Federal Public Key Infrastructure (PKI) X.509 Certificate and CRL Extensions Profile

October 12, 2005

Prepared By:

BOOZ•ALLEN & HAMILTON INC.
900 Elkridge Landing Road
Linthicum, Maryland 21090

Updated By:

National Institute of Standards and Technology
100 Bureau Dr.
Gaithersburg, MD 20899-8930

1. Introduction

This document specifies the X.509 version 3 certificate and version 2 certificate revocation list (CRL) profiles for Federal public key infrastructure (FPKI) systems. The profiles serve to identify unique parameter settings for Federal public key infrastructure systems.

In order to maximize interoperability among PKI communities within the Federal government, this profile was designed to complement the *NIST Recommendation for X.509 Path Validation*. The NIST Recommendation specifies a minimum set of features from X.509 that path validation software must implement in order to be considered a Bridge-enabled Path Validation Module (PVM). This profile recommends against the use of features from X.509 that are not included in this minimum set.

In the interest of establishing commonality and interoperability among PKI communities outside the Federal government, it was decided that this profile should be based on a "standard PKI profile" but still contain the unique parameter settings for Federal systems. The only widely accepted PKI profile currently on track to become a standard is the Internet Engineering Task Force (IETF) Public Key Infrastructure (PKI) profile developed by the PKIX working group. The PKIX profile, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, identifies the format and semantics of certificates and CRLs for the Internet PKI. Procedures are described for processing and validating certification paths in the Internet environment. Encoding rules are provided for all fields and extensions profiled in both the X.509 v3 certificate and v2 CRL. Encoding rules for cryptographic algorithms specified in this profile are specified in RFC 3279.

This FPKI profile complements the current PKIX profile. If a specific program needs to implement a subset of the FPKI certificate and/or CRL profile, the program should tailor their X.509 certificate and/or CRL profile using the parameters stipulated in this document together with the parameters stipulated in PKIX. Parameters stipulated in this document should take precedence. Any program deciding to tailor their FPKI-compliant X.509 certificates and/or CRLs to meet their specific needs must document the intended subset profile (referencing the FPKI profile as a basis) so that the certificate generation element will know how to populate the program-specific certificates.

The current version of this profile includes requirements that did not appear in previous versions. Operators of existing CAs are not encouraged to re-issue any certificates for the sole purpose of complying with this version of the profile. However, operators should make preparations to begin issuing all new certificates in conformance with the profile.

1.1. Structure

This document is divided into six sections. Section 1 includes this introduction. Sections 2 and 3 describe the v3 certificate and v2 CRL respectively. These sections specifically describe the differences in generation and processing requirements between the PKIX profile and FPKI profile. Unless otherwise noted in this profile, the reader should follow the PKIX generation and processing requirements for a particular field. Section 4 specifies rules for choosing character encoding sets for attribute values of type DirectoryString in distinguished names. Section 5 profiles the use of uniform resource identifiers (URIs) in certificates. Section 6 provides guidance on the selection of algorithms for signing certificates and CRLs and for the selection of public key types. Section 7 provides an overview of each of the certificate and CRL profiles included in the worksheets at the end of this document.

1.2. Acronyms

CA	Certification Authority
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules
DH	Diffie-Hellman
DN	Distinguished Name
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FBCA	Federal Bridge Certification Authority
FPKI	Federal Public Key Infrastructure
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
KEA	Key Exchange Algorithm
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure (X.509)
PVM	Path Validation Module
RDN	Relative Distinguished Name
RFC	Request For Comments
RSA	Rivest-Shamir-Adelman
URI	Uniform Resource Identifier
v2	version 2
v3	version 3

1.3. References

[1] *NIST Recommendation for X.509 Path Validation*, Version 0.5, May 2004.

[2] Russell Housley and Paul Hoffman. Internet X.509 Public Key Infrastructure: *Operational Protocols: FTP and HTTP*, RFC 2585, May 1999.

[3] Russell Housley, Tim Polk, Warwick Ford, and David Solo. Internet Public Key Infrastructure: *X.509 Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280, April 2002.

[4] Tim Howes and Mark Smith. *The LDAP URL Format*, RFC 2255, December 1997.

[5] Tim Berners-Lee, Larry Masinter, and Mark McCahill. *Uniform Resource Locators*

(URL), RFC 1738, December 1994.

[6] Steve Lloyd. *AKID/SKID Implementation Guideline*, September 2002.

[7] Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Carlisle Adams. Internet X.509 Public Key Infrastructure: *Online Certificate Status Protocol – OCSP*, RFC 2560, June 1999.

[8] Tim Polk, Russell Housley, and Larry Bassham. Internet Public Key Infrastructure: *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, RFC 3279, April 2002.

[9] Blake Ramsdell. *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*, RFC 3851, July 2004.

[10] Mark Wahl, Steve Kille, and Tim Howes. Lightweight Directory Access Protocol (v3): *UTF-8 String Representation of Distinguished Names*, RFC 2253, December 1997.

2. X.509 v3 Certificates

X.509 v3 certificates contain the identity and attribute data of a subject using the base certificate with applicable extensions. The base certificate contains such information as the version number of the certificate, the certificate's identifying serial number, the signature algorithm used to sign the certificate, the issuer's distinguished name, the validity period of the certificate, the distinguished name of the subject, and information about the subject's public key. To this base certificate are appended numerous certificate extensions. More detailed information about X.509 certificates can be found in Recommendation X.509 and RFC 3280.

CAs create certificates for user authentication procedures that require one user to obtain another user's public key. So that users trust the public key, the CA employs a digital signature to cryptographically sign the certificate in order to provide assurance that the information within the certificate is correct. The fields in a certificate identify the issuer (i.e., CA), subject (i.e., user), version number, subject's public key, validity period, and serial number of the certificate along with the public key algorithm used to certify the certificate. A CA may also add certificate extensions containing additional information about the user or the CA, depending on the implementation.

All certification paths start from a trust anchor. A trust anchor is a CA that a user trusts to issue certificates based on out-of-band knowledge. The public key of a trust anchor is distributed to certificate users in the form of a "trust anchor certificate." A trust anchor certificate:

- is self-signed, that is, signed with the private key corresponding to the public key contained in the subject public key field of the certificate;[1]
- contains any needed parameters in the subject public key info field, where the digital signature algorithm used in the certificate requires the use of parameters;

[1] NOTE: While in most cases, the public key of a CA that is to act as a trust anchor is distributed using self-signed certificates, this is not strictly necessary. Relying parties may obtain the public key of a trust anchor by other means.

