Independent Submission   M. Jenkins
Request for Comments: 8603   L. Zieglar
Category: Informational   NSA
ISSN: 2070-1721   May 2019

Commercial National Security Algorithm (CNSA) Suite Certificate and Certificate Revocation List (CRL) Profile

Abstract

This document specifies a base profile for X.509 v3 Certificates and X.509 v2 Certificate Revocation Lists (CRLs) for use with the United States National Security Agency's Commercial National Security Algorithm (CNSA) Suite. The profile applies to the capabilities, configuration, and operation of all components of US National Security Systems that employ such X.509 certificates. US National Security Systems are described in NIST Special Publication 800-59. It is also appropriate for all other US Government systems that process high-value information. It is made publicly available for use by developers and operators of these and any other system deployments.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document specifies a base profile for X.509 v3 Certificates and X.509 v2 Certificate Revocation Lists (CRLs) for use by applications that support the United States National Security Agency's Commercial National Security Algorithm (CNSA) Suite [CNSA]. The profile applies to the capabilities, configuration, and operation of all components of US National Security Systems that employ such X.509 certificates. US National Security Systems are described in NIST Special Publication 800-59 [SP80059]. It is also appropriate for all other US Government systems that process high-value information. It is made publicly available for use by developers and operators of these and any other system deployments.

This document does not define any new cryptographic algorithm suite; instead, it defines a CNSA-compliant profile of "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [RFC5280]. It applies to all CNSA Suite solutions that make use of X.509 v3 Certificates or X.509 v2 CRLs. The reader is assumed to have familiarity with RFC 5280. All MUST-level requirements of RFC 5280 apply throughout this profile and are generally not repeated here. In cases where a MUST-level requirement is repeated for emphasis, the text notes the requirement is "in adherence with RFC 5280". This profile contains changes that elevate some SHOULD-level options in RFC 5280 to MUST-level and also contains changes that elevate some MAY-level options in RFC 5280 to SHOULD-level or MUST-level. All options from RFC 5280 that are not listed in this profile remain at the requirement level of RFC 5280.

The reader is also assumed to have familiarity with these documents:

- [RFC5480] for the syntax and semantics for the Subject Public Key Information field in certificates that support Elliptic Curve Cryptography,

- [RFC5758] for the algorithm identifiers for Elliptic Curve Digital Signature Algorithm (ECDSA),

- [RFC3279] for the syntax and semantics for the Subject Public Key Information field in certificates that support RSA Cryptography, and

- [RFC4055] for the algorithm identifiers for RSA Cryptography with the SHA-384 hash function.

2. The Commercial National Security Algorithm Suite

The National Security Agency (NSA) profiles commercial cryptographic algorithms and protocols as part of its mission to support secure, interoperable communications for US Government National Security Systems. To this end, it publishes guidance both to assist with transitioning the United States Government to new algorithms and to provide vendors, and the Internet community in general, with information concerning their proper use and configuration.

Recently, cryptographic transition plans have become overshadowed by the prospect of the development of a cryptographically relevant quantum computer. The NSA has established the Commercial National Security Algorithm (CNSA) Suite to provide vendors and IT users near-term flexibility in meeting their cybersecurity interoperability requirements. The purpose behind this flexibility is to avoid vendors and customers making two major transitions in a relatively short time frame, as we anticipate a need to shift to quantum-resistant cryptography in the near future.

The NSA is authoring a set of RFCs, including this one, to provide updated guidance concerning the use of certain commonly available commercial algorithms in IETF protocols. These RFCs can be used in conjunction with other RFCs and cryptographic guidance (e.g., NIST Special Publications) to properly protect Internet traffic and data-at-rest for US Government National Security Systems.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "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.

4. General Requirements and Assumptions

The goal of this document is to define a base set of requirements for certificates and CRLs to support interoperability among CNSA Suite solutions. Specific communities, such as those associated with US National Security Systems, may define community profiles that further restrict certificate and CRL contents by mandating the presence of extensions that are optional in this base profile, defining new optional or critical extension types, or restricting the values and/or presence of fields within existing extensions. However, communications between distinct communities MUST conform with the requirements specified in this document when interoperability is desired. Applications may add requirements for additional non-critical extensions, but they MUST NOT assume that a remote peer will be able to process them.

