International Telecommunication Union

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

# X.509
## Corrigendum 2
(04/2016)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Directory

# Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks

## Technical Corrigendum 2

Recommendation ITU-T X.509 (2012) – Technical Corrigendum 2

ITU-T X-SERIES RECOMMENDATIONS

## DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

| | |
|---|---|
| PUBLIC DATA NETWORKS | |
| Services and facilities | X.1–X.19 |
| Interfaces | X.20–X.49 |
| Transmission, signalling and switching | X.50–X.89 |
| Network aspects | X.90–X.149 |
| Maintenance | X.150–X.179 |
| Administrative arrangements | X.180–X.199 |
| OPEN SYSTEMS INTERCONNECTION | |
| Model and notation | X.200–X.209 |
| Service definitions | X.210–X.219 |
| Connection-mode protocol specifications | X.220–X.229 |
| Connectionless-mode protocol specifications | X.230–X.239 |
| PICS proformas | X.240–X.259 |
| Protocol Identification | X.260–X.269 |
| Security Protocols | X.270–X.279 |
| Layer Managed Objects | X.280–X.289 |
| Conformance testing | X.290–X.299 |
| INTERWORKING BETWEEN NETWORKS | |
| General | X.300–X.349 |
| Satellite data transmission systems | X.350–X.369 |
| IP-based networks | X.370–X.379 |
| MESSAGE HANDLING SYSTEMS | X.400–X.499 |
| **DIRECTORY** | **X.500–X.599** |
| OSI NETWORKING AND SYSTEM ASPECTS | |
| Networking | X.600–X.629 |
| Efficiency | X.630–X.639 |
| Quality of service | X.640–X.649 |
| Naming, Addressing and Registration | X.650–X.679 |
| Abstract Syntax Notation One (ASN.1) | X.680–X.699 |
| OSI MANAGEMENT | |
| Systems management framework and architecture | X.700–X.709 |
| Management communication service and protocol | X.710–X.719 |
| Structure of management information | X.720–X.729 |
| Management functions and ODMA functions | X.730–X.799 |
| SECURITY | X.800–X.849 |
| OSI APPLICATIONS | |
| Commitment, concurrency and recovery | X.850–X.859 |
| Transaction processing | X.860–X.879 |
| Remote operations | X.880–X.889 |
| Generic applications of ASN.1 | X.890–X.899 |
| OPEN DISTRIBUTED PROCESSING | X.900–X.999 |
| INFORMATION AND NETWORK SECURITY | X.1000–X.1099 |
| SECURE APPLICATIONS AND SERVICES | X.1100–X.1199 |
| CYBERSPACE SECURITY | X.1200–X.1299 |
| SECURE APPLICATIONS AND SERVICES | X.1300–X.1399 |
| CYBERSECURITY INFORMATION EXCHANGE | X.1500–X.1599 |
| CLOUD COMPUTING SECURITY | X.1600–X.1699 |

*For further details, please refer to the list of ITU-T Recommendations.*

**INTERNATIONAL STANDARD ISO/IEC 9594-8**
**RECOMMENDATION ITU-T X.509**

# Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks

## Technical Corrigendum 2

### Summary

Technical Corrigendum 2 to Rec. ITU-T X.509 (2012) | ISO/IEC 9594-8:2014 covers resolution to defect reports 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419 and 420.