- contains few or no extensions;
- is kept in protected memory or otherwise protected from alteration by an intruder;
- is transferred to the application or certificate using system in an authenticated manner. The signature on the trust anchor certificate cannot authenticate the certificate.

There is no single trust anchor for the entire Federal Government. The trust anchor used by a certificate using application may be the CA that issued it a certificate or may be a CA that is at the top of a hierarchy of CAs. Which trust anchors may be used by agency certificate using systems to start certification paths is a matter of agency security policy.

The CAs in the FPKI may be cross-certified with each other. In order to facilitate secure intra-agency communication, CAs within the FPKI may either cross-certify with each other directly or may cross-certify with the Federal Bridge Certification Authority (FBCA). In general, cross-certification with the FBCA is the preferable option since it maximizes inter-agency connectivity while minimizing the number of cross-certifications that any given CA needs to maintain.

Any certificate using system in the FPKI can view any CA in the FPKI as the trust anchor for starting certification paths, provided:

1. the certificate using system has an authenticated copy of the trust anchor's self-signed certificate; and,
2. local agency security policy allows the use of that CA as a trust anchor.

Agencies will designate the CAs that may be used as trust anchors by certificate using systems within the agency, and will establish the approved mechanisms for obtaining the trust anchors' public keys in a secure, authenticated manner. The FBCA will make the self-signed certificate of the Common Policy Root CA available for use as a trust anchor, and it is expected that this CA will be used as the trust anchor for most users who are issued certificates under the Common Policy.

X.509 v3 certificates provide a mechanism for CAs to append additional information about the subject’s public key, issuer’s public key, and issuer’s CRLs. Standard certificate extensions are defined for X.509 v3 certificates. These extensions provide methods for increasing the amount of information the X.509 certificate conveys to facilitate automated certificate processing.

3. X.509 v2 Certificate Revocation Lists

CAs use CRLs to publicize the revocation of a subject’s certificate. The CRLs are stored in the directory as attributes and are checked by relying parties to verify that a user’s certificate has not been revoked. The fields in a CRL identify the issuer, the date the current CRL was generated, the date by which the next CRL will be generated, and the revoked users’ certificates.

If a CA issues a large number of certificates, then the use of distribution points is encouraged in order to ensure that CRLs do not become too large. However, distribution points should not be specified as nameRelativeToIssuer since the NIST Recommendation for X.509 Path Validation [1] does not require Bridge-enabled PVMs to be able to process this feature. For the same reason, the use of indirect CRLs and CRLs segmented by reason code is discouraged. CAs that issue segmented CRLs are strongly encouraged to also issue full CRLs in order to accommodate third parties that use CRLs to generate OCSP responses.

Since Bridge-enabled PVMs PP are not required to be able to process delta-CRLs, CAs should not assume that relying parties can process them. However, CAs may still choose to issue delta-CRLs in addition to complete for scope CRLs in order to improve efficiency for those relying parties that can process the delta-CRLs.

CAs may optionally supplement the CRL based revocation mechanisms with on-line revocation mechanisms.

4. Encoding Distinguished Names

X.509 certificates and CRLs include distinguished names (DNs) to identify issuers of certificates and CRLs, subjects of certificates, and CRL distribution points. A DN consists of a sequence of relative distinguished names (RDNs) where each RDN consists of a set of attribute type and value pairs. In most cases, an RDN will only include a single attribute type and value pair, but in some cases the use of a multi-valued RDN (i.e., an RDN that includes more than one attribute type and value pair) may be appropriate. In general, multi-valued RDNs should not be included in DN that identify CAs or CRL distribution points. When multi-valued RDNs are included in DNs that identify end entities, they should only be included in the final RDN in the DN (i.e., the RDN that appears first when the DN is written as specified in RFC 2253) and no multi-valued RDN should include two instances of an attribute type.

Many of the attributes in distinguished names use the DirectoryString syntax. DirectoryString permits encoding of names in a choice of character sets: PrintableString, TeletexString, BMPString, UniversalString, and UTF8String. PrintableString is currently the most widely used encoding for attribute values in distinguished names. PrintableString is a subset of ASCII; it does not include characters required for most international languages. UTF8String is an encoding that supports all recognized written languages, including some ancient languages (e.g., Runic).

Name comparison is an important step in X.509 path validation, particularly for name chaining and name constraints computation. Many legacy implementations are unable to perform name comparisons when names are encoded using different character sets. To simplify correct operation of path validation, CAs are strongly encouraged to honor the subject's chosen character set when issuing CA certificates or populating extensions. That is, if a subject CA encodes its own name in the issuer field of certificates and CRLs it generates using TeletexString, the cross-certificate should use the same character set to specify that CA's name.

Name constraints are specified in CA certificates. The names specified in name constraints must be compared with the subject names in subsequent certificates in a certification path. To help ensure that name constraints are applied correctly, CAs should encode each attribute value in a name constraint using the same encoding as is used to encode the corresponding attribute value in subject names in subsequent certificates. In general, this may be accomplished by encoding attribute values in the name constraints extension of a certificate in the same way as they are encoded in the subject name of the certificate.

Subject names in end entity certificates do not figure in name chaining, but are used to validate name constraints. In order to ensure that name constraints can be computed correctly, attribute values that are shared between an end entity and its certificate issuer should be encoded identically. Attribute values in end entity names that are unique to the end entity (e.g., the common name) may be encoded in PrintableString or UTF8String without concern for name comparison issues.

Attributes of type DirectoryString in the distinguished names of new CAs should be encoded using the PrintableString encoding wherever possible. When the attribute value cannot be encoding using PrinableString, the UTF8String encoding should be used. However, if the new CA is being added to an existing PKI, attribute values in the new CA's DN that are shared with the DNs of pre-existing CAs may use the same encoding as was used in the DNs of the pre-existing CAs. As products that compare names encoded in different character sets become available, CAs should transition to either PrintableString or UTF8String encodings when they roll over to new key pairs.

For CAs within the Federal PKI, all attributes of type DirectoryString should be encoded using the PrintableString encoding. However, the common name attribute of end entity certificates may be encoded in UTF8String if the subject's name cannot be encoded using PrintableString.

5. Use of URIs in Distribution Points, authorityInfoAccess, and subjectInfoAccess Extensions

Uniform Resource Identifiers (URIs) are used in five different extensions within the certificate and CRL profiles in this document: cRLDistributionPoints, issuingDistributionPoint, FreshestCRL, authorityInfoAccess, and subjectInfoAccess. Two different protocols are used in this document: LDAP and HTTP. The specifications for URIs for these protocols may be found in RFC 2255 and RFC 1738, respectively.

