Suite B Base Certificate and CRL Profile
27 May 2008

Executive Summary

This document specifies a base profile for X.509v3 Certificates and Certificate Revocation Lists (CRLs) for use by Transport Layer Security (TLS v1.2), IPsec Internet Key Exchange (IKE v1 and IKEv2), S/MIME, Cryptographic Message Syntax (CMS) and Secure Shell (SSH) implementations which support Suite B cryptography. This profile is also applicable to protocols under revision in the international standards environment such as XML Encryption or XML Signature which will incorporate Suite B cryptography. The format is applicable to the following Suite B certificate and CRL types:

- Root Certification Authority (CA) Self-Signed Certificate using P-256 signed with P-256
- Root CA Self-Signed Certificate using P-384 signed with P-384
- Subordinate CA Certificate using P-256 signed with P-256
- Subordinate CA Certificate using P-384 signed with P-384
- Subordinate CA Certificate using P-256 signed with P-384
- CA Cross-Certificate using P-256 signed with P-256
- CA Cross-Certificate using P-384 signed with P-384
- CA Cross-Certificate using P-256 signed with P-384
- End-Entity Signature Certificate using P-256 signed with P-256
- End-Entity Signature Certificate using P-384 signed with P-384
- End-Entity Signature Certificate using P-256 signed with P-384
- End Entity Key Establishment Certificate using P-256 signed with P-256
- End Entity Key Establishment Certificate using P-384 signed with P-384
- End Entity Key Establishment Certificate using P-256 signed with P-384
- Certificate Revocation List (CRL) signed with P-256
- Certificate Revocation List (CRL) signed with P-384

This document does not address implementation requirements. It does not address requirements on the use of specific protocols, equipment, or key-strength levels to protect information with specific sensitivity levels and/or in specific threat environments.

The goal of this document is to define a base set of certificate and CRL formats to support interoperability between distinct Suite B solutions. Specific communities, such the US National Security Systems, may define community profiles which further restrict certificate and CRL formats by mandating the presence of extensions which 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 use the formats specified in this document when interoperability is desired. (Applications may add additional non-critical extensions to these formats but they must not assume that a remote peer will be able to process them.)

In most applications, performance, scalability, and cost will require the infrastructure to delegate responsibility for generating end-entity key pairs to the nodes themselves. With the end-entities performing the key generation, the infrastructure is only required to support X.509v3 certificate issuance and signing functions. When necessary, key pairs may be generated centrally by the infrastructure. The infrastructure is responsible for revocation, protection of private keys if centrally generated, and other certificate management functions. In either key generation model, the format of the Suite B certificates shall remain as specified in this document.

Background

In 2006-2007, two papers were posted to the NSA Internet web site. The first paper[1] provided background and rationale, from a security and efficiency perspective, for moving to ECC. The second paper[2] provides a list of Suite B algorithms and the appropriate National Institutes of Standards and Technology (NIST) source documentation[3]. Since the original algorithm list was published, only EC Diffie-Hellman (ECDH) is considered the preferred Suite B compliant key establishment algorithm.

Implementing Suite B

Every Suite B certificate must use X.509v3 format, and contain either:

- an ECDSA-capable signing key, using group P-256 or P-384; or
- an ECDH-capable key establishment key, using group P-256 or P-384.

Every Suite B Certificate and CRL must be signed using ECDSA. The signing CA's key must be on the group P-256 or P-384 if the certificate contains a key on P-256. If the certificate contains a key on P-384, the signing CA's key must be on the group P-384. Any certificate must be hashed using SHA256 or SHA384, matched to the size of the signing CA's key.

Compliant Suite B implementations that use an ECDSA signing key or static ECDH key must be capable of issuing certificate requests and resolving certificate responses. The preferred protocol is RFC 2797bis, Certificate Management Messages over CMS[4]. The infrastructure supporting Suite B must be capable of issuing X.509v3 certificates and CRLs, signed with ECDSA with the appropriate P-256 or P-384 curve.

---

[1] The Case for Elliptic Curve Cryptography, www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm
[2] Fact Sheet NSA Suite B Cryptography www.nsa.gov/ia/industry/crypto_suite_b.cfm
[3] NIST SP80056A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised). csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf
[4] RFC 2797bis, "Certificate Management Messages over CMS (CMC), http://www.ietf.org/internet-drafts/draft-ietf-pkix-2797-bis-07.txt; CMC Compliance Requirements, http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmc-compl-05.txt; Transport Protocols, http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmc-trans-08.txt

Commercial Protocols

Transport Layer Security (TLS)[5]

TLS utilizes ECDHE (ephemeral) and ECDSA (with P-256 or P-384) or ECDH-ECDSA (with P-256 or P-384).

In the first case the ECDH key exchange is Ephemeral-Ephemeral (E-E) with the server supplying a certificate containing an ECDSA key, signed with ECDSA. At a minimum, ECDHE-ECDSA (with P-256) should be implemented. The IETF informational RFC4492 (see section 3, paragraph 1) disallows Static-Ephemeral (S-E) and it is likely that commercial client implementations will not support S-E DH. Suite B TLS implementations are ***required*** to follow this guidance from RFC4492[6].

In the second case RFC4492 allows the key exchange to be either E-S or S-S. However, Suite B TLS implementations explicitly require Ephemeral-Static. So the server needs two certificates: one with an ECDSA signing key and one with an ECDH key establishment key, both signed with ECDSA.[7]

In either case, if the client is to be authenticated it must acquire an X.509v3 certificate containing an ECDSA key, signed with ECDSA to be used in TLS exchanges. Note that the TLS protocol would indicate the server sent a request for a certificate of type "ECDSA_sign". Clients in a Suite B TLS exchange never require static key establishment keys.

