BS ISO/IEC 9594-8:2020

Incorporating corrigenda December 2021 and October 2023

BSI Standards Publication

Information technology — Open systems interconnection

Part 8: The Directory: Public-key and attribute certificate frameworks

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9594-8:2020

National foreword

This British Standard is the UK implementation of ISO/IEC 9594-8:2020, incorporating corrigenda December 2021 and October 2023. It supersedes BS ISO/IEC 9594-8:2017, which is withdrawn.

The start and finish of text introduced or altered by corrigendum is indicated in the text by tags. Text altered by ISO/IEC corrigendum December 2021 is indicated in the text by ▌AC1⟩ ⟨AC1▐.

The UK participation in its preparation was entrusted to Technical Committee IST/6, Data communications.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2024
Published by BSI Standards Limited 2024

ISBN 978 0 539 27168 3

ICS 35.100.70

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2020.

Amendments/corrigenda issued since publication

Date	Text affected
31 January 2023	Implementation of ISO/IEC corrigendum December 2021
31 March 2024	Implementation of ISO/IEC corrigendum October 2023

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9594-8:2020

INTERNATIONAL STANDARD

ISO/IEC 9594-8

Ninth edition
2020-11

Information technology — Open systems interconnection —

Part 8: The Directory: Public-key and attribute certificate frameworks

Reference number
ISO/IEC 9594-8:2020(E)

Documents
Document Sharing Hub
© ISO/IEC 2020

Full document details: BS ISO/IEC 9594-8:2020

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9594-8:2020

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by ITU-T as ITU-T X.509 (10/2019) and drafted in accordance with its editorial rules, in collaboration with Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

This ninth edition cancels and replaces the eighth edition (ISO/IEC 9594-8:2017), which has been technically revised.

A list of all parts in the ISO/IEC 9594 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9594-8:2020

CONTENTS

Page

SECTION 1 – General 1

1 Scope 1

2 Normative references 1
- 2.1 Identical Recommendations | International Standards 1
- 2.2 Paired Recommendations | International Standards equivalent in technical content 2
- 2.3 Recommendations 2
- 2.4 Other references 2

3 Definitions 3
- 3.1 OSI Reference Model security architecture definitions 3
- 3.2 Baseline identity management terms and definitions 3
- 3.3 Directory model definitions 3
- 3.4 Access control framework definitions 3
- 3.5 Public-key and attribute certificate definitions 3

4 Abbreviations 7

5 Conventions 8

6 Frameworks overview 8
- 6.1 Digital signatures 9
- 6.2 Public-key cryptography and cryptographic algorithms 10
- 6.3 Distinguished encoding of basic encoding rules 12
- 6.4 Applying distinguished encoding 12
- 6.5 Using repositories 13

SECTION 2 – PUBLIC-KEY CERTIFICATE FRAMEWORK 14

7 Public keys and public-key certificates 14
- 7.1 Introduction 14
- 7.2 Public-key certificate 14
- 7.3 Public-key certificate extensions 17
- 7.4 Types of public-key certificates 18
- 7.5 Trust anchor 18
- 7.6 Entity relationship 19
- 7.7 Certification path 19
- 7.8 Generation of key pairs 21
- 7.9 Public-key certificate creation 21
- 7.10 Certificate revocation list 22
- 7.11 Uniqueness of names 25
- 7.12 Indirect CRLs 25
- 7.13 Repudiation of a digital signing 26

8 Trust models 27
- 8.1 Three-cornered trust model 27
- 8.2 Four cornered trust model 27

9 Public-key certificate and CRL extensions 28
- 9.1 Policy handling 29
- 9.2 Key and policy information extensions 31
- 9.3 Subject and issuer information extensions 38
- 9.4 Certification path constraint extensions 41
- 9.5 Basic CRL extensions 45
- 9.6 CRL distribution points and delta CRL extensions 52
- 9.7 Authorization and validation list extensions 57
- 9.8 Alternative cryptographic algorithms and digital signature extensions 58

10 Delta CRL relationship to base 59

11 Authorization and validation lists 60
- 11.1 Authorization and validation list concept 60

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Page
11.2 The authorizer ... 60
11.3 Authorization and validation list syntax ... 61
11.4 Multiple cryptographic algorithms for authorization and validation list ... 62
12 Certification path processing procedure ... 63
12.1 Path processing inputs ... 63
12.2 Path processing outputs ... 63
12.3 Path processing variables ... 64
12.4 Initialization step ... 64
12.5 Public-key certificate processing ... 65
13 PKI directory schema ... 67
13.1 PKI directory object classes and name forms ... 67
13.2 PKI directory attributes ... 68
13.3 PKI directory matching rules ... 72
13.4 PKI directory syntax definitions ... 77
SECTION 3 – ATTRIBUTE CERTIFICATE FRAMEWORK ... 80
14 Attribute certificates ... 80
14.1 General ... 80
14.2 Attribute certificate syntax ... 81
14.3 Multiple cryptographic algorithms for attribute certificates ... 83
14.4 Delegation paths ... 83
14.5 Attribute certificate revocation lists ... 84
15 Attribute authority, source of authority and certification authority relationship ... 86
15.1 Privilege in attribute certificates ... 87
15.2 Privilege in public-key certificates ... 87
16 PMI models ... 87
16.1 General model ... 87
16.2 Control model ... 89
16.3 Delegation model ... 90
16.4 Group assignment model ... 90
16.5 Roles model ... 91
16.6 Recognition of Authority Model ... 93
16.7 XML privilege information attribute ... 96
16.8 Permission attribute and matching rule ... 97
17 Attribute certificate and attribute certificate revocation list extensions ... 97
17.1 Basic privilege management extensions ... 98
17.2 Privilege revocation extensions ... 101
17.3 Source of authority extensions ... 107
17.4 Role extensions ... 109
17.5 Delegation extensions ... 110
17.6 Recognition of authority extensions ... 114
17.7 Use of alternative digital signature algorithm and digital signature extensions ... 117
18 Delegation path processing procedure ... 118
18.1 Basic processing procedure ... 118
18.2 Role processing procedure ... 119
18.3 Delegation processing procedure ... 119
19 PMI directory schema ... 121
19.1 PMI directory object classes ... 121
19.2 PMI directory attributes ... 123
19.3 PMI general directory matching rules ... 125
19.4 PMI directory syntax definitions ... 127
Annex A – Public-key and attribute certificate frameworks ... 129
Annex B – Reference definition of cryptographic algorithms ... 157
Annex C – Certificate extension attribute types ... 164

© ISO/IEC 2020 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Page

C.1 Certificate extension attribute concept 164
C.2 Formal specification for certificate extension attribute types 164

Annex D – External ASN.1 modules 173

Annex E – CRL generation and processing rules 182
E.1 Introduction 182
E.2 Determine parameters for CRLs 183
E.3 Determine CRLs required 184
E.4 Obtain CRLs 185
E.5 Process CRLs 185

Annex F – Examples of delta CRL issuance 189

Annex G – Privilege policy and privilege attribute definition examples 191
G.1 Introduction 191
G.2 Sample syntaxes 191
G.3 Privilege attribute example 195

Annex H – An introduction to public key cryptography[2)] 196

Annex I – Examples of use of certification path constraints 198
I.1 Example 1: Use of basic constraints 198
I.2 Example 2: Use of policy mapping and policy constraints 198
I.3 Use of name constraints extension 198

Annex J – Guidance on determining for which policies a certification path is valid 207
J.1 Certification path valid for a user-specified policy required 207
J.2 Certification path valid for any policy required 208
J.3 Certification path valid regardless of policy 208
J.4 Certification path valid for a user-specific policy desired, but not required 208

Annex K – Key usage certificate extension issues 209

Annex L – Deprecated extensions 210
L.1 CRL scope extension 210

Annex M – Directory concepts 213
M.1 Scope 213
M.2 The directory attribute concept 213
M.3 Basic directory concepts 213
M.4 Subtrees 214
M.5 Directory distinguished names 214
M.6 Directory schema 215

Annex N – Considerations on strong authentication 216
N.1 Introduction 216
N.2 One-way authentication 217
N.3 Two-way authentication 217
N.4 Three-way authentication 218
N.5 Five-way authentication (initiated by A) 219
N.6 Five-way authentication (initiated by B) 220

Annex O – Alphabetical list of information item definitions 222

Annex P – Amendments and corrigenda 225

Bibliography 226

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Introduction

Many applications have requirements for security to protect against threats to the communication of information. Virtually all security services are dependent upon the identities of the communicating parties being reliably known, i.e., authenticated.

This Recommendation | International Standard defines a framework for public-key certificates. This framework includes the specification of data objects used to represent the public-key certificates themselves, as well as revocation notices for issued public-key certificates that should no longer be trusted. It defines some critical components of a public-key infrastructure (PKI), but it does not define a PKI in its entirety. However, this Recommendation | International Standard provides the foundation upon which full PKIs and their specifications can be built.

Similarly, this Recommendation | International Standard defines a framework for attribute certificates. This framework includes the specification of data objects used to represent the attribute certificates themselves, as well as revocation notices for issued attribute certificates that should no longer be trusted. It defines some critical components of a privilege management infrastructure (PMI), but it does not define a PMI in its entirety. However, this Recommendation | International Standard provides the foundation upon which full PMIs and their specifications can be built.

Directory schema definitions allow PKI and PMI information to be represented 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) or according to the lightweight directory access protocol (LDAP) specification.

This Recommendation | International Standard provides the foundation frameworks upon which industry profiles can be defined by other standards groups and industry forums. Many of the features defined as optional in these frameworks may be mandated for use in certain environments through profiles. This ninth edition technically revises and enhances the eighth edition of this Recommendation | International Standard.

This ninth edition specifies versions 1, 2 and 3 of public-key certificates, versions 1 and 2 of certificate revocation lists and version 2 of attribute certificates.

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.

Annex A, which is an integral part of this Recommendation | International Standard, provides the ASN.1 modules which contain all of the definitions associated with the frameworks.

Annex B, which is not an integral part of this Recommendation | International Standard, lists object identifiers assigned to cryptographic algorithms defined by other specifications. It is provided for easy reference and import into other ASN.1 modules.

Annex C, which is an integral part of this Recommendation | International Standard, provides definitions for how certificate extension types may be represented by directory attribute types.

Annex D, which is not an integral part of this Recommendation | International Standard, includes extracts of external ASN.1 modules referenced by this Recommendation | International Standard.

Annex E, which is an integral part of this Recommendation | International Standard, provides rules for generating and processing certificate revocation lists (CRLs).

Annex F, which is not an integral part of this Recommendation | International Standard, provides examples of delta certificate revocation list (CRL) issuance.