Except for the id-ad-ocsp access method of the authorityInfoAccess extension, all URIs should have a prefix of "ldap" or "http" to indicate that the relevant information is located in an LDAP accessible directory or via HTTP. For the id-ad-ocsp access method of the authorityInfoAccess, the URI should have a prefix of "http" to indicate that the transport protocol for the OCSP request/response messages is HTTP. The hostname of every URI should be specified as either a fully qualified domain name or an IP address. The port number of the server must be specified if it is not the default port number for the relevant protocol (80 for HTTP and 389 for LDAP).

In the cRLDistributionPoints and FreshestCRL extensions, the URI is a pointer to a current CRL that provides status information about the certificate. If LDAP is used, the URI must include the DN of the entry containing the CRL and specify the directory attribute in which the CRL is located (certificateRevocationList, authorityRevocationList, or deltaRevocationList). If the directory in which the CRL is stored expects the "binary" option to be specified, then the attribute type must be followed by ";binary" in the URI. If HTTP is used, the URI must point to a file that has an extension of ".crl" that contains the

DER encoded CRL (see RFC 2585). When a URI is used as the DistributionPointName in the issuingDistributionPoint extension in a CRL, the value must match the URI in the corresponding distribution points in the cRLDistributionPoints extensions in certificates covered by the CRL.

Some examples of URIs that may appear in a cRLDistributionPoints, FreshestCRL, or issuingDistributionPoint extension are:

ldap://smime2.nist.gov/cn=Good%20CA,o=Test%20Certificates,c=US?certificateRevocationList;binary

ldap://129.6.20.71/cn=onlyContainsCACerts%20CA,o=Test%20Certificates,c=US?authorityRevocationList;binary

http://CRLs.example.com/fictitiousCRLdirectory/fictitiousCRL1.crl

The authorityInfoAccess extension uses URIs for two purposes. When the id-ad-caIssuers access method is used, the access location specifies where certificates issued to the issuer of the certificate may be found. If LDAP is used, the URI must include the DN of the entry containing the relevant certificates and specify the directory attribute(s) (e.g., cACertificate and crossCertificatePair) in which the certificates are located. If the directory in which the certificates are stored expects the "binary" option to be specified, then the attribute type must be followed by ";binary" in the URI. If HTTP is used, the URI must point to a file that has an extension of ".p7c" that contains a certs-only CMS message (see RFC 3851). The CMS message should include all certificates issued to the issuer of this certificate, but must at least contain all certificates issued to the issuer of this certificate in which the subject public key may be used to verify the signature on this certificate.

For a certificate issued by “Good CA”, some examples of URIs that may appear as the access location in an authorityInfoAccess extension when the id-ad-caIssuers access method is used are:

ldap://129.6.20.71/cn=Good%20CA,o=Test%20Certificates,c=US?cACertificate;binary,crossCertificatePair;binary

http://Certs.example.com/fictitiousCertsOnlyCMSdirectory/certsIssuedToGoodCA.p7c

When the id-ad-ocsp access method is used, the access location specifies the location of an OCSP server that provides status information about the certificate (see RFC 2560). The URI may include a path. Where privacy is a requirement, the URI may have a prefix of "https" to indicate that the transport protocol for OCSP requests/responses is HTTP over SSL/TLS. In this case, the default port number is 443, and the URI must include the server's port number if this default port number is not used.

The id-ad-caRepository access method for the subjectInfoAccess extension uses URIs to specify the location where CA certificates issued by the subject of the certificate may be found. If LDAP is used, the URI must include the DN of the entry containing the relevant certificates and specify the directory attribute(s) (e.g., cACertificate and crossCertificatePair) in which the certificates are located. If the directory in which the certificates are stored expects the "binary" option to be specified, then the attribute type must be followed by ";binary" in the URI. If HTTP is used, the URI must point to a file that has an extension of ".p7c" that contains a certs-only CMS message (see RFC 3851). The CMS message should include all CA certificates issued by the subject of this certificate, but must at least contain all CA certificates issued by the subject of this certificate in which the signature on the certificate may be verified using the subject public key in this certificate.

If the subject of the certificate only issues end entity certificates, then the subjectInfoAccess extension may be excluded. If the subject of the certificate issues self-issued certificates (e.g., key rollover certificates), but does not issue certificates to other CAs, then the LDAP URI in the subjectInfoAccess extension only needs to specify the cACertificate attribute.

For a certificate issued to "Good CA", some examples of URIs that may appear as the access location in an subjectInfoAccess extension when the id-ad-caRepository access method is used are:

ldap://129.6.20.71/cn=Good%20CA,o=Test%20Certificates,c=US?cACertificate;binary,crossCertificatePair;binary

http://Certs.example.com/fictitiousCertsOnlyCMSdirectory/CAcertsIssuedByGoodCA.p7c

6. Certificate and CRL Signing Algorithms and Subject Public Key Types

This profile permits the use of any FIPS approved algorithm for signing certificates and CRLs. However, agencies are cautioned against use of the weakest or strongest of these algorithms. Specifically, cryptographic algorithms with 80 bits of security strength (e.g., 1024 RSA, 1024 bit DSA, 160 bit ECDSA, and SHA-1) are nearing the end of their useful cryptographic lifetime. PKIs using these algorithms are strongly encouraged to migrate to stronger algorithms as soon as possible. Additionally, cryptographic algorithms with more than 256 bits of security strength (e.g., 4096+ bit RSA, 384+ bit ECDSA, SHA-384, and SHA-512) will greatly impact performance. Such algorithms are currently appropriate for highly sensitive data (e.g., classified data) rather than for general use.

7. Worksheet Contents

This document contains six worksheets. Each worksheet lists mandatory contents of a particular class of certificates or CRLs. Optional features that will be widely supported in the Federal PKI are also identified. These features may be included at the issuer's option. Certificate and CRL issuers may include additional information in non-critical extensions for local use, but should not expect clients in the Federal PKI to process this additional information. Critical extensions that are not listed in these worksheets must not be included in certificates or CRLs used in the Federal PKI.

The six worksheets are:

1. The *Self-Signed CA Certificate* worksheet defines the mandatory and optional contents of self-signed certificates issued by CAs in the Federal PKI for use by PKI client systems when establishing trust anchors.
2. The *Key Rollover CA Certificate* worksheet defines the mandatory and optional contents of key rollover certificates (self-issued certificates that are not self-signed).
3. The *Cross-Certificate* worksheet defines the mandatory and optional contents of certificates issued by CAs in the Federal PKI where the subject is a CA and the public key will be used to verify the signature on certificates. This profile applies whether the subject CA is considered to be hierarchically subordinate to the issuer

or is considered to be a peer of the issuer. One optional feature in this worksheet is the use of the public key to verify the signature on CRLs.

