|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **itu-old** | INTERNATIONAL TELECOMMUNICATION UNION | | | | COM 17 – C 269 – E |
| **TELECOMMUNICATION STANDARDIZATION SECTOR**  STUDY PERIOD 2013-2016 | | | **September 2014** | |
| **English only**  **Original: English** | |
| **Question(s):** | | 11/17 |  | | |
| **STUDY GROUP 17 – CONTRIBUTION 269** | | | | | |
| **Source:** | | Denmark | | | |
| **Title:** | | Proposal for additional restructuring and updates of Rec. ITU-T X.509 | ISO/IEC 9594-8. | | | |

Justification

To prepare Rec. ITU-T X.509 | ISO/IEC 9594-8 for new areas, such a smart grid, it requires some basic internal alignment and some restructuring. In particular, the following should be noted:

a) As Rec. ITU-T X.509 | ISO/IEC 9594-8 has been under development since 1988 with changing editors and contributors, there has been a shift in terminology through the years. A consistent terminology adapted to the current usage should be applied.

b) Some updates are necessary to align the document with the current ITU-T editing guidelines (e.g., use lower case rather than upper case.

c) Other updates are necessary to align the document with the current ASN.1 terminology.

d) There needs to be a more clear separation between public-key certificates and attribute certificates.

e) Ambiguity should be clarified.

Inclusion of new features will be presented in separate proposals.

The actual proposal

Replace the summary with:

Summary

Recommendation ITU-T X.509 | ISO/IEC 9594-8 defines frameworks for public-key certificates and attribute certificates. The public-key certificate framework is the base specification for public-key certificates, for the different components going into a public-key infrastructure (PKI) for validation procedures and for public-key certificate revocation, etc. The attribute certificate framework is the base specification for attribute certificates and the different components going into the Privilege Management Infrastructure (PMI). These frameworks may be used by standards bodies to profile their application to PKIs and PMIs.

Replace the fourth paragraph of the introduction:

Information objects for holding PKI and PMI objects in the Directory and for comparing presented values with stored values are also defined.

with:

Schema definitions are defined for holding PKI and PMI information in a directory according to the specification found in the ITU-T .500 series of Recommendations | ISO/IEC 9594, all parts or according to the lightweight directory access protocol (LDAP) specification.

Change the seventh paragraph of Introduction as shown:

The extensibility function was added in an earlier edition with version 3 of the public-key certificate and with version 2 of the certificate revocation list and was incorporated into the attribute certificate from its initial inception. This function is specified in clause 8. It is anticipated that any enhancements to this edition can be accommodated using this function and avoid the need to create new versions.

Update the 2nd paragraph of clause 1 as shown:

The public-key certificate framework defined in this Recommendation | International Standard includes a definition of the information objects for a public-key infrastructure (PKI), including public-key certificates and cCertificate Rrevocation Llists (CRLs). The attribute certificate framework includes a definition of the information objects for a Pprivilege Mmanagement Iinfrastructure (PMI), including attribute certificates, and aAttribute Ccertificate Rrevocation Llists (ACRLs). This Recommendation | International Standard also provides the framework for issuing, managing, using and revoking certificates. An extensibility mechanism is included in the defined formats for both certificate types and for all revocation list schemes. This Recommendation | International Standard also includes a set of standard extensions for each, which is expected to be generally useful across a number of applications of PKI and PMI. The schema components (including object classes, attribute types and matching rules) for storing PKI and PMI objectsinformation in the Directorya directory, are included in this Recommendation | International Standard. Other elements of PKI and PMI, beyond these frameworks, such as key and certificate management protocols, operational protocols, additional certificate and CRL extensions are expected to be defined by other standards bodies (e.g., ISO TC 68, IETF, etc.).

Delete the fourth paragraph of clause 1 including the three bullet points.

Delete the last paragraph of clause 1.

In clause 2.4, replace:

– IETF RFC 1035 (1987), *Domain names – implementation* and *specification*.

with:

– IETF RFC 1123 (1989), Requirements for Internet Hosts – Application and Support.