### History

| Edition | Recommendation | Approval | Study Group | Unique ID[*] |
|---|---|---|---|---|
| 1.0 | ITU-T X.509 | 1988-11-25 | | 11.1002/1000/2999 |
| 2.0 | ITU-T X.509 | 1993-11-16 | 7 | 11.1002/1000/3000 |
| 3.0 | ITU-T X.509 | 1997-08-09 | 7 | 11.1002/1000/4123 |
| 3.1 | ITU-T X.509 (1997) Technical Cor. 1 | 2000-03-31 | 7 | 11.1002/1000/5033 |
| 3.2 | ITU-T X.509 (1997) Technical Cor. 2 | 2001-02-02 | 7 | 11.1002/1000/5311 |
| 3.3 | ITU-T X.509 (1997) Technical Cor. 3 | 2001-10-29 | 7 | 11.1002/1000/5559 |
| 3.4 | ITU-T X.509 (1997) Technical Cor. 4 | 2002-04-13 | 17 | 11.1002/1000/6025 |
| 3.5 | ITU-T X.509 (1997) Technical Cor. 5 | 2003-02-13 | 17 | 11.1002/1000/6236 |
| 3.6 | ITU-T X.509 (1997) Technical Cor. 6 | 2004-04-29 | 17 | 11.1002/1000/7285 |
| 4.0 | ITU-T X.509 | 2000-03-31 | 7 | 11.1002/1000/5034 |
| 4.1 | ITU-T X.509 (2000) Technical Cor. 1 | 2001-10-29 | 7 | 11.1002/1000/5560 |
| 4.2 | ITU-T X.509 (2000) Technical Cor. 2 | 2002-04-13 | 17 | 11.1002/1000/6026 |
| 4.3 | ITU-T X.509 (2000) Technical Cor. 3 | 2004-04-29 | 17 | 11.1002/1000/7284 |
| 4.4 | ITU-T X.509 (2000) Technical Cor. 4 | 2007-01-13 | 17 | 11.1002/1000/8637 |
| 5.0 | ITU-T X.509 | 2005-08-29 | 17 | 11.1002/1000/8501 |
| 5.1 | ITU-T X.509 (2005) Cor. 1 | 2007-01-13 | 17 | 11.1002/1000/9051 |
| 5.2 | ITU-T X.509 (2005) Cor. 2 | 2008-11-13 | 17 | 11.1002/1000/9591 |
| 5.3 | ITU-T X.509 (2005) Cor. 3 | 2011-02-13 | 17 | 11.1002/1000/11042 |
| 5.4 | ITU-T X.509 (2005) Cor. 4 | 2012-04-13 | 17 | 11.1002/1000/11577 |
| 6.0 | ITU-T X.509 | 2008-11-13 | 17 | 11.1002/1000/9590 |
| 6.1 | ITU-T X.509 (2008) Cor. 1 | 2011-02-13 | 17 | 11.1002/1000/11043 |
| 6.2 | ITU-T X.509 (2008) Cor. 2 | 2012-04-13 | 17 | 11.1002/1000/11578 |
| 6.3 | ITU-T X.509 (2008) Cor. 3 | 2012-10-14 | 17 | 11.1002/1000/11736 |
| 7.0 | ITU-T X.509 | 2012-10-14 | 17 | 11.1002/1000/11735 |
| 7.1 | ITU-T X.509 (2012) Cor. 1 | 2015-05-29 | 17 | 11.1002/1000/12474 |
| 7.2 | ITU-T X.509 (2012) Cor. 2 | 2016-04-29 | 17 | 11.1002/1000/12844 |

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

## CONTENTS

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

# Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks

# Technical Corrigendum 2

*(Covering resolution to defect reports 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419 and 420)*

## 1) Correction of the defects reported in defect report 406

*In clause 8.5.3.1 and Annex A, update the ASN.1 for* **`CRLReason`** *as follows:*

```
CRLReason ::= ENUMERATED {
  unspecified          (0),
  keyCompromise        (1),
  cAcompromise         (2),
  affiliationChanged   (3),
  superseded           (4),
  cessationOfOperation (5),
  certificateHold      (6),
  removeFromCRL        (8),
  privilegeWithdrawn   (9),
  aACompromise         (10),
  ... ,
  weakAlgorithmOrKey   (11) }
```

*Add a new bullet point:*

– **`weekAlgorithm`** indicates that the certificate was revoked due to a weak cryptographic algorithm and/or key (e.g., due to short key length or unsafe key generation).

*In clause 8.6.2.1 and Annex A, update the ASN.1 for* **`ReasonFlags`** *as follows:*

```
ReasonFlags ::= BIT STRING {
  unused                (0),
  keyCompromise         (1),
  cACompromise          (2),
  affiliationChanged    (3),
  superseded            (4),
  cessationOfOperation  (5),
  certificateHold       (6),
  privilegeWithdrawn    (7),
  aACompromise          (8),
  weakAlgorithmOrKey    (9) }(SIZE(0..9,...,10))
```

## 2) Correction of the defects reported in defect report 407

*Add the following abbreviations to clause 4:*

| | |
|---|---|
| DSA | Digital Signature Algorithm |
| ECC | Elliptic Curve Cryptography |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| RSA | Rivest-Shamir-Adelman |

*Delete the old DSA abbreviation*

*Update 6.1 as shown*

### 6.1 Digital signatures

*Replace this clause with:*

This subclause describes digital signatures in general. Sections 2 and 3 of this Directory Specification discuss the use of digital signatures within PKI and PMI specifically. This subclause is not intended to specify a specification for digital signatures in general, but to specify the means by which instances of the PKI and PMI specific data types are signed.