4. The *CRL* worksheet defines the mandatory and optional contents of CRLs issued by CRL issuers in the Federal PKI.
5. The *End Entity Signature Certificate* worksheet defines the mandatory and optional contents of certificates issued by CAs in the Federal PKI where the subject is an end entity and the public key will be used to verify the signatures.
6. The *Key Management Certificate* worksheet defines the mandatory and optional contents of certificates issued by CAs in the Federal PKI where the subject is a CA and the public key will be used to perform key management operations (e.g., key transport using RSA or Diffie-Hellman key agreement).

Note that the Federal PKI does not absolutely prohibit the use of dual-use end entity certificates, where an RSA or elliptic curve key is used for both digital signatures and key management. However, dual-use certificates are generally discouraged. As such, a worksheet for dual-use certificates is not supplied with this profile. CAs in the Federal PKI that issue dual-use certificates may use the End Entity Signature Certificate profile and assert the additional key usage bits as appropriate (i.e., key encipherment for RSA keys or key agreement for elliptic curve keys). Dual-use certificates must not assert the nonRepudiation bit.

Worksheet 1: Self-Signed CA Certificate Profile

Field	Criticality Flag	Value	Comments
Certificate			
tbsCertificate			Fields to be signed.
version		2	Integer value of "2" for version 3 certificate.
serialNumber		INTEGER	Unique positive integer
signature			
AlgorithmIdentifier			Must match Algorithm Identifier in signatureAlgorithm field. The parameters field is only populated when the algorithm is RSA.
algorithm		Choice of following algorithms:	
		1.2.840.113549.1.1.5	sha-1WithRSAEncryption
		1.2.840.113549.1.1.10	id-RSASSA-PSS (RSA with PSS padding; 800-78 requires use with SHA-256 hash algorithm)
		1.2.840.113549.1.1.11	sha256WithRSAEncryption
		1.2.840.10045.4.3.1	ecdsa-with-Sha224
		1.2.840.10045.4.3.2	ecdsa-with-Sha256
		1.2.840.10045.4.3.3	ecdsa-with-Sha384
		1.2.840.10045.4.3.4	ecdsa-with-Sha512
parameters		2.16.840.1.101.3.4.2.1 or 2.16.840.1.101.3.4.2.3	For id-RSASSA-PSS only, specify either the SHA-256 or SHA-512 hash algorithm as a parameter
		NULL	For all RSA algorithms except id-RSASSA-PSS
issuer			
Name			Will match the subject DN.
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
validity			
notBefore			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
notAfter			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
subject			
Name			Will match the issuer DN.
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			

Field	Criticality Flag	Value	Comments
AttributeType		OID	
AttributeValue		see comment	See section 4.
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			Public key algorithm associated with the public key. May be either RSA or elliptic curve.
algorithm		1.2.840.113549.1.1.1	RSA
		1.2.840.10045.2.1	Elliptic Curve
parameters		See comment	For RSA include NULL. For ECC include parameters.
subjectPublicKey		BIT STRING	For RSA, modulus must be at least 2048 bits. For ECC, public key must be at least 224 bits.
required extensions			
subjectKeyIdentifier	FALSE		This extension is required to assist in path development.
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the public key.
subjectInfoAccess	FALSE		subjectInfoAccess consists of a sequence of accessMethod and accessLocation pairs. Only one access method is defined for use in CA certificates.
AccessDescription			
accessMethod		id-ad-caRepository (1.3.6.1.5.5.7.48.5)	Self-signed certificates must include at least one instance of this access method that includes the URI name form to specify the location of an LDAP accessible directory server. Certificates may also include a URI name form to specify an HTTP accessible Web server. Each URI should point to a location where CA certificates issued by the subject of this certificate may be found.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
basicConstraints	TRUE		The contents of this extension are not used in the X.509 path validation algorithm. Path length constraints should not be included since they may not be enforced.
cA		TRUE	
keyUsage	TRUE		The contents of this extension are not used in the X.509 path validation algorithm. If the subject public key may be used for purposes other than certificate and CRL signing (e.g., signing OCSP responses), then the digitalSignature and/or nonRepudiation bits may be set as well.
digitalSignature		0	
nonRepudiation		0	
keyEncipherment		0	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		1	
cRLSign		1	
encipherOnly		0	
decipherOnly		0	
optional extensions			

Field	Criticality Flag	Value	Comments
issuerAltName	FALSE		Any name types may be present; only the most common is specified here.
GeneralNames			
GeneralName			
rfc822Name		IA5String	Electronic mail address of the PKI administration.
Designed by Robert Moskowitz (ICSA) and modified by Booz Allen & Hamilton and NIST			

Worksheet 2: Key Rollover CA Certificate Profile

Field	Criticality Flag	Value	Comments
Certificate			
tbsCertificate			Fields to be signed.
version		2	Integer value of "2" for version 3 certificate.
serialNumber		INTEGER	Unique positive integer.
signature			
AlgorithmIdentifier			Must match Algorithm Identifier in signatureAlgorithm field. The parameters field is only populated when the algorithm is RSA.
algorithm		Choice of following algorithms:	
		1.2.840.113549.1.1.5	sha-1WithRSAEncryption
		1.2.840.113549.1.1.10	id-RSASSA-PSS (RSA with PSS padding; 800-78 requires use with SHA-256 hash algorithm)
		1.2.840.113549.1.1.11	sha256WithRSAEncryption
		1.2.840.10040.4.3	id-dsa-with-sha1
		1.2.840.10045.4.1	ecdsa-with-SHA1
		1.2.840.10045.4.3.1	ecdsa-with-Sha224
		1.2.840.10045.4.3.2	ecdsa-with-Sha256
		1.2.840.10045.4.3.3	ecdsa-with-Sha384
		1.2.840.10045.4.3.4	ecdsa-with-Sha512
parameters		2.16.840.1.101.3.4.2.1 or 2.16.840.1.101.3.4.2.3	For id-RSASSA-PSS only, specify either the SHA-256 or SHA-512 hash algorithm as a parameter
		NULL	For all RSA algorithms except id-RSASSA-PSS
issuer			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
validity			
notBefore			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
notAfter			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
subject			
Name			Subject name should be encoded exactly as it is encoded in the issuer field of this certificate.

