ITU-T RECOMMENDATION X.509 (2000) | ISO/IEC 9594-8:2001

Technical Corrigendum 1 (DTC 2)

(covering resolutions to defect reports 272, 273, 274, 275, 276, 277, 278 & 279)

This corrects the defects reported in defect report 272

*In clause 8.4.2.1, add the following text to the end of the paragraph that begins with "The **pathLenConstraint** component shall be present only if…"*

The constraint takes effect beginning with the next certificate in the path. The constraint restricts the length of the segment of the certification path between the certificate containing this extension and the end-entity certificate. It has no impact on the number of CA-certificates in the certification path between the trust anchor and the certificate containing this extension. Therefore, the length of a complete certification path may exceed the maximum length of the segment constrained by this extension. The constraint controls the number of non self-issued CA certificates between the CA certificate containing the constraint and the end-entity certificate. Therefore the total length of this segment of the path, excluding self-issued certificates, may exceed the value of the constraint by as many as two certificates. (This includes the certificates at the two endpoints of the segment plus the CA certificates between the two endpoints that are constrained by the value of this extension.)

In clause 15.5.2.1, in the paragraph that begins with "The pathLenConstraint component is meaningful only if…", replace the last two sentences of this paragraph with the following:

The constraint restricts the length of the segment of the delegation path between the certificate containing this extension and the end-entity certificate. It has no impact on the number of AA-certificates in the delegation path between the trust anchor and the certificate containing this extension. Therefore, the length of a complete delegation path may exceed the maximum length of the segment constrained by this extension. The constraint controls the number of AA certificates between the AA certificate containing the constraint and the end-entity certificate. Therefore the total length of this segment of the path may exceed the value of the constraint by as many as two certificates. (This includes the certificates at the two endpoints of the segment plus the AA certificates between the two endpoints that are constrained by the value of this extension.)

This corrects the defects reported in defect report 273

Replace clause 8.4.2.2 with the following:

8.4.2.2 Name constraints extension

This field, which shall be used only in a CA-certificate, indicates a name space within which all subject names in subsequent certificates in a certification path must be located. This field is defined as follows:

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


NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees       [0]   GeneralSubtrees  OPTIONAL,
    excludedSubtrees        [1]   GeneralSubtrees  OPTIONAL,
```

```
    requiredNameForms    [2]   NameForms OPTIONAL   }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

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

BaseDistance ::= INTEGER (0..MAX)


NameForms  ::= SEQUENCE {
    basicNameForms  [0]    BasicNameForms  OPTIONAL,
    otherNameForms  [1]    SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ --none; i.e.:at least one component shall be present-- }))

