INTERNATIONAL STANDARD ISO/IEC 9594-8:2014
TECHNICAL CORRIGENDUM 1

Published 2015-10-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Open Systems Interconnection — The Directory —

Part 8: Public-key and attribute certificate frameworks

TECHNICAL CORRIGENDUM 1

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

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

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 9594-8:2014 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems.*

ICS 35.100.70 **Ref. No. ISO/IEC 9594-8:2014/Cor.1:2015(E)**

Published in Switzerland

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

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

Technical Corrigendum 1

(Covering resolution to defect reports 389, 390,393, 394, 395, 397, 398, 399, 400, 401, 402, 403, 404 and 405)

1) Correction of the defects reported in defect report 389

Replace clause 3.5.61 with the following:
3.5.61 self-issued attribute certificate: An attribute certificate where the issuer and the holder are the same attribute authority. An attribute authority might use a self-issued attribute certificate, for example, to publish policy information.

2) Correction of the defects reported in defect report 390

Delete the last paragraph of clause 8.6.2.

3) Correction of the defects reported in defect report 393

Replace the last paragraph of clause 8.5.2.9 with:
The scope of a CRL containing this extension is extended to include the revocation status of revoked certificates that expired after the date specified in **`ExpiredCertsOnCRL`** or at that date. The revocation status of a certificate shall not be updated once the certificate has expired.

4) Correction of the defects reported in defect report 394

Add the following references to clause 2.4

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

Add a new definition to clause 3.5:
3.5.68 trust anchor store: A trust anchor information collection at a relying party for one or more trust anchors.
Replace clause 7.5 with:

7.5 Trust anchor

An entity is a trust anchor for a particular relying party for one or more purposes, typically including certificate validation. A trust anchor is identified by trust anchor information. Trust anchor information includes a public key and some associated data. This trust anchor information is configured into the relying party in a trust anchor store. A relying party may have configured information about multiple trust anchors into one or more trust anchor stores.
A trust anchor may be a CA that issues public-key certificates and certificate revocation lists (CRLs) (see clause 7.10). The relying party may then use the trust anchor information for public-key certificate and CRL validation.
A trust anchor may also function as an end entity by signing other types of information such as software packages, time stamps, responses to online certificate status protocol (OCSP) requests (see IETF RFC 6960), etc.
A CA may be a trust anchor for some entities with respect to particular public-key certificates, but may otherwise be an ordinary CA.

NOTE 1 – As an example, entities within a company may trust all the public-key certificates issued by the company CA. This CA is then the trust anchor for these local relying parties with respect to locally issued public-key certificates. However, by use of name constraints, it might not be a trust anchor with respect to public-key certificates issued outside the company. Likewise, relying parties outside the company may not consider the company CA as the trust anchor for any public-key certificates.

NOTE 2 – The term trust anchor is seen as synonymous with the term root-CA. In a strict hierarchy, the CA at the top of the hierarchy may be the root CA and it may also be a trust anchor. However, in more complex environments, it may not be possible

to identify a root CA. Even when it is possible to identify a root CA, a relying party may not necessarily consider it a trust anchor. An intermediate CA may instead take that role.

IETF RFC 5914 defines trust anchor information as a choice between three alternatives:

```
TrustAnchorChoice ::= CHOICE {
  certificate     Certificate,
  tbsCert     [1] EXPLICIT TBSCertificate,
  taInfo      [2] EXPLICIT TrustAnchorInfo }
```

The **certificate** alternative specifies a public-key certificate that can be either a self-signed certificate or a public-key certificate.

The **tbsCert** alternative specifies an unsigned public-key certificate as defined in clause 7.2.

NOTE 3 – This alternative is deprecated by this Specification and therefore not considered further.

The **taInfo** alternative specify a special trust anchor information format defined by IETF RFC 5914.

In case the trust anchor information is not used for signing public-key certificates, it shall be an end-entity public-key certificate.

5) Correction of the defects reported in defect report 395

Add the following to the references in clause 2.4:

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

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

Add the following abbreviations to clause 4:

FQDN Fully-Qualified Domain Name

IDN Internationalized Domain Name

LDH Letters, Digits, Hyphen

*Replace the text for the **dNSName** in clause 8.3.2.1 with:*

– the **dNSName** alternative shall be a fully-qualified domain name (FQDN). The domain name shall be in the syntax as specified by section 2.3.1 of IETF RFC 5890 meaning that a domain name is a sequence of labels in the letters, digits, hyphen (LDH) format separated by dots.