Annex G, which is not an integral part of this Recommendation | International Standard, provides examples of privilege policy syntaxes and privilege attributes.

Annex H, which is not an integral part of this Recommendation | International Standard, is an introduction to public-key cryptography.

Annex I, which is not an integral part of this Recommendation | International Standard, contains examples of the use of certification path constraints.

Annex J, which is not an integral part of this Recommendation | International Standard, provides guidance for public-key infrastructure (PKI) enabled applications on the processing of certificate policy while in the certification path validation process.

Annex K, which is not an integral part of this Recommendation | International Standard, provides guidance on the use of the **`contentCommitment`** bit in the **`keyUsage`** certificate extension.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Annex L, which is not an integral part of this Recommendation | International Standard, includes public-key and attribute certificate extensions that have been deprecated.

Annex M, which is not an integral part of this Recommendation | International Standard, gives a short introduction to directory and distinguished name concepts.

Annex N, which is not an integral part of this Recommendation | International Standard, provides some general considerations on strong authentication.

Annex O, which is not an integral part of this Recommendation | International Standard, contains an alphabetical list of information item definitions in this Recommendation | International Standard.

Annex P, which is not an integral part of this Recommendation | International Standard, lists the amendments and defect reports that have been incorporated to form this edition of this Recommendation | International Standard.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9594-8:2020

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

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

SECTION 1 – GENERAL

1 Scope

This Recommendation | International Standard addresses some of the security requirements in the areas of authentication and other security services through the provision of a set of frameworks upon which full services can be based. Specifically, this Recommendation | International Standard defines frameworks for:

- public-key certificates; and
- attribute certificates.

The public-key certificate framework defined in this Recommendation | International Standard specifies the information objects and data types for a public-key infrastructure (PKI), including public-key certificates, certificate revocation lists (CRLs), trust broker and authorization and validation lists (AVLs). The attribute certificate framework specifies the information objects and data types for a privilege management infrastructure (PMI), including attribute certificates, and attribute certificate revocation lists (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 extensions, 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 information in a directory, are included in this Recommendation | International Standard.

This Recommendation | International Standard specifies the framework for strong authentication, involving credentials formed using cryptographic techniques. It is not intended to establish this as a general framework for authentication, but it can be of general use for applications which consider these techniques adequate.

Authentication (and other security services) can only be provided within the context of a defined security policy. It is a matter for users of an application to define their own security policy.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- Recommendation ITU-T X.411 (1999) | ISO/IEC 10021-4:2003, *Information technology – Message Handling Systems (MHS) – Message Transfer System: Abstract Service Definition and Procedures*.
- Recommendation ITU-T X.500 (2019) | ISO/IEC 9594-1:2020, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.
- Recommendation ITU-T X.501 (2019) | ISO/IEC 9594-2:2020, *Information technology – Open Systems Interconnection – The Directory: Models*.
- Recommendation ITU-T X.511 (2019) | ISO/IEC 9594-3:2020, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition*.
- Recommendation ITU-T X.518 (2019) | ISO/IEC 9594-4:2020, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation*.
- Recommendation ITU-T X.519 (2019) | ISO/IEC 9594-5:2020, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications*.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

- Recommendation ITU-T X.520 (2019) | ISO/IEC 9594-6:2020, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types*.
- Recommendation ITU-T X.521 (2019) | ISO/IEC 9594-7:2020, *Information technology – Open Systems Interconnection – The Directory: Selected object classes*.
- Recommendation ITU-T X.525 (2019) | ISO/IEC 9594-9:2020, Information technology – Open Systems Interconnection – The Directory: Replication.
- Recommendation ITU-T X.660 (2011) | ISO/IEC 9834-1:2012, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the International Object Identifier tree*.
- 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*.
- 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)*.
- Recommendation ITU-T X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework*.
- Recommendation ITU-T X.813 (1996) | ISO/IEC 10181-4:1997, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Non-repudiation framework*.
- Recommendation ITU-T X.841 (2000) | ISO/IEC 15816:2002, *Information technology – Security techniques – Security information objects for access control*.

2.2 Paired Recommendations | International Standards equivalent in technical content

- Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

  ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

2.3 Recommendations

- Recommendation ITU-T X.1252 (2010), *Baseline identity management terms and definitions*.

2.4 Other references

- IETF RFC 791 (1981), *Internet Protocol*.
- IETF RFC 822 (1982), *Standard for the Format of ARPA Internet Text Messages.*
- IETF RFC 1630 (1994), *Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web.*
- IETF RFC 3492 (2003), *Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA).*
- [AC2⟩ IETF RFC 3641 (2003), *Generic String Encoding Rules (GSER) for ASN.1 Types*. ⟨AC2]
- IETF RFC 4511 (2006), *Lightweight Directory Access Protocol (LDAP): The Protocol*.
- IETF RFC 4523 (2006), *Lightweight Directory Access Protocol (LDAP) Schema Definitions for X.509 Certificates*.
- IETF RFC 5280 (2008), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.
- IETF RFC 5890 (2010), *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*.
- IETF RFC 5914 (2010), *Trust Anchor Format*.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

– IETF RFC 6960 (2013), *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 OSI Reference Model security architecture definitions

The following terms are defined in Rec. ITU-T X.800 | ISO 7498-2:

a) authentication exchange;

b) authentication information;

c) confidentiality;

d) credentials;

e) cryptography;

f) data origin authentication;

g) decipherment;

h) digital signature;

i) encipherment; and

j) key;

3.2 Baseline identity management terms and definitions

The following term is defined in Rec. ITU-T X.1252:

a) trust.

3.3 Directory model definitions

The following terms are defined in Rec. ITU-T X.501 | ISO/IEC 9594-2:

a) attribute;

c) directory information tree (DIT);

b) directory system agent;

c) directory user agent (DUA);

d) distinguished name;

e) entry;

f) relative distinguished name;

g) root.

3.4 Access control framework definitions

The following terms are defined in Rec. ITU-T X.812 | ISO/IEC 10181-3:

a) access control decision function (ADF);

b) access control enforcement function (AEF).

3.5 Public-key and attribute certificate definitions

The following terms are defined in this Recommendation | International Standard:

3.5.1 ACRL distribution point: A directory entry or another distribution source for **attribute certificate revocation list**s (ACRLs); an ACRL distributed through an ACRL distribution point may contain revocation entries for only a subset of the full set of attribute certificates issued by one attribute authority (AA) or may contain revocation entries for multiple AAs.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

3.5.2 attribute authority certificate: An attribute certificate for one attribute authority (AA) issued by another AA or by the same AA.

3.5.3 attribute certificate: A data structure, digitally signed by an attribute authority that binds some attribute values with identification information about its holder.

3.5.4 attribute authority (AA): An authority which assigns privileges by issuing attribute certificates or by including them in public-key certificates.

3.5.5 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 AA.

3.5.6 attribute certificate revocation list (ACRL): A revocation list containing a list of references to attribute certificates that are no longer considered valid by the issuing attribute authority.

3.5.7 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.

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

3.5.9 authorization and validation list (AVL): A signed list containing information to an AVL entity about potential communications entities and possible restrictions on the communications with such entities.

3.5.10 authorization and validation list entity (AVL entity): An entity, when acting as a relying party, is dependent on an AVL issued by a designated authorizer.

3.5.11 authorizer: An entity trusted by one or more entities operating as authorization and validation list (AVL) entities to create, maintain and sign authorization and validation lists.

3.5.12 base attribute certificate revocation list (base ACRL): An attribute certificate revocation list (ACRL) that is used as the foundation in the generation of a delta attribute certificate revocation list (dACRL).

3.5.13 base certificate revocation list (base CRL): A certificate revocation list (CRL) that is used as the foundation in the generation of a delta certificate revocation list (dCRL).

3.5.14 CA certificate: A public-key certificate for one certification authority (CA) issued by another CA or by the same CA.

3.5.15 certificate policy: A named set of rules that indicates the applicability of a public-key certificate to a particular community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range.

3.5.16 certification practice statement (CPS): A statement of the practices that a certification authority (CA) employs in issuing certificates.

3.5.17 certificate revocation list (CRL): A signed list indicating a set of public-key certificates that are no longer considered valid by the issuing certification authority (CA). In addition to the generic term certificate revocation list (CRL), some specific CRL types are defined for CRLs that cover particular scopes.

3.5.18 certificate serial number: An integer value, unique within the issuing authority, which is unambiguously associated with a certificate issued by that authority.

3.5.19 certification authority (CA): An authority trusted by one or more entities to create and digital sign public-key certificates. Optionally the certification authority may create the subjects' keys.

3.5.20 certification authority revocation list (CARL): A revocation list containing a list of certification authority (CA) certificates issued to CAs that are no longer considered valid by the issuing CA.

3.5.21 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 end-entity public-key certificate to be validated. All intermediate public-key certificates, if any, are certification authority (CA) certificates in which the subject of the preceding public-key certificate is the issuer of the following public-key certificate.

3.5.22 CRL distribution point: A directory entry or other distribution source for certificate revocation lists (CRLs); a CRL distributed through a CRL distribution point may contain revocation entries for only a subset of the full set of public-key certificates issued by one certification authority (CA) or may contain revocation entries for multiple CAs.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

3.3.23 cross-certificate: A certification authority (CA) certificate where the issuer and the subject are different CAs. CAs issue cross-certificates to other CAs as a mechanism to authorize the subject CA's existence.

3.5.24 delegation: Conveyance of privilege from one entity that holds such privilege, to another entity.

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

3.5.26 delta attribute certificate revocation list (dACRL): A partial attribute certificate revocation list that only contains entries for attribute certificates that have had their revocation status changed since the issuance of the referenced base attribute certificate revocation list (base ACRL).

3.4.27 delta-CRL (dCRL): A partial revocation list that only contains entries for public-key certificates that have had their revocation status changed since the issuance of the referenced base CRL.

3.5.28 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, but uses its attributes only to gain access to a resource.

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

3.5.30 end-entity attribute certificate revocation list (EARL): A revocation list containing a list of attribute certificates issued to holders that are not also attribute authorities (AAs) that are no longer considered valid by the certificate issuer.

3.5.31 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.

3.5.32 end-entity public-key certificate revocation list (EPRL): A revocation list containing a list of public-key certificates issued to subjects that are not also certification authorities (CAs), that are no longer considered valid by the certificate issuer.

3.5.33 environmental variables: Those aspects of policy required for an authorization decision, that are not contained within static structures, but are available through some local means to a privilege verifier (e.g., time of day or current account balance).

3.5.34 full attribute certificate revocation list (full ACRL): A complete revocation list that contains entries for all attribute certificates that have been revoked for a given scope.

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

3.5.36 hash function: A (mathematical) function which maps data of arbitrary size into data of a fixed size called a digest.