Add the following references to clause 2.4:

– IETF RFC 3492 (2003), Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA).

– IETF RFC 4511 (2006), Lightweight Directory Access Protocol (LDAP): The Protocol.

– IETF RFC 5890 (2010), Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework.

– IETF RFC 5914 (2010), Trust Anchor Format.

– IETF RFC 5934 (2010), Trust Anchor Management Protocol (TAMP).

In clause 3.1, delete item l (password) and renumber subsequent items.

Change 3.5.2 as shown:

**3.5.2** Aa**ttribute** Aa**uthority (AA)**: An authority which assigns privileges by issuing attribute certificates.

Change 3.5.3 as shown:

**3.5.3 attribute authority revocation list (AARL)**: A revocation list containing a list of references to attribute certificates issued to attribute authorities (AAs) that are no longer considered valid by the issuing attribute authority.

Change 3.5.8 as shown:

**3.5.8 base** CRLcertificate revocation list: A CRLcertificate revocation list that is used as the foundation in the generation of a dCRLdelta certificate revocation list.

Delete 3.5.14 (certificate using system)

Change 3.5.18 as shown:

**3.5.18 certification path**: An ordered list of one or more public-key certificates, starting with a public-key certificate signed by the trust anchor, and ending with the public key certificate to be validated. All intermediate public-key certificates, if any, are CA-certificates in which the subject of the preceding public-key certificate is the issuer of the following public-key certificate.

Change 3.5.24 as shown:

**3.5.24 delegation path**: An ordered sequence of attribute certificates which together with the authentication of a privilege asserter's identity, can be processed to verify the authenticity of an asserter's privilege.

Change 3.5.33 as shown:

**3.5.33 full** CRLcertificate revocation list: A complete revocation list that contains entries for all public-key certificates that have been revoked for the given scope.

Change 3.5.34 as shown:

**3.5.34 hash function**: A (mathematical) function which maps values from a large (possibly very large) domain into a smaller range. A "good" hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range called a digest.

Change 3.5.61 as shown:

**3.5.61 self-issued attribute certificate**: An attribute certificate where the issuer and the subjectholder are the same Aattribute Aauthority. An Aattribute Aauthority might use a self-issued ACattribute certificate, for example, to publish policy information.

Add a new definition:

3.5.68 trust anchor store: A trust anchor information collection at a relying party for one or more trust anchors.

Add a new abbreviations clause 4:

IDN Internationalized Domain Name

LDAP Lightweight Directory Access Protocol

LDH Letters, Digits, Hyphen

RDN Relative Distinguished Name

Change the second paragraph of clause 5 as shown:

This Specification presents ASN.1 notation in the bold courier new typeface. When ASN.1 types and values are referenced in normal text, they are differentiated from normal text by presenting them in the bold Courier New typeface. The names of procedures, typically referenced when specifying the semantics of processing, are differentiated from normal text by displaying them in bold Ttimes Nnew Rroman. Access control permissions are presented in italicized Times New Roman. When a definition is referenced for the first time in normal text it is also presented in italicized Ttimes Nnew Rroman.

Change the first bullet of clause 6 as shown:

– establishing a trusted path of public-key certificates between a trusted entity called a trust anchor (see clause 8.5) and the public-key certificate subject, i.e., the entity for which the public-key certificate has been issued;

Change the second list of bullets of clause 6 as shown:

– ensuring that the privileges in the attribute certificate are sufficient when compared against the privilege policy;

– establishing a trusted delegation path of attribute certificates if necessary;

– verifying the digital signature on each attribute certificate in the path;

– ensuring that each issuer was authorized to delegate privileges; and

– validating that the attribute certificates have not expired or been revoked by their issuers.

Change the note in clause 6.1 as shown:

NOTE – The encipherment using the private key is intended to ensures that the signature cannot be forged. The one-way nature of the hash function ensures that false information, generated to have the same hash result (and thus signature) cannot be substituted.