Internet Key Exchange (IKE)[8]

IKE shared secrets always involve an ephemeral-ephemeral DH key exchange, so static ECDH keys are thus unnecessary for this protocol. Authentication via public key will require an ECDSA-capable certificate. These will be X.509v3 certificates containing a key from the Suite B elliptic curve P-256 or P-384, and signed with ECDSA.

S/MIME[9]

Suite B key establishment in S/MIME uses an ECDH key-establishment key and an ECDSA signing key, both signed using ECDSA. The ECDH involves Ephemeral-Static

[5] The IETF draft for Suite B in TLS is waiting for TLS v1.2 to be completed. https://datatracker.ietf.org/drafts/draft-rescorla-tls-suiteb replaced by www.ietf.org/internet-drafts/draft-ietf-tls-ecc-new-mac-04.txt

[6] Disallowing S-E does not preclude a client from sending the same DH public key every time: it just prevents the client from doing so *implicitly* by sending a DH contribution in a client certificate instead of a non-empty client_key_exchange message.

[7] NIST SP800-56A, Section 5.6.4.2, disallows the use of a single key for both digital signatures and key establishment.

[8] RFC 4869 "Suite B Cryptographic Suites for IPsec" available at http://www.ietf.org/rfc/rfc4869.txt

[9] RFC 5008 "Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME available at http://www.ietf.org/rfc/rfc5008.txt.

(E-S) mode using P-256 or P-384 curves. Any message peer in a Suite B compliant S/MIME environment must have an ECDH X.509v3 certificate, signed with ECDSA.

Some current RSA-based implementations of S/MIME certificates contain a key with key usage extension bits set for both key establishment and signing. However, Suite B implementations are not permitted to use a single certificate for multiple key usages. [10]

Secure Shell (SSH)

In the SSH environment, ECDH is always E-E, with the server authentication to the client using ECDSA. Therefore the only certificate required to support SSH contains an ECDSA key, signed with ECDSA. The Internet draft for SSH use of Suite B is expected to be completed in mid-2008, as there are dependencies on other IETF documents.

[10] NIST SP800-56A, Section 5.6.4.2, disallows the use of a single key for both digital signatures and key establishment.

Annex A: Suite B X.509 Certificate and Certificate Revocation List (CRL) Profile

RFC 5280 describes the basic structure of an X.509 version 3 certificate and CRL and provides a profile to facilitate the use of X.509 certificates and CRLs within the Internet community for various applications such as WWW (TLS), electronic email (S/MIME) and IPSec.

This Suite B certificate and CRL profile complements RFC 5280. If a specific program needs to implement a subset of the Suite B certificate and CRL profile, the program should tailor its X.509 certificate and CRL profile using the parameters stipulated in this document together with the parameters stipulated in RFC 5280. Parameters stipulated in this document should take precedence. When "no specific requirements" is stated for a particular field or extension in this profile, then no specific requirements apply except for those stated by RFC 5280. In case of discrepancies between the present profile and RFC 5280, the present document is the normative one for Suite B.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in the present document are to be interpreted as described in RFC2119 [RFC2119].

X.509 Certificate Description

The basic X.509 certificate can be displayed as follows [RFC 5280], Section 4.1:

```
Certificate  ::=  SEQUENCE  {
        tbsCertificate       TBSCertificate,
        signatureAlgorithm   AlgorithmIdentifier,
        signatureValue       BIT STRING  }

   TBSCertificate  ::=  SEQUENCE  {
        version         [0]  EXPLICIT Version DEFAULT v1,
        serialNumber         CertificateSerialNumber,
        signature            AlgorithmIdentifier,
        issuer               Name,
        validity             Validity,
        subject              Name,
        subjectPublicKeyInfo SubjectPublicKeyInfo,
        issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        extensions      [3]  EXPLICIT Extensions OPTIONAL
                             -- If present, version MUST be v3
        }
```

Certificate is REQUIRED and must include the three fields “tbsCertificate”, “signatureAlgorithm” and “signatureValue”. The support requirements for the subfields of the tbsCertificate follow.

Fields of tbsCertificate:

Version is REQUIRED and MUST be set to 2 to denote X.509 version 3 public key certificates [RFC 5280], Section 4.1.2.1.

SerialNumber MUST follow RFC 5280, Section 4.1.2.2.

Signature is REQUIRED and contains an algorithm identifier to denote the algorithm used by the issuer to sign the certificate. The two algorithm identifiers used by Suite B are ecdsa-with-SHA256 and ecdsa-with-SHA384 [X9.62], Section E.8.

Issuer is REQUIRED and MUST follow RFC 5280, Section 4.1.2.4.

Validity is REQUIRED and MUST follow RFC 5280, Section 4.1.2.5.

Subject is REQUIRED and MUST follow RFC 5280, Section 4.1.2.6.

SubjectPublicKeyInfo, described in RFC 5280, Section 4.1.2.7, is REQUIRED and contains the EC public key and usually the algorithm with which the key is used. The following conditions apply.

- For ECDSA signing keys, the algorithm ID MUST be id-ecPublicKey, indicating unrestricted algorithm usage of the public key [RFC3279], Section 2.3.5.
- For ECDH key establishment keys, the algorithm ID, id-ecPublicKey, MUST be supported. The algorithm ID, id-ecDH, MAY be supported [DRAFT3279bis], Section 2.1.

