding ECDSA/SHA-2 signatures in certificates and CRLs.

Conforming client implementations that process ECDSA signatures with any of the SHA-2 hash algorithms when processing certificates and CRLs MUST recognize the corresponding OIDs specified in Section 3.2.1. Conforming client implementations MAY also recognize the signature OIDs specified in Sections 3.2.2 and 3.2.3.

Encoding rules for ECDSA signature values are specified in [RFC 3279]. For completeness, this information is repeated below:

When signing, the ECDSA algorithm generates two values commonly referred to as r and s. To easily transfer these two values as one signature, they MUST be ASN.1 encoded using the following ASN.1 structure:

```
Ecdsa-Sig-Value  ::=  SEQUENCE  {
     r     INTEGER,
     s     INTEGER  }
```

The elliptic curve parameters in the subjectPublicKeyInfo field of the certificate of the issuer MUST be applied to the verification of the signature. The subjectPublicKeyInfo field must be compliant with

requirements for Subject Public Key Information field in [Elliptic Curve].

3.2.1. ECDSA with SHA-2 Hash Algorithms

The ASN.1 OIDs used to specify that an ECDSA signature was generated using SHA224, SHA256, SHA384 or SHA 512 respectively:

```
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 }
```

When the ecdsa-with-SHA224, ecdsa-with-SHA256, ecdsa-with-SHA384 or ecdsa-with-SHA512 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding MUST omit the parameters field. 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.

3.2.2. ECDSA with Recommended Hash Algorithm

The following object identifier identifies the hash function to be used for message digesting implicitly, based on the size of the signer's public key:

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

The recommended hash functions are given in [X9.62], and are determined as follows. Among the hash functions SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, the recommended one has the largest bit size that does not require bit truncation during the signing process. Bit truncation occurs when the hash output bit-length is greater than the bit length of n, the order of the base point G. (Note: even if bit truncation does not occur, modular reduction can occur.)

Conforming CA implementations MUST NOT specify the ecdsa-with-Recommended OID when encoding certificates and CRLs. To maximize interoperability, conforming client implementations MAY recognize the ecdsa-with-Recommended OID when processing certificates and CRLs.

3.2.3. ECDSA with Specified Hash Algorithm

The following object identifier identifies the hash function to be used for message digesting is the one specified in the parameters field of the algorithm identifier:

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

For signatures generated using one of the SHA-2 hash algorithms, the parameters field would contain the appropriate OID from Section 2.

Conforming CA implementations MUST NOT specify the ecdsa-with-Specified OID when encoding certificates and CRLs. To maximize interoperability, conforming client implementations MAY recognize the ecdsa-with-Specified OID when processing certificates and CRLs.

4. ASN.1 Module

```
DEFINITIONS EXPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL --
-- All types and values defined in this module are
-- exported for use in other ASN.1 modules.

IMPORTS

NONE
```

```
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 }

--
--   ECDSA Signatures with SHA-2 Hashes, from X9.62
--

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 }

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

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

--
--   DSA with SHA-224 and SHA-256 signature algorithms
--
```

```
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 }

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 }

END -- Definitions
```

5. Security Considerations

This specification supplements [RFC 3279]. The Security Considerations section of that document applies, but is specific to the RSA algorithm and this document covers the DSA and ECDSA algorithms and the associated considerations.

The appropriate use of the hash functions in terms of the algorithm strengths and expected time frames for secure use as defined by NIST can be found in Special Publications 800-78-1 [SP 800-78-1], 800-57 [SP 800-57] and 800-107 [SP 800-107].

For security reasons, in [FIPS 186-3] NIST recommends three types of elliptic curves for use in conjunction with one of the described hash functions: curves over prime fields, curves over binary fields, and Koblitz curves (anomalous binary curves). FIPS 186-3 provides a table listing the uses and time periods for each algorithm and key size combinations for various applications. For further details, see the referenced document.

The one-way hash algorithms discussed in this document, SHA-224, SHA-256, SHA-384, and SHA-512 each have a recommended lifetime when used in combination with a digital signature algorithm. NIST provides information on the appropriate time periods for which each combination should be used based upon the security needs of the service and information being protected in NIST Special Publication 800-57. A table outlines the year in which NIST deems it is no longer safe to use specific combinations of key lengths and algorithms of various strengths for RSA, DSA, and ECDSA. NIST also provides Recommendation for using NIST-approved hash algorithms in the digital signature applications in [SP 800-107].

