DRAFT AMENDMENT

ISO/IEC 9594-8:2017 DAM 1

ISO/IEC JTC **1**/SC **6**

Secretariat: **KATS**

Voting begins on:
2019-02-14

Voting terminates on:
2019-05-09

Information technology — Open Systems Interconnection — The Directory —

Part 8:
Public-key and attribute certificate frameworks

AMENDMENT 1: General updates

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — L'annuaire —

Partie 8: Cadre général des certificats de clé publique et d'attribut

AMENDEMENT 1: Titre manque

ICS: 35.100.70

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC 9594-8:2017/DAM 1:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

TITLE: Rec. ITU-T X.509 (2016) | ISO/IEC 9594-8:2017 Information Technology - Open systems Interconnection - The Directory: Public-key and attribute certificate frameworks – PDAM for Adm. 1: General updates

SOURCE: Collaborative ITU-T and ISO/IEC JTC1 meeting on the Directory, Tokyo, Japan, 27-31 August 2018

Introduction

Update the fourth paragraph shown:

Directory s~~S~~chema definitions are defined for holding PKI and PMI information in a directory according to the specification found in the Directory Specifications (Rec. ITU-T X.500 | ISO/IEC 9594-1, Rec. ITU-T X.501 | ISO/IEC 9594-2, Rec. ITU-T X.511 | ISO/IEC 9594-3, Rec. ITU-T X.518 | ISO/IEC 9594-4, Rec. ITU-T X.519 | ISO/IEC 9594-5, Rec. ITU-T X.520 | ISO/IEC 9594-6, Rec ITU-T X.521 | ISO/IEC 9594-7 and Rec. ITU-T X.525 | ISO/IEC 9594-9) ~~ITU-T X.500 series of Recommendations | ISO/IEC 9594 (all parts)~~ or according to the lightweight directory access protocol (LDAP) specification.

2.1 Identical Recommendations | International Standards

Move the following references from the Bibliography to here:

- Recommendation ITU-T X.680 (2015) | ISO/IEC 8824-1:2015, Information technology – Abstract Syntax Notation One (ASN.1): *Specification of basic notation*.
- Recommendation ITU-T X.681 (2015) | ISO/IEC 8824-2:2015, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- Recommendation ITU-T X.682 (2015) | ISO/IEC 8824-3:2015, Information technology – Abstract Syntax Notation One (ASN.1): *Constraint specification*.
- Recommendation ITU-T X.683 (2015) | ISO/IEC 8824-4:2015, Information technology – Abstract Syntax Notation One (ASN.1): *Parameterization of ASN.1 specifications*.

Add the following reference:

- Recommendation ITU-T X.690 (2015) | ISO/IEC 8825-1:2015, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.

3.5 Public-key and attribute certificate definitions

Add the following new definition (renumber later):

3.5.41a multiple cryptographic algorithms public-key certificate: Public-key certificate that for migration purposes includes extensions for an alternative public-key algorithm, an alternative digital signature algorithm and an alternative digital signature.

4 Abbreviations

Add the following new abbreviation:

ASN.1	Abstract Syntax Notation One

Remove the following abbreviations (moved to new part):

AVMP	Authorization Validation Management Protocol
CASP	Certification Authority Subscription Protocol

5 Conventions

Change the second paragraph as shown:

The term "The Directory Specifications" shall be taken to mean Rec. ITU-T X.500 | ISO/IEC 9594-1, Rec. ITU-T X.501 | ISO/IEC 9594-2, Rec. ITU-T X.511 | ISO/IEC 9594-3, Rec. ITU-T X.518 | ISO/IEC 9594-4, Rec. ITU-T X.519 | ISO/IEC 9594-5, Rec. ITU-T X.520 | ISO/IEC 9594-6, Rec ITU-T X.521 | ISO/IEC 9594-7 and Rec. ITU-T X.525 | ISO/IEC 9594-9~~the other parts of the ITU-T X.500 series of Recommendations | ISO/IEC 9594 (all parts), excluding this Specification~~.

Add two new paragraphs:

If an International Standard or ITU-T Recommendation is referenced within normative text without an indication of the edition, the edition shall be taken to be the latest one as specified in the normative references clause.

