Government of Canada
Public Key Infrastructure
(GoC PKI)

GOC PKI X.509 Certificate and CRL Fields and Extensions Profile

Draft Version 2.0, 26 October 1999

This document is in draft form and is subject to revision

Table of contents

1 Introduction....................1

1.1 Background....................1
1.2 Overview....................1

2 X.509 v3 Certificates....................3

2.1 Introduction....................3
2.2 Base X.509 Certificate Processing....................3
2.2.1 Base Certificate Settings....................3
2.2.2 Base Certificate Processing....................5
2.2.3 Time Format....................5
2.3 Certificate Extensions....................5
2.3.1 authorityKeyIdentifier....................5
2.3.1.1 ASN.1 Syntax....................6
2.3.1.2 Extension source and control in the GOC PKI....................6
2.3.1.3 Generation Requirements....................7
2.3.1.4 Processing Requirements....................7
2.3.2 subjectKeyIdentifier....................7
2.3.2.1 ASN.1 Syntax....................7
2.3.2.2 Extension source and control in the GOC PKI....................7
2.3.2.3 Generation Requirements....................7
2.3.2.4 Processing Requirements....................8
2.3.3 keyUsage....................8
2.3.3.1 ASN.1 Syntax....................8
2.3.3.2 Extension source and control in the GOC PKI....................8
2.3.3.3 Generation Requirements....................9
2.3.3.4 Processing Requirements....................9
2.3.4 extKeyUsage....................9
2.3.4.1 ASN.1 Syntax....................9
2.3.4.2 Extension source and control in the GOC PKI....................10
2.3.4.3 Generation Requirements....................10
2.3.4.4 Processing Requirements....................10
2.3.5 privateKeyUsagePeriod....................10
2.3.5.1 ASN.1 Syntax....................11
2.3.5.2 Extension source and control in the GOC PKI....................11
2.3.5.3 Generation Requirements....................11
2.3.5.4 Processing Requirements....................11
2.3.6 certificatePolicies....................11
2.3.6.1 ASN.1 Syntax....................11
2.3.6.2 Extension source and control in the GOC PKI....................12
2.3.6.3 Generation Requirements....................12
2.3.6.4 Processing Requirements....................12
2.3.7 policyMappings....................13
2.3.7.1 ASN.1 Syntax....................13
2.3.7.2 Extension source and control in the GOC PKI....................13
2.3.7.3 Generation Requirements....................13
2.3.7.4 Processing Requirements....................13
2.3.8 privateVersInfo....................13
2.3.8.1 ASN.1 Syntax....................13
2.3.8.2 Extension source and control in the GOC PKI....................14
2.3.8.3 Generation Requirements....................14
2.3.8.4 Processing Requirements....................14

2.3.9 subjectAltName.. 14
2.3.9.1 ASN.1 Syntax.. 14
2.3.9.2 Extension source and control in the GOC PKI.. 14
2.3.9.3 Generation Requirements.. 15
2.3.9.4 Processing Requirements.. 15
2.3.10 issuerAltName.. 15
2.3.10.1 ASN.1 Syntax.. 15
2.3.10.2 Extension source and control in the GOC PKI.. 16
2.3.10.3 Generation Requirements.. 16
2.3.10.4 Processing Requirements.. 16
2.3.11 subjectDirectoryAttributes.. 16
2.3.11.1 ASN.1 Syntax.. 16
2.3.11.2 Extension source and control in the GOC PKI.. 16
2.3.11.3 Generation Requirements.. 17
2.3.11.4 Processing Requirements.. 17
2.3.12 basicConstraints.. 17
2.3.12.1 ASN.1 Syntax.. 17
2.3.12.2 Extension source and control in the GOC PKI.. 17
2.3.12.3 Generation Requirements.. 17
2.3.12.4 Processing Requirements.. 17
2.3.13 nameConstraints.. 18
2.3.13.1 ASN.1 Syntax.. 18
2.3.13.2 Extension source and control in the GOC PKI.. 18
2.3.13.3 Generation Requirements.. 19
2.3.13.4 Processing Requirements.. 19
2.3.14 policyConstraints.. 19
2.3.14.1 ASN.1 Syntax.. 19
2.3.14.2 Extension source and control in the GOC PKI.. 19
2.3.14.3 Generation Requirements.. 19
2.3.14.4 Processing Requirements.. 19
2.3.15 cRLDistributionPoints.. 19
2.3.15.1 ASN.1 Syntax.. 20
2.3.15.2 Extension source and control in the GOC PKI.. 20
2.3.15.3 Generation Requirements.. 21
2.3.15.4 Processing Requirements.. 21
2.3.16 authorityInfoAccess.. 21
2.3.16.1 ASN.1 Syntax.. 21
2.3.16.2 Extension source and control in the GOC PKI.. 22
2.3.16.3 Generation Requirements.. 22
2.3.16.4 Processing Requirements.. 22

3 X.509 v2 CRLs.. 23

3.1 CRL Extensions.. 24
3.1.1 authorityKeyIdentifier.. 24
3.1.1.1 ASN.1 Syntax.. 24
3.1.1.2 Extension source and control in the GOC PKI.. 25
3.1.1.3 Generation Requirements.. 25
3.1.1.4 Processing Requirements.. 25
3.1.2 issuerAltName.. 25
3.1.2.1 ASN.1 Syntax.. 25
3.1.2.2 Extension source and control in the GOC PKI.. 26
3.1.2.3 Generation Requirements.. 26
3.1.2.4 Processing Requirements.. 26
3.1.3 cRLNumber.. 26

3.1.3.1 ASN.1 Syntax 26
3.1.3.2 Extension source and control in the GOC PKI 26
3.1.3.3 Generation Requirements 26
3.1.3.4 Processing Requirements 27
3.1.4 deltaCRLIndicator 27
3.1.4.1 ASN.1 Syntax 27
3.1.4.2 Extension source and control in the GOC PKI 27
3.1.4.3 Generation Requirements 27
3.1.4.4 Processing Requirements 27
3.1.5 issuingDistributionPoint 27
3.1.5.1 ASN.1 Syntax 27
3.1.5.2 Extension source and control in the GOC PKI 28
3.1.5.3 Generation Requirements 28
3.1.5.4 Processing Requirements 28
3.2 CRL Entry Extensions 29
3.2.1 reasonCode 29
3.2.1.1 ASN.1 Syntax 29
3.2.1.2 Extension source and control in the GOC PKI 29
3.2.1.3 Generation Requirements 29
3.2.1.4 Processing Requirements 29
3.2.2 holdInstructionCode 29
3.2.2.1 ASN.1 Syntax 29
3.2.2.2 Extension source and control in the GOC PKI 30
3.2.2.3 Generation Requirements 30
3.2.2.4 Processing Requirements 30
3.2.3 invalidityDate 30
3.2.3.1 ASN.1 Syntax 30
3.2.3.2 Extension source and control in the GOC PKI 30
3.2.3.3 Generation Requirements 30
3.2.3.4 Processing Requirements 30
3.2.4 certificateIssuer 30
3.2.4.1 ASN.1 Syntax 30
3.2.4.2 Extension source and control in the GOC PKI 31
3.2.4.3 Generation Requirements 31
3.2.4.4 Processing Requirements 31

4 PKIX Compliance 33

4.1 Minimum PKIX compliance 33
4.2 PKIX and X.509 Certificate Extension comparison 34
4.3 Extension support, recognition, and criticality 35
4.4 GOC PKI compliance to ASN.1 38

5 Glossary 55

6 References 57

List of tables

Table 1. Key usage combinations 9
Table 2. Minimum PKIX compliance 33
Table 3. Comparison of extension criticality 34
Table 4. Certification Authority: comparison of extension support 35
Table 5. End Entity: comparison of extension support 37
Table 6. Base certificate compliance 40

Table 7. Certificate extension compliance 42
Table 8. Base CRL compliance 49
Table 9. CRL extension/CRL entry extension compliance 51

1 Introduction

This document specifies the Government of Canada Public Key Infrastructure (GOC PKI) profile for X.509 v3 certificates and X.509 v2 Certificate Revocation Lists (CRL) as described in Section 6, Reference 1. Implementation guidance is provided for certificate generation entities (i.e., Certification Authorities (CAs)) and certificate processing entities (i.e., End Entities (EEs)).

Throughout this document the following terms and meanings are used:

- **GOC PKI CA** represents GOC PKI implementations that create public key certificates;
- **GOC PKI certificate processing entities** represent GOC PKI end entities that can process public key certificates; and
- **GOC PKI CRL processing entities** represent GOC PKI end entities that can process CRLs and ARLs.

1.1 Background

X.509 v3 certificates contain the identity and other 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 is appended numerous certificate extensions. This document describes those extensions that can be used in the GOC PKI. Detailed information about X.509 certificates can be found in Section 6, Reference 1.

1.2 Overview

The document is divided into five sections:

Section 1 (this section) provides introductory information.

Section 2 (X.509 v3 Certificates) describe X.509 v3 certificates and certificate extensions applicable to the GOC PKI and how they can be controlled by GOC PKI CAs and processed by GOC PKI certificate processing entities.

Section 3 (X.509 v2 CRLs) describes X.509 v2 CRLs and CRL extensions applicable to the GOC PKI and how they can be controlled by GOC PKI CAs and processed by GOC PKI CRL processing entities.

Section 4 (PKIX Compliance) provides a mapping between the PKIX Certificate and CRL Profile, Section 6, Reference 1, and the GOC PKI profile.

2 X.509 v3 Certificates

2.1 Introduction

CAs create certificates for user authentication and confidentiality public keys. So that users trust the public key, the CA employs a digital signature to cryptographically sign certificates and 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 or to control the trust placed in a certificate (see Section 2.3) depending on the implementation.

This document stipulates the required certificate and CRL format for GOC PKI-compliant programs. Any specific program implementing certificate-based public key cryptography, and claiming compliance to the GOC PKI requirements is required to tailor its X.509 certificates (as defined in Section 6, Reference 1) within the parameters outlined within this document. Through the remainder of this document, requirements for generation and processing of particular extensions are applied.

2.2 Base X.509 Certificate Processing

2.2.1 Base Certificate Settings

GOC PKI CAs shall, for all certificates:

1) include the **version field** with an integer value to indicate the certificate version.

- a value of 0 indicates a v1 certificate,
- a value of 1 indicates a v2 certificate, and
- a value of 2 indicates a v3 certificate;

2) include the **serialNumber field** with an integer value to indicate the certificate's serial number. This is always controlled by the CA;

3) include in the **signature field** the identifier (OID) of the algorithm used to sign the certificate and populate the parameters in this field for the DSA algorithm;

Note: May be specified by a PKIX-CMP client. If no algorithm is specified or PKIX-CMP is used, and the CA is set to sign with SHA-1, the algorithm will be set to **sha1WithRSAEncryption** if the CA key type is RSA and **dsa-with-sha1** if the CA key type is DSA.

If no algorithm is specified and the CA is set to sign with MD5 (CA key will be RSA) the default will be **md5WithRSAEncryption**.

If **dsa-with-sha1** is received and the CA key pair is RSA, the certificate will be signed with SHA-1 or MD5 (whichever is the default) and the algorithm will be changed to **md5WithRSAEncryption** or **sha1WithRSAEncryption**.

If **sha1WithRSAEncryption** is received and the CA key pair type is DSA, the certificate will be signed with SHA-1 and the algorithm will be changed to **dsa-with-sha1**.

- **md5WithRSAEncryption** (1.2.840.113549.1.1.4) for RSA/MD5 CA key pair;

- **sha1WithRSAEncryption** (1.2.840.113549.1.1.5) for RSA/SHA1 CA key pair; or
- **dsa-with-sha1** (1.2.840.10040.4.3) for DSA/SHA1 CA key pair.

4) include the **issuer field** with the X.500 distinguished name of the CA who created the certificate;

Note: May be specified by a PKIX-CMP client. If specified from a PKIX-CMP client, the issuer name must be that of the main CAs (or a virtual SET CA).

5) include the **validity field** with the time period for which the certificate is considered valid[1];

Note: For PKIX-CMP, the validity is either from the user specific settings (first) or the security policy (second). The PKIX-CMP client can specify the validity in the certificate template but the values will be overwritten if the PKIX-CMP user has custom validity periods or no key rollover set, or if the specified validity period is outside the allowed values or it is invalid.

The validity period in the certificate template will be used if the PKIX-CMP user is set to use the default security policy settings. The specified validity period must fall within the minimum and maximum values allowed by the GOC PKI CA. If the validity is longer or shorter than the maximum allowed, it will be lowered to the maximum or raised to the minimum, respectively, for the certificate type.

If the PKIX-CMP client does not specify a validity and if the user has custom settings, these will be used, otherwise the system default will be used for the specified certificate type.

The validity **notBefore** date is allowed to be up to 24 hours before the current time. When the **notAfter** date is calculated in this case, the validity period will be added to the current time, not the **notBefore** time.

6) include the **subject field** with the X.500 distinguished name of the subject to whom the certificate was issued;

Note: The subject can be filled in by the PKIX-CMP client but is not necessary. If it is filled in, it must match the DN in the PKIX message header (if any) or match the DN associated with the reference number in the PKIX message header.