- The intended application for the public key is indicated in the key usage extension that is described later.
- The parameters, [RFC3279], Section 2.3.5, of the AlgorithmIdentifier of the subjectPublicKeyInfo MUST use the namedCurve option; the ecParameters and implicitlyCA options MUST NOT be used [RFC3279], Section 2.3.5.
- The namedCurve MUST be either the OID for secp256r1(P-256 curve) or secp384r1(P-384 curve) [SEC2], Section A.2.1. The elliptic curve public key, ECPoint, SHALL be the octet string representation of an elliptic curve point following the conversion routine in [X9.63], Section 4.3.6, and [RFC3279], Section 2.3.5.
- Suite B implementations MUST support the uncompressed form of the elliptic curve point [X9.63], Section 4.3.6. Suite B certificates MAY support the compressed form of the elliptic curve point [X9.63], Section 4.3.6.
- The elliptic curve public key (an ECPoint which 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 [RFC3279], Section 2.3.5.

IssuerUniqueID and SubjectUniqueID MUST NOT be used in Suite B certificates [RFC 5280], Section 4.1.2.8.

The following extensions MUST be included in Suite B Root CA Self-Signed Certificates: subjectKeyIdentifier, keyUsage, basicConstraints [DRAFTRFC3280bis], Section 4.2.

- The subjectKeyIdentifier extension MUST be marked as non-critical. Its value will be computed following the guidance in RFC 5280, Section 4.2.1.2.
- The keyUsage extension MUST be marked as critical and MUST be set for keyCertSign and cRLSign [DRAFTRFC3280bis], Section 4.2.1.3.
- The basicConstraints extension MUST be marked as critical, the cA bit subfield MUST be set to indicate that the subject is a CA and the pathLenConstraint subfield MUST NOT be set [RFC 5280], Section 4.2.1.9.

The following extensions MUST be included in Suite B Subordinate CA Certificates: authorityKeyIdentifier, subjectKeyIdentifier, keyUsage, basicConstraints and certificatePolicies [DRAFTRFC3280bis], Section 4.2.

- The authorityKeyIdentifier extension MUST be marked as non-critical and MUST include the keyIdentifier field. The value of the keyIdentifier field will be computed following the guidance in RFC 5280, Section 4.2.1.1.
- The subjectKeyIdentifier extension MUST be marked as non-critical and its value will be computed following the guidance in [RFC 5280], Section 4.2.1.2.
- The keyUsage extension MUST be marked as critical and MUST be set for keyCertSign and cRLSign [RFC 5280], Section 4.2.1.3.
- The basicConstraints extension MUST be marked as critical, the cA bit subfield MUST be set to indicate that the subject is a CA and the pathLenConstraint subfield is OPTIONAL [RFC 5280], Section 4.2.1.9.

- The certificatePolicies extension MUST be marked as non-critical, MUST contain the OID for the applicable certificate policy and SHOULD NOT use the policyQualifiers option [RFC 5280], Section 4.2.1.4. Following the guidance in section 4.2.1.4 of RFC 5280, when a CA does not wish to limit the set of polices for certification paths that include this certificate, it MAY assert the special policy anyPolicy, with a value of {2 5 29 32 0}.

The following extensions MUST be included in Suite B CA Cross-Certificates: authorityKeyIdentifier, subjectKeyIdentifier, keyUsage, basicConstraints and certificatePolicies [DRAFTRFC3280bis], Section 4.2.

- The authorityKeyIdentifier extension MUST be marked as non-critical and MUST include the keyIdentifier field. The value of the keyIdentifier field will be computed following the guidance in RFC 5280, Section 4.2.1.1.
- The subjectKeyIdentifier extension MUST be marked as non-critical and its value will be computed following the guidance in [RFC 5280], Section 4.2.1.2.
- The keyUsage extension MUST be marked as critical and MUST be set for keyCertSign and cRLSign [RFC 5280], Section 4.2.1.3.
- The basicConstraints extension MUST be marked as critical, the cA bit subfield MUST be set to indicate that the subject is a CA and the pathLenConstraint subfield MUST NOT be set [RFC 5280], Section 4.2.1.9.
- The certificatePolicies extension MUST be marked as non-critical, MUST contain the OID for the applicable certificate policy and SHOULD NOT use the policyQualifiers option [RFC 5280], Section 4.2.1.4.

The following extensions are RECOMMENDED in CA Cross-Certificates: Policy Mappings, Policy Constraints and InhibitAnyPolicy [DRAFTRFC3280bis], Section 4.2.

- The policyMappings extension MUST NOT be marked as critical. Following the guidance in RFC 5280, Section 4.2.1.5, policies MUST NOT be mapped either to or from the special value anyPolicy.
- The policyConstraints extension MUST be marked as critical. The requireExplicitPolicy and inhibitPolicyMapping fields MUST be set to zero [RFC 5280], Section 4.2.1.11.
- The inhibitAnyPolicy extension MUST be marked as critical. SkipCerts MUST be set to zero [RFC 5280], Section 4.2.1.14.

The following extensions MUST be included in Suite B End Entity Signature and Key Establishment certificates: authorityKeyIdentifier, keyUsage and certificatePolicies [RFC 5280], Section 4.2.