There are different types of digital signatures, such as RSA digital signatures, DSA digital signatures and ECDSA digital signatures.

NOTE 1 – It is not within the scope of this Specification to describe these different techniques in details, but Annex F gives a short introduction with references to more detailed specifications.

Figure 1 illustrates how a signer of instances of PKI/PMI data types (public-key certificates, attribute certificates, revocation lists, etc.) creates a digital signature and then adds that to the data before transmission. It also illustrates how the recipient of the signed data (the validator) validates the digital signature.

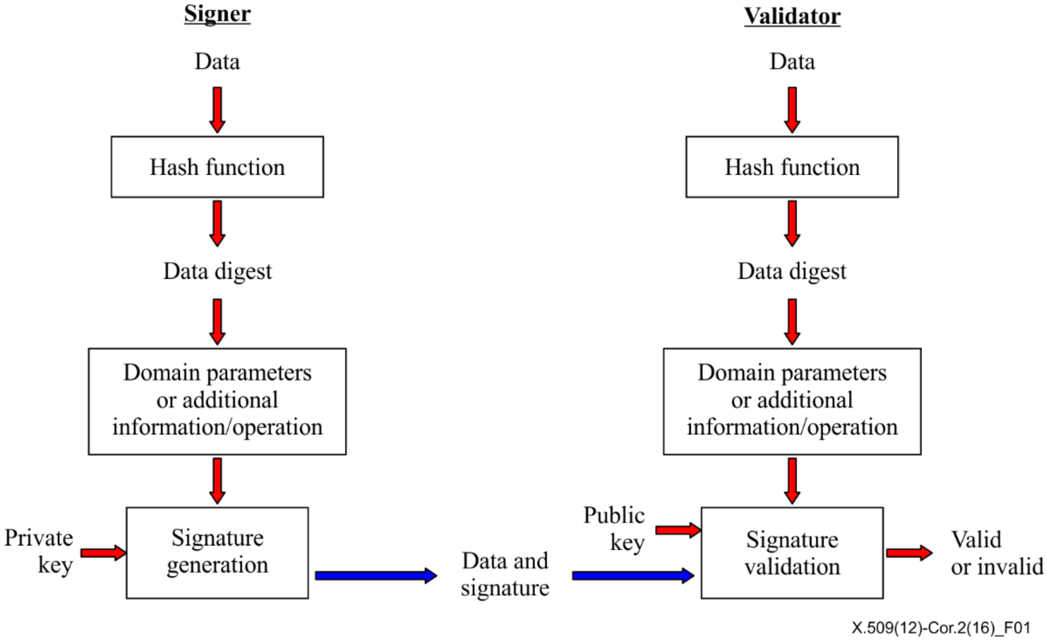

**Figure 1 – Digital signature generation and validation**

The digital signature signer goes through the following procedure:

1) The signer creates a hash digest over the PKI/PMI data using a secure hashing algorithm (see Annex F).
2) The hash digest is then supplemented with additional information in preparation for generating of the digital signature for improved security and for padding the hash digest to a length required by the asymmetric cryptographic function. For the RSA algorithms that can be adding some information to the hash digest and in some cases, to perform yet another hashing operation. For the DSA and ECDSA signature algorithms additional domain parameters are added.
3) The result from item 2) together with the private key of the signer and the use of a specific algorithm result in a bit string that together with an identity of the used algorithm constitute the digital signature.
4) The signature is appended to data to be signed.

Having received the data, the recipient (validator) goes through a similar procedure:

1) The validator goes through the same procedure as in steps 1) and 2) above, and if the received data is unmodified, the result will be the same as for the signer. If not, the next step will fail.
2) The result from item 1) together with the public key of the signer, the bit string of the signature and the use of a corresponding algorithm, the digital signature is evaluated as either valid or invalid.

If the digital signature proves valid, the validator has ensured that the data has not been modified and that the signer is in the position of the private key that corresponds to the public key used by the validator, i.e., the digital signature provides insurance of data integrity and authentication of the signer.