7) The **subjectPublicKeyInfo** field consists of an algorithm identifier and a subject public key. The CA or a PKIX-CMP client can specify this value. Allowed values are:

- 1.2.840.113549.1.1.1 (**rsaEncryption**),
- 1.2.840.10040.4.1 (**dsa**), and
- 1.2.840.10045.2.1 (**ecdsa**);

Note: A PKIX-CMP request will be refused if **subjectPublicKeyInfo** is filled in but the algorithm and key type are not one of the above on the list. The **rsaEncryption** algorithm should have an ASN1 NULL for the parameters. The **dsa** algorithm should have parameters filled in (same for everyone) or it generates an error.

PKIX-CMP cannot specify a key size when no **subjectPublicKeyInfo** is sent. The CA will automatically generate an RSA 1024 bit encryption key pair when the **subjectPublicKeyInfo** is empty.

[1] Certificate validity dates through the year 2049 shall be encoded as UTCTime; certificate validity dates in 2050 or later shall be encoded as GeneralizedTime.

8) omit the **issuerUniqueIdentifier** field. The **subjectUniqueIdentifier** field may be specified by a PKIX-CMP client. The specified certificate version cannot be v1. No syntax checking is done on this field;

9) use the Parametrized Type as defined in X.500 to sign the certificate (the DSA parameters shall not be included in the SIGNED MACRO algorithm identifier field); and;

10) include the extensions field and extensions, as required, as described in Section 2.3.

2.2.2 Base Certificate Processing

Certificate path processing begins with a trusted public key and associated parameters which are obtained in a trusted manner. The trusted key must be maintained in a manner to insure it integrity.

For each certificate in the path, certificate processing entities shall:

- attempt to validate the signature of the certificate;
- process fields generated by a GOC PKI (or other) CA as identified in the certificate profiles; and
- process any extension fields (see Section 2.3), if present.

2.2.3 Time Format

The Distinguished Encoding Rules (DER) allow several methods for formatting UTCTime and GeneralizedTime. It is important that all implementations use the same format to minimize signature verification problems. To ensure that UTCTimes are consistently formatted, GOC PKI-compliant software must format all UTCTimes included in ASN.1 syntaxes that are encoded using the DER using the 'Z' format and must never omit the "seconds" field, even when it is '00' (i.e., the format shall be YYMMDDHHMMSSZ). The system shall interpret the year field, YY, as 19YY when YY is greater than or equal to 50, and 20YY when YY is less than 50. The GeneralizedTime type for this profile shall be expressed using the “Z” format and shall include seconds (i.e., the format shall be YYMMDDHHMMSSZ) but not fractional seconds.

2.3 Certificate Extensions

X.509 v3 certificates provide a mechanism for CAs to append additional information about the subject’s public key, issuer’s public key, issuer’s CRLs or to impose business controls (e.g., name constraints). Standard certificate extensions are defined for v3 X.509 certificates. It is not required that all the extensions be used by GOC PKI, however all the extensions are described here to ensure completeness. The following sections describe how these extensions are implemented.

An extension is flagged as being either critical or non-critical. If an extension is flagged critical and a certificate-using system does not recognize the extension field type or does not implement the semantics of the extension, then that system shall consider the certificate invalid. If an extension is flagged non-critical, a certificate-using system that does not recognize or implement that extension type may process the remainder of the certificate ignoring the extension.

2.3.1 authorityKeyIdentifier

This extension identifies the public key used to verify the signature on a certificate. It enables distinct keys used by the same CA to be differentiated. This extension may hold an explicit key identifier, or an explicit certificate identifier. This extension is useful when a CA uses more than one key (e.g., when the CA key is updated).

2.3.1.1 ASN.1 Syntax

```
authorityKeyIdentifier EXTENSION ::= {
        SYNTAX                         AuthorityKeyIdentifier
        IDENTIFIED BY                  id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
        authorityKeyIdentifier         [0] KeyIdentifer
        OPTIONAL,
        authorityCertIssuer            [1] GeneralNames            OPTIONAL,
        authorityCertSerialNumber      [2] CertificateSerialNumber OPTIONAL }
        ( WITH COMPONENTS { …, authorityCertIssuer PRESENT,
                                       authorityCertSerialNumber PRESENT } |
          WITH COMPONENTS { …, authorityCertIssuer ABSENT,
                                       authorityCertSerialNumber ABSENT } )
KeyIdentifier ::= OCTET STRING

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                      [0]      INSTANCE OF OTHER-NAME,
        rfc822Name                     [1]      IA5String,
        dNSName                        [2]      IA5String,
        x400Address                    [3]      ORAddress,
        directoryName                  [4]      Name,
        ediPartyName                   [5]      EDIPartyName,
        uniformResourceIdentifier      [6]      IA5String,
        iPAddress                      [7]      OCTET STRING,
        registeredID                   [8]      OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                   [0]      DirectoryString OPTIONAL,
        partyName                      [1]      DirectoryString }

DirectoryString ::= CHOICE {
        teletexString                  TeletexString (SIZE (1..MAX)),
        printableString                PrintableString (SIZE (1..MAX)),
        universalString                UniversalString (SIZE (1..MAX)),
        utf8String                     UTF8String (SIZE (1.. MAX)),
        bmpString                      BMPString (SIZE (1..MAX)) }

CertificateSerialNumber ::= INTEGER
```

2.3.1.2 Extension source and control in the GOC PKI

The **authorityKeyIdentifier extension** is controlled only by the CA. It can only be **Non-Critical**. It cannot be modified by any other means. If it is received from a PKIX-CMP client, it will be ignored and changed to the CA value. Alternative Settings may be used to exclude this extension from CA certificates as well as encryption and verification certificates in other certificate categories. Alternative Settings may also be used to insert an **authorityKeyIdentifier** as per the PKIX profile.

2.3.1.3 Generation Requirements

GOC PKI CAs shall:

- automatically include the extension in CA and EE certificates or, optionally, manually exclude the extension from self-signed CA certificates using Alternative Settings;
- automatically set the criticality flag to “false”;
- automatically exclude the **authorityCertIssuer** and **authorityCertSerialNumber** fields; and
- optionally include the **authorityKeyIdentifier** field as a 20 byte SHA-1 hash of the **subjectPublicKeyInfo** in the CA certificate or, using Alternative Settings, as a hash of the **subjectPublicKey** as per the PKIX profile.

2.3.1.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process the **authorityCertIssuer** and **authorityCertSerialNumber** fields.

2.3.2 subjectKeyIdentifier

This extension identifies the public key being certified. It enables distinct keys used by the same subject to be differentiated. This extension may hold the explicit key identifier, and is useful when a subject uses more than one key. This extension is required in the self-signed certificate as a result of the possibility that several Root CAs[2] will coexist.

2.3.2.1 ASN.1 Syntax

```
subjectKeyIdentifier EXTENSION ::= {
        SYNTAX                  SubjectKeyIdentifier
        IDENTIFIED BY                   id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

KeyIdentifier ::= OCTET STRING
```

2.3.2.2 Extension source and control in the GOC PKI

The **subjectKeyIdentifier extension** is controlled by the CA. It can only be **Non-Critical**. It cannot be modified by any other means. If it is received from a PKIX-CMP client, it will be ignored and changed to the CA value. Alternative Settings may also be used to insert a **subjectKeyIdentifier** as per the PKIX profile.

2.3.2.3 Generation Requirements

GOC PKI CAs shall:

- automatically include this extension in all certificates;
- automatically set the criticality flag to “false”; and
- optionally, include the **subjectKeyIdentifier** as a 20 byte SHA-1 hash of the **subjectPublicKeyInfo** in the certificate or, using Alternative Settings, as a hash of the **subjectPublicKey** as per the PKIX profile.

[2] A Root CA is a CA which acts as the trust anchor for all CAs below it in a hierarchy. A hierarchy is an inverted tree structure that contains superior and subordinate CAs. At the top of a hierarchy is a single “root” CA.

2.3.2.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.3 keyUsage

This extension indicates the purposes for which the certified public key is used. The **KeyUsage field** includes bit values used for digital signature verification (for purposes other than non-repudiation, certificates, or CRLs), digital signature for non-repudiation, enciphering keys or other security information, enciphering user data, a key agreement mechanism, a CA to sign certificates, a CA to sign CRLs, encipher only, and decipher only.

The scope of this section is restricted to general purpose public key certificates. Some specific protocols (e.g., SSL/TLS and IPSec require the use of public keys in a manner which is not recommended for general information processing). That is, they use the same public key pair for both signature generation and symmetric key management and this usage is generally deprecated for general purpose information security.

2.3.3.1 ASN.1 Syntax

```
keyUsage EXTENSION ::= {
        SYNTAX              KeyUsage
        IDENTIFIED BY               id-ce-keyUsage }

KeyUsage ::= BIT STRING {
        digitalSignature            (0),
        nonRepudiation      (1),
        keyEncipherment     (2),
        dataEncipherment    (3),
        keyAgreement                (4),
        keyCertSign         (5),
        cRLSign             (6),
        encipherOnly        (7),
        decipherOnly        (8) }
```

2.3.3.2 Extension source and control in the GOC PKI

The **keyUsage extension** is initially controlled by the CA. It can, however, also originate through Alternative Settings or over PKIX-CMP. The CA automatically inserts this extension into certificates set to **Non-Critical** but can be set to **Critical** using Alternative Settings. The extension value can be changed using Alternative Settings. Alternative Settings may be used to exclude this extension from CA and EE certificates. **keyUsage** values in Alternative Settings or over PKIX-CMP override the default CA setting. If **keyUsage** values exist in both PKIX-CMP requests and Alternative Settings, the values will be merged. The Alternative Settings has values for both encryption and signature certificates and will not allow a single **keyUsage** to be specified for both encryption and verification certificates.

2.3.3.3 Generation Requirements

GOC PKI CAs shall:

- automatically include the extension in all CA and EE certificates, or optionally, manually exclude this extension from CA and EE certificates using Alternative Settings;
- automatically set the criticality flag to “false”, or manually override the criticality using Alternative Settings;
- automatically set the **keyEncipherment bit** for encryption certificates and the **digitalSignature bit** for verification certificates, or manually override the value using Alternative Settings;
- automatically set the **keyCertSign bit** and **cRLSign bit** for CA certificates and cross-certificates, or manually override the values using Alternative Settings; and
- if FPKI compliance (client setting) is turned on, enforce valid FPKI **keyUsage** bit combinations, as listed in Table 1.

Table 1. Key usage combinations

#	Key Usages	Valid Combinations					
1.	digitalSignature				x*		
2.	nonRepudiation				x*		
3.	keyEncipherment	x					
4.	dataEncipherment		x				
5.	keyAgreement			x		(x)	(x)
6.	keyCertSign				x*		
7.	cRLSign				x*		
8.	encipherOnly**	-	-	-	-	- (x)	-
9.	decipherOnly**	-	-	-	-	-	- (x)

Note: * indicates that any subset combination of these key usages is valid.

Note: ** indicates that under FPKI compliance, there are no requirements to support these key usages.

2.3.3.4 Processing Requirements

GOC PKI certificate processing entities shall:

- when processing a certificate chain, if **keyUsage** is present in any certificate except the last one in a chain, the **keyCertSign** bit must be set, or halt the processing; and
- when processing cross-certificates, if the **keyUsage** extension is present and Critical, the **keyCertSign** bit must be set, or halt the processing.

2.3.4 extKeyUsage

This extension indicates one or more purposes for which the certified public key may be used in addition to or in place of the basic purposes indicated in the key usage extension.

2.3.4.1 ASN.1 Syntax

extKeyUsage EXTENSION ::= {

```
          SYNTAX                SEQUENCE SIZE (1..MAX) of KeyPurposeID
          IDENTIFIED BY         id-ce-extKeyUsage

KeyPurposeID ::= OBJECT IDENTIFIER

id-kp-serverAuth            OBJECT IDENTIFIER ::= {id-kp 1}
-- TLS Web server authentication
-- Key usage bits that may be consistent: digitalSignature, keyEncipherment, or
-- keyAgreement

id-kp-clientAuth            OBJECT IDENTIFIER ::= {id-kp 2}
-- TLS Web client authentication
-- Key usage bits that may be consistent: digitalSignature and/or keyAgreement

id-kp-codeSigning           OBJECT IDENTIFIER ::= {id-kp 3}
-- Signing of downloadable executable code
-- Key usage bits that may be consistent: digitalSignature

id-kp-emailProtection       OBJECT IDENTIFIER ::= {id-kp 4}
-- E-mail protection
-- Key usage bits that may be consistent: digitalSignature, nonRepudiation, and/or
-- keyEncipherment or keyAgreement

id-kp-profileKeyEncryption      OBJECT IDENTIFIER ::= {1 2 840 113533 7 74 1}
-- Profile server key encryption
-- Key usage bits that may be consistent: digitalSignature, nonRepudiation, and/or
-- keyEncipherment or keyAgreement

id-kp-timeStamping   OBJECT IDENTIFIER ::= {id-kp 8}
-- Binding the hash of an object to a time from an agreed-upon time source.
-- Key usage bits that may be consistent: digitalSignature, nonRepudiation, and/or
-- keyEncipherment or keyAgreement
```

2.3.4.2 Extension source and control in the GOC PKI

The **extKeyUsage** extension can be set through Alternative Settings or through PKIX-CMP. It can be either a **Critical** or **Non-Critical** extension.

2.3.4.3 Generation Requirements

GOC PKI CAs shall:

- optionally, manually set the value and criticality using Alternative Settings or PKIX-CMP.

2.3.4.4 Processing Requirements

GOC PKI certificate processing entities shall:

- only recognize and process the **timeStamping** and **profileKeyEncryption** extended key usages.