This Specification makes extensive use of the Abstract Syntax Notation One (ASN.1) for the formal specification of data types and values, as it is specified in Rec. ITU-T X.680 | ISO/IEC 8824-1, Rec. ITU-T X.681 | ISO/IEC 8824-2, Rec. ITU-T X.682 | ISO/IEC 8824-3, Rec. ITU-T X.683 (2015) | ISO/IEC 8824-4 and Rec. ITU-T X.690 | ISO/IEC 8825-1.

Add a new subclause 6.2.3:

6.2.3 Migration of cryptographic algorithms

Cryptographic algorithms that were once considered safe to use later becomes vulnerable to attack as computer technology and mathematics evolves. Both within a PKI and PMI, migration to more safe cryptographic algorithms raises several practical issues, as it is unlike that it is possible simultaneously to change cryptographic algorithms all over a PKI or PMI.

To allow smooth migration of a PKI or a PMI from a legacy use of one set of cryptographic algorithms to an assumingly stronger set of cryptographic algorithms, this Specification includes provisions for an alternative set of cryptographic algorithms. This to allow some entities to move to a more safe set of algorithms while at the same time legacy entities during a migration period may keep using the old set of cryptographic algorithms.

Add a new heading just below the 7.2 heading:

7.2.1 Public-key certificate syntax

Add new subclause 7.2.2:

7.2.2 Multiple cryptographic algorithms for public-key certificates

It is possible for a public-key certificate to hold:

- subject's alternative public key information to be used instead of the information provided in the **`subjectPublicKeyInfo`** component;
- an alternative signature algorithm to be used instead of the algorithm specified in the **`signature`** component; and
- an alternative digital signature to be checked instead of the native digital signature.

Such alternative specifications are provided as public-key certificates extensions.

A public-key certificate that includes the alternative cryptographic algorithm extensions shall contain alternative public key information in the **`subjectAltPublicKeyInfo`** extension (see clause 9.8.2), an alternative signature algorithm value in the **`altSignatureAlgorithm`** extension (see clause 9.8.3) and an alternative signature value in the **`altSignatureValue`** extension (see clause 9.8.4).

A CA generating a public-key certificate with the alternative cryptographic algorithms and alternative digital signature shall:

- when generating the value in the **`altSignatureValue`** extension exclude the **`signature`** component and the **`altSignatureValue`** extension from the public-key certificate and generate the digital signature over the remaining DER encoded public-key certificate using the algorithm specified in the **`altSignatureAlgorithm`** extension; and
- when generating the value in the **`signature`** component, the **`subjectAltPublicKeyInfo`**, the **`altSignatureAlgorithm`** and the **`altSignatureValue`** extensions shall be included in the DER encoding of the public-key certificate.

A relying party that has not migrated to support alternative cryptographic algorithms and alternative digital signatures shall verify the native digital signature. The **subjectAltPublicKeyInfo**, **altSignatureAlgorithm** and **altSignatureValue** extensions shall then be included in the DER encoding of the public-key certificate.

A relying party that has migrated to support alternative cryptographic algorithms and alternative digital signatures shall verifying the alternative digital signature. The **signature** component and the **altSignatureValue** extension shall then be excluded from the DER encoding of the public-key certificate. Thus, the relying party shall decode the public-key certificate and then re-DER-encode the same public-key certificate after the above modifications have been made, otherwise the validation of the alternative signature will fail.

NOTE – The **subjectAltPublicKeyInfo** extension is only used a public-key certificate extension, while **altSignatureAlgorithm** and **altSignatureValue** extensions are also relevant as CRL extensions.

After the migration period, it is expected that new public-key certificates be issued without these extension and with the new set of cryptographic algorithms and the digital signature in the base part of the public-key certificate.

7.7 Certification path

After the first paragraph, add a new paragraph:

If the public-key certificates in the certification path include multiple cryptographic extensions and the relying party supports this extension, then the relying party shall validate the alternative signature stored in the **altSignatureValue** extensions of the public-key certificates in the path.

Change the note as shown:

NOTE – The **CertificationPath** data type was defined at an early stage ~~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 the 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 digital signature (see Rec. ITU-T X.511 | ISO/IEC 9594-3). It is recommended that new applications use the **PkiPath** data type.