3.5.37 holder: An entity to whom some privilege has been delegated either directly from the source of authority or indirectly through another attribute authority.

3.5.38 indirect attribute certificate revocation list (iACRL): A revocation list whose scope includes attribute certificates issued by one or more attribute authorities (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.

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

3.5.40 intermediate CA: A certification authority (CA) is acting as an intermediate CA within a certification path when it is the issuer of the next public-key certificate on that certification path.

3.5.41 key agreement: A method for negotiating a symmetric key value on-line without transferring the key, even in an encrypted form, e.g., the Diffie-Hellman technique (see ISO/IEC 11770-1 for more information on key agreement mechanisms).

3.5.42 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.

3.5.43 object method: An action that can be invoked on a resource (e.g., a file system may have read, write and execute object methods).

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

3.5.44 one-way function: A (mathematical) function which is easy to compute, but which for a general value *y* in the range, it is computationally difficult to find a value *x* in the domain such that f(*x*) = *y*. There may be a few values *y* for which finding *x* is not computationally difficult.

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

3.5.46 PMI end entity: An entity that is acting as an end entity in a privilege management infrastructure (PMI) environment, where the holder uses its privilege attributes to gain access to a resource.

3.5.47 policy decision point (PDP): The point where policy decisions are made (synonymous with access control decision function, ADF).

3.5.48 policy enforcement point (PEP): The point where the policy decisions are actually enforced (synonymous with access control enforcement function, AEF).

3.5.49 policy mapping: Recognizing that, when a certification authority (CA) in one domain certifies a CA in another domain, a particular certificate policy in the second domain may be considered by the authority of the first domain to be equivalent (but not necessarily identical in all respects) to a particular certificate policy in the first domain.

3.5.50 private key: (In a public-key cryptosystem) that key of an entity's key pair which is known only by that entity.

3.5.51 privilege: An attribute or property assigned to an entity by an attribute authority.

3.5.52 privilege asserter: A privilege holder using the attributes of the attribute certificate or public-key certificate to assert privilege.

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

3.5.54 privilege management infrastructure (PMI): The infrastructure able to support the management of privileges in support of a comprehensive authorization service and in relationship with a public-key infrastructure.

3.5.55 privilege policy: The policy that outlines conditions for privilege verifiers to provide/perform sensitive services to/for qualified privilege asserters. Privilege policy relates attributes associated with the service as well as attributes associated with privilege asserters.

3.5.56 privilege verifier: An entity verifying attributes of a public-key certificate or attribute certificate against a privilege policy.

3.5.57 public key: That key of an entity's key pair which is publicly known.

3.5.58 public-key certificate: The public key of an entity, together with some other information, rendered unforgeable by digital signature with the private key of the certification authority (CA) that issued it.

3.5.59 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.

3.5.60 public-key infrastructure (PKI): The infrastructure able to support the management of public keys able to support authentication, encryption, integrity or non-repudiation services.

3.5.61 registration authority (RA): Those aspects of the responsibilities of a certification authority that are related to identification and authentication of the subject of a public-key certificate to be issued by that certification authority. An RA may either be a separate entity or be an integrated part of the certification authority.

NOTE – This definition is different in scope from the one defined in Rec. ITU-T X.660 | ISO/IEC 9834-1.

3.5.62 relying party: An entity that relies on the data in a public-key certificate in making decisions.

3.5.63 role assignment certificate: A certificate that contains the role attribute, assigning one or more roles to the certificate subject/holder.

3.5.64 role specification certificate: An attribute certificate that contains the assignment of privileges to a role.

3.5.65 sensitivity: Characteristic of a resource that implies its value or importance.

3.5.66 security policy: The set of rules laid down by the security authority governing the use and provision of security services and facilities.

3.5.67 self-issued attribute certificate: An attribute certificate where the issuer and the holder are the same attribute authority (AA). An AA might use a self-issued attribute certificate, for example, to publish policy information.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

3.5.68 self-issued certificate: A certification authority (CA) certificate where the issuer and the subject are the same CA. A CA might use self-issued certificates, for example, during a key rollover operation to provide trust from the old key to the new key.

3.5.69 self-signed certificate: A special case of self-issued certificates where the private key used by the certification authority (CA) to sign the CA certificate corresponds to the public key that is certified within the CA certificate. A CA might use a self-signed certificate, for example, to advertise their public key or other information about their operations.

NOTE – Use of self-issued certificates and self-signed certificates issued by other than certification authorities (CAs) are outside the scope of this Recommendation | International Standard.

3.5.70 source of authority (SOA): An attribute authority that a privilege verifier for a particular resource trusts as the ultimate authority to assign a set of privileges for asserting that resource.

3.5.71 subject CA: A certification authority (CA) for which another CA has issued a CA certificate.

3.5.72 trust anchor: An entity that is trusted by a relying party and used for validating public-key certificates.

3.5.73 trust anchor information: Trust anchor information is at least the: distinguished name of the trust anchor, associated public key, algorithm identifier, public key parameters (if applicable), and any constraints on its use, including a validity period. The trust anchor information may be provided as a self-signed certification authority (CA) certificate or as a normal CA certificate (i.e., cross-certificate).

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

3.5.75 trust broker: A third party trusted by relying parties to provide quantitative information about the trustworthiness and qualitative information about the intended use of a public-key certificate based on information about the certification authority. Trust brokers are independent of certification authorities and have direct trust relationships with relying parties.

3.5.76 trust in a CA: Belief that the certification authority (CA) will act reliability and truthfully in the management of its public-key certificates and will comply with its published certification practise statement and relevant legislation.

3.5.77 trust in a public-key certificate: Belief that the public-key certificate belongs to the subject identified in the public-key certificate.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

AA	Attribute Authority
AARL	Attribute Authority Revocation List
ACRL	Attribute Certificate Revocation List
ADF	Access control Decision Function
AEF	Access control Enforcement Function
AIA	Authority Information Access
ASN.1	Abstract Syntax Notation One
AVL	Authorization and Validation List
BER	Basic Encoding Rules
CA	Certification Authority
CARL	Certification Authority Revocation List
CRL	Certificate Revocation List
dACRL	Delta Attribute Certificate Revocation List
DAP	Directory Access Protocol
dCRL	Delta Certificate Revocation List
DER	Distinguished Encoding Rules
DIT	Directory Information Tree
DS	Delegation Service
DSA	Digital Signature Algorithm
DUA	Directory User Agent

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

EARL	End-entity Attribute certificate Revocation List
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EPRL	End-entity Public-key Certificate Revocation List
FQDN	Fully Qualified Domain Name
HSM	Hardware Security Module
iACRL	Indirect Attribute Certificate Revocation List
iCRL	Indirect Certificate Revocation List
IDN	Internationalized Domain Name
IDP	Issuing Distribution Point
LDAP	Lightweight Directory Access Protocol
LDH	Letters, Digits, Hyphen
M2M	Machine-to-Machine
OCSP	Online Certificate Status Protocol
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIN	Personal Identification Number
PKCS	Public-Key Cryptography Standards
PKI	Public-Key Infrastructure
PMI	Privilege Management Infrastructure
RA	Registration Authority
RDN	Relative Distinguished Name
RoA	Recognition of Authority
RSA	Rivest-Shamir-Adelman
SOA	Source Of Authority
TB	Trust Broker

5 Conventions

The term "Specification" (as in "this Specification") shall be taken to mean Rec. ITU-T X.509 | ISO/IEC 9594-8.

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.

If an International Standard or ITU-T Recommendation is referenced within normal 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 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 | ISO/IEC 8824-4 and Rec. ITU-T X.690 | ISO/IEC 8825-1.

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. When a definition is referenced for the first time in normal text it is presented in italicized Times New Roman.

If the items in a list are numbered (as opposed to using "–" or letters), then the items shall be considered steps in a procedure.

6 Frameworks overview

This Specification defines a framework for obtaining and trusting a public key of an entity in order to verify the digital signature of that entity. If the key pair is of a type that allows encryption and decryption, the public key may also be used

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

for encrypting information to be decrypted by the entity owning the private key. The framework includes the issuance of a public-key certificate by a certification authority (CA) and the validation of that public-key certificate by the relying party, i.e., the entity relying on the content of the public-key certificate. The validation includes:

- establishing a trusted path of public-key certificates between a trusted entity called a trust anchor (see clause 7.5) and the public-key certificate subject, i.e., the entity for which the public-key certificate has been issued;
- verifying the digital signatures on each public-key certificate in the path; and
- validating all the public-key certificates along that path (i.e., that they obey restrictions, were not expired or not revoked at a given time); and
- optionally, asking a trust broker if the public-key certificate can be trusted for the intended purpose.

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

This Specification defines a framework for obtaining and trusting privilege attributes of an entity in order to determine whether they are authorized to access a particular resource. The framework includes the issuance of an attribute certificate by an attribute authority (AA) and the validation of that attribute certificate by a privilege verifier. The validation includes:

- 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.

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 providing privileges. Such aspects of privileges carried by public-key certificates are also covered by the attribute certificate framework within Section 3.

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

Although PKI and PMI are separate infrastructures and may be established independently from one another, they are related. This Specification recommends that holders and issuers of attribute certificates be identified within attribute certificates by pointers to their appropriate public-key certificates. Authentication of the attribute certificate issuers and holders, to ensure that entities claiming privilege and issuing privilege are who they claim to be, is done using the normal processes of the PKI to authenticate identities. This authentication process is not duplicated within the attribute certificate framework.

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.

This Specification makes extensive use of cryptographic algorithms, such as hashing algorithms, public-key algorithms and digital signature algorithms.

6.1 Digital signatures

This subclause describes digital signatures in general. Sections 2 and 3 of this 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 Rivest-Shamir-Adelman (RSA) digital signatures, digital signature algorithm (DSA) digital signatures and elliptic curve digital signature algorithm (ECDSA) digital signatures.