2.3.5 privateKeyUsagePeriod

This extension indicates the period of use of the private key corresponding to the certified public key. It is present only in verification certificates. This extension is used to compare the date of signature of a message to the validity period included within this extension.

2.3.5.1 ASN.1 Syntax

```
privateKeyUsagePeriod EXTENSION ::= {
        SYNTAX              PrivateKeyUsagePeriod
        IDENTIFIED BY               id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
        notBefore           [0]     GeneralizedTime OPTIONAL,
        notAfter            [1]     GeneralizedTime OPTIONAL }
        ( WITH COMPONENTS { ..., notBefore PRESENT } |
          WITH COMPONENTS { ..., notAfter PRESENT } )
```

2.3.5.2 Extension source and control in the GOC PKI

The **privateKeyUsagePeriod extension** value can be controlled by the security policy (set in GOC PKI/RA), user-specific settings (set in GOC PKI/RA), or through PKIX-CMP. The criticality is **Non-Critical** by default but can be changed to **Critical** through PKIX-CMP only. Alternative Settings may be used to exclude this extension from verification certificates only.

If set through PKIX-CMP, if the user has no key rollover set or a custom private key usage setting, the **privateKeyUsagePeriod extension** will be changed to the custom value. If the user is set to use the system default, the value received from PKIX-CMP will be used. If the **privateKeyUsagePeriod extension** is invalid, it will be changed by the CA.

2.3.5.3 Generation Requirements

GOC PKI CAs shall:

- automatically include this extension in EE verification certificates, or optionally, manually exclude the extension from EE verification certificates using Alternative Settings;
- automatically set the criticality flag to “false” or, optionally, set the criticality flag to “true” if set through PKIX-CMP; and
- ensure that if the **notBefore** date is set, that it is less than **notAfter**.

2.3.5.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.6 certificatePolicies

This extension lists certificate policies that the certificate is expressly recognized as supporting, together with optional qualifier information pertaining to these policies. The certificate policy indicates the procedures under which the certificate was created.

2.3.6.1 ASN.1 Syntax

```
certificatePolicies EXTENSION ::= {
        SYNTAX              CertificatePoliciesSyntax
        IDENTIFIED BY               id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
        policyIdentifier    CertPolicyId,
        policyQualifiers        SEQUENCE SIZE (1..MAX) OF
        PolicyQualifierInfo     OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
        policyQualifierId       CERT-POLICY-QUALIFIER.&id
                                        ({SupportedPolicyQualifiers}),
        qualifier               CERT-POLICY-QUALIFIER.&Qualifier

        ({SupportedPolicyQualifiers}{@policyQualifierId})
                                        OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }

CERT-POLICY-QUALIFIER ::= CLASS {
        &id                             OBJECT IDENTIFIER UNIQUE,
        &Qualifier                      OPTIONAL }
WITH SYNTAX {
        POLICY-QUALIFIER-ID  &id
        [QUALIFIER-TYPE &Qualifier] }
```

2.3.6.2 Extension source and control in the GOC PKI

The **certificatePolicies extension** is initially controlled through the security policy (value set in GOC PKI/RA) or in user-specific settings (value set in GOC PKI/RA). Only the **policyIdentifier field** can be set from GOC PKI/RA. The criticality can be set to **Critical** or **Non-Critical** through Alternative Settings. The value can be set through Alternative Settings and in a PKIX-CMP request (including the **policyIdentifier** and **policyQualifiers fields**).

No merging is done with the extension when there are conflicts. OIDs in an Alternative Settings or arriving in a PKIX-CMP request are not required to be in the master OID list. In the case where global security policy settings are to be used when creating a certificate, each category has a security policy setting which indicates if the global policy settings are to be used or not.

2.3.6.3 Generation Requirements

GOC PKI CAs shall:

- automatically include the OID(s) for the applicable certificate policy in the **policyIdentifier field(s)** in all CA and EE certificates if at least one certificate policy is included from GOC PKI/RA or, optionally, manually set values using Alternative Settings or through PKIX-CMP;
- automatically include the **PolicyInformation field(s)** with the applicable **policyIdentifier field(s)** if at least one certificate policy is included;
- automatically set the criticality flag to “false” or, optionally, manually override the criticality using Alternative Settings; and
- optionally, manually include the **policyQualifiers** field using Alternative Settings.

2.3.6.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.7 policyMappings

This extension, which is used in CA certificates only, allows a certificate issuer to indicate that, for the purposes of the user of a certification path containing this certificate, one of the issuer's certificate policies can be considered equivalent to a different certificate policy used in the subject CA's domain.

2.3.7.1 ASN.1 Syntax

```
policyMappings EXTENSION ::= {
        SYNTAX                  PolicyMappingsSyntax
        IDENTIFIED BY           id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
        issuerDomainPolicy      CertPolicyId,
        subjectDomainPolicy     CertPolicyId }
```

2.3.7.2 Extension source and control in the GOC PKI

The **policyMappings extension** value and criticality can be controlled only through Alternative Settings. The criticality can be set to **Critical** or **Non-Critical**.

2.3.7.3 Generation Requirements

GOC PKI CAs shall:

- optionally, manually include this extension in cross-certificates using Alternative Settings; and
- optionally, manually set values and criticality using Alternative Settings.

2.3.7.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.8 privateVersInfo

This extension is a private extension that indicates the version of the GOC PKI CA software and flags indicating if user key update is allowed and the user’s category.

2.3.8.1 ASN.1 Syntax

```
privateVersInfo EXTENSION ::= {
        SYNTAX                  PrivateVersInfoSyntax
        IDENTIFIED BY           id-ce-privateVersInfo }

PrivateVersInfoSyntax ::= SEQUENCE {
        privateVers             GeneralString
        privateVersInfoFlags    PrivateInfoFlags }

PrivateInfoFlags ::= BIT STRING {
        keyUpdateAllowed        (0),
        obsolete1               (1),
        obsolete2               (2),
        enterpriseCategory      (3),
        webCategory             (4),
        SETCategory             (5) }
```

2.3.8.2 Extension source and control in the GOC PKI

The **privateVersInfo extension** is controlled only the CA. It cannot be modified by any other means. This extension can only be **Non-Critical**.

2.3.8.3 Generation Requirements

GOC PKI CAs shall:

- automatically include the extension in all certificates and set the criticality flag to “false”; and
- automatically exclude the **obsolete1** and **obsolete2** flags.

2.3.8.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.9 subjectAltName

This extension provides one or more names that are bound by the CA to the subject’s certified public key.

2.3.9.1 ASN.1 Syntax

```
subjectAltName EXTENSION ::= {
        SYNTAX                  GeneralNames
        IDENTIFIED BY                   id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                       [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                      [1]     IA5String,
        dNSName                         [2]     IA5String,
        x400Address                     [3]     ORAddress,
        directoryName                   [4]     Name,
        ediPartyName                    [5]     EDIPartyName,
        uniformResourceIdentifier       [6]     IA5String,
        iPAddress                       [7]     OCTET STRING,
        registeredID                    [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                    [0]     DirectoryString OPTIONAL,
        partyName                       [1]     DirectoryString }

DirectoryString ::= CHOICE {
        teletexString                   TeletexString (SIZE (1..MAX)),
        printableString                 PrintableString (SIZE (1..MAX)),
        universalString                 UniversalString (SIZE (1..MAX)),
        utf8String                      UTF8String (SIZE (1.. MAX)),
        bmpString                       BMPString (SIZE (1..MAX)) }
```

2.3.9.2 Extension source and control in the GOC PKI

The **subjectAltName extension** can be controlled by the CA and has a default criticality of **Non-Critical** if populated through GOC PKI/RA, but its criticality may be changed to **Critical** through Alternative Settings for all certificates. For end-entity certificates, the value can be specified through PKIX-CMP or through user-specific settings set in GOC

PKI/RA. For CA certificates and cross-certificates, the value can be specified through Alternative Settings.

An administrator can indicate if a specific email attribute type is to be automatically inserted into **subjectAltName** as an **rfc822Name** choice when a new user is added. If the user has a value in the specified X.500 attribute, the value of the attribute is stored as a **rfc822Name** choice in that user's **subjectAltName** data. The administrator could have also entered other values in the **subjectAltName** data. If the administrator did not enter any values of the **rfc822Name** type, the administrator entered value(s) and the **rfc822Name** from the directory attribute will be merged together. If the administrator did enter a value in the **rfc822Name** type, the value from the directory attribute will not be added to the user's **subjectAltName** data.

2.3.9.3 Generation Requirements

GOC PKI CAs shall:

- optionally, automatically generate this extension for EE certificates based on a specified X.500 directory attribute or, per user, manually generate this extension for EE certificates based on a specified email address;
- optionally, manually include this extension and set the value for CA certificates and cross-certificates using Alternative Settings;
- optionally, manually set the criticality flag to "true" using Alternative Settings; and
- be capable, from the GOC PKI/RA interface, of populating **GeneralName** only with the type **rfc822Name**.

2.3.9.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension when required (e.g., for name constraints processing).

2.3.10 issuerAltName

This extension provides a name, in a form other than that of distinguished name, for the certificate issuer.

2.3.10.1 ASN.1 Syntax

```
issuerAltName EXTENSION ::= {
        SYNTAX                  GeneralNames
        IDENTIFIED BY                   id-ce-issuerAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                       [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                      [1]     IA5String,
        dNSName                         [2]     IA5String,
        x400Address                     [3]     ORAddress,
        directoryName                   [4]     Name,
        ediPartyName                    [5]     EDIPartyName,
        uniformResourceIdentifier       [6]     IA5String,
        iPAddress                       [7]     OCTET STRING,
        registeredID                    [8]     OBJECT IDENTIFIER }
```

```
EDIPartyName ::= SEQUENCE {
        nameAssigner               [0]     DirectoryString OPTIONAL,
        partyName                  [1]     DirectoryString }

DirectoryString ::= CHOICE {
        teletexString              TeletexString (SIZE (1..MAX)),
        printableString            PrintableString (SIZE (1..MAX)),
        universalString            UniversalString (SIZE (1..MAX)),
        utf8String                 UTF8String (SIZE (1.. MAX)),
        bmpString                  BMPString (SIZE (1..MAX)) }
```

2.3.10.2 Extension source and control in the GOC PKI

The **issuerAltName extension** value and criticality can be controlled only through Alternative Settings. It can be set to be **Critical** or **Non-Critical**.

2.3.10.3 Generation Requirements

GOC PKI CAs shall:

- optionally, manually set the value and criticality using Alternative Settings; and
- optionally, generate this extension for all applicable certificates.

2.3.10.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension when required (e.g., for name constraints processing).

2.3.11 subjectDirectoryAttributes

This extension may convey any desired attribute values for the subject of the certificate. CAs not needing to convey authorizations in X.509 certificates need not populate the **subjectDirectoryAttributes** field.

2.3.11.1 ASN.1 Syntax

```
subjectDirectoryAttributes EXTENSION ::= {
        SYNTAX                  AttributesSyntax
        IDENTIFIED BY                   id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute

Attribute ::= privateUserRole

privateUserRole ::= INTEGER
```

2.3.11.2 Extension source and control in the GOC PKI

The **subjectDirectoryAttributes** extension value and criticality is controlled by the CA for GOC PKI/RA administrator certificates but can also be controlled through Alternative Settings. It can be set to be **Critical** or **Non-Critical**. If this extension is added using Alternative Settings, it is merged with **privateUserRole** attributes.

2.3.11.3 Generation Requirements

GOC PKI CAs shall:

- automatically, generate this extension for all GOC PKI/RA administrator certificates and set the criticality flag to "false" or, optionally, manually set the value and criticality using Alternative Settings.

2.3.11.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension (including the **privateUserRole** attribute to determine which policy certificate to retrieve at login).

2.3.12 basicConstraints

This extension indicates whether the subject may act as a CA using the certified public key to sign certificates. If so, a certification path length constraint may also be specified.

2.3.12.1 ASN.1 Syntax

```
basicConstraints EXTENSION ::= {
        SYNTAX                  BasicConstraintsSyntax
        IDENTIFIED BY                   id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
        cA                      BOOLEAN DEFAULT FALSE,
        pathLenConstraint       INTEGER (0..MAX) OPTIONAL }
```

2.3.12.2 Extension source and control in the GOC PKI

The **basicConstraints extension** is controlled by the CA and through Alternative Settings. The Alternative Settings and the CA always ensure that the **cA** boolean is set properly (i.e., False for user certificates and True for CA certificates). The CA does not fill in the **pathLenConstraint** value, this can only be set through Alternative Settings intended for cross-certificates. The Alternative Settings can change the criticality of the extension from the default **Non-Critical** to **Critical**. If this extension is received through PKIX-CMP, it will be ignored and changed to the value dictated by the CA or Alternative Settings. Alternative Settings may be used to exclude this extension from encryption and verification certificates.

2.3.12.3 Generation Requirements

GOC PKI CAs shall:

- automatically set the **cA** criticality flag to "false" in CA certificates and self-signed certificates or, optionally, manually set the criticality flag to "true" though Alternative Settings;
- automatically set the **cA** value to "true" in CA certificates and self-signed certificates;
- optionally, manually set the **pathLenConstraint** value through Alternative Settings; and
- optionally, manually exclude the extension using Alternative Settings.

2.3.12.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension;
- if both the **keyUsage** and **basicConstraints** extensions are present, and the **keyUsage keyCertSign** bit is set, the **cA** field must be set to “true” or the processing fails; and
- if **basicConstraints** is not present with the **cA** field set to “true” in all certificates except the last one, the processing fails.