A label may be in one of two formats:

a) All characters in the label are from the Basic Latin collection as defined by ISO/IEC 10646 (i.e., having code points in the ranges 002D, 0030-0039, 0041-005A and 0061-007A) and it does not start with "xn--". The maximum length is 63 octets.

b) It is an A-label as defined in IETF RFC 5890, i.e., it starts with the "xn--" and is a U-label converted to valid ASCII characters as in item a) using the Punycode algorithm defined by IETF RFC 3492. The converted string shall be maximum 59 octets. To be valid, it shall be possible for an A-label to be converted to a valid U-label. The U-label is as also defined in IETF RFC 5890.

NOTE 1 – An A-label is normally not human-readable.

6) Correction of the defects reported in defect report 397

*In clause 7.10, replace the explanatory text for the **version** component with:*

The **version** field shall indicate the version of the encoded revocation list. If the **extensions** component is present in the revocation list, the version shall be **v2**. If the **extensions** component is not present, the version shall either be absent or present as **v2**.

NOTE 1 – In the first and the second editions of this specification, the version component was always absent. In the third, fourth, fifth and sixth editions of this specification, the version shall be v2, if the extensions component flagged as critical is present in the revocation list. Or the version may either be absent or present as v2, if no extensions component flagged as critical is present in the revocation list.

Delete current Note 4.

Renumber the remaining notes from clause 7.10.

7) Correction of the defects reported in defect report 398

Update the ASN.1 in clause 8.6.2.2 as shown:

```
IssuingDistPointSyntax ::= SEQUENCE {
  -- If onlyContainsUserPublicKeyCerts and onlyContainsCACerts are both FALSE,
  -- the CRL covers both public-key certificate types
  distributionPoint               [0]  DistributionPointName OPTIONAL,
  onlyContainsUserPublicKeyCerts  [1]  BOOLEAN DEFAULT FALSE,
  onlyContainsCACerts             [2]  BOOLEAN DEFAULT FALSE,
  onlySomeReasons                 [3]  ReasonFlags OPTIONAL,
  indirectCRL                     [4]  BOOLEAN DEFAULT FALSE,
  onlyContainsAttributeCerts      [5]  BOOLEAN OPTIONAL, -- Use is strongly deprecated
  ... }
```

After the first paragraph after the ASN.1, add a new paragraph:

If **onlyContainsAttributeCerts** is **TRUE**, the CRL only contains revocations for attribute certificates. This component is deprecated and should not be included. Instead, the **aAissuingDistributionPoint** extension should be used.

NOTE 1 – This component was introduced into the fourth edition of this Specification and removed again in the fifth edition. Each of these two actions has caused compatibility problems. This component has been reintroduced into the sixth edition in a way to remove any compatibility issues.

In the penultimate paragraph of clause 8.6.2.2, renumber current NOTE as NOTE 2.

8) Correction of the defects reported in defect report 399

C.1 Introduction

Replace the third paragraph of C.1:

This annex is written for revocation status checking of public-key certificates using CRLs, Full and Complete End-Entity CRLs (EPRLs) and CA Revocation Lists (CARLs). However, this description can also be applied to revocation status checking of attribute certificates using Attribute Certificate Revocation Lists (ACRL) and Attribute Authority Revocation Lists (AARL). For the purposes of this annex, ACRL can be considered in place of CRL, EPRL can be full and complete end-entity ACRL, and AARL in place of CARL. Similarly, the directory attributes identified in clause C.4 shall be mapped to those for the AARL and ACRL and the fields identifying certificate types in the Issuing Distribution Point extension can be mapped to those applicable to PMI.

with:

This annex is written for revocation status checking of public-key certificates using CRLs, full and complete end-entity certificate revocation lists (EPRLs) and certification authority revocation lists (CARLs). However, this description may also be applied to revocation status checking of attribute certificates. For the purposes of this annex, privilege verifier may be considered in place of relying party, attribute certificate revocation lists (ACRLs) may be considered in place of CRLs, full and complete end-entity attribute certifications lists (ACRLs) in place of EPRLs, and attribute authority revocation lists (AARLs) in place of CARLs. Similarly, the directory attributes types **certificateRevocationList** and **authorityRevocationList** identified in clause C.4 may be mapped into **attributeCertificateRevocationList** and **attributeAuthorityRevocationList** and the **issuingDistributionPoint** extension may be mapped into the **aAissuingDistributionPoint** extension.

C.1.1 CRL types

Update the following as shown:

CRLs of one or more of the following types may be available to a relying party, based on the revocation aspects of the policy of the certificate issuing authority:

- Full and complete CRL;
- Full and complete end-entity public-key certificate revocation list ~~CRL~~ (EPRL);
- Full and complete certification authority ~~CA R~~revocation ~~L~~list (CARL);
- Distribution Point CRL, EPRL or CARL;