7.9 Public-key certificate creation

Update the first paragraph as shown:

A public-key certificate associates the public key (or multiple public keys in the case of multiple cryptographic public-key certificates) and the ~~unique~~ name of the subject it describes. Thus:

7.10.2 Certificate revocation list syntax

Delete NOTE 1 and renumber subsequent notes (the history is not interesting and just confusing).

Change what is now NOTE 1 as shown:

NOTE ~~1~~2 – By including this component, the signature algorithm is protected by the signature. In addition, by having information about the used hash algorithm at an early stage, it is possible during the pass of the certificate revocation list to start generating the hash for validation.

Add new subclause 7.10.3:

7.10.3 Multiple cryptographic algorithms for certificate revocation lists

It is possible for a CRL to hold:

- an alternative signature algorithm to be used instead of the algorithm specified in the **signature** component; and
- an alternative digital signature to be checked instead of the native digital signature.

Such alternative specifications are provided as CRL extensions.

A CRL that includes the multiple cryptographic algorithm extensions shall have an alternative signature algorithm value in the **altSignatureAlgorithm** extension (see clauses 9.7.3) and an alternative digital signature value in the **altSignatureValue** extension (see clause 9.7.4).

A CRL issuer generating a CRL with the alternative cryptographic algorithms and alternative digital signature shall:

- when generating the value in the **altSignatureValue** extension exclude the **signature** component and the **altSignatureValue** extension from the CRL and generate the digital signature over the remaining DER encoded CRL using the algorithm specified in the **altSignatureAlgorithm** extension; and

– when generating the value in the **`signature`** component, the **`altSignatureAlgorithm`** and the **`altSignatureValue`** extensions shall be included in the DER encoding of the CRL.

A relying party that has not migrated to support alternative cryptographic algorithms and alternative digital signatures shall verify the native digital signature. The **`altSignatureAlgorithm`** and **`altSignatureValue`** extensions shall then be included in the DER encoding of the CRL.

A relying party that has migrated to support alternative cryptographic algorithms and alternative digital signatures shall verifying the alternative digital signature. The **`signature`** component and the **`altSignatureValue`** extension shall then be excluded from the DER encoding of the CRL. Thus, the relying party shall decode the CRL and then re-DER-encode the same CRL after the above modifications have been made, otherwise the validation of the alternative digital signature will fail.

9.2.2.3 Key usage extension

Add a new paragraph after the first paragraph:

For a multiple cryptographic algorithm public-key certificate, the **`keyUsage`** extension applies also to the alternative **`SubjectAltPublicKeyInfo`** in the **`subjectAltPublicKeyInfo`** extension.

9.2.2.4 Extended key usage extension

Add a new paragraph right after the ASN.1:

For a multiple cryptographic algorithm public-key certificate, the **`keyUsage`** extension applies also to the alternative **`SubjectAltPublicKeyInfo`** in the **`subjectAltPublicKeyInfo`** extension.

9.2.2.5 Private key usage period extension

A new paragraph to end of 9.2.2.5:

In a multiple cryptographic algorithm public-key certificate, the private key usage period shall also apply to the alternative private key.

9.6.2.2 Issuing distribution point extension

NOTE 1 – This component was introduced into the Rec. ITU-T X.509 (2000)| ISO/IEC 9594-8:2001 ~~fourth~~ edition of this Specification and removed again in the subsequent ~~fifth~~ edition. Each of these two actions has caused compatibility problems. This component has been re-introduced into the Rec. ITU-T X.509 (2008)| ISO/IEC 9594-8:2008 ~~sixth~~ edition in a way to remove any compatibility issues.

Add a new subclause 9.7

9.8 Alternative cryptographic algorithms and digital signature extensions

9.8.1 Introduction

There is a need to provide capabilities for easy migration within a PKI from one set of cryptographic algorithms to another more suitable set of cryptographic algorithms. The following cryptographic algorithm and digital signature extensions are defined for that purpose:

a) subject alternative public key information extension;

b) alternative signature algorithm extension; and

c) alternative signature value extension.

These extensions may be included in both CA certificates and end-entity public-key certificates. Alternative signature algorithm extension and alternative signature value extension may also be used as CRL extensions.