2.3.13 nameConstraints

This extension, which is for use only in cross-certificates, indicates a name space within which all subject names in subsequent certificates in the certification path must be located.

2.3.13.1 ASN.1 Syntax

```
nameConstraints EXTENSION ::= {
        SYNTAX                          NameConstraintsSyntax
        IDENTIFIED BY                           id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
        permittedSubtrees               [0]     GeneralSubtrees OPTIONAL,
        excludedSubtrees                [1]     GeneralSubtrees OPTIONAL }
GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
        base                            GeneralName,
        minimum                         [0]     BaseDistance DEFAULT 0,
        maximum                         [1]     BaseDistance OPTIONAL }

GeneralName ::= CHOICE {
        otherName                       [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                      [1]     IA5String,
        dNSName                         [2]     IA5String,
        x400Address                     [3]     ORAddress,
        directoryName                   [4]     Name,
        ediPartyName                    [5]     EDIPartyName,
        uniformResourceIdentifier       [6]     IA5String,
        iPAddress                       [7]     OCTET STRING,
        registeredID                    [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                    [0]     DirectoryString OPTIONAL,
        partyName                       [1]     DirectoryString }

DirectoryString ::= CHOICE {
        teletexString                   TeletexString (SIZE (1..MAX)),
        printableString                 PrintableString (SIZE (1..MAX)),
        universalString                 UniversalString (SIZE (1..MAX)),
        utf8String                      UTF8String (SIZE (1.. MAX)),
        bmpString                       BMPString (SIZE (1..MAX)) }
BaseDistance ::= INTEGER (0..MAX)
```

2.3.13.2 Extension source and control in the GOC PKI

The **nameConstraints extension** value and criticality can be controlled only through Alternative Settings. It can be set to be **Critical** or **Non-Critical**.

2.3.13.3 Generation Requirements

GOC PKI CAs shall:

- automatically exclude this extension from EE certificates;
- optionally, include this extension in cross-certificates;
- optionally, manually set the criticality flag and value;
- optionally, manually set the values for the **permittedSubtrees** and **excludedSubtrees** fields; and
- optionally, include the appropriate integer in the **minimum** and **maximum** fields of **GeneralSubtree** to indicate the name space.

2.3.13.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.14 policyConstraints

This extension specifies constraints which may require explicit certificate policy identification or inhibit policy mapping for the remainder of the certification path.

2.3.14.1 ASN.1 Syntax

```
policyConstraints EXTENSION ::= {
        SYNTAX                  PolicyConstraintsSyntax
        IDENTIFIED BY                   id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
        requireExplicitPolicy   [0] SkipCerts OPTIONAL,
        inhibitPolicyMapping    [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)
```

2.3.14.2 Extension source and control in the GOC PKI

The **policyConstraints extension** value and criticality can be controlled only through Alternative Settings. It can be set to be **Critical** or **Non-Critical**.

2.3.14.3 Generation Requirements

GOC PKI CAs shall:

- automatically exclude this extension from EE certificates;
- optionally, include this extension in cross-certificates; and
- optionally, manually set the criticality flag and value.

2.3.14.4 Processing Requirements

GOC PKI certificate processing entities shall:

- recognize and process this extension.

2.3.15 cRLDistributionPoints

This extension identifies the CRL distribution point or points to which a certificate user should refer to ascertain if the certificate has been revoked.

2.3.15.1 ASN.1 Syntax

```
cRLDistributionPoints EXTENSION ::= {
        SYNTAX                          CRLDistPointsSyntax
        IDENTIFIED BY                           id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
        distributionPoint               [0]     DistributionPointName OPTIONAL,
        reasons                         [1]     ReasonFlags OPTIONAL,
        cRLIssuer                       [2]     GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
        fullName                        [0]     GeneralNames,
        nameRelativeToCRLIssuer         [1]     RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
        unspecified                     (0),
        keyCompromise                   (1),
        cACompromise                            (2),
        affiliationChanged              (3),
        superseded                      (4),
        cessationOfOperation            (5),
        certificateHold                         (6)
        removeFromCRL                   (8) }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                       [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                      [1]     IA5String,
        dNSName                         [2]     IA5String,
        x400Address                     [3]     ORAddress,
        directoryName                   [4]     Name,
        ediPartyName                    [5]     EDIPartyName,
        uniformResourceIdentifier       [6]     IA5String,
        iPAddress                       [7]     OCTET STRING,
        registeredID                    [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                    [0]     DirectoryString OPTIONAL,
        partyName                       [1]     DirectoryString }

DirectoryString ::= CHOICE {
        teletexString                   TeletexString (SIZE (1..MAX)),
        printableString                 PrintableString (SIZE (1..MAX)),
        universalString                 UniversalString (SIZE (1..MAX)),
        utf8String                      UTF8String (SIZE (1.. MAX)),
        bmpString                       BMPString (SIZE (1..MAX)) }
```

2.3.15.2 Extension source and control in the GOC PKI

The **cRLDistributionPoints extension** is controlled by the CA and its default criticality is **Non-Critical**. Its criticality can be controlled through Alternative Settings and can be set to **Critical** or **Non-Critical**. Alternative Settings may be used to exclude this extension from CA and EE certificates. If this is received through PKIX-CMP, it will be ignored and changed to the CA value.

2.3.15.3 Generation Requirements

GOC PKI CAs shall:

- automatically include this extension in all CA and EE certificates or, optionally, manually exclude this extension from CA and EE certificates using Alternative Settings;
- automatically set the criticality flag to “false” or, optionally, manually set the criticality flag to “true” using Alternative Settings;
- automatically set the **DistributionPointName** as a **directoryName** and, if enabled, as a **uniformResourceIndicator** for Microsoft Windows 2000 clients; and
- automatically exclude the **ReasonFlags** field.

2.3.15.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process the **ReasonFlags** field; and
- interpret a missing **DistributionPoint** as meaning the **DistributionPointName** defaults to the CRL issuer name (i.e., CRL retrieved from issuing CA directory entry).

2.3.16 authorityInfoAccess

The **authorityInfoAccess** extension indicates how to access CA information and services for the issuer of the certificate in which the extension appears. Information and services may include on-line validation services and CA policy data.

2.3.16.1 ASN.1 Syntax

```
authorityInfoAccess EXTENSION ::= {
        SYNTAX                  AuthorityInfoAccessSyntax
        IDENTIFIED BY                   id-ce-authorityInfoAccess }

AuthorityInfoAccessSyntax  ::=
        SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription  ::=  SEQUENCE {
        accessMethod            OBJECT IDENTIFIER,
        accessLocation          GeneralName  }

GeneralName ::= CHOICE {
        otherName                       [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                      [1]     IA5String,
        dNSName                         [2]     IA5String,
        x400Address                     [3]     ORAddress,
        directoryName                   [4]     Name,
        ediPartyName                    [5]     EDIPartyName,
        uniformResourceIdentifier       [6]     IA5String,
        iPAddress                       [7]     OCTET STRING,
        registeredID                    [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                    [0]     DirectoryString OPTIONAL,
        partyName                       [1]     DirectoryString }
```

```
DirectoryString ::= CHOICE {
        teletexString           TeletexString (SIZE (1..MAX)),
        printableString         PrintableString (SIZE (1..MAX)),
        universalString         UniversalString (SIZE (1..MAX)),
        utf8String              UTF8String (SIZE (1.. MAX)),
        bmpString               BMPString (SIZE (1..MAX)) }

id-ad OBJECT IDENTIFIER ::= { id-pkix 48 }

id-ad-caIssuers OBJECT IDENTIFIER ::= { id-ad 2 }
```

2.3.16.2 Extension source and control in the GOC PKI

The **authorityInfoAccess extension** value and criticality can be controlled only through Alternative Settings. It can be set to be **Critical** or **Non-Critical**.

2.3.16.3 Generation Requirements

GOC PKI CAs shall:

- optionally, manually include this extension in CA or EE certificates;
- optionally, manually set the criticality flag to “false”; and
- optionally, set the **accessMethod** value to be id-ad-caIssuers (1.3.6.1.5.5.7.48.2) and set the **accessLocation** to be a **uRI** (IA5String).

2.3.16.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not recognize nor process this extension.

3 X.509 v2 CRLs

CAs use CRLs to publish notice of revocation of a subject's certificate. The CRLs are stored in the directory as attributes and are checked by users to verify that the other users' certificates are not revoked. The fields in a CRL identify the issuer (i.e., CA), the date/time the current CRL was generated, the date the next CRL will be generated, and the revoked users' certificates. A CA may also add extensions that contain additional information about a specific entry or extensions about the entire CRL (see Section 3.1).

The CRL shall use the syntax of the CertificateList as defined in the 1997 X.509 Specification, Section 6, Reference 1. The GOC PKI uses the CertificateList (i.e., v2 CRL) to revoke both user and CA certificates.

GOC PKI CAs shall generate and sign CRLs that:

1) include the **signature field** to indicate the algorithm used to certify the CRL (if parameters are associated with the signature algorithm, those parameters shall not be included);

2) include the **version field** to indicate that it is a v2 CRL only if there are critical CRL extensions present, otherwise it is absent;

3) include in the **signature field** the identifier (OID) of the algorithm used to sign the certificate, but not populate the parameters in this field:

- **md5WithRSAEncryption** (1.2.840.113549.1.1.4) for RSA/MD5 CA key pair;
- **sha1WithRSAEncryption** (1.2.840.113549.1.1.5) for RSA/SHA1 CA key pair; or
- **dsa-with-sha1** (1.2.840.10040.4.3) for DSA/SHA1 CA key pair;

4) include the **issuer field** to indicate the distinguished name of the CRL issuer;

5) include the **thisUpdate field** to indicate when the CRL was generated;

6) include the **nextUpdate field** to indicate when the next CRL update will be generated, if a scheduled time is known;

7) include the **revokedCertificates field** containing the sequence(s) of **userCertificate** (which may identify user or CA certificates) **field(s)**, **revocationDate field(s)**, and **crlEntryExtensions field(s)** to indicate the serial number of each revoked certificate, the time when it was revoked, and the entry extensions (as described in Section 3.1); and

8) include **crlExtensions field(s)** as specified in Section 3.1.

GOC PKI certificate processing entities shall:

1) verify the signature on the CRL by employing the public key from the issuer's certificate and parameters, if applicable;

2) verify the certification path of the CRL issuer's signature certificate;

3) verify that the version is v2;

4) if present, verify the present time falls within the **thisUpdate** and **nextUpdate** field(s);

5) if present, verify that the **CRLNumber** it is greater than that of the last CRL that the user possessed;

6) verify that the **CRL issuer** is the issuer of the certificate (or as indicated by the **cRLDistributionPoints extension**);

7) verify that the subject name in the CRL issuer's X.509 certificate matches the CRL issuer's name and the CRL issuer's certificate **basicConstraints** extension **cA** flag is set to "true";

8) if the **keyUsage extension** is present in the CRL issuer's certificate and is flagged critical, verify that the **keyUsage cRLSign** bit is set to 1; and

9) check whether the certificate serial number appears on the CRL. If a certificate that appears on the CRL is a CA certificate the user shall be notified and the certificate is rejected.

3.1 CRL Extensions

The following sections describe the standard CRL extensions. CRL entry extensions are described in Section 3.2. The CRL extensions add information about the CRL and the CRL issuer, and provide mechanisms to control the size of the CRLs. The CRL entry extensions add information about a specific entry within the CRL..

3.1.1 authorityKeyIdentifier

This extension identifies the public key to be used to verify the signature on this CRL. It enables distinct keys used by the same CA to be differentiated. This extension may hold the explicit key identifier or an explicit certificate identifier. This extension is useful when a CA uses more than one key (e.g., when the CA key is updated).

3.1.1.1 ASN.1 Syntax

```
authorityKeyIdentifier EXTENSION ::= {
        SYNTAX                    AuthorityKeyIdentifier
        IDENTIFIED BY             id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
        authorityKeyIdentifier    [0] KeyIdentifer
        OPTIONAL,
        authorityCertIssuer       [1] GeneralNames            OPTIONAL,
        authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
        ( WITH COMPONENTS { …, authorityCertIssuer PRESENT,
                                  authorityCertSerialNumber PRESENT } |
          WITH COMPONENTS { …, authorityCertIssuer ABSENT,
                                  authorityCertSerialNumber ABSENT } )

KeyIdentifier ::= OCTET STRING

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                 [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                [1]     IA5String,
        dNSName                   [2]     IA5String,
        x400Address               [3]     ORAddress,
        directoryName             [4]     Name,
        ediPartyName              [5]     EDIPartyName,
        uniformResourceIdentifier [6]     IA5String,
        iPAddress                 [7]     OCTET STRING,
        registeredID              [8]     OBJECT IDENTIFIER }

CertificateSerialNumber ::= INTEGER
```

```
EDIPartyName ::= SEQUENCE {
       nameAssigner                [0]     DirectoryString OPTIONAL,
       partyName                   [1]     DirectoryString }

DirectoryString ::= CHOICE {
       teletexString               TeletexString (SIZE (1..MAX)),
       printableString             PrintableString (SIZE (1..MAX)),
       universalString             UniversalString (SIZE (1..MAX)),
       utf8String                  UTF8String (SIZE (1.. MAX)),
       bmpString                   BMPString (SIZE (1..MAX)) }
```

3.1.1.2 Extension source and control in the GOC PKI

The **authorityKeyIdentifier CRL extension** value and criticality is controlled by the CA. It can only be **Non-Critical**. It cannot be modified by any other means. Alternative Settings may also be used to insert an **authorityKeyIdentifier** as per the PKIX profile.