Update clause 7 as shown:

# 7 Trust Models

The public-key certificate framework defined here is for use by applications with requirements for authentication, integrity, confidentiality and non-repudiation.

The binding of a public-key to an entity is provided by an authority through a digitally signed data structure called a public-key certificate. The format of public-key certificates is defined here, including an extensibility mechanism and a set of specific public-key certificate extensions. If, for any reason, an authority revokes a previously issued public-key certificate, users need to be able to learn that revocation has occurred so they do not use an untrustworthy public-key certificate. Revocation lists are one scheme that can be used to notify users of revocations. The format of revocation lists is defined here, including an extensibility mechanism and a set of revocation list extensions. In both the certificate and revocation list case, other bodies may also define additional extensions that are useful to their specific environments.

An entity that makes decision based on the validity of a public-key certificates is called the relying party. A relying party needs to validate a public-key certificate prior to using that public-key certificate for a particular transaction in an application. Procedures for performing that validation are also defined here, including verifying the integrity of the public-key certificate itself, its revocation status, and its validity with respect to the intended use. In the three cornered trust model described below, the relying party acts on its own behalf. In the four-cornered trust model the relying party engages the services of a trust broker to validate public-key certificates on its behalf. When the term relying party is used in this specification, if the four-cornered trust model is being employed, then it is assumed that the trust broker is acting on behalf of the relying party.

Update clause 8.1 as shown:

## 8.1 Introduction

In order for a relying party to be able to trust a public-key of another userentity, for instance to authenticate the identity of that userentity, the public- key shall be obtained from a trusted source. Such a source, called a Ccertification Aauthority (CA), certifies a public key by issuing a public-key certificate which binds the public-key to the entity which holds the corresponding private-key. The procedures used by a CA to ensure that an entity is in fact in possession of the private key and other procedures related to the issuance of public-key certificates are outside the scope of this Specification. However, trust brokers and relying parties are the entities that should review these procedures and make trust decisions based upon them. The public-key certificate, the form of which is specified later in this clause, has the following properties:

– any relying party with access to the public key of the CA can recover the public key which was certified;

– no party other than the CA can modify the public-key certificate without this being detected (public-key certificates are unforgeable).

Because public-key certificates are unforgeable, they can be published by being placed in the Directory, without the need for the latter to make special efforts to protect them.

NOTE – Although the CAs are unambiguously defined by a distinguished name in the DIT, this does not imply that there is any relationship between the organization of the CAs and the DIT.

Replace the signature component specification clause 8.2 with:

The signature component shall contain the algorithm identifier for the signature algorithm used by the CA in signing the certificate (e.g., sha256WithRSAEncryption, sha384WithRSAEncryption, dsa-with-sha256, etc.). It shall be the same value as used in the algorithmIdentifier component of the SIGNATURE data type when signing the public-key certificate.

NOTE 2 – By including this component, signature algorithm is protected by the signature.

Update the last paragraph of clause 8.2 as shown:

A userAn entity may obtain one or more public-key certificates from one or more CAs. Each public-key certificate bears the name of the CA which issued it.

Update the first paragraph of 8.3 as shown:

The extensions fieldcomponent of a public-key certificate allows for the addition of new fieldsextensions to the structure without modification to the basic ASN.1 definitiondata type. An extension field consists of an extension identifier, a criticality flag, and an encoding of a data value of an ASN.1 type associated with the identified extension. For those extensions where ordering of individual extensions within the SEQUENCE is significant, the specification of those individual extensions shall include the rules for the significance of the order therein. When a relying party processing a public-key certificate does not recognize an extension and the criticality flag is FALSE, it may ignore that extension. If the criticality flag is TRUE, unrecognized extensions shall cause the structure to be considered invalid, i.e., in a public-key certificate, an unrecognized critical extension wouldshall cause validation of a signature using that public-key certificate to fail. When a relying party recognizes and is able to fully process an extension, then the relying party shall process the extension regardless of the value of the criticality flag. When a relying party recognizes and is able to partially process an extension for which the criticality flag is TRUE, then its behaviour in the presence of unrecognized elements is extension specific and may be documented in each extension. However, the default behaviour, when not specified specifically for an extension, is to treat the entire extension as unrecognized. If unrecognized elements appear within the extension, and the extension is not marked critical, those unrecognized elements shall be ignored according to the rules of extensibility documented in clause 12.2.2 in Rec. ITU‑T X.519 | ISO/IEC 9594-5.