Field	Criticality Flag	Value	Comments
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	Encoding of name must use the encoding of the issuer field in certificates and CRLs issued by this subject CA.
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			Public key algorithm associated with the public key. May be RSA, DSA, or elliptic curve.
algorithm		1.2.840.113549.1.1.1	RSA
		1.2.840.10040.4.1	DSA
		1.2.840.10045.2.1	Elliptic Curve
parameters		See comment	For RSA include NULL; for DSA and ECC include parameters if subject and issuer have different parameters. If parameters are inherited, omit field for DSA and include NULL for ECC.
subjectPublicKey		BIT STRING	
required extensions			
authorityKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the signer's public key.
subjectKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the subject public key.
keyUsage	TRUE		If the subject public key may be used for purposes other than certificate and CRL signing (e.g., signing OCSP responses), then the digitalSignature and/or nonRepudiation bits may be set as well.
digitalSignature		0	
nonRepudiation		0	
keyEncipherment		0	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		1	
cRLSign		1	
encipherOnly		0	
decipherOnly		0	
certificatePolicies	FALSE		
PolicyInformation			The inclusion of policy qualifiers is discouraged.
policyIdentifier		OID	
basicConstraints	TRUE		This extension must appear in all CA certificates. The pathLenConstraint field should not appear in self-issued certificates.

Field	Criticality Flag	Value	Comments
cA		TRUE	
cRLDistributionPoints	FALSE		This extension must appear in all certificates and must include at least an LDAP URI distribution point name. This profile recommends against the use of indirect CRLs or CRLs segmented by reason code.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
authorityInfoAccess	FALSE		authorityInfoAccess consists of a sequence of accessMethod and accessLocation pairs. Two access methods are defined: one for locating certificates issued to the certificate issuer and one for locating an OCSP server that provides status information about this certificate.
AccessDescription			Access Method #1
accessMethod		id-ad-caIssuers (1.3.6.1.5.5.7.48.2)	All certificates must include at least one instance of this access method that uses the URI name form to specify the location of an LDAP accessible directory server where certificates issued to the issuer of this certificate may be found. Certificates may also include an instance of this access method that uses the URI name form to specify the location of an HTTP accessible Web server.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
AccessDescription			Access Method #2
accessMethod		id-ad-ocsp (1.3.6.1.5.5.7.48.1)	For this access method, the access location should use the URI name form to specify the location of an OCSP server that provides status information about this certificate.
accessLocation			
GeneralName			
uniformResourceIdentifier		http://…	See section 5.
subjectInfoAccess	FALSE		subjectInfoAccess consists of a sequence of accessMethod and accessLocation pairs. Only one access methods is defined for use in CA certificates.

Field	Criticality Flag	Value	Comments
AccessDescription			
accessMethod		id-ad-caRepository (1.3.6.1.5.5.7.48.5)	All CA certificates must include at least one instance of this access method that uses the URI name form to specify the location of an LDAP accessible directory server where CA certificates issued by the subject of this certificate may be found. CA certificates may also include an instance of this access method that uses the URI name form to specify the location of an HTTP accessible Web server.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
optional extensions			
issuerAltName	FALSE		Any name types may be present; only the most common is specified here.
GeneralNames			
GeneralName			
rfc822Name		IA5String	Electronic mail address of the PKI administration
FreshestCRL	FALSE		If delta-CRLs are issued that cover this certificate then either this certificate or the complete for scope CRLs that correspond to the delta-CRLs should include the FreshestCRL extension.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
Designed by Robert Moskowitz (ICSA) and modified by Booz Allen & Hamilton and NIST			

Worksheet 3: Cross-Certificate Profile

Field	Criticality Flag	Value	Comments
Certificate			
tbsCertificate			Fields to be signed.
version		2	Integer value of "2" for version 3 certificate.
serialNumber		INTEGER	Unique positive integer.
signature			
AlgorithmIdentifier			Must match Algorithm Identifier in signatureAlgorithm field. The parameters field is only populated when the algorithm is RSA.
algorithm		Choice of following algorithms:	
		1.2.840.113549.1.1.5	sha-1WithRSAEncryption
		1.2.840.113549.1.1.10	id-RSASSA-PSS (RSA with PSS padding; 800-78 requires use with SHA-256 hash algorithm)
		1.2.840.113549.1.1.11	sha256WithRSAEncryption
		1.2.840.10040.4.3	id-dsa-with-sha1
		1.2.840.10045.4.1	ecdsa-with-SHA1
		1.2.840.10045.4.3.1	ecdsa-with-Sha224
		1.2.840.10045.4.3.2	ecdsa-with-Sha256
		1.2.840.10045.4.3.3	ecdsa-with-Sha384
		1.2.840.10045.4.3.4	ecdsa-with-Sha512
parameters		2.16.840.1.101.3.4.2.1 or 2.16.840.1.101.3.4.2.3	For id-RSASSA-PSS only, specify either the SHA-256 or SHA-512 hash algorithm as a parameter
		NULL	For all RSA algorithms except id-RSASSA-PSS
issuer			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
validity			
notBefore			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
notAfter			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
subject			
Name			X.500 Distinguished name of the owner of the subject public key in the certificate. Subject name should be encoded exactly as it is encoded in the

Field	Criticality Flag	Value	Comments
			issuer field of certificates an issued by the subject.
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	Encoding of name must use the encoding of the issuer field in certificates and CRLs issued by this subject CA.
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			Public key algorithm associated with the public key. May be RSA, DSA, or elliptic curve.
algorithm		1.2.840.113549.1.1.1	RSA
		1.2.840.10040.4.1	DSA
		1.2.840.10045.2.1	Elliptic Curve
parameters		See comment	For RSA include NULL; for DSA and ECC include parameters if subject and issuer have different parameters. If parameters are inherited, omit field for DSA and include NULL for ECC.
subjectPublicKey		BIT STRING	
required extensions			
authorityKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the signer's public key.
subjectKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	The value in this field must be the same as the value that the subject CA uses in the authority key identifier extension of the certificates and CRLs that it signs with the private key that corresponds to the subject public key included in this certificate.
keyUsage	TRUE		If the subject public key may be used for purposes other than certificate and CRL signing (e.g., signing OCSP responses), then the digitalSignature and/or nonRepudiation bits may be set as well.
digitalSignature		0	
nonRepudiation		0	
keyEncipherment		0	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		1	Always asserted in CA certificates.
cRLSign		1	Asserted if this key is also used to sign CRLs.
encipherOnly		0	
decipherOnly		0	
certificatePolicies	FALSE		
PolicyInformation			The inclusion of policy qualifiers is discouraged.
policyIdentifier		OID	