BasicNameForms  ::= BIT STRING {
    rfc822Name                (0),
    dNSName                   (1),
    x400Address               (2),
    directoryName             (3),
    ediPartyName              (4),
    uniformResourceIdentifier (5),
    iPAddress                 (6),
    registeredID              (7) }  (SIZE (1..MAX))
```

If present, the **permittedSubtrees** and **excludedSubtrees** components each specify one or more naming subtrees, each defined by the name of the root of the subtree and optionally, within that subtree, an area that is bounded by upper and/or lower levels. If **permittedSubtrees** is present, subject names within these subtrees are acceptable. If **excludedSubtrees** is present, any certificate issued by the subject CA or subsequent CAs in the certification path that has a subject name within these subtrees is unacceptable. If both **permittedSubtrees** and **excludedSubtrees** are present and the name spaces overlap, the exclusion statement takes precedence for names within that overlap. If neither permitted nor excluded subtrees are specified for a name form, then any name within that name form is acceptable. If **requiredNameForms** is present, all subsequent certificates in the certification path must include a name of at least one of the required name forms.

If **permittedSubtrees** is present, the following applies to all subsequent certificates in the path. If any certificate contains a subject name (in the **subject** field or **subjectAltNames** extension) of a name form for which permitted subtrees are specified, the name must fall within at least one of the specified subtrees. If any certificate contains only subject names of name forms other than those for which permittee subtrees are specified, the subject names are not required to fall within any of the specified subtrees. For example, assume that two permitted subtrees are specified, one for the DN name form and one for the rfc822 name form, no excluded subtrees are specified, but **requiredNameForms** is specified with the **directoryName** bit and **rfc822Name** bit present. A certificate that contained only names other than a directory name or rfc822 name would be unacceptable. If requiredNameForms were not specified, however, such a certificate would be acceptable. For example, assume that two permitted subtrees are specified, one for the DN name form and one for the rfc822 name form, no excluded subtrees are specified, and **requiredNameForms** is not present. A certificate that only contained a DN and where the DN is within the specified permitted subtree, would be acceptable. A certificate that contained both a DN and an rfc822 name and where only one of them is within its specified permitted subtree, would be unacceptable. A certificate that contained only names other than a DN or rfc822 name would also be acceptable.

If **excludedSubtrees** is present, any certificate issued by the subject CA or subsequent CAs in the certification path that has a subject name (in the **subject** field or **subjectAltNames** extension) within these subtrees is unacceptable. For example, assume that two excluded subtrees are specified, one for the DN name form and one for the rfc822 name form. A certificate that only contained a DN and where the DN is within the specified excluded subtree, would be unacceptable. A certificate that contained both a DN and an rfc822 name and where at least one of them is within its specified excluded subtree, would be unacceptable.

When a certificate subject has multiple names of the same name form (including, in the case of the **directoryName** name form, the name in the subject field of the certificate if non-null) then all such names shall be tested for consistency with a name constraint of that name form.

If **requiredNameForms** is present, all subsequent certificates in the certification path must include a subject name of at least one of the required name forms.

Of the name forms available through the **GeneralName** type, only those name forms that have a well-defined hierarchical structure may be used in the **permittedSubtrees** and **excludedSubtrees** fields. The **directoryName** name form satisfies this requirement; when using this name form a naming subtree corresponds to a DIT subtree.

The **minimum** field specifies the upper bound of the area within the subtree. All names whose final name component is above the level specified are not contained within the area. A value of **minimum** equal to zero (the default) corresponds to the base, i.e. the top node of the subtree. For example, if **minimum** is set to one, then the naming subtree excludes the base node but includes subordinate nodes.

The **maximum** field specifies the lower bound of the area within the subtree. All names whose last component is below the level specified are not contained within the area. A value of **maximum** of zero corresponds to the base, i.e. the top of the subtree. An absent **maximum** component indicates that no lower limit should be imposed on the area within the subtree. For example, if **maximum** is set to one, then the naming subtree excludes all nodes except the subtree base and its immediate subordinates.

This extension may, at the option of the certificate issuer, be either critical or non-critical. It is recommended that it be flagged critical, otherwise a certificate user may not check that subsequent certificates in a certification path are located in the name space intended by the issuing CA.

Conformant implementations are not required to recognize all possible name forms.

If the extension is present and is flagged critical, a certificate-using implementation must recognize and process all name forms for which there is both a subtree specification (permitted or excluded) in the extension and a corresponding value in the **subject** field or **subjectAltNames** extension of any subsequent certificate in the certification path. If an unrecognized name form appears in both a subtree specification and a subsequent certificate, that certificate shall be handled as if an unrecognized critical extension was encountered. If any subject name in the certificate falls within an excluded subtree, the certificate is unacceptable. If a subtree is specified for a name form that is not contained in any subsequent certificate, that subtree can be ignored. If the **requiredNameForms** component specifies only unrecognized name forms, that certificate shall be handled as if an unrecognized critical extension was encountered. Otherwise, at least one of the recognized name forms must appear in all subsequent certificates in the path.

If the extension is present and is flagged non-critical and a certificate-using implementation does not recognize a name form used in any **base** component, then that subtree specification may be ignored. If the extension is flagged non-critical and any of the name forms specified in the **requiredNameForms** component are not recognized by the certificate-using implementation, then the certificate shall be treated as if the **requiredNameForms** component was absent.

In clause 10.3 add a new path processing variable as follows and renumber subsequent bullets accordingly:

d) *required-name-forms:* A (possibly empty) set of sets of name forms. For each set of name forms, every subsequent certificate must contain a name of one of the name forms in the set.

In clause 10.4 add a new initialization step as follows and renumber subsequent bullets accordingly:

d) Initialize the *required-name-forms* to an empty set;

In clause 10.5, add a step to the checks applied to all certificates as follows:

h) If the certificate is not an intermediate self-issued certificate, and if required-name-forms *is not an empty set, for each set of name forms in* required-name-forms *check that there is a subject name in the certificate of one of the name forms in the set.*

In clause 10.5, add a step to the constraint recording actions applied to intermediate certificates as follows:

c) If the ***nameConstraints*** *extension with a* ***requiredNameForms*** *component is present in the certificate, set the* required-name-forms *variable to the union of its previous value and the set consisting of the set of name forms specified in the certificate extension. If the* ***requiredNameForms*** *component contains more than one name form, the* required-name-forms *variable shall signal that a name of at least one of the indicated name forms in this extension shall be present in all subsequent certificates. The union of a previous value of* the required-name-forms *variable with the value from the current certificate extension is a set of sets signalling requirements for all subsequent certificates. For example if the current* required-name-forms *is set to requiring that either a DN or an rfc822 name must be present in certificates and the current extension in the certificate being processed indicates that either rfc822 names or DNS names are required, the resulting union that is the new* required-name-forms *indicates that each of the subsequent certificates must have either an rfc822 name or both a DN and a DNS name.*

In Annex A, ***certificateExtensions*** *module update the asn.1 for* ***nameConstraints*** *extension as above*

In Annex A, ***certificateExtensions*** *module add the following:*