Update the paragraph of 8.3 following the list items as shown:

Some extensions shall always can only be marked critical. In these cases, a relying party that understands the extension processes it; the acceptance/rejection of the public-key certificate is dependent (at least in part) on the content of the extension. A relying party that does not understand the extension shall reject the public-key certificate.

Some extensions can onlyshall always be marked non-critical. In these cases, a relying party that understands the extension shall process it and acceptance/rejection of the public-key certificate is dependent (at least in part) on the content of the extension. A relying party that does not understand the extension accepts the public-key certificate (unless factors other than this extension cause it to be rejected).

Some extensions may be marked either as critical or as non-critical. In these cases, a relying party that understands the extension processes it: the acceptance/rejection of the public-key certificate is dependent (at least in part) on the content of the extension, regardless of the criticality flag. A relying party that does not understand the extension accepts the public-key certificate if the extension is marked non-critical (unless factors other than this extension cause it to be rejected) and rejects the public-key certificate if the extension is marked critical.

When a CA considers including an extension in a public-key certificate it does so with the expectation that its intent will be adhered to wherever possible. If it is necessary that the content of the extension be considered prior to any reliance on the public-key certificate, a CA shall flag the extension critical. This is done with the realization that any relying party that does not process the extension will reject the public-key certificate (probably limiting the set of applications that can verify the public-key certificate). The CA may mark certain extensions non-critical to achieve backward compatibility with validation applications that cannot process the extensions. Where the need for backward compatibility and interoperability with validation applications incapable of processing the extensions is more vital than the ability of the CA to reinforce the extensions, then these optionally critical extensions would be marked non-critical. It is most likely that CAs would set optionally critical extensions as non-critical during a transition period while the verifiers' public-key certificate processing applications are upgraded to ones that can process the extensions.

Specific extensions may be defined in ITU-T Recommendations | International Standards or by any organization which has a need. The object identifier which identifies an extension shall be defined in accordance with Rec. ITU-T X.660 | ISO/IEC 9834-1. StandardSome extensions for public-key certificates are defined in clause 9 of this Specification.

Update the third paragraph of clause 8.7 as shown:

The issuer and subject fields of each public-key certificate are used, in part, to identify a valid path. For each pair of adjacent public-key certificates in a valid certification path, the value of the subject field in one certificate shall match the value of the issuer field in the subsequent certificate. In addition, the value of the issuer field in the public-key certificate issued by the trust anchor shall match the distinguished name of the trust anchor. Only the names in these fields are used when checking the validity of a certification path. Names in certificate extensions are not used for this purpose. The distinguishedNameMatch matching rule, defined in clause 13.5.2 of Rec. ITU‑T X.501 | ISO/IEC 9594-2, shall be used to compare the distinguished name in the issuer fieldsubject component of one public-key certificate with the distinguished name in the subject fieldissuer component of anothera subsequent public-key certificate.

Update the note in clause 8.7 as shown:

NOTE – The CertificationPath data type had already been defined by the first edition of this Specification before the concept of certification path was fully developed. The order of elements in a CertificationPath instance is opposite of that of a certification path. This data type is used, as an example, by the directory protocols for the support of strong authentication and electronic signature (see Rec. ITU-T X.511 | ISO/IEC 9594-3). It is recommended that new applications use the PkiPath data type.