3.1.1.3 Generation Requirements

GOC PKI CAs shall:

- automatically include this extension in all CRLs;
- automatically set the criticality flag to “false”;
- automatically exclude the **authorityCertIssuer** and **authorityCertSerialNumber** fields; and
- optionally, include the **authorityKeyIdentifier field** as a 20 byte SHA-1 hash of the **subjectPublicKeyInfo** in the CA certificate or, using Alternative Settings, as a hash of the **subjectPublicKey** as per the PKIX profile.

3.1.1.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process the **authorityCertIssuer** and **authorityCertSerialNumber** fields.

3.1.2 issuerAltName

This field contains one or more alternative names, using any of a variety of name forms, for the certificate or CRL issuer.

3.1.2.1 ASN.1 Syntax

```
issuerAltName EXTENSION ::= {
       SYNTAX               GeneralNames
       IDENTIFIED BY               id-ce-issuerAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
       otherName                   [0]     INSTANCE OF OTHER-NAME,
       rfc822Name                  [1]     IA5String,
       dNSName                     [2]     IA5String,
       x400Address                 [3]     ORAddress,
       directoryName               [4]     Name,
       ediPartyName                [5]     EDIPartyName,
       uniformResourceIdentifier   [6]     IA5String,
       iPAddress                   [7]     OCTET STRING,
       registeredID                [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
       nameAssigner                [0]     DirectoryString OPTIONAL,
       partyName                   [1]     DirectoryString }

DirectoryString ::= CHOICE {
       teletexString               TeletexString (SIZE (1..MAX)),
       printableString             PrintableString (SIZE (1..MAX)),
       universalString             UniversalString (SIZE (1..MAX)),
       utf8String                  UTF8String (SIZE (1.. MAX)),
       bmpString                   BMPString (SIZE (1..MAX)) }
```

3.1.2.2 Extension source and control in the GOC PKI

The **issuerAltName CRL extension** is not supported by the GOC PKI for use in CRLs.

3.1.2.3 Generation Requirements

GOC PKI CAs shall:

- not include this extension in CRLs.

3.1.2.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process this CRL extension.

3.1.3 cRLNumber

This CRL extension conveys a monotonically increasing sequence number for each CRL issued by a given CA through a given CA directory attribute or CRL distribution point directory attribute.

3.1.3.1 ASN.1 Syntax

```
cRLNumber EXTENSION ::= {
        SYNTAX                  CRLNumber
        IDENTIFIED BY                       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)
```

3.1.3.2 Extension source and control in the GOC PKI

The **cRLNumber extension** value and criticality is controlled by the CA only. It can only be **Non-Critical**.

3.1.3.3 Generation Requirements

GOC PKI CAs shall:

- automatically include this extension in all CRLs; and
- automatically set the criticality flag to “false”.

3.1.3.4 Processing Requirements

GOC PKI CRL processing entities shall:

- recognize and process this CRL extension.

3.1.4 deltaCRLIndicator

The **deltaCRLIndicator** CRL extension identifies a CRL as a delta-CRL.

3.1.4.1 ASN.1 Syntax

```
deltaCRLIndicator EXTENSION ::= {
        SYNTAX          BaseCRLNumber
        IDENTIFIED BY           id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber
```

3.1.4.2 Extension source and control in the GOC PKI

The **deltaCRLIndicator CRL extension** is not supported by the GOC PKI for use in CRLs.

3.1.4.3 Generation Requirements

GOC PKI CAs shall:

- not include this extension in CRLs.

3.1.4.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process this CRL extension.

3.1.5 issuingDistributionPoint

This CRL extension identifies the CRL distribution point for this particular CRL, and indicates if the CRL is limited to revocations for end-entity certificates only, for CA-certificates only, or for a limited set of reasons only. This extension indicates that the CRL may contain entries from CAs other than the authority that signed and issued the CRL.

3.1.5.1 ASN.1 Syntax

```
issuingDistributionPoint EXTENSION ::= {
        SYNTAX                          IssuingDistPointSyntax
        IDENTIFIED BY                           id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
        distributionPoint               [0]     DistributionPointName OPTIONAL,
        onlyContainsUserCerts           [1]     BOOLEAN DEFAULT FALSE,
        onlyContainsCACerts             [2]     BOOLEAN DEFAULT FALSE,
        onlySomeReasons                 [3]     ReasonFlags OPTIONAL,
        indirectCRL                     [4]     BOOLEAN DEFAULT FALSE }

DistributionPointName ::= CHOICE {
        fullName                        [0]     GeneralNames,
        nameRelativeToCRLIssuer         [1]     RelativeDistinguishedName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
```

```
GeneralName ::= CHOICE {
        otherName                   [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                  [1]     IA5String,
        dNSName                     [2]     IA5String,
        x400Address                 [3]     ORAddress,
        directoryName               [4]     Name,
        ediPartyName                [5]     EDIPartyName,
        uniformResourceIdentifier   [6]     IA5String,
        iPAddress                   [7]     OCTET STRING,
        registeredID                [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                [0]     DirectoryString OPTIONAL,
        partyName                   [1]     DirectoryString }

DirectoryString ::= CHOICE {
        teletexString               TeletexString (SIZE (1..MAX)),
        printableString             PrintableString (SIZE (1..MAX)),
        universalString             UniversalString (SIZE (1..MAX)),
        utf8String                  UTF8String (SIZE (1.. MAX)),
        bmpString                   BMPString (SIZE (1..MAX)) }

ReasonFlags ::= BIT STRING {
        unspecified                 (0),
        keyCompromise               (1),
        cACompromise                        (2),
        affiliationChanged          (3),
        superseded                  (4),
        cessationOfOperation        (5),
        certificateHold                     (6)
        removeFromCRL               (8) }
```

3.1.5.2 Extension source and control in the GOC PKI

The **issuingDistributionPoint CRL extension** value and criticality is controlled by the CA only. It can only be **Critical**.

3.1.5.3 Generation Requirements

GOC PKI CAs shall generate this extension only for CRLs and shall:

- automatically include this extension in all CRLs/ARLs or automatically exclude this extension if Combined CRLs are used;
- automatically set the criticality flag to “true”;
- automatically use **directoryName** only for **GeneralName**;
- automatically use **onlyContainsUserCerts** for CRLs
- automatically use **onlyContainsCACerts** for ARLs; and
- never use **onlySomeReasons** or **indirectCRL**.

3.1.5.4 Processing Requirements

GOC PKI CRL processing entities shall:

- recognize and process this CRL extension.

3.2 CRL Entry Extensions

3.2.1 reasonCode

This CRL entry extension field identifies a reason for the certificate revocation.

3.2.1.1 ASN.1 Syntax

```
reasonCode EXTENSION ::= {
        SYNTAX                  CRLReason
        IDENTIFIED BY                   id-ce-reasonCode }

CRLReason ::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        cACompromise            (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        certificateHold                 (6),
        removeFromCRL           (8) }
```

3.2.1.2 Extension source and control in the GOC PKI

The **reasonCode CRL entry extension** criticality is controlled by the CA. The value is controlled through a setting in GOC PKI/RA. It can only be **Non-Critical**. Advanced Alternative Settings may be used to exclude the revocation reason from revocation lists if the revocation reason is unspecified.

3.2.1.3 Generation Requirements

GOC PKI CAs that generate this extension shall:

- automatically include this extension in all CRLs or, optionally, manually using Alternative Settings, exclude the extension if the revocation reason is unspecified;
- automatically set the criticality flag to “false”; and
- manually include **CRLReason** bits for **unspecified**, **keyCompromise**, **affiliationChanged**, **superseded**, or **cessationOfOperation** only.

3.2.1.4 Processing Requirements

GOC PKI CRL processing entities shall:

- recognize and process this CRL extension.

3.2.2 holdInstructionCode

The **holdInstructionCode** CRL entry extension provides a registered instruction identifier which indicates the action to be taken after encountering a certificate that has been placed on hold.

3.2.2.1 ASN.1 Syntax

```
holdInstructionCode EXTENSION ::= {
        SYNTAX          HoldInstruction
        IDENTIFIED BY id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER
```

3.2.2.2 Extension source and control in the GOC PKI

The **holdInstructionCode CRL entry extension** is not supported by the GOC PKI for use in CRLs.

3.2.2.3 Generation Requirements

GOC PKI CAs shall:

- not include this extension in CRLs.

3.2.2.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process this CRL extension.

3.2.3 invalidityDate

This CRL entry extension indicates the date at which it is known or suspected that the private key was compromised or that the certificate should otherwise be considered invalid. This date may be earlier than the revocation date in the CRL entry, which is the date at which the CA processed the revocation. GOC PKI shall use this extension to identify the date at which the certificate should be considered invalid only if the revocation reason is **keyCompromise**.

3.2.3.1 ASN.1 Syntax

```
invalidityDate EXTENSION ::= {
        SYNTAX                  GeneralizedTime
        IDENTIFIED BY                   id-ce-invalidityDate }
```

3.2.3.2 Extension source and control in the GOC PKI

The **invalidityDate CRL entry extension** value and criticality is controlled by the CA only. It can only be **Non-Critical**.

3.2.3.3 Generation Requirements

GOC PKI CAs that generate this extension shall:

- automatically include this extension in a CRL only if the revocation reason is **keyCompromise**; and
- automatically set the criticality flag to “false”;

3.2.3.4 Processing Requirements

GOC PKI CRL processing entities shall:

- recognize and process this CRL extension.

3.2.4 certificateIssuer

This CRL entry extension identifies the certificate issuer associated with an entry in an indirect CRL (i.e., a CRL that has the indirectCRL indicator bit set in its issuing distribution point extension).

3.2.4.1 ASN.1 Syntax

```
certificateIssuer EXTENSION ::= {
        SYNTAX                  GeneralNames
        IDENTIFIED BY                   id-ce-certificateIssuer }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                       [0]     INSTANCE OF OTHER-NAME,
        rfc822Name                      [1]     IA5String,
        dNSName                         [2]     IA5String,
        x400Address                     [3]     ORAddress,
        directoryName                   [4]     Name,
        ediPartyName                    [5]     EDIPartyName,
        uniformResourceIdentifier       [6]     IA5String,
        iPAddress                       [7]     OCTET STRING,
        registeredID                    [8]     OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
        nameAssigner                    [0]     DirectoryString OPTIONAL,
        partyName                       [1]     DirectoryString }

DirectoryString ::= CHOICE {
        teletexString                   TeletexString (SIZE (1..MAX)),
        printableString                 PrintableString (SIZE (1..MAX)),
        universalString                 UniversalString (SIZE (1..MAX)),
        utf8String                      UTF8String (SIZE (1.. MAX)),
        bmpString                       BMPString (SIZE (1..MAX)) }
```

3.2.4.2 Extension source and control in the GOC PKI

The **certificateIssuer CRL extension** is not supported by the GOC PKI for use in CRLs.

3.2.4.3 Generation Requirements

GOC PKI CAs shall:

- not include this extension in CRLs.

3.2.4.4 Processing Requirements

GOC PKI certificate processing entities shall:

- not process this CRL extension.

4 PKIX Compliance

This profile was developed in the interest of interoperability with communities outside the Government of Canada. Of specific interest is the community represented by the PKIX working group of the Internet Engineering Task Force (IETF). The PKIX working group has developed its own certificate and CRL profile (Section 6, Reference 2).

4.1 Minimum PKIX compliance

In general, CAs conforming to the PKIX profile must support a minimum set of certificate and CRL extensions. This minimum set of certificate and CRL extensions is summarized in Table 2. This minimum set excludes those extensions that the PKIX profile defines as not recommended or optional (without a recommendation).

The set of extensions **not recommended** for use in the PKIX profile are:

- privateKeyUsagePeriod

The set of extensions merely suggested as **optional (without any recommendation)** for use in the PKIX profile are:

- subjectDirectoryAttributes
- policyMappings
- issuerAltName
- authorityInfoAccess
- deltaCRLIndicator
- issuingDistributionPoint
- holdInstructionCode
- certificateIssuer

In this table, “Yes” means that the extension must be supported by the CA or the End Entity and “Optional” means that the extension is not mandated to be supported. Where applicable, where support is marked as “Optional”, if support is recommended, it is so indicated with **bold text**.

Table 2. Minimum PKIX compliance

Element	CA Support	End Entity Recognize
Certificate extensions		
authorityKeyIdentifier	Yes	**Optional**
subjectKeyIdentifier	Yes	**Optional**
keyUsage	Yes	Yes
certificatePolicies	Yes	Yes
subjectAltName	Yes[3]	Yes
basicConstraints	Yes	Yes
nameConstraints	Optional	Yes
policyConstraints	Optional	Yes
extKeyUsage	Optional	Yes
cRLDistributionPoints	**Optional**	**Optional**
CRL extensions		
authorityKeyIdentifier	Yes	Optional
cRLNumber	Yes	Optional
CRL entry extensions		
reasonCode	**Optional**	Optional

[3] Must only be supported by a PKIX-compliant CA if the CA issues certificates with an empty sequence for the subject field in the base certificate.

Element	CA Support	End Entity Recognize
invalidityDate	**Optional**	Optional

4.2 PKIX and X.509 Certificate Extension comparison

Table 3 summarizes the possible and recommended choices of criticality for each certificate and CRL extension specified in X.509 and in PKIX. This table also compares the possible choices for criticality supported by the GOC PKI. Where applicable, any recommendation on criticality is indicated with **bold text**. In some cases, "**n/a**" is used to indicate that the profile in question does not specify the extension (e.g., **privateVersInfo** is a Non-Critical private extension).