NOTE 1 – It is not within the scope of this Specification to describe these different techniques in detail, 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.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

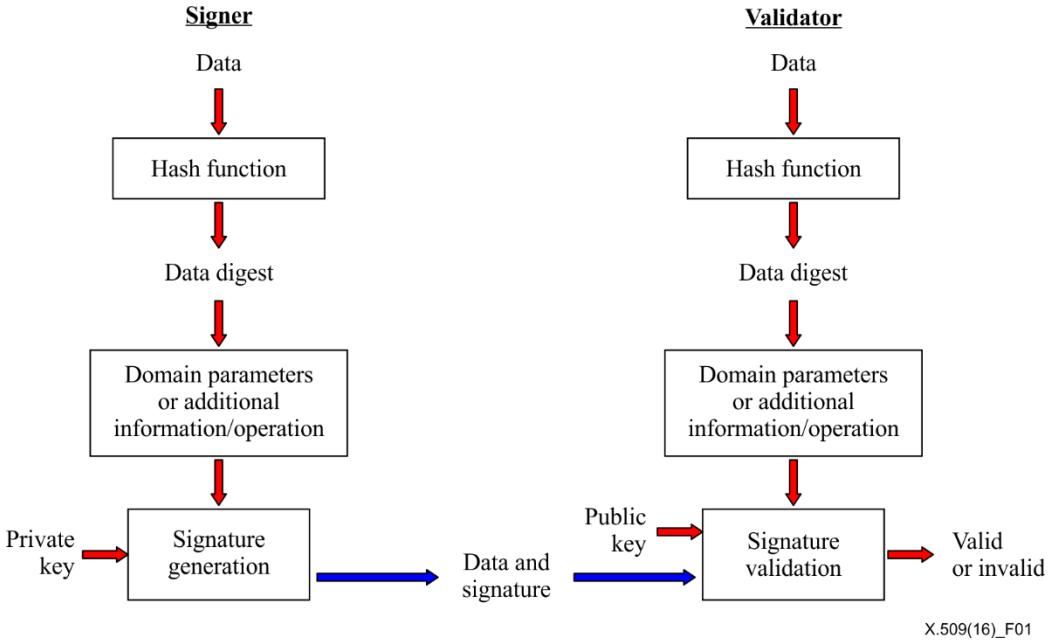

Figure 1 – Digital signature generation and validation

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 by additional information in preparation for generation 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 supplementation can be the addition of some information to the hash digest and in some cases, to perform yet another hashing operation. For the DSA and ECDSA digital 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 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) From the result from item 1) together with the public key of the signer, the bit string of the signature and the use of an associated 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.

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 digital signature algorithm.

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


© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

If one or two signatures are to be appended to the data, the following ASN.1 may be used to define the data type resulting from applying one or two signatures to the given data type.

NOTE – The components after the extension mark were added from Rec. ITU-T X.509 (2016) | ISO/IEC 9594-8:2017.

```
SIGNED{ToBeSigned} ::= SEQUENCE {
  toBeSigned    ToBeSigned,
  COMPONENTS OF SIGNATURE,
  ...,
[[4:
  altAlgorithmIdentifier  AlgorithmIdentifier{{SupportedAltAlgorithms}} OPTIONAL,
  altSignature            BIT STRING OPTIONAL]]
  } (WITH COMPONENTS {..., altAlgorithmIdentifier PRESENT, altSignature PRESENT } |
     WITH COMPONENTS {..., altAlgorithmIdentifier ABSENT,  altSignature ABSENT } ) }
```

The following data type is no longer 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 no longer used 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

AC1 The following ASN.1 information object class is used to specify cryptographic algorithms.

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

The **ALGORITHM** information object class has the following fields.

a) The **&Type** field is used to specify those fixed parameters that are necessary for specifying the exact procedure for deploying the cryptographic algorithm being defined. Not all cryptographic algorithms require such parameters. The field is then absent or has the value **NULL**, as determined by the individual cryptographic algorithm specifications.

b) The **&DynParms** field is used to specify those dynamic parameters that determine the value(s) to be exchanged between two communicating entities when invoking the cryptographic algorithm. Not all cryptographic algorithms require dynamic parameters. In this case the **&DynParms** field shall be absent.

c) The **&id** field is used to uniquely identify the class of cryptographic algorithm being defined.

The **AlgorithmWithInvoke** parameterized data type defined as follows is used in situations where the type of cryptographic algorithm is signalled together with its invocation.

```
AlgorithmWithInvoke{ALGORITHM:SupportedAlgorithms} ::= SEQUENCE {
  algorithm       ALGORITHM.&id({SupportedAlgorithms}),
  parameters  [0] ALGORITHM.&Type({SupportedAlgorithms}{@algorithm}) OPTIONAL,
  dynamParms  [1] ALGORITHM.&DynParms({SupportedAlgorithms}{@algorithm}) OPTIONAL,
  ... }
```

© ISO/IEC 2020 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

The **AlgorithmWithInvoke** parameterized data type has the following components.

a) The **algorithm** component shall hold the object identifier that uniquely identify the cryptographic algorithm being defined.

b) The **parameters** component, when present, shall hold the values of the fixed parameters that further identify the cryptographic algorithm in question. This component shall be present when the **&Type** field is present in the information object for the cryptographic algorithm in question. Otherwise, it shall be absent.

c) The **dynamParms** component, when present, shall hold the value(s) required by the dynamic parameters for the cryptographic algorithm. This component shall be present when the **&DynParms** field is present in the information object for the cryptographic algorithm. Otherwise, it shall be absent.

The **AlgorithmIdentifier** parameterized data type defined as follows is used in situations where the type of cryptographic algorithm is signalled without a corresponding invocation.

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

The components of **AlgorithmIdentifier** data type shall be as specified for the corresponding components of the **AlgorithmWithIvoke** parameterized data type.

The **AlgoInvoke** parameterized data type defined as follows is used when the cryptographic algorithm has previously been determined and where only invocation information is required.

```
AlgoInvoke{ALGORITHM:SupportedAlgorithms} ::=
    ALGORITHM.&DynParms({SupportedAlgorithms}) ⟨AC1⟩
```

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 sets are deferred to referencing
specifications having a requirement for specific information object sets.*/

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

The following parameterized data type specifies the syntax for fingerprint algorithms:

```
FingerPrint {ToBeFingerprinted} ::= SEQUENCE {
  algorithmIdentifier  AlgorithmIdentifier{{Sup
  fingerprint          BIT STRING,
  ... }
```

Typically, the algorithm used for generating a fingerprint is a hash algorithm, but the above definition is open to other types of algorithms.

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 elliptic curve cryptography (ECC) public key algorithm:

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

/* 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}
```

Annex B lists, for easy reference, some cryptographic algorithms specified in other specifications.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

6.2.3 Migration of cryptographic algorithms

Cryptographic algorithms that were once considered safe to use later become vulnerable to attack as computer technology and mathematics evolve. Both within a PKI and PMI, migration to more safe cryptographic algorithms raises several practical issues, as it is unlikely that it is possible to change cryptographic algorithms simultaneously for all entities within 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 allows some entities to move to a safer set of algorithms while at the same time legacy entities may keep using the old set of cryptographic algorithms during the migration period.

6.3 Distinguished encoding of basic encoding rules

In order to enable the validation of **SIGNED** and **SIGNATURE** types in a distributed environment, either the original signed message or, in its absence, a canonical encoding is required. The canonical encoding that shall be used by this specification is called the distinguished encoding of a **SIGNED** or **SIGNATURE** data value and shall be obtained by applying the basic encoding rules defined in Rec. ITU-T X.690 | ISO/IEC 8825-1, with the following restrictions:

a) the definite form of length encoding shall be used, encoded in the minimum number of octets;

b) for string types, the constructed form of encoding shall not be used;

c) if the value of a type is its default value, it shall be absent;

d) the components of a Set type shall be encoded in ascending order of their tag values;

e) the components of a Set-of type shall be encoded in ascending order of their octet values;

f) if the value of a Boolean type is **TRUE**, the encoding shall have its contents octet set to "FF";

g) each unused bit in the final octet of the encoding of a Bit String value, if there are any, shall be set to zero;

h) the encoding of a Real type shall be such that bases 8, 10 and 16 shall not be used, and the binary scaling factor shall be zero;

i) the encoding of a UTCTime shall be as specified in Rec. ITU-T X.690 | ISO/IEC 8825-1;

j) the encoding of a GeneralizedTime shall be as specified in Rec. ITU-T X.690 | ISO/IEC 8825-1.

NOTE – These restrictions are the same as the distinguished encoding rules specified in Rec. ITU-T X.690 | ISO/IEC 8825-1.

6.4 Applying distinguished encoding

Generating distinguished encoding requires the abstract syntax of the data to be encoded to be fully understood. An entity may be required to sign data or check the signature of data that is already signed or contains unknown protocol extensions or unknown attribute syntaxes. The entity shall follow these rules:

a) It shall preserve the encoding of received information whose abstract syntax it does not fully know and which it expects to subsequently sign.

b) When signing data for sending, it shall send data whose syntax it fully knows with a distinguished encoding and any other data with its preserved encoding, and shall sign the actual encoding it sends.

c) When checking signatures in received data, it shall check the signature against the actual data received rather than its conversion of the received data to a distinguished encoding.

6.5 Using repositories

PKI and PMI data objects of different kind need to be made available in in some kind of repository. This Specification does not mandate a specific repository technology, but describes repository requirements in term of directories as specified by the Directory Specifications or by the lightweight directory access protocol (LDAP) as specified by IETF RFC 4511.

This Specification also uses directory related terms, like distinguished name and subtree.

A short introduction to directory concepts is given in Annex M.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

SECTION 2 – PUBLIC-KEY CERTIFICATE FRAMEWORK

7 Public keys and public-key certificates

7.1 Introduction

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 issued by an authority called a certification authority (CA). 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, a CA revokes a previously issued public-key certificate, entities 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 entities of revocations. The format of revocation lists is defined here, including an extensibility mechanism and a set of revocation list extensions. In both the public-key certificate and the revocation list case, other bodies may also define additional extensions that are useful to their specific environments.

An entity that makes decisions based on the validity of a public-key certificates is called a relying party. A relying party needs to validate a public-key certificate prior to using it for a particular transaction in an application. Procedures for performing validation are also defined by this Specification, including verifying the integrity of the public-key certificate itself, its revocation status, and its validity with respect to the intended use.

In order for a relying party to be able to trust a public-key of another entity, for instance to authenticate the identity of that entity, the public key shall be obtained from a trusted source. Such a source, called a certification authority (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, relying parties 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 provided that secure cryptographic algorithms are being used in a secure way).

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

NOTE – Although the CAs are unambiguously defined by distinguished names having a hierarchical structure, this does not imply that there is any relationship between these distinguished names and the hierarchy of the CAs.

7.2 Public-key certificate

7.2.1 Public-key certificate syntax

A CA issues a public-key certificate of an entity by digital signing (see clause 6.1) a collection of information, including its distinguished name, the entity's distinguished name, a validity period, the value of a public-key algorithm and public key, as well as an optional additional information like for the permitted usage of the user's public key. The following ASN.1 data type specifies the syntax of public-key certificates:

```
Certificate ::= SIGNED{TBSCertificate}

TBSCertificate ::= SEQUENCE {
  version                  [0]  Version DEFAULT v1,
  serialNumber                  CertificateSerialNumber,
  signature                     AlgorithmIdentifier{{SupportedAlgorithms}},
  issuer                        Name,
  validity                      Validity,
  subject                       Name,
  subjectPublicKeyInfo          SubjectPublicKeyInfo,
  issuerUniqueIdentifier   [1] IMPLICIT UniqueIdentifier OPTIONAL,
  ...,
  [[2: -- if present, version shall be v2 or v3
  subjectUniqueIdentifier  [2] IMPLICIT UniqueIdentifier OPTIONAL]],
  [[3: -- if present, version shall be v2 or v3
  extensions               [3]  Extensions OPTIONAL]]
```


© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

```
  -- If present, version shall be v3]]
}

Version ::= INTEGER {v1(0), v2(1), v3(2)}

CertificateSerialNumber ::= INTEGER

Validity ::= SEQUENCE {
  notBefore  Time,
  notAfter   Time,
  ... }

SubjectPublicKeyInfo ::= SEQUENCE {
  algorithm         AlgorithmIdentifier{{SupportedAlgorithms}},
  subjectPublicKey  BIT STRING,
  ... }

Time ::= CHOICE {
  utcTime          UTCTime,
  generalizedTime  GeneralizedTime }

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE {
  extnId     EXTENSION.&id({ExtensionSet}),
  critical   BOOLEAN DEFAULT FALSE,
  extnValue  OCTET STRING
    (CONTAINING EXTENSION.&ExtnType({ExtensionSet}{@extnId})
       ENCODED BY der),
  ... }

der OBJECT IDENTIFIER ::=
  {joint-iso-itu-t asn1(1) ber-derived(2) distinguished-encoding(1)}

ExtensionSet EXTENSION ::= {...}
```

When generating a digital signature using the **SIGNED** parameterized data type, only one digital signature shall be generated, i.e., the **altAlgorithmIdentifier** and the **altSignature** components shall be absent.

Before a value of **Time** is used in any comparison operation, e.g., as part of a matching rule in a search, and if the syntax of **Time** has been chosen as the **UTCTime** type, the value of the two digit year component shall be rationalized into a four digit year value as follows:

– If the 2-digit value is 00 through to 49 inclusive, the value shall have 2000 added to it.

– If the 2-digit value is 50 through to 99 inclusive, the value shall have 1900 added to it.

NOTE 1 – The use of **GeneralizedTime** may prevent interworking with implementations unaware of the possibility of choosing either **UTCTime** or **GeneralizedTime**. It is the responsibility of those specifying the domains in which public-key certificates defined in this Specification will be used, e.g., profiling groups, as to when the **GeneralizedTime** may be used. In no case shall **UTCTime** be used for representing dates beyond 2049.

The **TBSCertificate** data type is the unsigned public-key certificate and is referred to as a to-be-signed public-key certificate. It shall be encoded using the distinguished encoding rules (DER).

NOTE 2 – Some specifications may specify that a public-key certificate may be transmitted in a non-DER encoding, i.e., in basic encoding rules (BER) encoding without the DER restrictions, but the signature then has to be generated over a DER encoded value of the **TBSCertificate** data type. An otherwise valid signature will then fail the signature validation if the relying party does not decode the public-key certificate and then DER encode it before validating the signature. It is a local policy decision whether in that case to fail the validation or to re-encode the public-key certificate.

The **version** component shall hold the version of the encoded public-key certificate. If the **extensions** component is present in the public-key certificate, **version** shall be **v3**. If the **issuerUniqueIdentifier** or **subjectUniqueIdentifier** component is present **version** shall be **v2** or **v3**.

The **serialNumber** component shall hold an integer assigned by the CA to the public-key certificate. The value of **serialNumber** shall be unique for each public-key certificate issued by a given CA (i.e., the issuer name and serial number together identify a unique public-key certificate).

NOTE 3 – Serial numbers do not have to be in numeric order.

The **signature** component shall contain the identifier for the digital signature algorithm used by the CA in signing the public-key certificate (e.g., **sha256WithRSAEncryption**, **sha384WithRSAEncryption**, dsa-**with-sha256**, **ecdsa-**

© ISO/IEC 2020 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

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 4 – By including this component, the digital signature algorithm is protected by the signature.

The **issuer** component shall hold the distinguished name of the CA that issued the public-key certificate. It shall hold a non-empty distinguished name.

The **validity** component shall hold the time interval during which the CA warrants that it will maintain information about the status of this public-key certificate.

The **subject** component shall identify the entity associated with the public-key found in the **subjectPublicKey** component of the **subjectPublicKeyInfo** component. If the public-key certificate is an end-entity public-key certificate (see clause 7.4), then the distinguished name may be an empty sequence providing that the **subjectAltName** extension is present and is flagged as critical (see clause 9.3.2.1). Otherwise, it shall be a non-empty distinguished name.

The **subjectPublicKeyInfo** component consists of two subcomponents:

- the **algorithm** subcomponent shall hold the algorithm which this public key is an instance of (e.g., **rsaEncryption**, **dhpublicnumber**, **id-dsa**, **id-ecPublicKey**, etc.); and
- the **subjectPublicKey** subcomponent shall hold the public key being certified.

The **issuerUniqueIdentifier** component is used uniquely to identify an issuer in case of name reuse (deprecated).

The **subjectUniqueIdentifier** component is used uniquely to identify a subject in case of name reuse (deprecated).

NOTE 5 – The use of **issuerUniqueIdentifier** and the **subjectUniqueIdentifier** is deprecated. These components were added because at one time there was some fear of the reuse of distinguished names. IETF RFC 5280 forbids the use of these components.

The **extensions** component, when present, shall hold one or more extensions as defined in clause 7.3.

An entity may obtain one or more public-key certificates from one or more CAs.

7.2.2 Multiple cryptographic algorithms for public-key certificates

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

- a 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 certificate 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 verify 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

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

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 as 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 extensions and with the new set of cryptographic algorithms and the digital signature in the base part of the public-key certificate.

7.3 Public-key certificate extensions

The **extensions** component of a public-key certificate allows for the addition of extensions to the structure without modification to the basic ASN.1 data type. An extension 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 OF** 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 shall 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 flagged as 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.

Any extension that is flagged as non-critical will cause inconsistent behaviour among relying parties that will process the extension and relying parties that do not recognize the extension and will ignore it. The same may be true for extensions that are flagged as critical, between relying parties that can fully process the extension and those that can partially process the extension, depending upon the extension.

A CA issuing a public-key certificate has three options with respect to an extension:

a) it can exclude the extension from the public-key certificate;

b) it can include the extension and flag it as non-critical;

c) it can include the extension and flag it as critical.

A relying party has three possible actions to take with respect to an extension:

a) if the extension is unrecognized and is flagged as non-critical, the relying party shall ignore the extension and accept the public-key certificate (all other things being equal);

b) if the extension is unrecognized and flagged as critical, the relying party shall consider the public-key certificate as invalid;

c) if the extension is recognized, the relying party shall process the extension and accept or consider the public-key certificate invalid, depending on the content of the extension and the conditions under which processing is occurring (e.g., the current values of the path processing variables).

Some extensions shall always be flagged as 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 consider the public-key certificate as invalid.

Some extensions shall always be flagged as 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 flagged 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 flagged as non-critical (unless factors other than this extension cause it to be rejected) and rejects the public-key certificate if the extension is flagged as 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 as critical. This is done with the realization that any relying party

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

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 flagged as 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. Some extensions for public-key certificates are defined in clause 9 of this Specification. Some extensions related to privilege information as defined in clause 17 may also be included in public-key certificates.

The following information object class is used to define specific extensions.