- The authorityKeyIdentifier extension MUST be marked as non-critical and MUST include the keyIdentifier field. The value of the keyIdentifier field will be computed following the guidance in [RFC 5280], Section 4.2.1.1.
- The keyUsage extension MUST be marked as critical and MUST be set for digitalSignature for end-entity signature certificates or for keyAgreement for end entity key establishment certificates [RFC 5280], Section 4.2.1.3.
- The certificatePolicies extension MUST be marked as non-critical, MUST contain the OID for the applicable certificate policy and SHOULD NOT use the policyQualifiers option [RFC 5280], Section 4.2.1.4. Following the guidance in section 4.2.1.4 of

RFC 5280, the special policy, anyPolicy, with a value of {2 5 29 32 0} MAY be included in this certificate.

If the subject name is an empty sequence, then the subjectAltName extension MUST be added in Suite B End Entity Signature and Key Establishment Certificates and MUST be marked as critical [RFC 5280], Section 4.2.1.6. The subjectAltName extension is OPTIONAL otherwise and if included, MUST be marked as non-critical.

The following extension is RECOMMENDED in Suite B End Entity Signature and Key Establishment Certificates: subjectKeyIdentifier.

- The subjectKeyIdentifier extension MUST be marked as non-critical and its value will be computed following the guidance in [RFC 5280], Section 4.2.1.2.

All other extensions not described in this profile should be considered OPTIONAL; their inclusion or exclusion and their values will depend upon the particular application or profile incorporating this Suite B Certificate and CRL profile as a base.

Signature Algorithm and Value Fields of Certificate:

The signatureAlgorithm field contains the identifier for the cryptographic algorithm used by the CA to sign this certificate [DRAFTRFC3280bis], Section 4.1.1.2. The two algorithm identifiers used by Suite B are ecdsa-with-SHA256 and ecdsa-with-SHA384 [X9.62], Section E.8.

The signatureValue field, for Suite B certificates, contains a digital signature computed upon the ASN.1 DER encoded tbsCertificate [RFC 5280], Section 4.1.1.3. The ECDSA digital signature MUST be used for Suite B certificates [X9.62], Section 7. The ECDSA signature value is comprised of two unsigned integers, denoted *r* and *s*. ASN.1 encoding of INTEGER is used, however, to represent *r* and *s* in the signature value field. If the high order bit of the unsigned integer is a 1, a byte with value 0x00 must be prepended to the binary representation before encoding it as an ASN.1 INTEGER. Unsigned integers for the P-256 and P-384 curves can be a maximum of 32 and 48 bytes, respectively. Therefore converting to an ASN.1 integer will mean a maximum of 33 bytes for the P-256 curve and 49 bytes for the P-384 curve.
The ECDSA signature value is encoded as a BIT STRING value of a DER encoded SEQUENCE of the two INTEGERS and stored in the signatureValue field of the Certificate.

X.509 Certificate Revocation List (CRL)

The basic X.509 Certificate Revocation List (CRL) can be displayed as follows [DRAFTRFC3280bis], Section 5.1:

```
CertificateList  ::=  SEQUENCE  {
     tbsCertList          TBSCertList,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       BIT STRING  }

   TBSCertList  ::=  SEQUENCE  {
        version                 Version OPTIONAL;if present, MUST be v2
        signature               AlgorithmIdentifier,
        issuer                  Name,
        thisUpdate              Time,
        nextUpdate              Time OPTIONAL,
        revokedCertificates     SEQUENCE OF SEQUENCE {
             userCertificate         CertificateSerialNumber,
             revocationDate          Time,
             crlEntryExtensions      Extensions OPTIONAL;
                                           --- if present, must be v2
                                  } OPTIONAL
  crlExtensions                [0] EXPLICIT Extensions OPTIONAL
                                         -- If present, MUST be v2
                                  }
```

CertificateList is REQUIRED and MUST include the three fields “tbsCertList”, “signatureAlgorithm” and “signatureValue”. The support requirements for the subfields of the tbsCertList follow.

Fields of tbsCertList:

Version is REQUIRED and MUST be set to 1 to denote version 2 CRLs [RFC 5280], Section 5.1.2.1.

Signature is REQUIRED and contains an algorithm identifier to denote the algorithm used by the issuer to sign the CRL. The two algorithm identifiers used by Suite B are ecdsa-with-SHA256 and ecdsa-with-SHA384 [X9.62], Section E.8.

Issuer is REQUIRED and MUST follow RFC 5280, Section 5.1.2.3.

ThisUpdate follows RFC 5280 and indicates the date that this CRL was issued [RFC 5280], Section 5.1.2.4.

NextUpdate follows RFC 5280 and indicates the date by which the next CRL will be issued [RFC 5280], Section 5.1.2.5.

Revoked Certificates follows RFC 5280, Section 5.1.2.6.

CrlExtensions MUST include the authority key identifier and the CRL Number extensions [RFC 5280], Section 5.2.

- The authorityKeyIdentifier extension MUST be marked as non-critical and MUST use the keyIdentifier field [RFC 5280], Section 5.2.1. The value of the keyIdentifier field will be computed following the guidance in RFC 5280, Section 4.2.1.1.
- The crlNumber extension MUST be marked as non-critical and follows RFC 5280, Section 5.2.3.

All extensions not described in this profile should be considered OPTIONAL; their inclusion or exclusion and their values will depend upon the particular application or environment incorporating this Suite B Certificate and CRL profile as a base.