If the digital signature proves invalid, either the data has been modified or the signing private key does not corresponds to the public key used by the validator.

NOTE 2 – Data to be stored in a database or a directory may also be appended a digital signature, which then can evaluated at the retrieval of the data.

*Replace clause 6.2 with:*

## 6.2 Public-key cryptography and cryptographic algorithms

### 6.2.1 Formal specification of public-key cryptography

The digital signature of a data item may expressed by the following ASN.1 data type, where the **`signature`** component is a bit string resulting from using the appropriate signature algorithm.

```
SIGNATURE ::= SEQUENCE {
  algorithmIdentifier  AlgorithmIdentifier{{SupportedAlgorithms}},
  signature            BIT STRING,
  ... }
```

In the case where a signature is appended to the data, the following ASN.1 may be used to define the data type resulting from applying a signature to the given data type.

```
SIGNED{ToBeSigned} ::= SEQUENCE {
  toBeSigned    ToBeSigned,
  COMPONENTS OF SIGNATURE,
  ... }
```

The following data type are is not used anymore by this Specification. It may be useful in other areas and is retained for possible import by referencing specifications.

```
HASH{ToBeHashed} ::= SEQUENCE {
  algorithmIdentifier  AlgorithmIdentifier{{SupportedAlgorithms}},
  hashValue            BIT STRING (CONSTRAINED BY {
   -- shall be the result of applying a hashing procedure to the DER-encoded
   -- octets of a value of -- ToBeHashed } ),
  ... }
```

The following data types are deprecated and are not used anymore by this Specification. They are retained for possible import by referencing specifications.

```
ENCRYPTED{ToBeEnciphered} ::= BIT STRING (CONSTRAINED BY {
  -- shall be the result of applying an encipherment procedure
  -- to the BER-encoded octets of a value of -- ToBeEnciphered } )

ENCRYPTED-HASH{ToBeSigned} ::= BIT STRING (CONSTRAINED BY {
  -- shall be the result of applying a hashing procedure to the DER-encoded (see 6.2)
  -- octets of a value of -- ToBeSigned -- and then applying an encipherment procedure
  -- to those octets -- } )
```

### 6.2.2 Formal definitions of cryptographic algorithms

The following ASN.1 information object class is used for specifying cryptographic algorithms.

```
ALGORITHM ::= CLASS {
  &Type          OPTIONAL,
  &id            OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
  [PARMS        &Type]
  IDENTIFIED BY &id }
```

The following general data type specifies the syntax of an algorithm specification:

```
AlgorithmIdentifier{ALGORITHM:SupportedAlgorithms} ::= SEQUENCE {
  algorithm   ALGORITHM.&id({SupportedAlgorithms}),
  parameters  ALGORITHM.&Type({SupportedAlgorithms}{@algorithm}) OPTIONAL,
  ... }
```

The **`algorithm`** component shall be an object identifier that uniquely identifies the cryptographic algorithm being defined.

The **`parameters`** component, when present, shall specify the parameters associated with the algorithm. Some, but not all algorithms require associated parameters.

```
/* The definitions of the following information object set is deferred to referencing
specifications having a requirement for specific information object sets.*/
```

```
SupportedAlgorithms ALGORITHM ::= {...}
```

The elliptic curve algorithms use the `parameters` component to specify the object identifier identifying a particular curve. The following object gives a general specification for an ECC public key algorithm:

```
ecPublicKey ALGORITHM ::= {
  PARMS       SupportedCurves
  IDENTIFIED  BY id-ecPublicKey }
```

The `ecPublicKey` algorithm is defined in IETF RFC 5480 and provided here for easy reference. The associated object identifier is defined as:

```
id-ecPublicKey OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) ansi-X9-62(10045)
                                      keyType(2) 1 }

/* The definitions of the following information value set is deferred to referencing
specifications having a requirement for specific value sets.*/

SupportedCurves OBJECT IDENTIFIER ::= {dummyCurv, ...}

dummyCurv OBJECT IDENTIFIER ::= {2 5 5}
```

NOTE – The ASN.1 requires that a value set shall hold at least one value. Not to make a preference for a specific curve, a dummy value is used here that might be replaced by a referencing specifications or implementers' agreement. The shown object identifier is not otherwise used by this specification.