However, not all of the extensions presented in Table 3 are mandatory for PKIX compliance. Refer to Section 4.1 for a full description of mandatory and non-mandatory extensions for PKIX compliance.

Table 3. Comparison of extension criticality

Element	PKIX	X.509	GOC PKI
Certificate extensions			
authorityKeyIdentifier	Non-Critical	Non-Critical	Non-Critical
subjectKeyIdentifier	Non-Critical	Non-Critical	Non-Critical
keyUsage	**Critical** or Non-Critical	Critical or Non-Critical	Critical or Non-Critical
privateKeyUsagePeriod	Non-Critical	Critical or Non-Critical	Critical or Non-Critical
certificatePolicies	Critical or Non-Critical	Critical or Non-Critical	Critical or Non-Critical
policyMappings	Non-Critical	Non-Critical	Critical or Non-Critical
privateVersInfo	n/a	n/a	Non-Critical
subjectAltName[4]	Critical or Non-Critical	Critical or Non-Critical	Critical or Non-Critical
issuerAltName	Critical or **Non-Critical**	Critical or Non-Critical	Critical or Non-Critical
subjectDirectoryAttributes	Non-Critical	Non-Critical	Critical or Non-Critical
basicConstraints	Critical	**Critical** or Non-Critical	Critical or Non-Critical
nameConstraints	Critical	**Critical** or Non-Critical	Critical or Non-Critical
policyConstraints	Critical or Non-Critical	**Critical** or Non-Critical	Critical or Non-Critical
extKeyUsage	Critical or Non-Critical	n/a	Critical or Non-Critical
CRLDistributionPoints	Critical or **Non-Critical**	Critical or **Non-Critical**	Critical or Non-Critical
authorityInfoAccess	Non-Critical	n/a	Critical or Non-Critical
CRL extensions			
authorityKeyIdentifier	Non-Critical	Non-Critical	Non-Critical
issuerAltName	Critical or **Non-Critical**	Critical or Non-Critical	n/a
cRLNumber	Non-Critical	Non-Critical	Non-Critical
deltaCRLIndicator	Critical	Critical	n/a
issuingDistributionPoint	Critical	Critical	Critical
CRL entry extensions			
reasonCode	Non-Critical	Non-Critical	Non-Critical
holdInstructionCode	Non-Critical	Non-Critical	n/a
invalidityDate	Non-Critical	Non-Critical	Non-Critical
certificateIssue	Critical	Critical	n/a

[4] Must only be supported by a PKIX-compliant CA if the CA issues certificates with an empty sequence for the subject field in the base certificate.

4.3 Extension support, recognition, and criticality

This section summarizes the extent of support, recognition and criticality of certificate and CRL extensions relating to the GOC PKI Certification Authority and End Entity.

Table 4 summarizes the PKIX requirements for CA support, population, and criticality of certificate and CRL extensions. In this table, there are three main columns: **PKIX CA**, **GOC PKI CA**, and **Comment**. Under the **PKIX CA** and **GOC PKI CA** columns there are three additional columns: **Support**, **Populate**, and **Criticality**. Under the PKIX CA column, these additional columns are PKIX requirements. Under the **GOC PKI CA** column, these additional columns describe GOC PKI CA capabilities.

The **Support** column indicates whether or not the CA must be able to include the extension in a certificate. The only possible values are: "Yes", "No", "Optional", or "n/a". That is, for PKIX compliance, an extension with **Support** marked "Yes" means that a PKIX-compliant CA must be capable of including the extension in issued certificates. An extension with **Support** marked "No" means that the PKIX profile does not use or does not recommend the use of the extension. An extension with **Support** marked "Optional" means that the PKIX profile does not mandate the use of the extension and may be optionally supported. An extension with **Support** marked "n/a" means that the extension is not a PKIX extension (e.g., **privateVersInfo**).

The **Populate** column indicates whether or not the CA must populate the extension with a value in a CA certificate. The only possible values are "Yes", "No", "Optional", or "n/a". Hence, for PKIX compliance, an extension with **Populate** marked "Yes" means that the extension must be populated in a CA certificate. An extension with **Populate** marked "No" means that the extension must not be populated in a CA certificate. An extension with **Populate** marked "Optional" means that the PKIX profile does not mandate the use of the extension and may be optionally populated. An extension with **Populate** marked "n/a" means that the extension is not a PKIX extension (e.g., **privateVersInfo**).

The **Criticality** column indicates whether or not the extension is marked Critical or Non-Critical in a CA certificate. An extension with **Criticality** marked "n/a" means that the extension is not a PKIX extension. Where applicable, any recommendation on criticality is indicated with **bold text** (e.g., a value of "**Critical** or Non-Critical" implies that the extension is recommended to be Critical).

The **Comment** column contains additional text about CA compliance to the PKIX profile.

Table 4. Certification Authority: comparison of extension support

Element	PKIX CA			GOC PKI CA			Comment
	Support	Populate	Criticality	Suppor t	Populat e	Criticality	
Certificate extension							
authorityKeyIdentifier	Yes	Yes	Non-Critical	Yes	Optional	Non-Critical	Can be automatically added by CA or excluded using Alternative Settings.
subjectKeyIdentifier	Yes	Yes	Non-Critical	Yes	Yes	Non-Critical	
keyUsage	Yes	Yes	**Critical** or Non-Critical	Yes	Optional	Critical or Non-Critical	Can set to Critical through Alternative Settings.
privateKeyUsagePeriod	No	No	Non-Critical	Yes	Optional	Critical or Non-Critical	Can be automatically added by CA but can be excluded using Alternative Settings.
certificatePolicies	Yes	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	Can be added automatically by CA or through Alternative Settings.
policyMappings	Optional	Optional	Non-Critical	Yes	Optional	Critical or Non-Critical	

Element	PKIX CA			GOC PKI CA			Comment
	Support	Populate	Criticality	Support	Populate	Criticality	
privateVersInfo	n/a	n/a	n/a	Yes	Yes	Non-Critical	Private extension added automatically by CA and is always Non-Critical.
subjectAltName	Optional[5]	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	Can be added by the CA through Alternative Settings.
issuerAltName	Optional	Optional	Critical or **Non-Critical**	Yes	Optional	Critical or Non-Critical	Can be added automatically by CA and can set to Non-Critical through Alternative Settings.
subjectDirectoryAttributes	Optional	Optional	Non-Critical	Yes	Optional	Critical or Non-Critical	
basicConstraints	Yes	Yes	Critical	Yes	Optional	Critical or Non-Critical	Added automatically by CA and can change criticality or exclude through Alternative Settings.
nameConstraints	Optional	Optional	Critical	Yes	Optional	Critical or Non-Critical	Can set to Critical through Alternative Settings.
policyConstraints	Optional	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	
extKeyUsage	Optional	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	
cRLDistributionPoints	**Optional**	**Optional**	Critical or **Non-Critical**	Yes	Optional	Critical or Non-Critical	Can be excluded using Alternative Settings. Can set criticality through Alternative Settings.
authorityInfoAccess	Optional	Optional	Non-Critical	Yes	Optional	Critical or Non-Critical	Can set to Non-Critical through Alternative Settings.
CRL extension							
authorityKeyIdentifier	Yes	Yes	Non-Critical	Yes	Yes	Non-Critical	Added automatically by CA.
issuerAltName	Optional	Optional	Critical or **Non-Critical**	No	No	n/a	Optional extension not supported or populated in CRLs.
cRLNumber	Yes	Yes	Non-Critical	Yes	Yes	Non-Critical	Added automatically by CA.
deltaCRLIndicator	Optional	Optional	Critical	No	No	n/a	Optional extension not supported or populated in CRLs.
issuingDistributionPoint	Optional	Optional	Critical	Yes	Yes	Critical	Added automatically by CA.
CRL entry extension							
reasonCode	**Optional**	**Optional**	Non-Critical	Yes	Yes	Non-Critical	Recommended option populated and set to Non-Critical.
holdInstructionCode	Optional	Optional	Non-Critical	No	No	n/a	Optional extension not supported or populated in CRLs.
invalidityDate	**Optional**	**Optional**	Non-Critical	Yes	Yes	Non-Critical	Recommended option populated and set to Non-Critical.
certificateIssuer	Optional	Optional	Non-Critical	No	No	n/a	Optional extension not supported or populated in CRLs.

Table 5 summarizes the PKIX requirements for End Entity recognition, population, and criticality of certificate and CRL extensions. In this table, there are three main columns: **PKIX EE**, **GOC PKI EE**, and **Comment**. Under the **PKIX EE** and **GOC PKI EE** columns there are three additional columns: **Recognize**, **Populate**, and **Criticality**. Under the **PKIX EE** column, these additional columns are PKIX requirements. Under the **GOC PKI EE** column, these additional columns describe GOC PKI EE capabilities.

The **Recognize** column indicates whether or not the End Entity must be capable of recognizing and processing the certificate extension. The only possible values are "Yes",

[5] Must only be supported by a PKIX-compliant CA if the CA issues certificates with an empty sequence for the subject field in the base certificate.

“No”, “Optional”, or “n/a”. That is, for PKIX compliance, an extension with **Recognize** marked “Yes” means that the End Entity must be capable of recognizing and processing the extension. An extension with **Recognize** marked “No” means that the PKIX profile does not use or does not recommend the recognition or processing of the extension. An extension with **Recognize** marked “Optional” means that the PKIX profile does not mandate the use of the extension and may be optionally recognized and processed by End Entities. An extension with **Recognize** marked “n/a” means that the extension is not a PKIX extension (e.g., **privateVersInfo**).

The **Populate** column indicates whether or not the extension is populated in End Entity certificates. The only possible values are “Yes”, “No”, “Optional”, or “n/a”. That is, for PKIX compliance, an extension with **Populate** marked “Yes” means that the extension must be populated in an End Entity certificate. An extension with **Populate** marked “Optional” means that the PKIX profile does not mandate the use of the extension and may be optionally populated. An extension with **Populate** marked “No” means that the PKIX profile mandates that the extension not be populated in End Entity certificates.

The **Criticality** column indicates whether or not the extension is marked Critical or Non-Critical in an EE certificate. An extension with **Criticality** marked “n/a” means that the extension is not a PKIX extension. Where applicable, any recommendation on criticality is indicated with **bold text** (e.g., a value of “**Critical** or Non-Critical” implies that the extension is recommended to be Critical).

The **Comment** column contains additional text about End Entity compliance to the PKIX profile.

Table 5. End Entity: comparison of extension support

Element	PKIX End Entity (EE)			GOC PKI End Entity (EE)			Comment
	Recognize	Populate	Criticality	Recognize	Populate	Criticality	
Certificate extension							
authorityKeyIdentifier	**Optional**	Yes	Non-Critical	Yes	Optional	Non-Critical	Can be automatically added by CA or excluded using Alternative Settings.
subjectKeyIdentifier	**Optional**	Yes	Non-Critical	Yes	Yes	Non-Critical	Automatically added by CA.
keyUsage	Yes	Yes	**Critical** or Non-Critical	Yes	Optional	Critical or Non-Critical	Can set to Critical through Alternative Settings and can be excluded using Alternative Settings.
privateKeyUsagePeriod	No	No	Non-Critical	Yes	Optional	Critical or Non-Critical	Can be automatically added by CA but can be excluded using Alternative Settings.
certificatePolicies	Yes	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	Can be added automatically by CA or through Alternative Settings.
policyMappings	Yes	No	Non-Critical	Yes	Optional	Critical or Non-Critical	Not for use in EE certificates.
privateVersInfo	n/a	n/a	n/a	Yes	Yes	Non-Critical	Added automatically by CA and is always Non-Critical.
subjectAltName	Yes	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	Can be added by the CA through GOC PKI/RA only.
issuerAltName	Optional	Optional	Critical or **Non-Critical**	Yes	Optional	Critical or Non-Critical	Can be added through Alternative Settings.

Element	PKIX End Entity (EE)			GOC PKI End Entity (EE)			Comment
	Recognize	**Populate**	**Criticality**	**Recognize**	**Populate**	**Criticality**	
subjectDirectoryAttributes	Optional	Optional	Non-Critical	Yes	Optional	Critical or Non-Critical	Added automatically by CA to administrator certificates to indicate role.
basicConstraints	Yes	Optional	Critical	Yes	Optional	Critical or Non-Critical	Added automatically by CA and can change criticality or exclude through Alternative Settings.
nameConstraints	Yes	No	Critical	Yes	Optional	Critical or Non-Critical	Not for use in EE certificates.
policyConstraints	Yes	No	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	Not for use in EE certificates.
extKeyUsage	Yes	Optional	Critical or Non-Critical	Yes	Optional	Critical or Non-Critical	Can be added through Alternative Settings.
cRLDistributionPoints	**Optional**	**Optional**	Critical or **Non-Critical**	Yes	Optional	Critical or Non-Critical	Can set to Non-Critical through Alternative Settings and can be excluded using Alternative Settings.
authorityInfoAccess	Optional	Optional	Non-Critical	No	Optional	Critical or Non-Critical	Can set to Non-Critical through Alternative Settings.
CRL extension							
authorityKeyIdentifier	Optional	Yes	Non-Critical	Yes	Yes	Non-Critical	Added automatically by CA.
issuerAltName	Optional	Optional	Critical or **Non-Critical**	No	No	n/a	Optional extension not supported or populated in CRLs.
cRLNumber	Optional	Yes	Non-Critical	Yes	Yes	Non-Critical	Added automatically by CA.
deltaCRLIndicator	Optional	Optional	Critical	No	No	n/a	Optional extension not supported or populated in CRLs.
issuingDistributionPoint	Optional	Optional	Critical	Yes	Yes	Critical	Added automatically by CA.
CRL entry extension							
reasonCode	Optional	**Optional**	Non-Critical	Yes	Yes	Non-Critical	Recommended option populated and set to Non-Critical.
holdInstructionCode	Optional	Optional	Non-Critical	No	No	n/a	Optional extension not supported or populated in CRLs.
invalidityDate	Optional	**Optional**	Non-Critical	Yes	Yes	Non-Critical	Recommended option populated and set to Non-Critical.
certificateIssuer	Optional	Optional	Non-Critical	No	No	n/a	Optional extension not supported or populated in CRLs.