Update clause 9.2.27 as shown:

### 9.2.2 Public-key certificate and CRL extension fields

The following extension fields are defined:

a) Aauthority key identifier;

b) Ssubject key identifier;

c) Kkey usage;

d) Eextended key usage;

e) Pprivate key usage period;

f) Ccertificate policies;

g) Ppolicy mappings.

These extension fields shall be used only as public-key certificate extensions, except for authority key identifier which may also be used as a CRL extension. Unless otherwise noted, these extensions may be used in both CA-certificates and end-entity public-key certificates.

Update 9.2.2.4 as shown:

#### 9.2.2.4 Extended key usage extension

This fieldxtension 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 field. This fieldextension is defined as follows:

extKeyUsage EXTENSION ::= {

SYNTAX SEQUENCE SIZE (1..MAX) OF KeyPurposeId

IDENTIFIED BY id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER

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

anyExtendedKeyUsage OBJECT IDENTIFIER ::= { id-ce-extKeyUsage 0 }

Key purposes may be defined by any organization with a need. Object identifiers used to identify key purposes shall be assigned in accordance with Rec. ITU-T X.660 | ISO/IEC 9834-1.

This extension may, at the option of the issuing CA, be either critical or non-critical.

If the extension is flagged critical, then the public-key certificate shall be used only for one of the purposes indicated.

If the extension is flagged non-critical, then it indicates the intended purpose or purposes of the key, and may be used in finding the correct key/certificate of an entity that has multiple keys/certificates. If this extension is present, and the relying party recognizes and processes the extendedKeyUsage extension type, then the relying party shall ensure that the public-key certificate shall be used only for one of the purposes indicated. (Using applications(A relying party may nevertheless require that a particular purpose be indicated in order for the public-key certificate to be acceptable to that application.)

If a public-key certificate contains both a critical key usage fieldextension and a critical extended key usage fieldextension, then both fieldsextensions shall be processed independently and the public-key certificate shall only be used for a purpose consistent with both fields. If there is no purpose consistent with both fieldsextensions, then the public-key certificate shall not be used for any purpose.

This Specification defines the following key purpose that can be included in the extended key usage extension. Other purposes that can also be included are defined in other specifications, such as IETF RFC 5280.

keyPurposes OBJECT IDENTIFIER ::= {id-kp 1}

Update first paragraph of clause 9.2.2.6 as shown:

#### 9.2.2.6 Certificate policies extension

This fieldextension lists certificate policies, recognized by the issuing CA, that apply to the public-key certificate, together with optional qualifier information pertaining to these certificate policies. The list of certificate policies is used in determining the validity of a certification path, as described in clause 11. The optional qualifiers are not used in the certification path processing procedure, but relevant qualifiers are provided as an output of that process to the certificate using applicationrelying party to assist in determining whether a valid path is appropriate for the particular transaction. Typically, different certificate policies will relate to different applications which may use the certified key. The presence of this extension in an end-entity public-key certificate indicates the certificate policies for which this public-key certificate is valid. The presence of this extension in a CA-certificate (cross-certificate) issued by one CA to another CA indicates the certificate policies for which certification paths containing this public-key certificate may be valid. This fieldextension is defined as follows:

Change the first paragraph after the ASN.1 of clause 9.2.2.6 as shown:

A value of the PolicyInformation type identifies and conveys qualifier information for one certificate policy. The component policyIdentifier component contains an identifier of a certificate policy and the component policyQualifiers component contains policy qualifier values for that element.

Update the bullet list of 9.3.2.1 as shown:

#### 9.3.2.1 Subject alternative name extension

– the x400Address alternative is an O/R address defined in accordance with by Rec. ITU-T X.411 | ISO/IEC 10021-4;

– the directoryName alternative is a distinguished name defined in accordance withby Rec. ITU-T X.501 | ISO/IEC 9594-2;

– the ediPartyName alternative is a name of a form agreed between communicating Electronic Data Interchangeelectronic data interchange (EDI) partners; the nameAssigner component identifies an authority that assigns unique values of names in the partyName component;

– the uniformResourceIdentifier alternative is a Uniform Resource Identifieruniform resource identifier for the worldwide web defined in accordance with IETF RFC 1630;

– the iPAddress alternative is an Internet Protocol address defined in accordance with IETF RFC 791, represented as a binary string. for IPv4 (four octets) or in accordance with IETF 2460 for IPv6 (16 octets).