4.1. Implementing the CNSA Suite

Every CNSA Suite certificate MUST use the X.509 v3 format and contain one of the following:

- o An ECDSA-capable signature verification key using curve P-384, or
- o An ECDH-capable (Elliptic Curve Diffie-Hellman) key establishment key using curve P-384, or
- o An RSA-capable signature verification key using RSA-3072 or RSA-4096, or
- o An RSA-capable key transport key using RSA-3072 or RSA-4096.

The signature applied to all CNSA Suite certificates and CRLs MUST be made with a signing key that is either generated on the curve P-384, or is an RSA-3072 or RSA-4096 key. The SHA-384 hashing algorithm MUST be used for all certificate and CRL signatures irrespective of the type of key used.

The RSA exponent "e" MUST satisfy $2^{16}<e<2^{256}$ and be odd per [FIPS186].

The requirements of this document are not intended to preclude use of RSASSA-PSS signatures. However, Certification Authorities (CAs) conforming with this document will not issue certificates specifying that algorithm for subject public keys. Protocols that use RSASSA-PSS should be configured to use certificates that specify rsaEncryption as the subject public key algorithm. Protocols that use these keys with RSASSA-PSS signatures must use the following parameters: the hash algorithm (used for both mask generation and signature generation) must be SHA-384, the mask generation function 1 from [RFC8017] must be used, and the salt length must be 48 octets.

4.2. CNSA Suite Object Identifiers

4.2.1. CNSA Suite Object Identifiers for ECDSA

The primary Object Identifier (OID) structure for the CNSA Suite is as follows per [X962], [SEC2], [RFC5480], and [RFC5758].

```
ansi-X9-62 OBJECT IDENTIFIER ::= {
   iso(1) member-body(2) us(840) 10045 }

certicom-arc OBJECT IDENTIFIER ::= {
   iso(1) identified-organization(3) certicom(132) }

id-ecPublicKey OBJECT IDENTIFIER ::= {
   ansi-X9-62 keyType(2) 1 }

secp384r1 OBJECT IDENTIFIER ::= {
   certicom-arc curve(0) 34 }

id-ecSigType OBJECT IDENTIFIER ::= {
   ansi-X9-62 signatures(4) }

ecdsa-with-SHA384 OBJECT IDENTIFIER ::= {
   id-ecSigType ecdsa-with-SHA2(3) 3 }
```

4.2.2. CNSA Suite Object Identifiers for RSA

The primary OID structure for CNSA Suite is as follows per [RFC3279].

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

rsaEncryption OBJECT IDENTIFIER ::= {
   pkcs-1 1}
```

The rsaEncryption OID is intended to be used in the algorithm field of a value of type AlgorithmIdentifier. The parameters field MUST have ASN.1 type NULL for this algorithm identifier.

The object identifier used to identify the PKCS #1 version 1.5 signature algorithm with SHA-384 is per [RFC4055]:

```
sha384WithRSAEncryption  OBJECT IDENTIFIER  ::=  {
   pkcs-1 12 }
```

5. CNSA Suite Base Certificate Required Values

This section specifies changes to the basic requirements in [RFC5280] for applications that create or use CNSA Suite certificates. Note that RFC 5280 has varying mandates for marking extensions as critical or non-critical. This profile changes some of those mandates for extensions that are included in CNSA Suite certificates.

5.1. signatureAlgorithm

5.1.1. ECDSA

For ECDSA, the algorithm identifier used by the CNSA Suite is as described in [RFC5758] and [X962]:

1.2.840.10045.4.3.3 for ecdsa-with-SHA384

The parameters MUST be absent as per [RFC5758].

5.1.2. RSA

For RSA, the algorithm identifier used by the CNSA Suite is as described in [RFC4055]:

1.2.840.113549.1.1.12 for sha384WithRSAEncryption.

Per [RFC4055], the parameters MUST be NULL. Implementations MUST accept the parameters being absent as well as present.

5.2. signatureValue

5.2.1. ECDSA

ECDSA digital signature generation is described in [FIPS186]. An ECDSA signature value is composed of two unsigned integers, denoted as "r" and "s". "r" and "s" MUST be represented as ASN.1 INTEGERs. If the high-order bit of the unsigned integer is a 1, an octet with the value 0x00 MUST be prepended to the binary representation before encoding it as an ASN.1 INTEGER. Unsigned integers for the P-384 curves can be a maximum of 48 bytes. Therefore, converting each "r" and "s" to an ASN.1 INTEGER will result in a maximum of 49 bytes for the P-384 curve.

The ECDSA signatureValue in an X.509 certificate is encoded as a BIT STRING value of a DER-encoded SEQUENCE of the two INTEGERS.

5.2.2. RSA

The RSA signature generation process and the encoding of the result is RSASSA-PKCS1-v1_5 as described in detail in PKCS #1 version 2.2 [RFC8017].

5.3. Version

For this profile, Version MUST be v3, which means the value MUST be set to 2.

5.4. SubjectPublicKeyInfo

5.4.1. Elliptic Curve Cryptography

For ECDSA signature verification keys and ECDH key agreement keys, the algorithm ID id-ecPublicKey MUST be used.

The parameters of the AlgorithmIdentifier in this field MUST use the namedCurve option. The specifiedCurve and implicitCurve options described in [RFC5480] MUST NOT be used. The namedCurve MUST be the OID for secp384r1 (curve P-384) [RFC5480].

The elliptic curve public key, ECPoint, SHALL be the OCTET STRING representation of an elliptic curve point following the conversion routine in Section 2.2 of [RFC5480] and Sections 2.3.1 and 2.3.2 of [SEC1].

CNSA Suite implementations MAY use either the uncompressed form or the compressed form of the elliptic curve point [RFC5480]. For interoperability purposes, all relying parties MUST be prepared to process the uncompressed form.

The elliptic curve public key (an ECPoint that is an OCTET STRING) is mapped to a subjectPublicKey (a BIT STRING) as follows: the most significant bit of the OCTET STRING becomes the most significant bit of the BIT STRING, and the least significant bit of the OCTET STRING becomes the least significant bit of the BIT STRING [RFC5480].

5.4.2. RSA

For RSA signature verification keys and key transport keys, the algorithm ID, rsaEncryption, MUST be used.

The parameters field MUST have ASN.1 type NULL for this algorithm identifier [RFC3279].

The RSA public key MUST be encoded using the ASN.1 type RSAPublicKey per Section 2.3.1 of [RFC3279].

6. Certificate Extensions for Particular Types of Certificates

Different types of certificates in this profile have different required and recommended extensions. Those are listed in this section. Those extensions from RFC 5280 not explicitly listed in this profile remain at the requirement levels of RFC 5280.

6.1. CNSA Suite Self-Signed CA Certificates

In adherence with [RFC5280], self-signed CA certificates in this profile MUST contain the subjectKeyIdentifier, keyUsage, and basicConstraints extensions.

The keyUsage extension MUST be marked as critical. The keyCertSign and cRLSign bits MUST be set. The digitalSignature and nonRepudiation bits MAY be set. All other bits MUST NOT be set.

In adherence with [RFC5280], the basicConstraints extension MUST be marked as critical. The cA boolean MUST be set to indicate that the subject is a CA, and the pathLenConstraint MUST NOT be present.

6.2. CNSA Suite Non-Self-Signed CA Certificates

Non-self-signed CA Certificates in this profile MUST contain the authorityKeyIdentifier, keyUsage, and basicConstraints extensions. If there is a policy to be asserted, then the certificatePolicies extension MUST be included.

The keyUsage extension MUST be marked as critical. The keyCertSign and CRLSign bits MUST be set. The digitalSignature and nonRepudiation bits MAY be set. All other bits MUST NOT be set.