The Special Publication 800-57 also discusses the "best practices" for key management to be used by both developers and system administrators. The document covers the aspects of key management from algorithm selection and key sizes with associated key usage period to key usage (preventing key overlap), the compromise of keys and keying material, and key destruction. Specific guidelines are offered for key usage periods such as the lifetime of a private signature key may be shorter than the lifetime of the public verification key for practical applications. The specification also provides recommendations on the number of years various key types should be used such as public and private signature keys, public and private authentication keys, etc.

NIST Special Publication 800-78-1 also lists time frames for the use of combined hash algorithms and digital signature algorithms for specific key types, including the

- O Personal Identity Verification (PIV) authentication key,
- O Card authentication key,
- O Digital signature key, and
- O Key management key.

Specific requirements on the PIV can be found in [FIPS 201].

The recommendation for the size of digital signatures and key management keys is more restrictive than that of authentication keys, because they are used to protect data for longer periods of time. Therefore, the transition dates to larger key sizes are earlier in general.

Guidelines for the protection of domain parameters, initialization vectors (IVs), and per message secret numbers for use with digital signature algorithms, DSA and ECSDA are provided in [FIPS 186-3]. An assurance of integrity should be obtained prior to using all keying material for the generation of digital signatures using DSA and ECDSA. Recommendation for Obtaining Assurances for Digital Signature Applications can be found in [SP 800-89]. The purpose of this is to ensure the keying material is in the proper format, the domain parameters are valid, the possession of the private key, the validity of the public key, and that the request is coming from an authorized source.

Algorithm implementations MUST follow the appropriate specification to ensure the generation of secure keys. The SHA-2 algorithms are fully defined in [FIPS 180-3]. FIPS 186-3 defines the requirements for the digital signature standard specifying the requirements for both DSA and ECDSA. ECDSA is fully specified in [X9.62].

Certificate Authorities (CAs) that issue certificates using the DSA and ECDSA algorithms for key generation SHOULD adhere to the recommended security guidelines for key management in the NIST Special Publication 800-57. A CA should use the same size or greater hash function than what is used when generating keys for subscriber signature certificates.

6. References

6.1. Normative references:

[RFC 2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

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

[RFC 5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[X9.62] X9.62-2005, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Standard (ECDSA)", December, 2005.

[Elliptic Curve] Turner S., Brown D., Yiu K., Housley R., and Polk W., "Elliptic Curve Cryptography Subject Public Key Information" draft-ietf-pkix-ecc-subpubkeyinfo-05.txt (work in progress), April 2008.

[FIPS 180-3] Federal Information Processing Standards Publication (FIPS PUB) 180-3, Secure Hash Standard (SHS), (draft) June 2007.

[FIPS 186-3] Federal Information Processing Standards Publication (FIPS PUB) 186-3, Digital Signature Standard (DSS), (draft) March 2006.

6.2. Informative references

[SP 800-107] Q. Dang, NIST, "Recommendation for Applications Using Approved Hash Algorithm",(draft) July 2007.

[SP 800-78-1] W. Timothy Polk, Donna, F. Dodson, William E. Burr, NIST, "Cryptographic Standards and Key Sizes for Personal Identity Verification", August 2007.

[SP 800-57] Elaine Barker, William Barker, William E. Burr, NIST, "Recommendation for Key Management", August 2005.

[SP 800-89] Elaine Barker, NIST, "Recommendation for Obtaining Assurances for Digital Signature Applications", December 2006.

[RFC 4055] Schaad, J., Kaliski, B., and Housley, R., "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, June 2005.

[FIPS 201] Federal Information Processing Standards Publication (FIPS PUB) 201, Personal Identity Verification (PIV) of Federal Employees and Contractors, 25 February 2005.

7. Authors' Addresses

Quynh Dang
NIST
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20899-8930
USA
Email: quynh.dang@nist.gov

Stefan Santesson
Microsoft
Tuborg Boulevard 12
2900 Hellerup
Denmark
EMail: stefans@microsoft.com

Kathleen M. Moriarty
RSA, The Security Division of EMC
174 Middlesex Turnpike
Bedford, MA 01730
Email: kathleen.moriarty@rsa.com

Daniel R. L. Brown
Certicom Corp.
5520 Explorer Drive
Mississaug, ON L4W 5L1
Email: dbrown@certicom.com

Tim Polk
NIST
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20899-8930
USA
Email: tim.polk@nist.gov

8. IANA Considerations

This document has no actions for IANA.

9. Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

10. Copyright Statement

Expires December 2008.