*In clause 7.2, remove the* ***AlgorithmIdentifier*** *datatype, the* ***SupportedAlgorithms*** *object set and the* ***ALGORITHM*** *object class.*

*In Annex A, replace:*

```
ALGORITHM ::= CLASS {
  &Type          OPTIONAL,
  &id            OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
                [&Type]
  IDENTIFIED BY &id }
```

*with:*

```
ALGORITHM ::= CLASS {
  &Type          OPTIONAL,
  &id            OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
  [PARMS        &Type]
  IDENTIFIED BY &id }
```

*After this new definition of ALGORITHM, add:*

```
ecPublicKey ALGORITHM ::= {
  PARMS       SupportedCurves
  IDENTIFIED  BY id-ecPublicKey }

id-ecPublicKey OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) ansi-X9-62(10045)
                                      keyType(2) 1 }

/* The definitions of the following information value set is deferred to referencing
specifications having a requirement for specific value sets.*/

SupportedCurves OBJECT IDENTIFIER ::= {dummyCurv, ...}

dummyCurv OBJECT IDENTIFIER ::= {2 5 5}
```

## 3) Correction of the defects reported in defect report 408

*Update 3.5.26 as shown:*

**3.5.26** **end entity**: Either a public-key certificate subject that uses its private key for purposes other than signing public-key certificates, or an attribute certificate holder that cannot delegate privileges of the attribute certificate, ~~that~~ but uses its attributes only to gain access to a resource.

*Update 3.5.27 as shown:*

**3.5.27 end-entity attribute certificate**: An attribute certificate issued to an entity, which then acts as an end entity within a privilege management infrastructure.

*Delete 3.5.29.*

*Update 3.5.30 as shown:*

**3.5.30 end-entity public-key certificate**: A public-key certificate issued to an entity, which then acts as an end entity within a public-key infrastructure.

## 4) Correction of the defects reported in defect report 409

*Replace 3.5.2 with:*

**3.5.2 attribute authority (AA)**: An authority which assigns privilege attributes to entities by either issuing attribute certificates or including them in public-key certificates. In the latter case, the authority is also a certification authority.

## 5) Correction of the defects reported in defect report 410

*Add the following new definition to clause 3.5 in alphabetical order, and reorder subsequent clauses accordingly:*

**3.5.x privilege holder**: An entity that has been assigned privilege. A privilege holder may assert its privilege for a particular purpose

## 6) Correction of the defects reported in defect report 411

*Change 11.3.5 to say:*

The `distributionPoint` component, when present, matches if the stored attribute value contains an issuing distribution point extension and the value of this component in the presented value equals the corresponding value, in at least one name form, in that extension

## 7) Correction of the defects reported in defect report 412

*Delete item g) of clause 8.4.1.*

## 8) Correction of the defects reported in defect report 413

*Replace 3.5.36 with:*

**3.5.36 indirect CRL (iCRL)**: A revocation list whose scope includes public-key certificates issued by one or more CAs other than the issuer of the revocation list. The same indirect CRL is also authoritative for the public-key certificates, if any, issued by the CRL issuer.

*Add the following new definition to clause 3.5 in alphabetical order, and reorder subsequent clauses accordingly:*

**3.5.x indirect ACRL (iACRL)**: A revocation list whose scope includes attribute certificates issued by one or more AAs other than the issuer of the revocation list. The same indirect ACRL is also authoritative for the attribute certificates, if any, issued by the ACRL issuer.

*Add new clauses 7.11 and 7.12 and renumber current clause 7.11 to clause 7.13*

### 7.11 Uniqueness of names

A PKI is requires that CAs uniquely and unambiguously named. If CRL issuing authorities are not uniquely named, it may result in incorrect use of revocation information.

It is outside the scope of this Specification specify procedures that ensure unique and unambiguous names for CA CRL issuing authorities.

## 7.12 Indirect CRLs

### 7.12.1 Introduction

The only mechanism defined for CRL delegation by this Specification (and IETF RFC 5280) is for the public-key certificate issuing CA to include a **`cRLDistributionPoint`** extension in a public-key certificate and include the **`cRLIssuer`** component in this extension. The public-key certificate issuing CA will have do this for each certificate whose revocation status the CA wishes to delegate via CRL to a CRL issuing authority.

There is no mechanism (i.e., public-key certificate or CRL extension) for a public-key certificate issuing CA to delegate CRL issuance for all its public-key certificate to another authority using a mechanism (similar to the delegated OCSP Responder public-key certificate as specified in IETF RFC 6960).