```
EXTENSION ::= CLASS {
  &id           OBJECT IDENTIFIER UNIQUE,
  &ExtnType }
WITH SYNTAX {
  SYNTAX        &ExtnType
  IDENTIFIED BY &id }
```

7.4 Types of public-key certificates

There are two primary types of public-key certificates, end-entity public-key certificates and CA certificates.

An end-entity public-key certificate is a public-key certificate issued by a CA to an entity acting as a PKI end entity that cannot use the corresponding private key to sign other public-key certificates.

A CA certificate is a public-key certificate issued by a CA to an entity that is also acting as a CA and therefore is capable of issuing and signing public-key certificates. A CA certificate shall include the **basicConstraints** extension with the **cA** components set to **TRUE** (see clause 9.4.2.1).

CA certificates can themselves be categorized by the following types:

- Self-issued certificate – This is a CA certificate where the issuer and the subject are the same CA. A CA might use self-issued certificates, for example, during a key rollover operation to provide trust from the old key to the new key.
- Self-signed certificate – This is a special case of self-issued certificates where the private key used by the CA to sign the certificate corresponds to the public key that is certified within the CA certificate. A CA might use a self-signed certificate, for example, to advertise their public key or other information about their operations.
- Cross-certificate – This is a CA certificate where the issuer and the subject are different CAs. CAs issue certificates to other CAs either as a mechanism to authorize the subject CA's existence (e.g., in a strict hierarchy) or to recognize the existence of the subject CA (e.g., in a distributed trust model). The cross-certificate structure is used for both of these.

7.5 Trust anchor

An entity is a trust anchor for a particular relying party for one or more purposes, typically including public-key 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 act as a CA that issues public-key certificates and certification 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 act as a PKI 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), AVLs, 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 local issued public-key certificates. However, 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.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

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. The term root CA is confusing and therefore not used by this Specification.

IETF RFC 5914 defines trust anchor information as a choice among 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 specifies a special trust anchor information format defined by IETF RFC 5914.

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

7.6 Entity relationship

There may be several CAs between the trust anchor recognized by the relying party and an entity acting as an end entity. Each CA has issued one or more cross-certificates for the next CA on the path between the trust anchor and the end entity. The CA that issues a cross-certificate to another CA takes the role of intermediate CA. The CA that is the subject for a cross-certificate takes the role of subject CA. This is illustrated in Figure 2. The same CA may take both the roles of an intermediate CA and a subject CA.

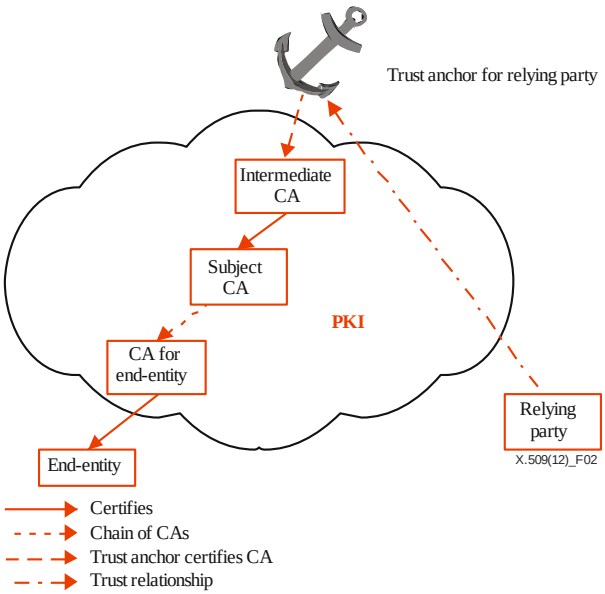

Figure 2 – Entity relationships

In some situations, conflicting or overlapping requirements for constraints, such as name constraints, may require a CA to issue more than one cross-certificate to another CA. In this case, multiple, different paths of public-certificate certificates are established between the trust anchor and the entity.

7.7 Certification path

Before a public-key certificate can be securely used by a relying party, it shall be validated. In order to validate such a public-key certificate, a chain of public-key certificates, called a certification path, shall be established between the public-key certificate signed by a trust anchor recognized by the relying party and the public-key certificate to be validated. Every public-key certificate within that path shall be checked. A certification path is thus an ordered list of public-key certificates starting with a public-key certificate signed by the trust anchor, and ending with the end-entity 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.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

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.

Each public-key certificate in a certification path shall be unique. A path that contains the same public-key certificate multiple times is not a valid certification path.

The **issuer** and **subject** components 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** component in one public-key certificate shall match the value of the **issuer** component in the subsequent public-key certificate. In addition, the value of the **issuer** component in the public-key certificate issued by the trust anchor shall match the distinguished name of the trust anchor. Only the names in these components are used when checking the validity of a certification path. Names in public-key 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 **subject** component of one public-key certificate with the distinguished name in the **issuer** component of the subsequent public-key certificate.

Figure 3 illustrates the situation where a relying party needs to check the validity of an end-entity public-key certificate and the relying party is able to construct a certification path between one of its trust anchors and the end-entity public-key certificate.

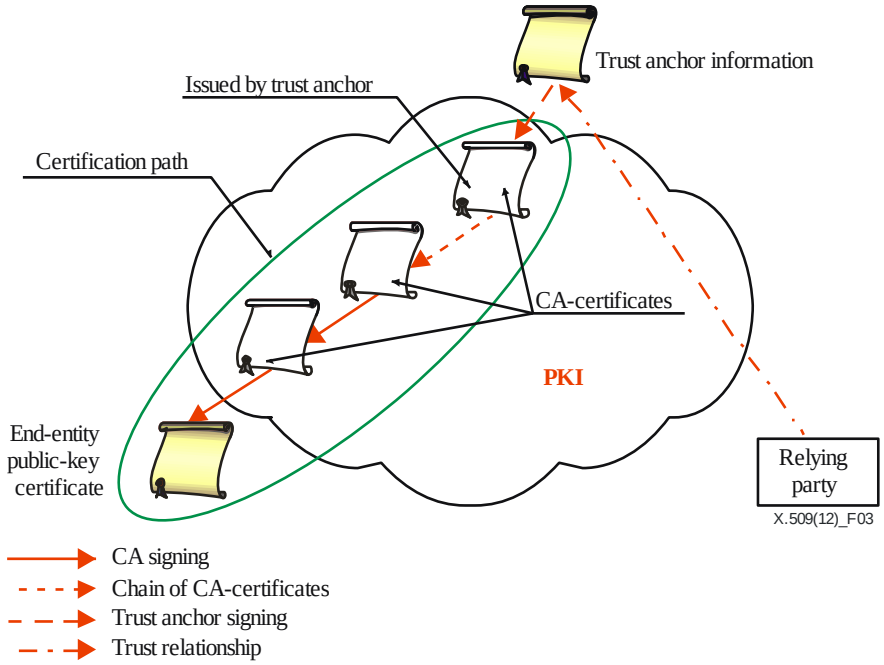

Figure 3 – Certification path

Trust suffers dilution as certification paths grow in length. The **basicConstraints** extension (see clause 9.4.2.1) allows restrictions to be put onto the length of the path. The validation of a public-key certificate may be affected by extensions in the chain of a public-key certificate, such as the **certificatePolicies** extension (see clause 9.2.2.6) and **nameConstraints** (see clause 9.4.2.2). It is the responsibility of the relying party to check that the restrictions are observed.

The ASN.1 data type **PkiPath** is used to represent a certification path. Within the sequence, the order of public-key certificates is such that the subject of the first public-key certificate is the issuer of the second public-key certificate, etc.

```
PkiPath ::= SEQUENCE OF Certificate
```

Each public-key certificate in a PKI path shall be unique.

The following data types are deprecated and should not be referenced by new specifications. They are retained here for backward compatibility not to invalidate existing specifications referencing these types.

NOTE – The **CertificationPath** data type was defined at an early stage, 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 as it is defined now. 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.

© ISO/IEC 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

```
Certificates ::= SEQUENCE {
  userCertificate    Certificate,
  certificationPath  ForwardCertificationPath OPTIONAL,
  ... }
```

The following ASN.1 data type can be used to represent the forward certification path. This component contains the certification path that can point back to the originator.

```
ForwardCertificationPath ::= SEQUENCE OF CrossCertificates

CrossCertificates ::= SET OF Certificate

CertificationPath ::= SEQUENCE {
  userCertificate    Certificate,
  theCACertificates  SEQUENCE SIZE (1..MAX) OF CertificatePair OPTIONAL,
  ... }
```

The **userCertificate** component shall hold the end-entity public-key certificate.

The **theCACertificates** component may hold an element for each CA from the end entity up to and including the CA which has been certified by the trust anchor. If the end-entity public-key certificate has been issued directly by the trust anchor, this component shall be absent.

7.8 Generation of key pairs

The overall security management policy of an implementation shall define the lifecycle of key pairs, and is, thus, outside the scope of this Specification. However, it is vital to the overall security that all private keys remain known only to the entity (subject) to whom they belong.

Key data is not easy for a human user to remember, so a suitable method for storing it in a convenient transportable manner shall be employed. One possible mechanism when an entity is associated with a human user would be to use a "Smart Card". This would hold the private and (optionally) public keys of the user, the user's public-key certificate, and a copy of the CA's public key. The use of this card shall additionally be secured by, e.g., at least the use of a personal identification number (PIN), increasing the security of the system by requiring the user to possess the card and to know how to access it. In other environments, e.g., in a machine-to-machine environment (M2M), a hardware security module (HSM) may be used for storing critical data. The exact method chosen for storing such data, however, is beyond the scope of this Specification.

Three ways in which a key pair of an entity may be produced are:

a) The entity generates its own key pair. This method has the advantage that the private key of an entity is never released to another entity, but requires a certain level of competence by the entity.

b) The key pair is generated by a third party. The third party shall release the private key to the entity in a physically secure manner, and then actively destroy all information relating to the creation of the key pair plus the keys themselves. Suitable physical security measures shall be employed to ensure that the third party and the data operations are free from tampering.

c) The key pair is generated by the CA. This is a special case of b), and the considerations there apply.

NOTE – The CA already exhibits trusted functionality with respect to the entity, and shall be protected by the necessary physical security measures. This method has the advantage of not requiring secure data transfer to the CA for certification.

The cryptosystem in use imposes particular (technical) constraints on key generation.

7.9 Public-key certificate creation

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

a) a CA shall be satisfied of the identity of a subject before creating a public-key certificate for it;

b) a CA shall not issue public-key certificates for two different subjects with the same subject name.

It is important that the transfer of information to the CA is not compromised, and suitable physical security measures shall be taken. In this regard:

a) It would be a serious breach of security if the CA issued a public-key certificate for a subject with a public key that had been tampered with.

b) If the means of generation of key pairs of clause 7.8, item b) or item c) is employed, the subject's private key shall be transferred to the entity in a secure manner.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

c) If the means of generation of key pairs of clause 7.8, item a) or item b) is employed, the subject may use different methods (online or offline) to communicate its public key to the CA in a secure manner. Online methods may provide some additional flexibility for remote operations performed between the entity and the CA.

A public-key certificate is a publicly available piece of information, and no specific security measures need to be employed with respect to its transmission e.g., to a directory server (directory system agent) according to the Directory Specifications or to the LDAP specification. As it is produced by an offline CA on behalf of a subject who shall be given a copy of it, the subject needs only store this information in its directory entry on a subsequent directory access. Alternatively, the CA could lodge the public-key certificate for the subject, in which case the CA shall be given suitable access rights to the entity's directory entry.

7.10 Certificate revocation list

7.10.1 Certificate revocation list principles

The CA that issues public-key certificates also has the responsibility to indicate the validity of the public-key certificates that it issues. Generally, public-key certificates are subject to possible subsequent revocation. This revocation and a notification of the revocation may be done directly by the same CA that issued the public-key certificate, or indirectly by another authority duly authorized by the CA that issued the public-key certificate. A CA that issues public-key certificates is required to state, possibly through a published statement of their practices, through the public-key certificates themselves, or through some other identified means, whether:

– the public-key certificates cannot be revoked;

– the public-key certificates may be revoked by the same CA directly; or

– the issuing CA authorizes a different entity to perform revocation.

CAs that do revoke public-key certificates are required to state, through similar means, what mechanism(s) can be used by relying parties to obtain revocation status information about public-key certificates issued by that CA. This Specification defines a certificate revocation list (CRL) mechanism and validation and authorization list (AVL) mechanism, but does not preclude the use of alternative mechanisms. One such alternative mechanism is the online certificate status protocol (OCSP) specified in IETF RFC 6960. Using this protocol, a relying party (client) requests the revocation status of a public-key certificate from an OCSP server. The server may use CRLs, or other mechanisms to check the status of the public-key certificate and respond to the client accordingly. If OCSP can be used by relying parties to check the status of a public-key certificate, IETF RFC 5280 contains a certificate extension (Authority Info Access) that would be included in such public-key certificates and would provide sufficient information to access an appropriate OCSP server. Relying parties check revocation status information, as appropriate, for all public-key certificates considered during the path processing procedure described in clause 12 to validate a public-key certificate.

Only a CA that is authorized to issue CRLs may choose to delegate that authority to another entity. If this delegation is done, it shall be verifiable at the time of public-key certificate/CRL verification. The **`cRLDistributionPoints`** extension (see clause 9.6.2.1) can be used for this purpose. The **`cRLIssuer`** component of this extension would be populated with the name(s) of any entities, other than the CA itself, that have been authorized to issue CRLs concerning the revocation status of the public-key certificate in question.

Public-key certificates shall have a lifetime associated with them, at the end of which they expire. In order to provide continuity of service, the authority shall ensure timely availability of replacement public-key certificates to supersede expired/expiring public-key certificates. Revocation notice date is the date/time that a revocation notice for a public-key certificate first appears on a CRL, regardless of whether it is a base or dCRL. In the CRL, revocation notice date is the value contained in the **`thisUpdate`** component. Revocation date is the date/time the CA actually revoked the public-key certificate, which could be different from the first time it appears on a CRL. In the CRL, revocation date is the value contained in the **`revocationDate`** component. Invalidity date is the date/time at which it is known or suspected that the private key was compromised or that the public-key certificate should otherwise be considered invalid. This date may be earlier that the revocation date. In the CRL, invalidity date is the value contained in the **`invalidityDate`** entry extension.

Two related points are:

– Validity of public-key certificates may be designed so that each becomes valid at the time of expiry of its predecessor, or an overlap may be allowed. The latter prevents the CA from having to install and distribute a large number of public-key certificates that may run out at the same expiration date.

– Expired public-key certificates will normally be removed from a directory. It is a matter for the security policy and responsibility of the CA to keep old public-key certificates for a period if a non-repudiation of data service is provided.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Public-key certificates may be revoked prior to their expiration time, e.g., if the entity's private key is assumed to be compromised, the entity is no longer to be certified by the CA, or if the CA's certificate is assumed to be compromised. The revocation of an end-entity public-key certificate or a CA certificate shall be made known by the issuing CA, and a new public-key certificate shall be made available, if appropriate. The CA may then inform the holder of the public-key certificate about its revocation by an offline procedure.

A CA that issues and subsequently revokes public-key certificates:

a) may be required to maintain an audit record of its revocation events for all public-key certificate types issued by that CA:

b) shall provide revocation status information to relying parties using CRLs, online certificate status protocol (OCSP), AVLs or another mechanism for the publication of revocation status information;

c) if using CRLs, it shall maintain and publish CRLs even if the lists of revoked public-key certificates are empty;

d) if using only partitioned CRLs, it shall issue a full set of partitioned CRLs covering the complete set of public-key certificates whose revocation status will be reported using the CRL mechanism. Thus, the complete set of partitioned CRLs shall be equivalent to a full CRL for the same set of public-key certificates, if the CRL issuer was not using partitioned CRLs.

Relying parties may use a number of mechanisms to locate revocation status information provided by an authority. For example, there may be a pointer in the public-key certificate itself that directs the relying party to a location where revocation information is provided. There may be a pointer in a revocation list that redirects the relying party to a different location. The relying party may locate revocation information in a repository (e.g., a directory) or through other means outside the scope of this Specification (e.g., locally configured).

If revocation lists are published in a directory, they are held within directory entries as attributes of the following types:

– **`certificateRevocationList`**;

– **`authorityRevocationList`**; and

– **`deltaRevocationList`**.

7.10.2 Certificate revocation list syntax

The following ASN.1 data type specifies the syntax of a CRL.