– the registeredID alternative is an object identifier of any registered object assigned in accordance with Rec. ITU-T X.660 | ISO/IEC 9834-1.

In clause 9.3.2.2, change the paragraph after the ASN.1 as follows:'

This extension may, at the option of the certificate or CRL issuer, be either critical or non-critical. An implementationA relying party which supports this extension is not required to be able to process all name forms. If the extension is flagged critical, at least one of the name forms that are present shall be recognized and processed, otherwise the public-key certificate or CRL shall be considered invalid. Apart from the preceding restriction, a relying party is permitted to ignore any name with an unrecognized or unsupported name form. It is recommended that, provided the issuer field of the certificate or CRL contains a distinguished name that unambiguously identifies the issuing authority, this fieldextension be flagged non-critical.

Change the fifth paragraph from the end of clause 9.4.2.3 as shown:

Conformant implementationsrelying parties are not required to recognize all possible name forms. If the extension is flagged as being critical and a certificate-using implementationrelying party does not recognize a name form used in any base component, the public-key certificate shall be handled as if an unrecognized critical extension had been encountered. If the extension is flagged as being non-critical and a certificate-using implementationrelying party does not recognize a name form used in any base component, then that subtree may be ignored.

In clause 9.5.1, change item b) as shown:

b) Some CRL usersrelying parties may wish to respond differently to a revocation, depending upon the reason for the revocation. There is therefore a requirement for a CRL entry to indicate the reason for the revocation.

Change the last paragraph of clause 9.5.2.9 as shown:

The scope of a CRL containing this extension is extended to include the revocation status of revoked certificates that expired atafter the exact timedate specified in the ExpiredCertsOnCRL extension or afterat that timedate. The revocation status of a certificate shall not be updated once the certificate has expired. If limitations in the CRL's scope are specified (by either reason codes or by distribution points), that applies to expired certificates as well. The revocation status of a certificate shall not be updated once the certificate has expired.

Move clause 9.5.2.2, CRL scope extension to Annex K as K.1.

In clause 9.6.2, move item c) to Section 3

Move clause 9.6.2.7 to Section 3 as 16.7.1

In SECTION 3, change third paragraph as shown:

An A privilege verifier needs to validate an attribute certificate-using system needs to validate a certificate prior to using that attribute certificate for an application. Procedures for performing that validation are also defined here, including verifying the integrity of the attribute certificate itself, its revocation status, and its validity with respect to the intended use.

Add a new subclause 16.1.2.7

#### 16.1.2.7 Authority key identifier extension

This extension is defined for public-key certificates in clause 9.2.2.1, but may also be applicable to attribute certificates and ACRLs.

The extension enables distinct keys used by the same attribute certificate and/or ACRL signing entity to be distinguished.

The ASN.1 in clause 9.2.2.1 applies without change.

The key may be identified by an explicit key identifier in the keyIdentifier component, by the identification of a public-key certificate for the key (giving certificate issuer in the authorityCertIssuer component and public-key certificate serial number in the authorityCertSerialNumber component), or by both explicit key identifier and identification of a public-key certificate for the key. If both forms of identification are used they shall identify the same public-key certificate. A key identifier shall be unique with respect to all key identifiers for the issuing authority for the attribute certificate or ACRL containing the extension. An implementation which supports this extension is not required to be able to process all name forms in the authorityCertIssuer component. (See clause 9.3.2.1 for details of the GeneralNames type.)

NOTE – An attribute certificate has the baseCertificateId components in the holder component of the AttributeCertificateInfo, there is no need for this extension.

Add a new second level heading:

## 16.7 Revocation list distribution point and delta revocation list extensions