For example, if a certificate issuing CA has issued a large number of public-key certificates and its wishes to delegate CRL issuance for all of these public-key certificate to a CRL issuing authority, the CA shall assert the **`cRLIssuer`** component in the **`cRLDistributionPoint`** extension of each of the issued public-key certificates. If the CA wishes to delegate issuance only for some of the issued public-key certificates, the CA shall assert the **`cRLIssuer`** component in the **`cRLDistributionPoint`** extension of the delegated public-key certificates and shall not assert the **`cRLIssuer`** component in the remaining public-key certificates that contain the **`cRLDistributionPoint`** extension.

The relationship of CRL delegation may be as follows:

a) A CA can delegate issuance of CRL for a given public-key certificate to multiple CRL issuance authorities. The CA might delegate to multiple authorities for the sake of redundancy by asserting multiple CRL issuers in a single distribution point in the **`cRLDistributionPoint`** extension or by asserting multiple distribution points in the **`cRLDistributionPoint`** extension with each distribution point containing one or more CRL issuer(s). Another example is a CA delegating CRL issuance to different authorities for different reason codes. In this case, the CRL Distribution Point extension must contain two or more distribution points with each distribution point containing applicable reason code(s) and CRL Issuer(s).

b) A CA can delegate issuance of CRL for different batch of public-key certificates to different CRL issuance authorities. The CA could create these batches stochastically or using a deterministic algorithm such as based on type of public-key certificate, reason code, issuance time, expiration time, subject organization, etc.

c) A CRL issuance authority can be authoritative for revocation information for public-key certificates issued by multiple CAs.

### 7.12.2 Indirect CRL contents

If a CRL issuance authority is a CA the CRLs it issues are authoritative for the public-key certificates issued by the CRL issuance authority as the CA and the public-key certificates whose revocation status is delegated to the CRL issuance authority. Thus, a CRL issued by a CRL issuance authority which has been delegated CRL issuance by m CAs, is authoritative for m or m + 1 CAs depending on whether the CRL issuance authority is a CA or not.

The CRL issued by the CRL issuance authority can be partitioned like any other CRLs using **`distributionPoint`** component of the **`cRLDistributionPoint`** extension. Furthermore, the CRL issuance authority may or may not choose to partition the CRL based on the public-key certificate issuer. If it chooses the former, it creates a partitioned CRL for each CA. But, the partitioned CRL discussion is outside the scope of this Specification.

Since the iCRL is authoritative for the CA(s) other that the CRL issuer, serial number alone in the CRL entry does not uniquely identity a public-key certificate that has been revoked. You also need to identify the CA that issued the public-key certificate placed on the iCRL. This is achieved by adding the **`certificateIssuer`** CRL entry extension. This extension shall always be flagged as critical to ensure that the relying parties process it and associated the CRL entry with the appropriate public-key certificate.

If each entry on a CRL contained the **`certificateIssuer`** extension (which is a directory distinguished name), it would make the CRL size large. Thus, in order to reduce the CRL size, the iCRL issuing authority should sort the CRL entries by issuing CA. Using this approach, only the first public-key certificate appearing on the CRL for a given CA needs to contain the **`certificateIssuer`** extension. All subsequent entries are assumed to for the same public-key certificate issuing CA until another **`certificateIssuer`** CRL entry extension is encountered. To further reduce the size of the CRL, if the iCRL issuing authority is a CA, it should contain its revoked public-key certificate first, obviating the need for **`certificateIssuer`** extension for any of its certificates.

NOTE 1 – The following example illustrates the use of iCRLs. In the example, there is single CRL issuing authority that issues revocation lists for multiple CAs. The issuing distribution point extension is present in that CRL is and flagged as critical. The **`indirectCRL`** component of this extension is set to **`TRUE`**. If the CRL issuing authority name is same as the name for one the CAs it serves, entries should then be placed first on the CRL without the certificate issuer extension. Entries for other CAs are kept together and the first CRL entry for a particular CA has included the certificate issuer extension flagged as critical.

> NOTE 2 – A relying party needs to develop and validate the certification path for the iCRL issuance authority. This is no different for building certification path for regular CRL with one difference. In the case of regular CRL, there is a probability that the CRL is signed using the same key as the public-key certificate, obviating the need for building a CRL certification path. However, for the indirect CRL, the CRL certification path will always differ from the certification path for the certificate whose revocation status is being checked.