Field	Criticality Flag	Value	Comments
basicConstraints	TRUE		This extension must appear in all CA certificates.
cA		TRUE	
pathLenConstraint		INTEGER	The use of a path length constraint is optional.
cRLDistributionPoints	FALSE		This extension must appear in all certificates and must include at least an LDAP URI distribution point name. This profile recommends against the use of indirect CRLs or CRLs segmented by reason code.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
authorityInfoAccess	FALSE		authorityInfoAccess consists of a sequence of accessMethod and accessLocation pairs. Two access methods are defined: one for locating certificates issued to the certificate issuer and one for locating an OCSP server that provides status information about this certificate.
AccessDescription			Access Method #1
accessMethod		id-ad-caIssuers (1.3.6.1.5.5.7.48.2)	All certificates must include at least one instance of this access method that uses the URI name form to specify the location of an LDAP accessible directory server where certificates issued to the issuer of this certificate may be found. Certificates may also include an instance of this access method that uses the URI name form to specify the location of an HTTP accessible Web server.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
AccessDescription			Access Method #2
accessMethod		id-ad-ocsp (1.3.6.1.5.5.7.48.1)	For this access method, the access location should use the URI name form to specify the location of an OCSP server that provides status information about this certificate.
accessLocation			
GeneralName			
uniformResourceIdentifier		http://…	See section 5.
subjectInfoAccess	FALSE		subjectInfoAccess consists of a sequence of

Field	Criticality Flag	Value	Comments
			accessMethod and accessLocation pairs. Only one access methods is defined for use in CA certificates.
AccessDescription			
accessMethod		id-ad-caRepository (1.3.6.1.5.5.7.48.5)	All CA certificates must include at least one instance of this access method that uses the URI name form to specify the location of an LDAP accessible directory server where CA certificates issued by the subject of this certificate may be found. CA certificates may also include an instance of this access method that uses the URI name form to specify the location of an HTTP accessible Web server.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
optional extensions			
issuerAltName	FALSE		Any name types may be present; only the most common is specified here.
GeneralNames			
GeneralName			
rfc822Name		IA5String	Electronic mail address of the PKI administration.
policyMappings	See comment		This extension may appear in a CA certificate. This extension may be set to noncritical to support legacy applications that cannot process the policy mappings extension.
issuerDomainPolicy		OID	OID of policy from issuing CA's domain that maps to the equivalent policy in the subject CA's domain.
subjectDomainPolicy		OID	OID of policy in the subject CA's domain that may be accepted in lieu of the issuer domain policy (above).
policyConstraints	See comment		If this extension appears, at least one of requireExplicitPolicy or inhibitPolicyMapping must be present. When present, this extension should be marked as critical.
requireExplicitPolicy			
SkipCerts		INTEGER	
inhibitPolicyMapping			Should be included if local policy prohibits policy mapping.
SkipCerts		INTEGER	
nameConstraints	TRUE		If present, any combination of permitted and excluded subtrees may appear. If permitted and excluded subtrees overlap, the excluded subtree takes precedence. It is recommended that name constraints only be imposed on the directoryName name form.
permittedSubtrees			Minimum is always zero. Maximum is never present.
GeneralSubtrees			
GeneralSubtree			
base			

Field	Criticality Flag	Value	Comments
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
minimum		0	
excludedSubtrees			Minimum is always zero. Maximum is never present.
GeneralSubtrees			
GeneralSubtree			
base			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
minimum		0	
FreshestCRL	FALSE		If delta-CRLs are issued that cover this certificate, then either this certificate or the complete for scope CRLs that correspond to the delta-CRLs should include the FreshestCRL extension.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
Designed by Robert Moskowitz (ICSA) and modified by Booz Allen & Hamilton and NIST			

Worksheet 4: CRL Profile

Field	Criticality Flag	Value	Comments
CertificateList			
tbsCertList			Fields to be signed.
version		1	Integer value of "1" for version 2 CRL. (CAs that are not capable of issuing version 2 CRLs may issue version 1 CRLs. Other than the omission of extensions all other fields should be populated as specified in this profile.)
signature			
AlgorithmIdentifier			Must match Algorithm Identifier in signatureAlgorithm field. The parameters field is only populated when the algorithm is RSA.
algorithm		Choice of following algorithms:	
		1.2.840.113549.1.1.5	sha-1WithRSAEncryption
		1.2.840.113549.1.1.10	id-RSASSA-PSS (RSA with PSS padding; 800-78 requires use with SHA-256 hash algorithm)
		1.2.840.113549.1.1.11	sha256WithRSAEncryption
		1.2.840.10040.4.3	id-dsa-with-sha1
		1.2.840.10045.4.1	ecdsa-with-SHA1
		1.2.840.10045.4.3.1	ecdsa-with-Sha224
		1.2.840.10045.4.3.2	ecdsa-with-Sha256
		1.2.840.10045.4.3.3	ecdsa-with-Sha384
		1.2.840.10045.4.3.4	ecdsa-with-Sha512
parameters		2.16.840.1.101.3.4.2.1 or 2.16.840.1.101.3.4.2.3	For id-RSASSA-PSS only, specify either the SHA-256 or SHA-512 hash algorithm as a parameter
		NULL	For all RSA algorithms except id-RSASSA-PSS
issuer			
Name			Issuer name should be encoded exactly as it is encoded in the issuer fields of the certificates that are covered by this CRL (or the cRLIssuer field of the cRLDistributionPoints extensions in the case of indirect CRLs).
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		See Comment.	See section 4.
thisUpdate			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
nextUpdate			RFC 3280 requires all CRLs to include a nextUpdate time.
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.

Field	Criticality Flag	Value	Comments
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
revokedCertificates			
userCertificate		INTEGER	Serial number of certificate being revoked
revocationDate			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
crlEntryExtensions			
Extensions			The inclusion of CRL entry extensions is optional.
reasonCode	FALSE		This extension may be included if the issuer wishes to provide information about the reason that a certificate was revoked.
CRLReason			Any one of these CRL reasons may be asserted: keyCompromise, cAcompromise, affiliationChanged, superseded, or cessationOfOperation. If the revocation reason is unspecified, then the reasonCode extension should not be included. The removeFromCRL reason code may only be used in delta CRLs and the use of certificateHold is discouraged.
invalidtyDate	FALSE		This extension may be included if the invalidity date precedes the revocation date.
GeneralizedTime		YYYYMMDDHHMMSSZ	Use this format for all dates.
crlExtensions			
Extensions			
authorityKeyIdentifier	FALSE		Must be included in all CRLs.
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the signer's public key.
cRLNumber	FALSE	INTEGER	Monotonically increasing sequential number. Must be included in all CRLs.
issuingDistributionPoint	TRUE		This extension appears in segmented CRLs. If the CRL covers all unexpired certificates issued by the CRL issuer (i.e., all unexpired certificates in which the issuer field contains the same name as the issuer field of the CRL), then this extension does not need to be included. This profile recommends against the use of indirect CRLs or CRLs that do not cover all reason codes.
distributionPoint			
DistributionPointName			If the issuer generates segmented CRLs (i.e., CRLs that do not cover all unexpired certificates in which the issuer field contains the same name as the issuer field in the CRL), this field must be present and must specify the same names as are specified in the distributionPoint field of the cRLDistributionPoints extensions of certificates covered by this CRL.
fullName			
GeneralNames			
GeneralName			
directoryName			