In adherence with [RFC5280], the basicConstraints extension MUST be marked as critical. The cA boolean MUST be set to indicate that the subject is a CA, and the pathLenConstraint subfield is OPTIONAL.

If a policy is asserted, the certificatePolicies extension MUST be marked as non-critical, MUST contain the OIDs for the applicable certificate policies, and SHOULD NOT use the policyQualifiers option. If a policy is not asserted, the certificatePolicies extension MUST be omitted.

Relying party applications conforming to this profile MUST be prepared to process the policyMappings, policyConstraints, and inhibitAnyPolicy extensions, regardless of criticality, following the guidance in [RFC5280] when they appear in non-self-signed CA certificates.

6.3. CNSA Suite End-Entity Signature and Key Establishment Certificates

In adherence with [RFC5280], end-entity certificates in this profile MUST contain the authorityKeyIdentifier and keyUsage extensions. If there is a policy to be asserted, then the certificatePolicies extension MUST be included. End-entity certificates SHOULD contain the subjectKeyIdentifier extension.

The keyUsage extension MUST be marked as critical.

For end-entity digital signature certificates, the keyUsage extension MUST be set for digitalSignature. The nonRepudiation bit MAY be set. All other bits in the keyUsage extension MUST NOT be set.

For end-entity key establishment certificates, in ECDH certificates, the keyUsage extension MUST be set for keyAgreement; in RSA certificates, the keyUsage extension MUST be set for keyEncipherment. The encipherOnly or decipherOnly bit MAY be set. All other bits in the keyUsage extension MUST NOT be set.

If a policy is asserted, the certificatePolicies extension MUST be marked as non-critical, MUST contain the OIDs for the applicable certificate policies, and SHOULD NOT use the policyQualifiers option. If a policy is not asserted, the certificatePolicies extension MUST be omitted.

7. CNSA Suite CRL Requirements

This CNSA Suite CRL profile is a profile of [RFC5280]. There are changes in the requirements from [RFC5280] for the signatures on CRLs of this profile.

The signatures on CRLs in this profile MUST follow the same rules from this profile that apply to signatures in the certificates. See Section 4.

8. Security Considerations

The security considerations in [RFC3279], [RFC4055], [RFC5280], [RFC5480], [RFC5758], and [RFC8017] apply.

A single key pair SHOULD NOT be used for both signature and key establishment per [SP80057].

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[CNSA] Committee for National Security Systems, "Use of Public Standards for Secure Information Sharing", CNSSP 15, October 2016, <https://www.cnss.gov/CNSS/Issuances/Policies.htm>.

[FIPS186] National Institute of Standards and Technology (NIST), "Digital Signature Standard (DSS)", FIPS PUB 186-4, DOI 10.6028/NIST.FIPS.186-4, July 2013, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>.

[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>.

[RFC3279] 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, DOI 10.17487/RFC3279, April 2002, <https://www.rfc-editor.org/info/rfc3279>.

[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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <https://www.rfc-editor.org/info/rfc5480>.

[RFC5758] Dang, Q., Santesson, S., Moriarty, K., Brown, D., and T. Polk, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", RFC 5758, DOI 10.17487/RFC5758, January 2010, <https://www.rfc-editor.org/info/rfc5758>.

[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>.

[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>.

[SEC1] Standards for Efficient Cryptography Group, "SEC1: Elliptic Curve Cryptography", May 2009, <https://www.secg.org/sec1-v2.pdf>.

10.2. Informative References

[SEC2] Standards for Efficient Cryptography Group, "SEC 2: Recommended Elliptic Curve Domain Parameters", January 2010, <https://www.secg.org/sec2-v2.pdf>.

[SP80057] National Institute of Standards and Technology, "Recommendation for Key Management - Part 1: General", NIST Special Publication 800-57 Revision 4, DOI 10.6028/NIST.SP.800-57pt1r4, January 2016, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf>.

[SP80059] National Institute of Standards and Technology, "Guideline for Identifying an Information System as a National Security System", NIST Special Publication 800-59, DOI 10.6028/NIST.SP.800-59, August 2003, <https://csrc.nist.gov/publications/detail/sp/800-59/final>.

[X962] American National Standards Institute, "Public Key Cryptography for the Financial Services Industry; The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI X9.62, November 2005.

Authors' Addresses

Michael Jenkins
National Security Agency

Email: mjjenki@nsa.gov

Lydia Zieglar
National Security Agency

Email: llziegl@tycho.ncsc.mil