*In 8.6.2.1 – CRL distribution points extension, add the following note right after* **`reasons`** *component description, and renumber subsequent notes accordingly:*

> NOTE 1 – While this components allows for shorter CRLs, it has the side-effect that a relying party has to search multiple CRLs to ensure that a particular public-key certificate has not been revoked. The possibility for segmenting CRLs should be used with caution.

## 9) Correction of the defects reported in defect report 414

*Add a new paragraph after the first bullet list of clause 6:*

The public-key infrastructure (PKI) is the infrastructure established to support the issuing, revocation and validation of public-key certificates.

*Add a new paragraph after the second bullet list of clause 6:*

The privilege management infrastructure (PMI) is the infrastructure established to support the issuing, revocation and validation of attribute-key certificates.

*Add a new paragraph after the penultimate paragraph:*

An entity may take one more roles in a PKI and/or a PMI. It may act as a CA, as an AA, as an end entity in a PKI environment (PKI end entity), as an end entity in a PMI environment (PMI end entity), as a relying party, etc.

A PKI end entity is an entity that has been assigned an end-entity public-key certificate, where the private key cannot be used to sign other public-key certificates, but may be used for signing for other purposes. A PMI end entity is an entity that uses its end-entity attribute certificate to assess privilege, but it cannot be used for delegating such privilege to other entities.

An entity may have multiple roles. As an example, an entity acting as a PKI end entity may also act as an AA using it private key to sign attribute certificates.

*Change 3.5.6 as shown:*

**3.5.6 authority**: An entity, responsible for the issuance of certificates or of revocation lists. Four ~~Two~~ types are defined in this Recommendation | International Standard; a certification authority which issues public-key certificates, ~~and~~ an attribute authority which issues attribute certificates, a CRL issuer which issues CRL and an ACRL issuer which issues ACRLs.

*Add the following two new definitions in clause 3.5 in alphabetical order, and renumber subsequent clauses accordingly:*

**3.5.x PKI end entity**: An entity that is acting as an end entity in a PKI environment, where the subject uses it private key for other purposes than signing public-key certificates.

**3.5.x PMI end entity**: An entity that is acting as an end entity in a PMI environment, where the holder uses its privilege attributes to gain access to a resource.

## 10) Correction of the defects reported in defect report 415

*Add the following definition in clause 3.5 in alphabetical order, and renumber subsequent clauses accordingly:*

**3.5.x attribute authority certificate**: An attribute certificate for one attribute authority issued by another attribute authority or by the same attribute authority.

## 11) Correction of the defects reported in defect report 416

*In clause 6, just after the second bullet list, add:*

Privileges are provided in directory attributes as defined by Rec. ITU-T X.501 | ISO/IEC 9594-2.

Public-key certificates may also include directory attributes for carrying privileges. Such aspects of privileges carried by public-key certificates are covered by the attribute certificate framework.

*In clause 8.3.2.3, update the text below the ASN.1 as shown:*

This extension may, at the option of the issuing CA ~~certificate issuer~~, be flagged either as critical or as non-critical. A relaying party ~~certificate-using system~~ processing this extension is not required to understand all attribute types included in the extension. If the extension is flagged as critical, at least one of the attribute types contained in the extension shall be understood for the certificate to be accepted. If the extension is flagged as critical and none of the contained attribute types is understood, the certificate shall be considered invalid~~rejected~~.

A relying party may require that it understands all the attribute types to accept the public-key certificate.

This extension is intended to associate identity and/or privilege with the subject of the public-key certificate. When the subject accesses an entity by including identity information, the accessed entity may, based on local rules, assign privilege to the subject, e.g., assign privilege used for access control. Such identity information may be different from or a supplement to the identity information supplied in the `subject` component and/or in the `subjecatAltName` extension (if present).

If privilege is included in the extension, this privilege has to be supplied by some authority, which may be the CA itself or some other associated authority. Section 3 expands on this issue.

If this extension is present in a public-key certificate and flagged as critical, some of the extensions defined in clause 15 may also be present as stated in the individual extensions.

## 12) Correction of the defects reported in defect report 417

*In clause 8.4.2.1, update from the third paragraph after ASN.1 as shown:*

This extension shall be supported by a conformant relying party.