Field	Criticality Flag	Value	Comments
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		See comment.	See section 4.
uniformResourceIdentifier		IA5String	See section 5.
onlyContainsUserCerts		BOOLEAN	If set to TRUE, this CRL only covers end entity certificates. (NOTE: If onlyContainsUserCerts is set to TRUE and the CRL covers all end entity certificates issued by the issuer of this CRL, then the distributionPoint field may be omitted.)
onlyContainsCACerts		BOOLEAN	If set to TRUE, this CRL only covers CA certificates. If onlyContainsUserCerts is TRUE, this field must be FALSE. (NOTE: If onlyContainsCACerts is set to TRUE and the CRL covers all CA certificates issued by the issuer of this CRL, then the distributionPoint field may be omitted.)
IndirectCRL		FALSE	This profile recommends against the use of indirect CRLs. However, if this CRL covers certificates that were not issued by the issuer of this CRL, then this field must be set to TRUE.
FreshestCRL	FALSE		If this is a complete for scope CRL, and delta-CRLs are issued for the same scope, then either this CRL or the certificates that it covers should include the FreshestCRL extension. When the FreshestCRL extension is used in a CRL, only the distributionPoint field is used. The reasons and cRLIssuer fields must be omitted.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
deltaCRLIndicator	TRUE		This extension is included if and only if the CRL is a delta CRL.
BaseCRLNumber		INTEGER	This value shall be identical to the value in the cRLNumber extension of the base certificate.
Designed by Robert Moskowitz (ICSA) and modified by Booz Allen & Hamilton and NIST			

Worksheet 5: End Entity Signature Certificate Profile

Field	Criticality Flag	Value	Comments
Certificate			
tbsCertificate			Fields to be signed.
version		2	Integer value of "2" for version 3 certificate.
serialNumber		INTEGER	Unique positive integer.
signature			
AlgorithmIdentifier			Must match Algorithm Identifier in signatureAlgorithm field. The parameters field is only populated when the algorithm is RSA.
algorithm		Choice of following algorithms:	
		1.2.840.113549.1.1.5	sha-1WithRSAEncryption
		1.2.840.113549.1.1.10	id-RSASSA-PSS (RSA with PSS padding; 800-78 requires use with SHA-256 hash algorithm)
		1.2.840.113549.1.1.11	sha256WithRSAEncryption
		1.2.840.10040.4.3	id-dsa-with-sha1
		1.2.840.10045.4.1	ecdsa-with-SHA1
		1.2.840.10045.4.3.1	ecdsa-with-Sha224
		1.2.840.10045.4.3.2	ecdsa-with-Sha256
		1.2.840.10045.4.3.3	ecdsa-with-Sha384
		1.2.840.10045.4.3.4	ecdsa-with-Sha512
parameters		2.16.840.1.101.3.4.2.1 or 2.16.840.1.101.3.4.2.3	For id-RSASSA-PSS only, specify either the SHA-256 or SHA-512 hash algorithm as a parameter
		NULL	For all RSA algorithms except id-RSASSA-PSS
issuer			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
validity			
notBefore			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
notAfter			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
subject			
Name			X.500 Distinguished name of the owner of the subject public key in the certificate.

Field	Criticality Flag	Value	Comments
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			Public key algorithm associated with the public key. May be RSA, DSA, or elliptic curve.
algorithm		1.2.840.113549.1.1.1	RSA
		1.2.840.10040.4.1	DSA
		1.2.840.10045.2.1	Elliptic Curve
parameters		See comment	For RSA include NULL; for DSA and ECC include parameters if subject and issuer have different parameters. If parameters are inherited, omit field for DSA and include NULL for ECC.
subjectPublicKey		BIT STRING	
required extensions			
authorityKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the signer's public key.
subjectKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the subject public key.
keyUsage	TRUE		Assert either the digitalSignature bit, the nonRepudiation bit, or both bits.
digitalSignature		1	
nonRepudiation		1	
keyEncipherment		0	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		0	
cRLSign		0	
encipherOnly		0	
decipherOnly		0	
certificatePolicies	FALSE		
PolicyInformation			The inclusion of policy qualifiers is discouraged.
policyIdentifier		OID	
cRLDistributionPoints	FALSE		This extension must appear in all certificates and must include at least an LDAP URI distribution point name. This profile recommends against the use of indirect CRLs or CRLs segmented by reason code.
DistributionPoint			
distributionPoint			
DistributionPointName			

Field	Criticality Flag	Value	Comments
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
authorityInfoAccess	FALSE		authorityInfoAccess consists of a sequence of accessMethod and accessLocation pairs. Two access methods are defined: one for locating certificates issued to the certificate issuer and one for locating an OCSP server that provides status information about this certificate.
AccessDescription			Access Method #1
accessMethod		id-ad-caIssuers (1.3.6.1.5.5.7.48.2)	All certificates must include at least one instance of this access method that uses the URI name form to specify the location of an LDAP accessible directory server where certificates issued to the issuer of this certificate may be found. Certificates may also include an instance of this access method that uses the URI name form to specify the location of an HTTP accessible Web server.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
AccessDescription			Access Method #2
accessMethod		id-ad-ocsp (1.3.6.1.5.5.7.48.1)	For this access method, the access location should use the URI name form to specify the location of an OCSP server that provides status information about this certificate.
accessLocation			
GeneralName			
uniformResourceIdentifier		http://…	See section 5.
optional extensions			
extKeyUsage	BOOLEAN		This extension may be included as either a critical or non-critical extension if its inclusion is required by the application(s) for which the certificate will be used. If the inclusion of this extension is not intended to limit acceptable uses of the subject public key, then the extension should be marked non-critical and the anyExtendedKeyUsage value should be included.
KeyPurposeId		OID	
issuerAltName	FALSE		Any name types may be present; only the most common is specified here.
GeneralNames			