```
CertificateList ::= SIGNED{CertificateListContent}

CertificateListContent ::= SEQUENCE {
  version              Version OPTIONAL,
  -- if present, version shall be v2
  signature            AlgorithmIdentifier{{SupportedAlgorithms}},
  issuer               Name,
  thisUpdate           Time,
  nextUpdate           Time OPTIONAL,
  revokedCertificates  SEQUENCE OF SEQUENCE {
    serialNumber         CertificateSerialNumber,
    revocationDate       Time,
    crlEntryExtensions   Extensions OPTIONAL,
    ...} OPTIONAL,
  ...,
  ...,
  crlExtensions   [0]  Extensions OPTIONAL }
```

When generating a digital signature using the **`SIGNED`** parameterized data type, only one digital signature shall be generated, i.e., the **`altAlgorithmIdentifier`** and the **`altSignature`** components shall be absent.

The **`version`** component 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 be present as **`v2`**.

The **`signature`** component shall contain the identifier for the algorithm used by the CA to sign the revocation list. It shall be the same value as used in the **`algorithmIdentifier`** component of the **`SIGNATURE`** data type when signing the revocation list.

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

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

The **issuer** component shall identify the entity that signed and issued the revocation list.

The **thisUpdate** component shall indicate the date/time at which this revocation list was issued.

The **nextUpdate** component, when present, shall indicate the date/time by which the next revocation list in this series will be issued. The next revocation list could be issued before the indicated date, but it shall not be issued any later than the indicated time.

The **revokedCertificates** component shall identify public-key certificates that have been revoked. The revoked public-key certificates are identified by their serial numbers. If none of the public-key certificates covered by this CRL has been revoked, it is strongly recommended that the **revokedCertificates** parameter be omitted from the CRL, rather than being included with an empty **SEQUENCE**.

The **crlExtensions** component, when present, shall contain one or more CRL extensions.

NOTE 2 – The checking of the entire list of public-key certificates is a local matter. The list shall not be assumed to be in any particular order unless specific ordering rules have been specified by the issuing CA, e.g., in that CA's policy.

NOTE 3 – If a non-repudiation of data service is dependent on keys provided by the CA, the service should ensure that all relevant keys of the CA (revoked or expired) and the time stamped revocation lists are archived and certified by a current authority.

When an implementation processing a CRL encounters the serial number of the public-key certificate of interest in a CRL entry, but does not recognize a critical extension in the **crlEntryExtensions** component from that CRL entry, that CRL cannot be used to determine the status of the public-key certificate. When an implementation does not recognize a critical extension in the **crlExtensions** component, that CRL cannot be used to determine the status of the public-key certificate, regardless of whether the serial number of the public-key certificate of interest appears in that CRL or not.

NOTE 4 – In these cases, local policy may dictate actions in addition to and/or stronger than those stated in this Specification, such as seeking revocation status information from other sources.

Public-key certificates for which revocation status cannot be determined should not be considered valid public-key certificates.

If an extension affects the treatment of the list (e.g., multiple CRLs need to be scanned to examine the entire list of revoked public-key certificates, or an entry may represent a range of public-key certificates), then either that extension or a related extension shall be indicated as critical in the **crlExtensions** component. Therefore, a critical extension in the **crlEntryExtensions** component of an entry shall affect only the public-key certificate specified in that entry, unless there is a related critical extension in the **crlExtensions** component that advertises a special treatment for it. The only example of this situation defined in this Directory Specification is the **certificateIssuer** CRL entry extension and the related **issuingDistributionPoint** CRL extension when the **indirectCRL** Boolean from that extension is set to **TRUE**.

NOTE 5 – Extensions for CRLs are defined in clause 9 of this Specification.

If unknown components appear within an extension, and the extension is not flagged as critical, those unknown components shall be ignored according to the rules of extensibility documented in clause 12.2.2 of Rec. ITU-T X.519 | ISO/IEC 9594-5.

7.10.3 Multiple cryptographic algorithms for certificate revocation lists

It is possible for a CRL to hold:

- an alternative digital 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.

In the case of migration of the digital signature algorithm and the corresponding signature value, the CRL issuer shall include in a CRL the **altSignatureAlgorithm** extension holding an alternative digital signature algorithm and shall include the **altSignatureValue** extension holding an alternative signature value'. The following applies for the CRL issuer:

- when generating the value in the **altSignatureValue** extension, the CRL issuer shall exclude the **signature** component and the **altSignatureValue** extension from the CRL and shall 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 CRL issuer shall include the **altSignatureAlgorithm** and the **altSignatureValue** extensions in the DER encoding of the CRL.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

A relying party that has not migrated to support the alternative digital signature algorithm 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 the digital signature algorithm and alternative digital signatures shall verify 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 even if valid.

7.11 Uniqueness of names

A PKI requires that CAs are 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 to specify procedures that ensure unique and unambiguous names for CA and 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 public-key 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 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 x CAs, is authoritative for $x + 1$ or x 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 the **distributionPoint** component of the **cRLDistributionPoint** extension (see clause 9.6.2.1). 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.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Since the iCRL is authoritative for the CA(s) other that the CRL issuer, serial number alone in the CRL entry does not uniquely identify 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 (see clause 9.6.2.3). This extension shall always be flagged as critical to ensure that the relying parties process it and associate the CRL entry with the appropriate CA.

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 CAs. 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 be 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 a single CRL issuing authority that issues revocation lists for multiple CAs. The **issuingDistributionPoint** extension (see clause 9.6.2.2) is present in that CRL and is flagged as critical. The **indirectCRL** component of this extension is set to **TRUE**. If the CRL issuing authority name is the same as that of the CAs it serves, entries should then be placed first on the CRL without the **certificateIssuer** CRL entry extension. Entries for other CAs are kept together and the first CRL entry for a particular CA includes the **certificateIssuer** CRL entry 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 a certification path for a regular CRL with one difference. In the case of a 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 public-key certificate whose revocation status is being checked.

7.13 Repudiation of a digital signing

Participants in an event may subsequently decide to repudiate anything that they digitally signed in that event. For example, one can dispute one's participation in a key establishment or being the originator of a signed email message as easily as one can dispute one's signing of a document with the intent to be bound to the content of that document. The repudiation may not be successful. The Non-repudiation Framework, Rec. ITU-T X.813 | ISO/IEC 10181-4, describes a dispute resolution process as follows:

1) evidence generation;
2) evidence transfer, storage and retrieval;
3) evidence verification; and
4) dispute resolution.

The generated evidence may include, but is not limited to:

- audit records pertinent to the event and an assertion of intent;
- statements made by third party notaries;
- policy statements;
- digitally signed information, including audit records and notary statements;
- timestamps of the digitally signed information;
- the public-key certificates supporting the digital signature;
- the appropriate revocation information published and available at the time of the disputed event; and,
- any public-key certificate revocations subsequent to the time of the event which indicate a key compromise occurred before the time of the event.

The integrity of stored data that might be presented as evidence may be maintained in a variety of ways, e.g., access control, storage of hashes by a trusted third party, digital signature. It may also be necessary periodically to strengthen the protection of that stored data to counteract improvements in computer processing and/or crypto-analysis.

NOTE – Neither the type and amount of evidence generated nor the level of integrity is specified by this Specification. However, it is expected that the level of effort will be commensurate with the risk involved.

Evidence verification may require the revalidation of the digital signatures of data, e.g., messages, documents, certificates, CRLs, and timestamps that were used in the initial validation process. The fact that a public-key certificate has expired shall not preclude its use for revalidating signatures created during the validity period of that public-key certificate. A public-key certificate that has been revoked may be used if it can be determined that the public-key certificate was valid at the time of the disputed event.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Even if all the digital evidence described above is considered technically valid, other conditions, e.g., the intent, understanding or competence of the signer, may allow the signer successfully to repudiate it.