This extension shall be present in a CA certificate with the `cA` component set to `TRUE` and flagged as critical (which requires that a CA certificate shall be a version 3 public-key certificate).

This extension may, when included in an end-entity public-key certificate at the option of the issuing CA, be either flagged as critical or as non-critical. ~~It is recommended that it be flagged critical, otherwise, an entity which is not authorized to be a CA may issue certificates and a relying party may unwittingly use such a certificate.~~

When ~~If~~ this extension is present ~~and is flagged critical, or is flagged non-critical but is recognized by the relying party~~, then:

– if the value of `cA` is not set to `TRUE` then the certified public key shall not be used to verify a public-key certificate signature;

– if the value of `cA` is set to `TRUE` and `pathLenConstraint` is present then the relying party shall check that the certification path being processed is consistent with the value of `pathLenConstraint`.

If this extension is not present, ~~or is flagged non-critical and is not recognized by a relying party,~~ then the public-key certificate is to be considered an end-entity public-key certificate and cannot be used to verify public-key certificate signatures.

NOTE – To constrain a public-key certificate subject to act ~~being~~ only as an end-entity, i.e., not a CA, the issuer may include this extension field containing only an empty `SEQUENCE` value.

## 13) Correction of the defects reported in defect report 418

*Update the second paragraph of clause 12 as shown:*

Public-key certificates, used in combination with the entity authentication service, can provide an authorization service directly, if privileges are associated with the subject ~~through the practices of the issuing CA~~. Public-key certificates may contain a `subjectDirectoryAttributes` extension that contains privileges associated with the subject of the public-key certificate (see 9.3.2.3). This mechanism is appropriate in situations where the CA ~~authority~~ issuing the public-key certificate ~~(CA) is~~ also has an associated ~~the~~ authority for delegating the privilege (i.e., AA) and the validity period of the privilege corresponds to the validity period of the public-key certificate. ~~End-entities cannot act as AAs.~~ If any of the extensions defined in clause 15 are included in a public-key certificate, those extensions apply equally to all privileges assigned in the `subjectDirectoryAttributes` extension of that public-key certificate.

## 14) Correction of the defects reported in defect report 419

*In clause 3.5, delete definition 3.5.15, add the following two new definitions in alphabetical order, and renumber subsequent clauses accordingly:*

**3.5.x public-key certificate validation**: The process of ensuring that a public-key certificate was valid at a given time, including possibly the construction and processing of a certification path, and ensuring that all public-key certificates in that path were valid (e.g., were not expired or revoked) at that given time.

**3.5.x attribute certificate validation**: The process of ensuring that an attribute certificate was valid at a given time, including possibly the construction and processing of a delegation path, and ensuring that all attribute certificates in that path were valid (e.g., were not expired or revoked) at that given time.

## 15) Correction of the defects reported in defect report 420

*In 12.2, replace "attribute certification path" with "delegation path" in the heading and in the first paragraph.*

*In 16, replace "privilege path" with "delegation path" in the header and the main text.*

## SERIES OF ITU-T RECOMMENDATIONS

| | |
|---|---|
| Series A | Organization of the work of ITU-T |
| Series D | General tariff principles |
| Series E | Overall network operation, telephone service, service operation and human factors |
| Series F | Non-telephone telecommunication services |
| Series G | Transmission systems and media, digital systems and networks |
| Series H | Audiovisual and multimedia systems |
| Series I | Integrated services digital network |
| Series J | Cable networks and transmission of television, sound programme and other multimedia signals |
| Series K | Protection against interference |
| Series L | Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant |
| Series M | Telecommunication management, including TMN and network maintenance |
| Series N | Maintenance: international sound programme and television transmission circuits |
| Series O | Specifications of measuring equipment |
| Series P | Terminals and subjective and objective assessment methods |
| Series Q | Switching and signalling |
| Series R | Telegraph transmission |
| Series S | Telegraph services terminal equipment |
| Series T | Terminals for telematic services |
| Series U | Telegraph switching |
| Series V | Data communication over the telephone network |
| **Series X** | **Data networks, open system communications and security** |
| Series Y | Global information infrastructure, Internet protocol aspects and next-generation networks, Internet of Things and smart cities |
| Series Z | Languages and general software aspects for telecommunication systems |

Printed in Switzerland
Geneva, 2016