4.4 GOC PKI compliance to ASN.1

This section summarizes the compliance of the certificate and CRL profile with the PKIX certificate and CRL profile at the ASN.1 level.

Table 6 summarizes the compliance of the GOC PKI base certificate profile with the PKIX base certificate profile, including any actions required to ensure that a GOC PKI-issued base certificate is compliant with the PKIX certificate profile.

Table 7 summarizes the compliance of the GOC PKI certificate profile with the PKIX certificate profile, including any actions required to ensure that a GOC PKI-issued certificate is compliant with the PKIX certificate profile

Table 8 summarizes the compliance of the GOC PKI base CRL profile with the PKIX base CRL profile, including any actions required to ensure that a GOC PKI-issued CRL is compliant with the PKIX CRL profile.

Table 9 summarizes the compliance of the GOC PKI CRL extension profile with the PKIX CRL extension profile, including any actions required to ensure that a GOC PKI-issued CRL is compliant with the PKIX CRL profile.

Table 6. Base certificate compliance

Field	PKIX type	GOC PKI		
		Type or Value	Notes	Compliance
Version	Version indicator=2 for v3 (INTEGER)	2	Must be v3 (value=2).	Compliant. Only v3 certificates are issued.
Serial Number	CertificateSerialNumber (INTEGER)	Serial number (INTEGER)	Must be unique for each certificate issued by a given CA.	Compliant.
Signature	SEQUENCE	SEQUENCE	Must contain same algorithm identifier as the **signatureAlgorithm** field in the **signatureAlgorithm** outside the certificate (used to sign the certificate). No algorithms are stipulated.	Compliant.
algorithm	Algorithm Identifier (OID)	1.2.840.113549.1.1.4 (for RSA/MD5) 1.2.840.113549.1.1.5 (for RSA/SHA-1) 1.2.840.10040.4.3 (for DSA/SHA-1)		
parameters	ANY DEFINED BY algorithm OPTIONAL	Null for RSA SEQUENCE of INTEGERS p, q, g for DSA		
Issuer	DN	DN	Must contain non-empty DN. UTF8String is the preferred encoding. Until Dec. 31 2003, PrintableString or BMPString may be used, depending on sufficiency of character set. TeletexString and UniversalString should not be used for certificates for new subjects.	Compliant. Encoded using UTF8String.
Validity	SEQUENCE	SEQUENCE	Until 2049, encode as UTCTime expressed in GMT (Zulu) including seconds for all UTCTime. After 2049, encode as GeneralizedTime.	Compliant. Encoded as UTCTime.
notBefore	Date/Time (UTCTime to 2049)	UTCTime		
notAfter	Date/Time (UTCTime to 2049)	UTCTime		
Subject	DN	DN	Must be present in all CA certificates as a non-empty DN, matching the value used in the issuer field of all certificates issued by that CA (includes self-signed certificates, CA certificates, and EE certificates).	Compliant.
Subject Public Key Information	SEQUENCE	SEQUENCE	Must contain same algorithm identifier as the **signatureAlgorithm** field in the **signatureAlgorithm** outside the certificate (used to sign the certificate). No algorithms are stipulated.	Compliant.
algorithm	Algorithm Identifier (OID)	1.2.840.113549.1.1.1 (RSA) 1.2.840.10040.4.1 (DSA) 1.2.840.10045.2.1 (ECDSA)		
subjectPublicKey	Public key (BIT STRING)	Public key (BIT STRING)		
Issuer Unique ID	Not Used	Not Used	Not used.	Compliant. Not used.

Field	PKIX type	GOC PKI		
		Type or Value	Notes	Compliance
Subject Unique ID	Not Used	Not Used	Not Used	Compliant. Not used.
extension	SEQUENCE	SEQUENCE	Present if extensions used.	Compliant.
extnId	Extension identifier (OID	OID		
critical	Extension criticality (Boolean)	Criticality		
extn value	Extension value (STRING)	Value		
issuer's signature	Digital signature (BITSTRING)	Digital signature (BITSTRING)	Mandatory issuer's digital signature.	Compliant.

Table 7. Certificate extension compliance

Extension	PKIX type	GOC PKI			
		Type or Value	Notes	Certification Authority	End Entity
authorityKeyIdentifier	SEQUENCE	SEQUENCE	**keyIdentifier** field of **authorityKeyIdentifier** extension included by default in all certificates. Extension can be excluded using Alternative Settings.	Compliant. Derived from 160-bit SHA-1 hash of **subjectPublicKeyInfo**, or optionally, as hash of **subjectPublicKey**.	Compliant. Derived from 160-bit SHA-1 hash of **subjectPublicKeyInfo**, or optionally, as hash of **subjectPublicKey**.
keyIdentifier	OCTET STRING	OCTET STRING			
authorityCertIssuer	GeneralNames	Not used.			
GeneralName	CHOICE	Not used.			
otherName	TYPE-IDENTIFIER	Not used.			
rfc822Name	[1] (IA5String)	Not used.			
dNSName	[2] (IA5String)	Not used.			
x400Address	[3] (ORAddress)	Not used.			
directoryName	[4] (DN)	Not used.			
ediPartyName	[5] (EDIPartyName)	Not used.			
uRI	[6] (IA5String)	Not used.			
iPAddress	[7] (OCTET STRING)	Not used.			
registeredID	[8] (OID)	Not used.			
authorityCertSerialNumber	Serial number (INTEGER)	Not used.			
subjectKeyIdentifier	SEQUENCE	SEQUENCE	**subjectKeyIdentifier** extension must be included in all CA certificates. Must be the same value as the value placed in the **keyIdentifier** field of **authorityKeyIdentifier** extension.	Compliant. Derived from 160-bit SHA-1 hash of **subjectPublicKeyInfo**, or optionally, as hash of **subjectPublicKey**.	Compliant. Derived from 160-bit SHA-1 hash of **subjectPublicKeyInfo**, or optionally, as hash of **subjectPublicKey**.
keyIdentifier	OCTET STRING	OCTET STRING			
keyUsage	BIT STRING	BIT STRING	Recommended as Critical.	Compliant. By default, the **keyCertSign** and **CrlSign** bits are set but use Alternative Settings to set to Critical.	Compliant. By default, the **keyEncipherment** bit and **digitalSignature** bits are set but use Alternative Settings to set to Critical.
digitalSignature	(0)	(0)			
nonRepudiation	(1)	(1)			
keyEncipherment	(2)	(2)			
dataEncipherment	(3)	(3)			
keyAgreement	(4)	(4)			
keyCertSign	(5)	(5)			
cRLSign	(6)	(6)			
encipherOnly	(7)	(7)			
decipherOnly	(8)	(8)			
privateKeyUsagePeriod	SEQUENCE	SEQUENCE	Not recommended . May exclude the **privateKeyUsagePeriod** extension using Alternative Settings.	Compliant. By default, present from security policy or user-specific setting and is Non-Critical.	Compliant. By default, present from security policy or user-specific setting and is Non-Critical.
notBefore	GeneralizedTime	GeneralizedTime			

Extension	PKIX type	GOC PKI			
		Type or Value	Notes	Certification Authority	End Entity
notAfter	GeneralizedTime	GeneralizedTime			
certificatePolicies	SEQUENCE	SEQUENCE	Recommends use of **policyIdentifier** only. If **policyQualifiers** used, recommends **cPSuri** only.	Compliant. By default, only **policyIdentifier** present.	Compliant. By default, only policy OIDs present. Can use Alternative Settings to add **policyQualifer**.
PolicyInformation	SEQUENCE	SEQUENCE			
policyIdentifier	OID	OID			
policyQualifiers	SEQUENCE	SEQUENCE			
PolicyQualifierInfo	SEQUENCE	SEQUENCE			
policyQualifierId	OID	OID			
id-qt	OID	id-pkix-2			
id-qt-cps	OID	id-qt-1			
id-qt-notice	OID	id-qt-1			
qualifier	CHOICE	CHOICE			
cPSuri	CPSuri	CPSuri			
CPSuri	IA5String	IA5String			
UserNotice	SEQUENCE	SEQUENCE			
noticeRef	NoticeReference	NoticeReference			
NoticeReference	SEQUENCE	SEQUENCE			
organization	DisplayText	DisplayText			
DisplayText	CHOICE	CHOICE			
visibleString	VisibleString	”Government of Canada - Gouvernment du Canada”			
bmpString	BMPString	BMPString			
utf8String	UTF8String	UTF8String			
noticeNumbers	SEQUENCE	SEQUENCE			
explicitText	DisplayText	DisplayText			
DisplayText	CHOICE	CHOICE			
visibleString	VisibleString	”Limited liability. See CP - Responsabilite limitee. Voir PC.”			
bmpString	BMPString	BMPString			
utf8String	UTF8String	UTF8String			
policyMappings	SEQUENCE	SEQUENCE	Only in cross-certificates.	Compliant. Use Alternative Settings to add policy mappings.	Compliant. Not used in EE certificates.
issuerDomainPolicy	CertPolicyId	CertPolicyId			
certPolicyId	OID	OID			
subjectDomainPolicy	CertPolicyId	CertPolicyId			
certPolicyId	OID	OID			
privateVersInfo	Not used.	SEQUENCE	Private extension. Must be Non-Critical.	Compliant.	Compliant.
privateVers		5.0			
privateVersInfoFlags		PrivateInfoFlags			
keyUpdateAllowed		(0)			

Extension	PKIX type	GOC PKI			
		Type or Value	Notes	Certification Authority	End Entity
obsolete1		Not used.			
obsolete2		Not used.			
enterpriseCategory		(3)			
webCategory		(4)			
SETCategory		(5)			
subjectAltName	GeneralNames	GeneralNames	Supported by the CA, although not strictly required for PKIX.	Compliant.	Compliant.
GeneralNames	SEQUENCE	SEQUENCE			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	Not used.			
rfc822Name	[1] (IA5String)	[1] (IA5String)			
dNSName	[2] (IA5String)	[2] (IA5String)			
x400Address	[3] (ORAddress)	[3] (ORAddress)			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	[5] (EDIPartyName)			
uri	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	[7] (OCTET STRING)			
registeredID	[8] (OID)	[8] (OID)			
issuerAltName	GeneralNames	GeneralNames	Encoded as per **subjectAltName**. Non-Critical is recommended.	Compliant.	Compliant.
GeneralNames	SEQUENCE	SEQUENCE			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	TYPE-IDENTIFIER			
rfc822Name	[1] (IA5String)	[1] (IA5String)			
dNSName	[2] (IA5String)	[2] (IA5String)			
x400Address	[3] (ORAddress)	[3] (ORAddress)			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	[5] (EDIPartyName)			
uri	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	[7] (OCTET STRING)			
registeredID	[8] (OID)	[8] (OID)			
subjectDirectoryAttributes	AttributesSyntax	AttributesSyntax	Only present in Administrator certificates to determine correct policy certificate to retrieve.	Compliant.	Compliant.
AttributesSyntax	SEQUENCE	SEQUENCE			
Attribute	n/a	privateUserRole			
privateUserRole	n/a	INTEGER			
basicConstraints	SEQUENCE	SEQUENCE	**PathLenConstraint** field meaningful only if **cA** is set to true and must be greater than or equal to zero, if used.	Compliant. Use Alternative Settings to set **pathLenContraint** value and set to Critical.	Compliant. Not used in EE certificates.
cA	BOOLEAN	TRUE (for CAs) FALSE (for EEs)			