9.8.2 Subject alternative public key information extension

This public-key certificate extension, when present, shall contain the subject's alternative public key information

```
subjectAltPublicKeyInfo EXTENSION ::= {
  SYNTAX          SubjectAltPublicKeyInfo
  IDENTIFIED BY   id-ce-subjectAltPublicKeyInfo }

SubjectAltPublicKeyInfo ::= SEQUENCE {
  algorithm             AlgorithmIdentifier{{SupportedAlgorithms}},
  subjectAltPublicKey   BIT STRING }
```

The **SubjectAltPublicKeyInfo** data type has the following components:

– the **algorithm** subcomponent shall hold the algorithm which this public key is an instance of; and

– the **subjectAltPublicKey** subcomponent shall hold the alternative public key.

This extension may be flagged as critical or as non-critical.

NOTE – It is recommended that it be flagged as non-critical. Flagging it as critical would require relying parties to understand this extension and the alternative public-key signature algorithm.

9.8.3 Alternative signature algorithm extension

This extension, which may be used as either a public-key certificate extension and CRL extension. It shall contain the algorithm identifier for the alternative signature algorithm used by the signer when creating an alternative digital signature and by the relying party when validating the alternative digital signature.

```
altSignatureAlgorithm EXTENSION ::= {
 SYNTAX           AltSignatureAlgorithm
 IDENTIFIED BY    id-ce-altSignatureAlgorithm }

AltSignatureAlgorithm ::= AlgorithmIdentifier
```

When the **altSignatureAlgorithm** extension is included in a particular value being an instance of a data type supporting extensions, the **altSignatureValue** extension shall also be included.

NOTE 1 – By having a separate **altSignatureAlgorithm** extension, instead of combining it with the **altSignatureValue** extension, the alternative signature algorithm is protected by the alternative signature.

This extension may be flagged either as critical or as non-critical.

NOTE 2 – It is recommended that it be flagged as non-critical. Flagging it as critical would require all relying parties to understand the extension and the alternative public-key signature algorithms.

9.8.4 Alternative signature value

This alternative signature shall be created by the issuer using the alternative private key of the issuer and it shall be validated using the alternative public key of the issuer.

```
altSignatureValue EXTENSION ::= {
  SYNTAX           AltSignatureValue
  IDENTIFIED BY    id-ce-altSignatureValue }

AltSignatureValue ::= BIT STRING
```


This extension can only be created by a signer holding a multiple cryptographic algorithms public-key certificate. When creating the alternative digital signature on an issued public-key certificate, the signer shall use its alternative private key.

The procedures for creating and validating alternative digital signatures are specified in:

– clause 7.2.2 for public-key certificates; and

– clause 7.10.3 for CRLs.

Add a new subclause 11.4:

12.5.1 Basic public-key certificate checks

Replace item a) with:

a) If the relying party has not migrated alternative digital signatures, then check that the signature verifies. Otherwise, check that the alternative signature verifies. Check that dates are valid, that the public-key certificate subject and public-key certificate issuer names chain correctly, and that the public-key certificate has not been revoked.

Add a new subclause 13.2.11:

13.2.11 Supported public-key algorithms attribute type

The **supportedPublicKeyAlgorithms** multivalued attribute type may be used by a CA to inform a subject that applies for a public-key certificate what types of public-key algorithms that are acceptable to the CA.

The **supportedPublicKeyAlgorithms** attribute type has the following syntax:

```
supportedPublicKeyAlgorithms ATTRIBUTE ::= {
```

```
  WITH SYNTAX             SupportedPublicKeyAlgorithms
  EQUALITY MATCHING RULE algorithmIdentifierMatch
  LDAP-SYNTAX             x509SupportedPublicKeyAlgorithms
  LDAP-NAME               {"supportedPublicKeyAlgorithms"}
  LDAP-DESC               "X.509 supported publiv key algorithms"
  ID                      id-at-supportedPublicKeyAlgorithms }

SupportedPublicKeyAlgorithms ::= SEQUENCE {
  algorithmIdentifier      AlgorithmIdentifier{{SupportedPublicKeyAlgorithms}},
  minKeySize               INTEGER,
  extensions           [0]  SEQUENCE SIZE (1..MAX) OF OidOrAttr OPTIONAL,
  ... }

OidOrAttr ::= CHOICE {
  oid       ATTRIBUTE.&id {{ ExtAttributes }},
  attribute Attribute {{ ExtAttributes }},
  ... }
```