Signature Algorithm and Value Fields of CRL:

The signatureAlgorithm and signatureValue fields of the Suite B CRL follow the description previously listed for those fields in the X.509 Suite B Certificate Profile.

The 'signatureAlgorithm' and 'signatureValue' in the Certificate and CertificateList sequences, the 'signature' and 'subjectPublicKeyInfo' in the TBSCertificate sub-component and the 'signature' in the TBSCertList sub-component MUST identify Suite B through the use of algorithm identifiers.

The primary OID structure for Suite B is as follows [X9.62], [SEC2], [RFC3279] and [DRAFT3279bis]:

ansi-X9-62 OID ::= {iso(1) member-body(2) us(840) 10045}
certicom-arc OID ::= {iso(1) identified-organization(3) certicom(132)}

id-ecPublicKey OID ::= {ansi-X9-62 keyType(2) 1}
id-ecDh OID ::={certicom-arc schemes(1) ecdh(12)}

secp256r1 OID ::= {ansi-X9-62 curves(3) prime(1) 7}
secp384r1 OID ::= {certicom-arc curve(0) 34}

id-ecSigType OID ::= {ansi-X9-62 signatures(4)}
ecdsa-with-SHA256 OID ::= {id-ecSigType specified(3) 2}
ecdsa-with-SHA384 OID ::= {id-ecSigType specified(3) 3}

References

[CASE] "The Case for Elliptic Curve Cryptography", www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm.

[DRAFT3279bis] Turner, S., Brown, D., Yiu, K., Housley, R. Polk, T., "Elliptic Curve Cryptography Subject Public Key Information", draft-ietf-pkix-ecc-subpubkeyinfo-04.txt, March 2008.

[RFC 5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W. "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", draft-ietf-pkix-rfc3280bis-11.txt, February 2008.

[DRAFT2797bis] Schaad, J., Myers, M. "Certificate Management Messages over CMS", draft-ietf-pkix-2797-bis-07.txt, March 2008.

[DRAFTCMCTRANS] Schaad, J., Myers, M., "Certificate Management Messages over CMS (CMC): Transport Protocols", draft-ietf-pkix-cmc-trans-08.txt, March 2008.

[DRAFTCMCCOMPL] Schaad, J., Myers, M., "Certificate Management Messages over CMS (CMC): Compliance Requirements ", draft-ietf-pkix-cmc-compl-05, December 2007.

[RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, March 1997.

[RFC2797] Myers, M., Liu, X., Schaad, J., Weinstein, J., "Certificate Management Messages over CMS", April 2000.

[RFC3279] Polk, W., Housley, R., Bassham, L., "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", April 2002.

[RFC3280] Housley, R., Polk, W., Ford, W. and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", April 2002.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., Moeller, B., "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", May 2006.

[RFC4251] Ylonen, T., Lonvick, C., "The Secure Shell (SSH) Protocol Architecture", January 2006.

[RFC4869] Law, L., Solinas, J., "Suite B Cryptographic Suites for IPsec", May 2007.

[RFC5008] Housley, R., "Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME)", September 2007.

[SEC2] Standards for Efficient Cryptography, "SEC 2: Recommended Elliptic Curve Domain Parameters ", September 2000.

[SP800-56A] Barker, E., Johnson, D., Smid, M., "NIST SP80056A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised) ", March 2007.

[SP800-57] Barker, E., Barker, W., Burr, W., Polk, W., Smid, M. "NIST SP800-57: Recommendation for Key Management- Part 1: General", March 2007.

[SUITEB] "Fact Sheet NSA Suite B Cryptography", www.nsa.gov/ia/industry/crypto_suite_b.cfm.

[TLSECC] Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode", draft-ietf-tls-ecc-new-mac-04.txt, February 2008.

[X9.62] ANS X9.62, "Public Key Cryptography for the Financial Services Industry; The Elliptic Curve Digital Signature Algorithm (ECDSA)", November 2005.

[X9.63] ANS X9.63, "Public Key Cryptography for the Financial Services Industry; Key Agreement and Key Transport Using Elliptic Curve Cryptography", November 2001

A.0.0 REFERENCE SECTIONS FOR CERTIFICATE FIELDS

Component	Referenced Standard	Section	Requirement or Recommendation
version	RFC 5280	4.1.2.1	Set to 2 for Version 3 certificates.
serialNumber	RFC 5280	4.1.2.2	SHALL be a positive (non-negative integer) and SHALL NOT be longer than 20 octets.
signature	X9.62	E.8	OIDs for ECDSA with SHA-256 or ECDSA with SHA-384.
issuer	RFC 5280	4.1.2.4	Follows the guidance in RFC 5280.
validity	RFC 5280	4.1.2.5	Follows the guidance in RFC 5280.
subject	RFC 5280	4.1.2.6	Follows the guidance in RFC 5280.
subjectPublicKeyInfo: AlgorithmIdentifier	RFC 5280 RFC3279 X9.62 DRAFT3279bis	4.1.2.7 2.3.5 E.7 2.1.2	The algorithm identifier uses the OIDs for EC and ECDH public keys and OIDs for P-256 and P-384 curves in the algorithm and parameters subfields.
subjectPublicKeyInfo: subjectPublicKey	X9.63 RFC3279	4.3.6 2.3.5	First byte is 0x00 (number of unused bits in last octet); 2nd byte is 0x04 for uncompressed; followed by 256(384) bits for x-coordinate; 256(384) bits for y-coordinate for P-256 and P-384 curves, respectively.
authorityKeyIdentifier	RFC 5280	4.2.1.1	Follows the guidance in RFC 5280; criticality is FALSE.
subjectKeyIdentifier	RFC 5280	4.2.1.2	Follows the guidance in RFC 5280; criticality is FALSE.