```
id-ce-nameConstraint            OBJECT IDENTIFIER ::=  {id-ce 30 1}
```

In Annex A, ***certificateExtensions*** *module, delete the following:*

```
id-ce-nameConstraints                  OBJECT IDENTIFIER ::=  {id-ce 30}
```

In Annex A, ***certificateExtensions*** *module, add the following to the set of OIDs not used in this specification:*

```
id-ce 30
```

This corrects the defects reported in defect report 274

In clause 12.1 and Annex A, in the ***AttributeCertificateInfo*** *ASN.1 production, replace:*

```
version          AttCertVersion DEFAULT v1,
```

with:

```
version          AttCertVersion  --version is v2,
```

In clause 12.1 and Annex A, replace the ***AttCertVersion*** *ASN.1 production with:*

```
AttCertVersion ::=  INTEGER  {v2(1) }
```

In clause 12.1 and Annex A, replace the ***AttCertIssuer*** *ASN.1 production with:*

```
AttCertIssuer  ::=  [0]   SEQUENCE {
     issuerName           GeneralNames   OPTIONAL,
      baseCertificateID [0]   IssuerSerial   OPTIONAL,
      objectDigestInfo  [1]   ObjectDigestInfo   OPTIONAL  }
```

In clause 12.1 and Annex A, in the ASN.1 Holder production:

Replace the comment under objectDigestInfo that reads "—if present, version must be v2" with the following asn.1 comment "—*used to directly authenticate the holder, eg. an executable*"

In clause 12.1, replace the first paragraph that follows the ASN.1 with the following:

The **version** differentiates between different versions of the attribute certificate. For attribute certificates issued in accordance with the syntax in this specification, **version** must be **v2**.

This corrects the defects reported in defect report 275

*In clause 8.2.2.4, add the following as a new second paragraph following the ASN.1 for the **extendedKeyUsage** extension.*

A CA may assert any-extended-key-usage by using the **anyExtendedKeyUsage** identifier. This enables a CA to issue a certificate that contains OIDs for extended key usages that may be required by certificate-using applications, without restricting the certificate to only those key usages. If extended key usage would restrict key usage, then the inclusion of this OID removes that restriction.

anyExtendedKeyUsage OBJECT IDENTIFIER ::= { 2 5 29 37 0 }

This corrects the defects reported in defect report 276

In clause 8.1.5,

In the last sentence, replace "and explicit-policy-pending indicators" with "explicit-policy-pending and inhibit-any-policy indicators".

In clause 8.4.2.4, in the first sentence

Replace "for all certificates in the certification path" with "for all non-self-issued certificates in the certification path".

In clause 10.5, in the first bullet list, step e),

Replace "or if the *inhibit-any-policy-indicator* is set, then delete" with "or if the *inhibit-any-policy-indicator* is set and the certificate is not a self-issued intermediate certificate, then delete".

This corrects the defects reported in defect report 277

In clause 8.4.2.3, in the last sentence of the first paragraph,

Replace "which is the subject of a subsequent certificate" with "which is the issuer of a subsequent certificate".

This corrects the defects reported in defect report 278

In clause 8.6.2.6, in the first sentence,

Replace "shall be used only as a certificate extension and may be…" with "may be used either as a certificate or CRL extension. Within certificates, this extension may be…"

This corrects the defects reported in defect report 279

In clause 7, add the following immediately after the ASN.1 ***CrossCertificates*** *production:*

PkiPath ::= SEQUENCE OF Certificate

PkiPath is used to represent a certification path. Within the sequence, the order of certificates is such that the subject of the first certificate is the issuer of the second certificate, etc.

In clause 11.1.6,

Replace "object class **pkiCA**" with "**pkiCA** or **pkiUser**".

In the last sentence of the last paragraph of clause 7,

Replace "component of **CertPath**" with "component of **CertPath** or a value of **Certificate** in **PkiPath**."

In clause 11.2.10,

Delete the **PkiPath** ASN.1 production.

In the first sentence of 11.2.10,

Replace "cross-certificates" with "certificates".

In clause 11.2.10, replace the text following the ASN.1 with the following:

This attribute can be stored in a directory entry of object class **pkiCA** or **pkiUser**.

When stored in **pkiCA** entries, values of this attribute contain certification paths excluding end-entity certificates. As such, the attribute is used to store certification paths that are frequently used by relying parties associated with that CA. A value of this attribute can be used in conjunction with any end-entity certificate issued by the last certificate subject in the attribute value.

When stored in **pkiUser** entries, values of this attribute contain certification paths that include the end-entity certificate. In this case, the end-entity is the user whose entry holds this attribute. The values of the attribute represent complete certification paths for certificates issued to this user.

In clause 11.3.9, in the last sentence of the first paragraph,

Replace "issued to the CA that issued the end-entity certificate being validated." with "issued to the specified subject".