Field	Criticality Flag	Value	Comments
GeneralName			
rfc822Name		IA5String	Electronic mail address of the PKI administration.
subjectAltName	FALSE		Any name types may be present; only the most common are specified here. Other names may be included to support local applications.
GeneralNames			
GeneralName			
rfc822Name		IA5String	This field contains the e-mail address of the subject.
dNSName		IA5String	For devices, this field contains the DNS name of the subject.
iPAddress		IA5String	For devices, this field contains the IP address of the subject.
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
FreshestCRL	FALSE		If delta-CRLs are issued that cover this certificate, then either this certificate or the complete for scope CRLs that correspond to the delta-CRLs should include the FreshestCRL extension.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
Designed by Robert Moskowitz (ICSA) and modified by Booz Allen & Hamilton and NIST			

Worksheet 6: Key Management Certificate Profile

Field	Criticality Flag	Value	Comments
Certificate			
tbsCertificate			Fields to be signed.
version		2	Integer value of "2" for version 3 certificate.
serialNumber		INTEGER	Unique positive integer.
signature			
AlgorithmIdentifier			Must match Algorithm Identifier in signatureAlgorithm field. The parameters field is only populated when the algorithm is RSA.
algorithm		Choice of following algorithms:	
		1.2.840.113549.1.1.5	sha-1WithRSAEncryption
		1.2.840.113549.1.1.10	id-RSASSA-PSS (RSA with PSS padding; 800-78 requires use with SHA-256 hash algorithm)
		1.2.840.113549.1.1.11	sha256WithRSAEncryption
		1.2.840.10040.4.3	id-dsa-with-sha1
		1.2.840.10045.4.1	ecdsa-with-SHA1
		1.2.840.10045.4.3.1	ecdsa-with-Sha224
		1.2.840.10045.4.3.2	ecdsa-with-Sha256
		1.2.840.10045.4.3.3	ecdsa-with-Sha384
		1.2.840.10045.4.3.4	ecdsa-with-Sha512
parameters		2.16.840.1.101.3.4.2.1 or 2.16.840.1.101.3.4.2.3	For id-RSASSA-PSS only, specify either the SHA-256 or SHA-512 hash algorithm as a parameter
		NULL	For all RSA algorithms except id-RSASSA-PSS
issuer			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
validity			
notBefore			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
notAfter			
Time			
utcTime		YYMMDDHHMMSSZ	Use for dates up to and including 2049.
generalTime		YYYYMMDDHHMMSSZ	Use for dates after 2049.
subject			
Name			X.500 Distinguished name of the owner of the subject public key in the certificate.

Field	Criticality Flag	Value	Comments
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			Public key algorithm associated with the public key. May be RSA, Diffie-Hellman, elliptic curve, or KEA.
algorithm		1.2.840.113549.1.1.1	RSA
		1.2.840.10046.2.1	Diffie-Hellman
		1.2.840.10045.2.1	Elliptic Curve
		2.16.840.1.101.2.1.1.22	Key Exchange Algorithm
parameters		See comment	For RSA include NULL. For DH and KEA, parameters must always be included. For ECC include parameters if subject and issuer have different parameters; if parameters are inherited, use a value of NULL.
subjectPublicKey		BIT STRING	
required extensions			
authorityKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the signer's public key.
subjectKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Typically derived using the SHA-1 hash of the subject public key.
keyUsage	TRUE		As specified below according to algorithm.
digitalSignature		0	
nonRepudiation		0	
keyEncipherment		1	Used when subject public key is RSA.
dataEncipherment		0	
keyAgreement		1	Used when subject public key is DH, ECC, or KEA.
keyCertSign		0	
cRLSign		0	
encipherOnly		0	There is no requirement to support this key usage.
decipherOnly		0	There is no requirement to support this key usage.
certificatePolicies	FALSE		
PolicyInformation			The inclusion of policy qualifiers is discouraged.
policyIdentifier		OID	
cRLDistributionPoints	FALSE		This extension must appear in all certificates and must include at least an LDAP URI distribution point name. This profile recommends against the use of indirect CRLs or CRLs segmented by reason code.
DistributionPoint			
distributionPoint			

Field	Criticality Flag	Value	Comments
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
authorityInfoAccess	FALSE		authorityInfoAccess consists of a sequence of accessMethod and accessLocation pairs. Two access methods are defined: one for locating certificates issued to the certificate issuer and one for locating an OCSP server that provides status information about this certificate.
AccessDescription			Access Method #1
accessMethod		id-ad-caIssuers (1.3.6.1.5.5.7.48.2)	All certificates must include at least one instance of this access method that uses the URI name form to specify the location of an LDAP accessible directory server where certificates issued to the issuer of this certificate may be found. Certificates may also include an instance of this access method that uses the URI name form to specify the location of an HTTP accessible Web server.
accessLocation			
GeneralName			
uniformResourceIdentifier		ldap://… or http://…	See section 5.
AccessDescription			Access Method #2
accessMethod		id-ad-ocsp (1.3.6.1.5.5.7.48.1)	For this access method, the access location should use the URI name form to specify the location of an OCSP server that provides status information about this certificate.
accessLocation			
GeneralName			
uniformResourceIdentifier		http://…	See section 5.
optional extensions			
extKeyUsage	BOOLEAN		This extension may be included as either a critical or non-critical extension if its inclusion is required by the application(s) for which the certificate will be used. If the inclusion of this extension is not intended to limit acceptable uses of the subject public key, then the extension should be marked non-critical and the anyExtendedKeyUsage value should be included.
KeyPurposeId		OID	
issuerAltName	FALSE		Any name types may be present; only the most common is specified here.

Field	Criticality Flag	Value	Comments
GeneralNames			
GeneralName			
rfc822Name		IA5String	Electronic mail address of the PKI administration.
subjectAltName	FALSE		Any name types may be present; only the most common are specified here. Other names may be included to support local applications.
GeneralNames			
GeneralName			
rfc822Name		IA5String	This field contains the e-mail address of the subject
dNSName		IA5String	For devices, this field contains the DNS name of the subject
iPAddress		IA5String	For devices, this field contains the IP address of the subject
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
FreshestCRL	FALSE		If delta-CRLs are issued that cover this certificate, then either this certificate or the complete for scope CRLs that correspond to the delta-CRLs should include the FreshestCRL extension.
DistributionPoint			
distributionPoint			
DistributionPointName			
fullName			
GeneralNames			
GeneralName			
directoryName			
Name			
RDNSequence			
RelativeDistinguishedName			
AttributeTypeAndValue			
AttributeType		OID	
AttributeValue		see comment	See section 4.
uniformResourceIdentifier		ldap://… or http://…	See section 5.
Designed by Robert Moskowitz (ICSA) and modified by Booz Allen & Hamilton and NIST			