keyUsage	RFC 5280	4.2.1.3	Set keyCertSign and cRLSign bits for Root CA Self-Signed certificate, Subordinate CA certificate and Cross-Certificate; sets digitalSignature for end entity Signature certificate; sets keyAgreement for end entity key establishment certificate; criticality is TRUE.
certificatePolicies	RFC 5280	4.2.1.4	Set to OID for particular certificate policy in use; can use the anyPolicy OID; policyQualifiers is NOT RECOMMENDED; criticality is FALSE.
basicConstraints	RFC 5280	4.2.1.9	Sets the cA bit in root CA self-signed certificate, subordinate CA certificate and cross-certificate; criticality is TRUE; pathLenConstraint is not set in CA root self-signed nor CA cross-certificates; is optional in CA subordinate certificates.
policyMappings	RFC 5280	4.2.1.5	Recommended for CA cross-certificates; criticality is FALSE; policies MUST not be mapped either to or from anyPolicy.
policyConstraints	RFC 5280	4.2.1.11	Recommended for CA cross-certificates; criticality is TRUE; requireExplicitPolicy and inhibitPolicyMapping MUST be set to zero.
inhibitAnyPolicy	RFC 5280	4.2.1.14	Recommended for CA cross-certificates; criticality is TRUE; SkipCerts MUST be set to zero.
subjectAltName	RFC 5280	4.2.1.6	In end entity certificates, if subject name is empty sequence, must include subjectAltName; criticality is then TRUE otherwise is FALSE.
signatureAlgorithm	RFC 5280 X9.62	4.1.1.2 E.8	OIDs for ECDSA with SHA-256 or ECDSA with SHA-384.
signatureValue	RFC 5280 X9.62 ASN.1	4.1.1.3 7 5.4, 5.7	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 (49) bytes for P-256 and P-384, respectively.

A.0.1 REFERENCE SECTIONS FOR CERTIFICATE REVOCATION LISTS (CRL) FIELDS

Component	Referenced Standard	Section	Requirement or Recommendation
version	RFC 5280	5.1.2.1	Set to 1 for Version 2 certificates.
signature	X9.62	E.8	OIDs for ECDSA with SHA-256 or ECDSA with SHA-384.
issuer	RFC 5280	5.1.2.3	Follows the guidance in RFC 5280.
thisUpdate	RFC 5280	5.1.2.4	Indicates the issue date of this CRL.
nextUpdate	RFC 5280	5.1.2.5	Indicates date by which the next CRL will be issued.
revokedCertificates	RFC 5280	5.1.2.6	List of revoked certificates.
authorityKeyIdentifier	RFC 5280	5.2.1 4.2.1.1	Follows the guidance in RFC 5280; criticality is FALSE.
cRLNumber	RFC 5280	5.2.3	A montonically increasing sequence number for a given CRL scope and issuer; criticality is FALSE.
signatureAlgorithm	RFC 5280 X9.62	5.1.1.2 E.8	OIDs for ECDSA with SHA-256 or ECDSA with SHA-384.
signatureValue	RFC 5280 X9.62 ANS.1	5.1.1.3 7 5.4, 5.7	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 (49) bytes for P-256 and P-384, respectively.

A.1 ROOT CA SELF-SIGNED CERTIFICATE USING P-256 SIGNED WITH P-256

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.2	ECDSA with SHA-256.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; must match the issuer name; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 01 06	keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
basicConstraints Identifier Critical Value:cA Value:pathLenConstraint	 2.5.29.19 TRUE TRUE	cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint MUST NOT be set.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.2	ECDSA with SHA-256.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 bytes.

(U) A.2 ROOT CA SELF-SIGNED CERTIFICATE USING P-384 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; must match the issuer name; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.3.132.0.34	P-384 curve (named curve option).
subjectPublicKey	BIT STRING (784 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 384-bit x-coordinate; 384-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 01 06	keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
basicConstraints Identifier Critical Value:cA Value:pathLenConstraint	 2.5.29.19 TRUE TRUE	cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint MUST NOT be set.
end tbCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.

A.3 SUBORDINATE CA CERTIFICATE USING P-256 SIGNED WITH P-256

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.2	ECDSA with SHA-256.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 01 06	keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
basicConstraints Identifier Critical Value:cA Value:pathLenConstraint	 2.5.29.19 TRUE TRUE	cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint is OPTIONAL.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.2	ECDSA with SHA-256.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 bytes.

A.4 SUBORDINATE CA CERTIFICATE USING P-384 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.3.132.0.34	P-384 curve (named curve option).
subjectPublicKey	BIT STRING (784 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 384-bit x-coordinate; 384-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 01 06	keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
basicConstraints Identifier Critical Value:cA Value:pathLenConstraint	 2.5.29.19 TRUE TRUE	cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint is OPTIONAL.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.

A.5 SUBORDINATE CA CERTIFICATE USING P-256 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 01 06	keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
basicConstraints Identifier Critical Value:cA Value:pathLenConstraint	 2.5.29.19 TRUE TRUE	cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint is OPTIONAL.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.