Extension	PKIX type	GOC PKI			
		Type or Value	Notes	Certification Authority	End Entity
pathLenConstraint	INTEGER	INTEGER			
nameConstraints	SEQUENCE	SEQUENCE	Used only in CA certificates and is optional for PKIX CAs. Restrictions defined in terms of permitted or excluded name subtrees.	Compliant. Can use Alternative Settings to add name constraints extension and set to Critical.	Compliant. Not used in EE certificates.
permittedSubTrees	GeneralSubTrees	GeneralSubTrees			
GeneralSubTrees	GeneralSubTree	GeneralSubTree			
GeneralSubTree	SEQUENCE	SEQUENCE			
base	GeneralName	GeneralName			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	TYPE-IDENTIFIER			
rfc822Name	[1] (IA5String)	[1] (IA5String)			
dNSName	[2] (IA5String)	[2] (IA5String)			
x400Address	[3] (ORAddress)	[3] (ORAddress)			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	[5] (EDIPartyName)			
uri	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	[7] (OCTET STRING)			
registeredID	[8] (OID)	[8] (OID)			
minimum	INTEGER	INTEGER			
maximum	INTEGER	INTEGER			
excludedSubTrees	GeneralSubTree	GeneralSubTree			
GeneralSubTree	SEQUENCE	SEQUENCE			
base	GeneralName	GeneralName			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	TYPE-IDENTIFIER			
rfc822Name	[1] (IA5String)	[1] (IA5String)			
dNSName	[2] (IA5String)	[2] (IA5String)			
x400Address	[3] (ORAddress)	[3] (ORAddress)			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	[5] (EDIPartyName)			
uri	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	[7] (OCTET STRING)			
registeredID	[8] (OID)	[8] (OID)			
minimum	INTEGER	INTEGER			
maximum	INTEGER	INTEGER			
policyConstraints	SEQUENCE	SEQUENCE	Used only in CA certificates and is optional for PKIX CAs. Must not be a null sequence, so at least one of **inhibitPolicyMapping** or **requireExplicitPolicy** must be present if used.	Compliant. Can use Alternative Settings to add policy constraints extension and set to Critical or Non-Critical.	Compliant. Not used in EE certificates.
requireExplicitPolicy	[0] SkipCerts	[0] SkipCerts			
SkipCerts	INTEGER	INTEGER			
inhibitPolicyMapping	[1] SkipCerts	[1] SkipCerts			

Extension	PKIX type	GOC PKI			
		Type or Value	Notes	Certification Authority	End Entity
SkipCerts	INTEGER	INTEGER			
extKeyUsage	SEQUENCE	SEQUENCE	Defines usages serverAuth, clientAuth, codeSigning, emailProtection, and timeStamping.	Compliant. Use Alternative Settings to add **extKeyUsage** and set to Critical or Non-Critical.	Compliant. Use Alternative Settings to add **extKeyUsage** and set to Critical or Non-Critical. EEs recognize and process only timeStamping and profileKeyEncryption key usages.
KeyPurposeId	CHOICE	CHOICE			
serverAuth	OID	Not used.			
clientAuth	OID	Not used.			
codeSigning	OID	Not used.			
emailProtection	OID	Not used.			
profileKeyEncryption	Not used.	1 2 840 113533 7 74 1			
timeStamping	OID	id-kp 8			
cRLDistribution Points	distributionPoint	distributionPoint	Support recommended and should be Non-Critical.	Compliant. Use Alternative Settings to set to Critical. Populate uRI for Microsoft Windows 2000 clients.	Compliant. Use Alternative Settings to set to Critical.
distributionPoint	SEQUENCE	SEQUENCE			
DistributionPointName	CHOICE	CHOICE			
fullName	GeneralName	GeneralName			
nameRelativeToCRLIssuer	Relative DN	Relative DN			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	Not used.			
rfc822Name	[1] (IA5String)	Not used.			
dNSName	[2] (IA5String)	Not used.			
x400Address	[3] (ORAddress)	Not used.			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	Not used.			
uRI	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	Not used.			
registeredID	[8] (OID)	Not used.			
reasons	ReasonFlags	Not used.			
ReasonFlags	BIT STRING	Not used.			
unspecified	(0)	Not used.			
keyCompromise	(1)	Not used.			
cACompromise	(2)	Not used.			
affiliationChanged	(3)	Not used.			
superseded	(4)	Not used.			
cessationOfOperation	(5)	Not used.			
certificateHold	(6)	Not used.			
removeFromCRL	(8)	Not used.			
cRLIssuer	GeneralNames	GeneralNames			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	Not used.			

Extension	PKIX type	GOC PKI			
		Type or Value	Notes	Certification Authority	End Entity
rfc822Name	[1] (IA5String)	Not used.			
dNSName	[2] (IA5String)	Not used.			
x400Address	[3] (ORAddress)	Not used.			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	Not used.			
uRI	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	Not used.			
registeredID	[8] (OID)	Not used.			
authorityInfoAccess	AccessDesciption	AccessDesciption	Support is optional. If used, must be Non-Critical.	Compliant. Can be added as Non-Critical using Alternative Settings.	Compliant. Can be added as Non-Critical using Alternative Settings. Not recognized nor processed by EEs.
AccessDescription	SEQUENCE	SEQUENCE			
accessMethod	OID	OID			
id-at	OID	id-at			
id-ad-ocsp	OID	1.3.6.1.5.5.7.48.1			
id-ad-caIssuers	OID	1.3.6.1.5.5.7.48.2			
accessLocation	GeneralName	GeneralName			
GeneralName	CHOICE	CHOICE			
otherName	TYPE-IDENTIFIER	TYPE-IDENTIFIER			
rfc822Name	[1] (IA5String)	[1] (IA5String)			
dNSName	[2] (IA5String)	[2] (IA5String)			
x400Address	[3] (ORAddress)	[3] (ORAddress)			
directoryName	[4] (DN)	[4] (DN)			
ediPartyName	[5] (EDIPartyName)	[5] n/a			
uRI	[6] (IA5String)	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)	[7] (OCTET STRING)			
registeredID	[8] (OID)	[8] (OID)			

Table 8. Base CRL compliance

Field	PKIX type	GOC PKI		
		Type or value	Notes	CRL
Version	Version indicator=1 for v2 (INTEGER)	1	When extensions are used, as required by this profile, this field must be present and must specify v2 (the integer value is 1).	Compliant. Only v2 CRLs are issued.
Signature	SEQUENCE	SEQUENCE	Must contain same algorithm identifier as the **signatureAlgorithm** field in the **signatureAlgorithm** outside the certificate (used to sign the certificate). No algorithms are stipulated.	Compliant.
algorithm	Algorithm Identifier (OID)	1.2.840.113549.1.1.4 (for RSA/MD5) 1.2.840.113549.1.1.5 (for RSA/SHA-1) 1.2.840.10040.4.3 (for DSA)		
parameters	ANY DEFINED BY algorithm OPTIONAL	Null for RSA SEQUENCE of INTEGERS p, q, g for DSA		
Issuer	DN	DN	Must contain non-empty DN. **UTF8String** is the preferred encoding. Until Dec. 31 2003, **PrintableString** or **BMPString** may be used, depending on sufficiency of character set. **TeletexString** and **UniversalString** should not be used for certificates for new subjects.	Compliant. Encoded using UTF8String.
This update	Date/Time (UTCTime to 2049)	UTCTime	Until 2049, encode as **UTCTime** expressed in GMT (Zulu) including seconds for all **UTCTime**. After 2049, encode as **GeneralizedTime**.	Compliant.
Next update	Date/Time (UTCTime to 2049)	UTCTime	Until 2049, encode as **UTCTime** expressed in GMT (Zulu) including seconds for all **UTCTime**. After 2049, encode as **GeneralizedTime**.	Compliant.
Revoked certificates	SEQUENCE	SEQUENCE	For revocation date, until 2049, encode as **UTCTime** expressed in GMT (Zulu) including seconds for all **UTCTime**. After 2049, encode as **GeneralizedTime**.	Compliant.
userCertificate	CertificateSerialNumber (INTEGER)	Serial number		
revocationDate	Date/Time (UTCTime to 2049)	UTCTime		
crlEntryExtensions	Extensions	Extensions		

Table 9. CRL extension/CRL entry extension compliance

Extension/Entry Extension	PKIX value	GOC PKI		
		Type or value	Notes	CRL
CRL extensions				
authorityKeyIdentifier	SEQUENCE	SEQUENCE	Derived from 160-bit SHA-1 hash of **subjectPublicKeyInfo** or, optionally using Advanced Alternative Settings, from hash of **subjectPublicKey** as per PKIX profile. Only **authorityKeyIdentifier field** supported.	Compliant.
keyIdentifier	OCTET STRING	OCTET STRING		
authorityCertIssuer	GeneralNames	Not used.		
GeneralName	CHOICE	Not used.		
otherName	TYPE-IDENTIFIER	Not used.		
rfc822Name	[1] (IA5String)	Not used.		
dNSName	[2] (IA5String)	Not used.		
x400Address	[3] (ORAddress)	Not used.		
directoryName	[4] (DN)	Not used.		
ediPartyName	[5] (EDIPartyName)	Not used.		
uRI	[6] (IA5String)	Not used.		
iPAddress	[7] (OCTET STRING)	Not used.		
registeredID	[8] (OID)	Not used.		
authorityCertSerialNumber	Serial number (INTEGER)	Not used.		
issuerAltName	GeneralNames	Not used.	Support is optional.	Compliant. Not used.
GeneralNames	SEQUENCE			
GeneralName	CHOICE			
otherName	TYPE-IDENTIFIER			
rfc822Name	[1] (IA5String)			
dNSName	[2] (IA5String)			
x400Address	[3] (ORAddress)			
directoryName	[4] (DN)			
ediPartyName	[5] (EDIPartyName)			
uri	[6] (IA5String)			
iPAddress	[7] (OCTET STRING)			
registeredID	[8] (OID)			
cRLNumber	INTEGER	INTEGER	Support is mandatory.	Compliant.
issuingDistributionPoint	SEQUENCE	SEQUENCE	Only **directoryName** populated for **GeneralName**.	Compliant.
distributionPoint	SEQUENCE	SEQUENCE		
DistributionPointName	CHOICE	CHOICE		
fullName	GeneralName	GeneralName		
nameRelativeToCRLIssuer	Relative DN	Relative DN		
GeneralName	CHOICE	CHOICE		
otherName	TYPE-IDENTIFIER	Not used.		
rfc822Name	[1] (IA5String)	Not used.		

Extension/Entry Extension	PKIX value	GOC PKI		
		Type or value	Notes	CRL
dNSName	[2] (IA5String)	Not used.		
x400Address	[3] (ORAddress)	Not used.		
directoryName	[4] (DN)	[4] (DN)		
ediPartyName	[5] (EDIPartyName)	Not used.		
uRI	[6] (IA5String)	Not used.		
iPAddress	[7] (OCTET STRING)	Not used.		
registeredID	[8] (OID)	Not used.		
onlyContainsUserCerts	BOOLEAN	TRUE for CRLs		
onlyContainsCACerts	BOOLEAN	TRUE for ARLs		
onlySomeReasons	ReasonFlags	Not used.		
ReasonFlags	BIT STRING	BIT STRING		
unspecified	(0)	(0)		
keyCompromise	(1)	(1)		
cACompromise	(2)	Not used.		
affiliationChanged	(3)	(3)		
superseded	(4)	(4)		
cessationOfOperation	(5)	(5)		
certificateHold	(6)	Not used.		
removeFromCRL	(8)	Not used.		
indirectCRL	BOOLEAN	Not used.		
deltaCRLIndicator	CRLNumber	Not used.	Support is optional.	Compliant.
CRLNumber	INTEGER			Not used.
CRL Entry extensions				
reasonCode	CRLReason	CRLReason	Support is optional. Can exclude extension using Alternative Settings if unspecified **reasonCode** is specified.	Compliant.
CRLReason	CHOICE	CHOICE		
unspecified	(0) (BIT STRING)	(0) (BIT STRING)		
keyCompromise	(1) (BIT STRING)	(1) (BIT STRING)		
cACompromise	(2) (BIT STRING)	Not used.		
affiliationChanged	(3) (BIT STRING)	(3) (BIT STRING)		
superseded	(4) (BIT STRING)	(4) (BIT STRING)		
cessationofOperation	(5) (BIT STRING)	(5) (BIT STRING)		
certificateHold	(6) (BIT STRING)	Not used.		
removeFromCRL	(8) (BIT STRING)	Not used.		
holdInstructionCode	CHOICE	Not used.	Support is optional.	Compliant.
HoldInstruction	OID			
id-holdinstruction-none	OID			Not used.
id-holdinstruction-callissuer	OID			

Extension/Entry Extension	PKIX value	GOC PKI		
		Type or value	Notes	CRL
id-holdinstruction-reject	OID			
invalidityDate	Date/Time (GeneralizedTime)	GeneralizedTime	Only included in the CRL if the revocation reason is **keyCompromise**.	Compliant.
certificateIssuer	GeneralNames	Not used.	Support is optional.	Compliant. Not used.
GeneralNames	SEQUENCE			
GeneralName	CHOICE			
OtherName	TYPE-IDENTIFIER			
rfc822Name	[1] (IA5String)			
DNSName	[2] (IA5String)			
x400Address	[3] (ORAddress)			
DirectoryName	[4] (DN)			
EdiPartyName	[5] (EDIPartyName)			
Uri	[6] (IA5String)			
IPAddress	[7] (OCTET STRING)			
RegisteredID	[8] (OID)			

5 Glossary

ARL	Authority Revocation List
ASN.1	Abstract Syntax Notation 1
CA	Certification Authority
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules
DN	Distinguished Name
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EE	End Entity
ECDSA	Elliptic Curve Digital Signature Algorithm
GOC PKI	Government of Canada Public Key Infrastructure
GUI	Graphical User Interface
IETF	Internet Engineering Task Force
IPSec	Internet Protocol Security
ITU-T	International Telecommunications Union Telecommunications Sector
KEA	Key Exchange Algorithm or Key Encryption Algorithm
LDAP	Lightweight Directory Access Protocol
MD5	Message Digest 5
OID	Object Identifier
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509 based
SCA	Subordinate CA
SHA-1	Secure Hash Algorithm 1
SSL	Secure Socket Layer
TLS	Transport Layer Security
v1, v2, v3	Version1, Version 2, Version 3

6 References

1. ITU-T Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Authentication Framework. June 1997.

2. RFC 2459. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. R. Housley, W. Ford, W. Polk, and D. Solo. January 1999.