The **SupportedPublicKeyAlgorithms** data type has the following components:

a) The **algorithmIdentifier** component shall specify a public-key algorithm acceptable for the CA.

b) The **minKeySize** component shall specify the minimum key size for the key-pair generated based on the algorithm.

c) The **extensions** component, if present, shall specify one or more extensions that are associated with the particular public-key algorithm. Each extension is defined by the **OidOrAttr** data type, which is a choice with the following alternatives:

  – the **oid** alternative specifies the object identifier of the attribute type for the extension to be included, but not the content of the extension;

  – the **attribute** alternative specifies an attribute containing the complete specification of the required extension.

Add new clause 13.4.12:

13.4.12 Supported public-key algorithms syntax


```
x509SupportedPublicKeyAlgos SYNTAX-NAME ::= {
  DESC               "X.509 supported public key algorithms"
  DIRECORY SYNTAX    SupportedPublicKeyAlgorithms
  ID                 id-asx-x509SupportedPublicKeyAlgos }
```

Delete SECTION 4 (moved to a new part)

Annex A

Public-key and attribute certificate frameworks

Replace the first part of the ***AuthenticationFramework*** *module of A.1:*

```
AuthenticationFramework
  {joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 9}
DEFINITIONS ::=
BEGIN

-- EXPORTS All
-- The types and values defined in this module are exported for use in the other ASN.1
-- modules contained within the Directory Specifications, and for the use of other
-- applications which will use them to access Directory services. Other applications may
-- use them for their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Directory service.

IMPORTS
  id-asx, id-at, id-avr, id-ldx, id-lsx, id-mr, id-nf, id-oa, id-oc, id-sc
    FROM UsefulDefinitions
      {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 9} WITH SUCCESSORS

  ATTRIBUTE, DistinguishedName, distinguishedNameMatch, MATCHING-RULE, Name, NAME-FORM,
OBJECT-CLASS,
  objectIdentifierMatch, RelativeDistinguishedName, SYNTAX-NAME, top
    FROM InformationFramework
      {joint-iso-itu-t ds(5) module(1) informationFramework(1) 9} WITH SUCCESSORS

  bitStringMatch, boolean, booleanMatch, caseExactMatch, commonName,
  directoryString, generalizedTime,
  generalizedTimeMatch, generalizedTimeOrderingMatch, integer, integerMatch,
  integerOrderingMatch, octetString, octetStringMatch,
  UnboundedDirectoryString, UniqueIdentifier, uri
    FROM SelectedAttributeTypes
      {joint-iso-itu-t ds(5) module(1) selectedAttributeTypes(5) 9} WITH SUCCESSORS

  algorithmIdentifierMatch, certificateExactMatch, certificateListExactMatch,
  certificatePairExactMatch, CertificatePoliciesSyntax, CertPolicyId, GeneralNames,
  KeyUsage, pkiPathMatch, policyMatch,
  CertificateAssertion, CertificateExactAssertion, CertificateListAssertion,
  CertificateListExactAssertion, CertificatePairAssertion,
  CertificatePairExactAssertion
      FROM CertificateExtensions
        {joint-iso-itu-t ds(5) module(1) certificateExtensions(26) 9} WITH SUCCESSORS ;
```

Add near the end of A. 1:

```
id-at-supportedPublicKeyAlgos  OBJECT IDENTIFIER ::= {id-at 103}

id-lsx-x509SupportedPublicKeyAlgorithms   OBJECT IDENTIFIER ::= {id-lsx 59}
```

Replace the first part of the ***CertificateExtensions*** *module of A.2:*

```
CertificateExtensions
  {joint-iso-itu-t ds(5) module(1) certificateExtensions(26) 9}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL

IMPORTS

  id-at, id-ce, id-ldx, id-mr
    FROM UsefulDefinitions
      {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 9} WITH SUCCESSORS

  Name, RelativeDistinguishedName, Attribute{}, MATCHING-RULE,
  SupportedAttributes, SYNTAX-NAME
    FROM InformationFramework
```