A.6 CA CROSS-CERTIFICATE USING P-256 SIGNED WITH P-256

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.2	ECDSA with SHA-256.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier		Follows RFC 5280.
Identifier	2.5.29.35	
Critical	FALSE	
Value	OCTET STRING	
subjectKeyIdentifier		Follows RFC 5280.
Identifier	2.5.29.14	
Critical	FALSE	
Value	OCTET STRING	
keyUsage		keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
Identifier	2.5.29.15	
Critical	TRUE	
Value	03 02 01 06	
basicConstraints		cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint MUST NOT be set.
Identifier	2.5.29.19	
Critical	TRUE	
Value:cA	TRUE	
Value:pathLenConstraint		
certificatePolicies		Follows RFC 5280.
Identifier	2.5.29.32	
Critical	FALSE	
Value	OID	
Recommended Extensions		
policyMappings		Follows RFC 5280.
Identifier	2.5.29.33	
Critical	FALSE	
Value	SEQUENCE	
policyConstraints		The requireExplicityPolicy and inhibitPolicyMapping fields MUST be set to zero.
Identifier	2.5.29.36	
Critical	TRUE	
Value	SEQUENCE	
inhibitAnypolicy		SkipCerts MUST be set to zero.
Identifier	2.5.29.54	
Critical	TRUE	
Value	INTEGER	
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.2	ECDSA with SHA-256.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a max of 33 bytes.

A.7 CA CROSS-CERTIFICATE USING P-384 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.3.132.0.34	P-384 curve (named curve option).
subjectPublicKey	BIT STRING (784 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 384-bit x-coordinate; 384-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier		Follows RFC 5280.
Identifier	2.5.29.35	
Critical	FALSE	
Value	OCTET STRING	
subjectKeyIdentifier		Follows RFC 5280.
Identifier	2.5.29.14	
Critical	FALSE	
Value	OCTET STRING	
keyUsage		keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
Identifier	2.5.29.15	
Critical	TRUE	
Value	03 02 01 06	
basicConstraints		cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint MUST NOT be set.
Identifier	2.5.29.19	
Critical	TRUE	
Value:cA	TRUE	
Value:pathLenConstraint		
certificatePolicies		Follows RFC 5280.
Identifier	2.5.29.32	
Critical	FALSE	
Value	OID	
Recommended Extensions		
policyMappings		Follows RFC 5280.
Identifier	2.5.29.33	
Critical	FALSE	
Value	SEQUENCE	
policyConstraints		The requireExplicityPolicy and inhibitPolicyMapping fields MUST be set to zero.
Identifier	2.5.29.36	
Critical	TRUE	
Value	SEQUENCE	
inhibitAnypolicy		SkipCerts MUST be set to zero.
Identifier	2.5.29.54	
Critical	TRUE	
Value	INTEGER	
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a max of 49 bytes

A.8 CA CROSS-CERTIFICATE USING P-256 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; MUST be non-empty field.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier		Follows RFC 5280.
Identifier	2.5.29.35	
Critical	FALSE	
Value	OCTET STRING	
subjectKeyIdentifier		Follows RFC 5280.
Identifier	2.5.29.14	
Critical	FALSE	
Value	OCTET STRING	
keyUsage		keyCertSign and cRLSign bits are set. Value is a DER encoded BIT STRING.
Identifier	2.5.29.15	
Critical	TRUE	
Value	03 02 01 06	
basicConstraints		cA bit is set to TRUE to indicate that the subject is a CA; pathLenConstraint MUST NOT be set.
Identifier	2.5.29.19	
Critical	TRUE	
Value:cA	TRUE	
Value:pathLenConstraint		
certificatePolicies		Follows RFC 5280.
Identifier	2.5.29.32	
Critical	FALSE	
Value	OID	
Recommended Extensions		
policyMappings		Follows RFC 5280.
Identifier	2.5.29.33	
Critical	FALSE	
Value	SEQUENCE	
policyConstraints		The requireExplicityPolicy and inhibitPolicyMapping fields MUST be set to zero.
Identifier	2.5.29.36	
Critical	TRUE	
Value	SEQUENCE	
inhibitAnypolicy		SkipCerts MUST be set to zero.
Identifier	2.5.29.54	
Critical	TRUE	
Value	INTEGER	
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a max of 49 bytes

A.9 END ENTITY SIGNATURE CERTIFICATE USING P-256 SIGNED WITH P-256

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.2	ECDSA with SHA-256.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; if empty, must contain the subjectAltName extension.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic Curve (EC).
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 07 80	digitalSignature bit is set. Value is a DER encoded BIT STRING.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
subjectAltName Identifier Critical Value	 2.5.29.17 TRUE or FALSE OID	If the subject name is an empty sequence, then the subjectAltName extension must be included and the criticality flag marked TRUE. Otherwise, this extension is optional and if included, the criticality flag is marked FALSE.
Recommended Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.2	ECDSA with SHA-256.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 bytes.

A.10 END ENTITY SIGNATURE CERTIFICATE USING P-384 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; if empty, must contain the subjectAltName extension.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.3.132.0.34	P-384 curve (named curve option).
subjectPublicKey	BIT STRING (784 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 384-bit x-coordinate; 384-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 07 80	digitalSignature bit is set. Value is a DER encoded BIT STRING.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
subjectAltName Identifier Critical Value	 2.5.29.17 TRUE or FALSE OID	If the subject name is an empty sequence, then the subjectAltName extension must be included and the criticality flag marked TRUE. Otherwise, this extension is optional and if included, the criticality flag is marked FALSE.
Recommended Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum 49 bytes.

A.11 END ENTITY SIGNATURE CERTIFICATE USING P-256 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; if empty, must contain the subjectAltName extension.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key.
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 07 80	digitalSignature bit is set. Value is a DER encoded BIT STRING.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
subjectAltName Identifier Critical Value	 2.5.29.17 TRUE or FALSE OID	If the subject name is an empty sequence, then the subjectAltName extension must be included and the criticality flag marked TRUE. Otherwise, this extension is optional and if included, the criticality flag is marked FALSE.
Recommended Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.

A.12 END ENTITY KEY ESTABLISHMENT CERTIFICATE USING P-256 SIGNED WITH P-256

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.2	ECDSA with SHA-256.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; if empty, must contain the subjectAltName extension.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key. MAY use id-ecDH = (1.3.132.1.12).
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 03 08	Key agreement bit is set. Value is a DER encoded BIT STRING.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
subjectAltName Identifier Critical Value	 2.5.29.17 TRUE or FALSE OID	If the subject name is an empty sequence, then the subjectAltName extension must be included and the criticality flag marked TRUE. Otherwise, this extension is optional and if included, the criticality flag is marked FALSE.
Recommended Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.2	ECDSA with SHA-256.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 bytes.

A.13 END ENTITY KEY ESTABLISHMENT CERTIFICATE USING P-384 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; if empty, must contain the subjectAltName extension.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key. MAY use id-ecDH = (1.3.132.1.12).
AlgorithmIdentifier:parameters	1.3.132.0.34	P-384 curve (named curve option).
subjectPublicKey	BIT STRING (784 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 384-bit x-coordinate; 384-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 03 08	Key agreement bit is set. Value is a DER encoded BIT STRING.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
subjectAltName Identifier Critical Value	 2.5.29.17 TRUE or FALSE OID	If the subject name is an empty sequence, then the subjectAltName extension must be included and the criticality flag marked TRUE. Otherwise, this extension is optional and if included, the criticality flag is marked FALSE.
Recommended Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.

A.14 END ENTITY KEY ESTABLISHMENT CERTIFICATE USING P-256 SIGNED WITH P-384

Field	Value	Comments
begin tbsCertificate		
version	2	Value=0x02 for version 3 certificates.
serialNumber	INTEGER	Follows RFC 5280.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
validity		Follows RFC 5280.
subject		Follows RFC 5280; if empty, must contain the subjectAltName extension.
subjectPublicKeyInfo		
AlgorithmIdentifier:algorithm	1.2.840.10045.2.1	Elliptic curve (EC) public key. MAY use id-ecDH = (1.3.132.1.12).
AlgorithmIdentifier:parameters	1.2.840.10045.3.1.7	P-256 curve (named curve option).
subjectPublicKey	BIT STRING (528 bits)	First byte is 0x00 for number of unused bits in last octet; second byte is 0x04 to denote uncompressed format. Followed by 256-bit x-coordinate; 256-bit y-coordinate.
Unique Identifiers		
issuerUniqueID		Not present.
subjectUniqueID		Not present.
Required Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
keyUsage Identifier Critical Value	 2.5.29.15 TRUE 03 02 03 08	Key agreement bit is set. Value is a DER encoded BIT STRING.
certificatePolicies Identifier Critical Value	 2.5.29.32 FALSE OID	Follows RFC 5280.
subjectAltName Identifier Critical Value	 2.5.29.17 TRUE or FALSE OID	If the subject name is an empty sequence, then the subjectAltName extension must be included and the criticality flag marked TRUE. Otherwise, this extension is optional and if included, the criticality flag is marked FALSE.
Recommended Extensions		
subjectKeyIdentifier Identifier Critical Value	 2.5.29.14 FALSE OCTET STRING	Follows RFC 5280.
end tbsCertificate		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.

A.15 CERTIFICATE REVOCATION LIST (CRL) SIGNED WITH P-256

Field	Value	Comments
begin tbsCertList		
version	1	Value=0x01 for version 2 CRL.
signature	1.2.840.10045.4.3.2	ECDSA with SHA-256.
issuer		Follows RFC 5280.
thisUpdate		Follows RFC 5280.
nextUpdate		Follows RFC 5280.
revokedCertificates		Follows RFC 5280.
Required CRL Extensions RFC 5280		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
cRLNumber Identifier Critical Value	 2.5.29.20 FALSE INTEGER	Follows RFC 5280.
end tbsCertList		
signatureAlgorithm	1.2.840.10045.4.3.2	ECDSA with SHA-256.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 33 bytes.

A.16 CERTIFICATE REVOCATION LIST (CRL) SIGNED WITH P-384

Field	Value	Comments
begin tbsCertList		
version	1	Value=0x01 for version 2 CRL.
signature	1.2.840.10045.4.3.3	ECDSA with SHA-384.
issuer		Follows RFC 5280.
thisUpdate		Follows RFC 5280.
nextUpdate		Follows RFC 5280.
revokedCertificates		Follows RFC 5280.
Required CRL Extensions		
authorityKeyIdentifier Identifier Critical Value	 2.5.29.35 FALSE OCTET STRING	Follows RFC 5280.
cRLNumber Identifier Critical Value	 2.5.29.20 FALSE INTEGER	Follows RFC 5280.
end tbsCertList		
signatureAlgorithm	1.2.840.10045.4.3.3	ECDSA with SHA-384.
signatureValue	BIT STRING	Encoded BIT STRING value of a DER encoded SEQUENCE of two INTEGERS; each a maximum of 49 bytes.