8 Trust models

Procedures for determining the trustworthiness of a public-key certificate for a particular transaction are not defined here. In the three cornered trust model described below, the relying party acts on its own behalf to determine both the validity and trustworthiness of a public-key certificate. In the four-cornered trust model, the relying party engages the services of a trust broker to determine the trustworthiness of a public-key certificate on its behalf.

8.1 Three-cornered trust model

The three-cornered trust model applies to an environment where the entity to which a public-key certificate has been issued (the subject) and the relying party both have some relationship with the CAs within a closed environment.

This is illustrated by the simplified example shown in Figure 4, where the subject of the public-key certificate and the relying party are associated with the same CA. In this environment, the trustworthiness of the issuing CA is relatively easy to establish.

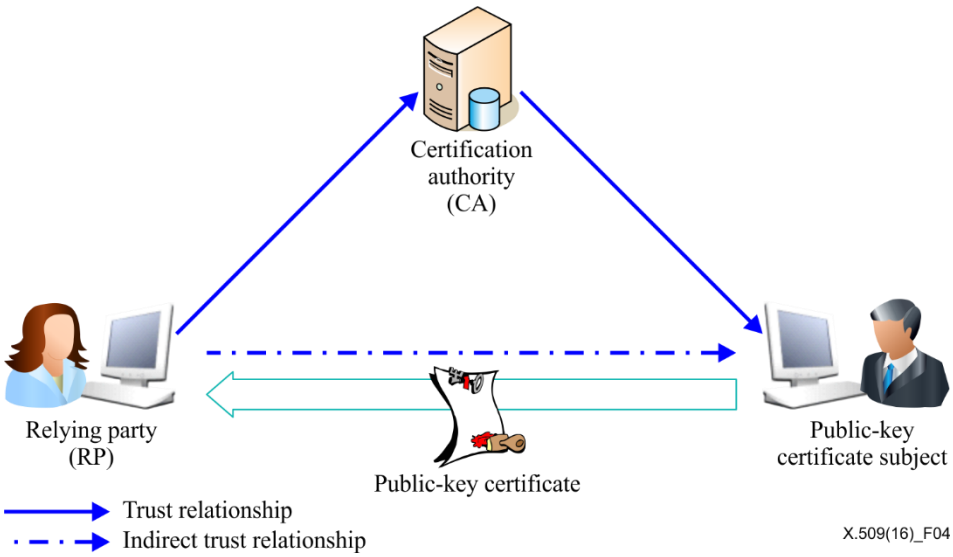

Figure 4 – The three cornered trust model

In this three cornered trust model, the public-key certificate subject trusts the CA and has asked it to issue a public-key certificate. Upon positive verification of the identity of the public-key certificate subject, the CA issues it with a public key certificate containing the subject's identifier. The relying party, being a public-key certificate subject of a CA within the same closed environment, also knows the trustworthiness of the issuing CA. Consequently the relying party can determine whether to trust the public-key certificate subject for the current transaction.

8.2 Four cornered trust model

The four cornered trust model may be used in open public-key infrastructures, where the relying party does not have a close relationship with the CA that issued the public-key certificate under validation. See Figure 5.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

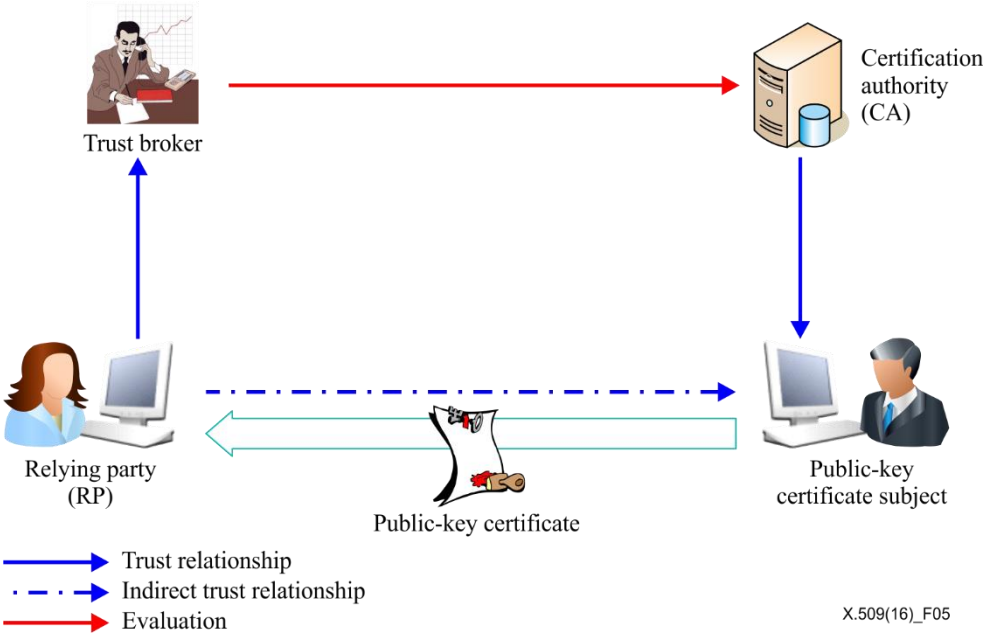

Figure 5 – The four cornered trust model

In the four cornered model, the relying party trusts an entity called the trust broker. The relying party is still responsible for validating public-key certificates and public-key certificate chains. However, the trust broker evaluates the trustworthiness of the CA for all intended transactions and relays this to the relying party so that it can decide about the trustworthiness of the public-key certificate subject for the current transaction. If the relying party's decision is positive, this means that the relying party trusts the CA and its public-key certificate for this transaction. The CA trusts the identity of the public-key certificate subject. Hence the relying party can indirectly trust the public-key certificate of the subject for this particular transaction.

9 Public-key certificate and CRL extensions

Some extensions defined in this clause are for use with public-key certificates. Other extensions are defined for CRLs (including CARLs and EPRLs). Extensions for use with attribute certificates and ACRLs (including AARLs and EARLs) are defined in clause 17.

This clause specifies extensions in the following areas:

a) Key and policy information: These public-key certificate and CRL extensions convey additional information about the keys involved, including key identifiers for subject and issuer keys, indicators of intended or restricted key usage, and indicators of a certificate policy.

b) Subject and issuer attributes: These public-key certificate and CRL extensions support alternative names, of various name forms, for a public-key certificate subject, a public-key certificate issuer, or a CRL issuer. These extensions can also convey additional attribute information about the public-key certificate subject, to assist a relying party in being confident that the public-key certificate subject is a particular person or entity.

c) Certification path constraints: These public-key certificate extensions allow constraint specifications to be included in CA certificates, i.e., CA certificates for CAs issued by other CAs, to facilitate the automated processing of certification paths when multiple certificate policies are involved. Multiple certificate policies arise when policies vary for different applications in an environment or when interoperation with external environments occurs. The constraints may restrict the types of public-key certificates that can be issued by the subject CA or that may occur subsequently in a certification path.

d) Basic CRL extensions: These CRL extensions allow a CRL to include indications of revocation reason, to provide for temporary suspension of a public-key certificate, and to include CRL-issue sequence numbers to allow relying parties to detect missing CRLs in a sequence from one CRL issuer.

e) CRL distribution points and delta CRLs: These public-key certificate and CRL extensions allow the complete set of revocation information from one CA to be partitioned into separate CRLs and allow revocation information from multiple CAs to be combined in one CRL. These extensions also support the use of partial CRLs indicating only changes since an earlier CRL issue.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Inclusion of any extension in a public-key certificate or CRL is at the option of the CA issuing that public-key certificate or the issuing authority for that CRL.

In a public-key certificate or CRL, an extension is flagged as being either critical or as non-critical. If an extension is flagged as critical and a relying party does not recognize the extension type or does not implement the semantics of the extension, then that relying party shall consider the public-key certificate as invalid. If an extension is flagged as non-critical, a relying party that does not recognize or implement that extension type may process the remainder of the public-key certificate ignoring the extension. If an extension is flagged as non-critical, a relying party that does recognize the extension, shall process the extension. Extension type definitions in this Specification indicate whether the extension shall always be flagged as critical, always be flagged as non-critical, or whether criticality can be decided by the public-key certificate or CRL issuer. The reason for requiring some extensions to be always flagged as non-critical is to allow relying parties which do not need to use such extensions to omit support for them without jeopardizing the ability to interoperate with all CAs.

NOTE – A relying party may require certain non-critical extensions to be present in a public-key certificate in order for that public-key certificate to be considered acceptable. The need for inclusion of such extensions may be implied by local policy rules of the relying party or may be a CA policy rule indicated to the relying party by inclusion of a particular public-key certificate policy identifier in the public-key certificate policies extension with that extension being flagged as critical.

For all public-key certificate extensions, CRL extensions, and CRL entry extensions defined in this Specification, there shall be no more than one instance of each extension type in any public-key certificate, CRL, or CRL entry, respectively.

9.1 Policy handling

9.1.1 Certificate policy

This framework contains four types of entities: The relying party, the CA, the trust broker, and the public-key certificate subject. Each entity operates under obligations to the other entities and, in return, enjoys limited warranties offered by them. The obligations and warranties of a CA are defined in its certificate policy. A certificate policy is a document (usually in plain-language, but it could be machine readable). It may be referenced by an object identifier and a URL, which may be included in the certificate policies extension of the public-key certificate issued by the CA to the entity and upon which the relying party relies. A public-key certificate may be issued in accordance with one or more policies. The definition of the policy and assignment of the object identifier is performed by a policy authority. The set of policies administered by a policy authority is called a policy domain. All public-key certificates are issued in accordance with a policy, even if the policy is neither recorded anywhere nor referenced in the public-key certificate. This Specification does not prescribe the style or contents of the certificate policy.

The relying party may be bound to its obligations under the certificate policy by the act of importing a CA public key and using it as trust anchor information, or by relying on a public-key certificate that includes the associated policy identifier. The CA may be bound to its obligations under the policy by the act of issuing a public-key certificate that includes the associated policy identifier. The public-key certificate subject and any relying party who is also a public-key certificate subject may be bound to its obligations under the policy by the act of requesting and accepting a public-key certificate that includes the associated policy identifier and by using the corresponding private key. Implementations that do not use the certificate policies extension should achieve the required binding by other means.

The relying party and trust broker may be bound by any contractual agreement that they have or by any trust policy that the trust broker issues.

For an entity simply to declare conformance to a policy does not generally satisfy the assurance requirements of the other entities in the framework. They require some reason to believe that the other parties operate a reliable implementation of their policy. However, if explicitly stated in the certificate policy, relying parties may accept the CA's assurances that its subjects agree to be bound by their obligations under the policy, without having to confirm this directly with them. This aspect of certificate policy is outside the scope of this Specification.

A CA may place limitations on the use of its public-key certificates, in order to control the risk that it assumes as a result of issuing public-key certificates. For instance, it may restrict the community of relying parties, the purposes for which they may use its public-key certificates and/or the type and extent of damages that it is prepared to make good in the event of a failure on its part, or that of its entities acting as PKI end entities. These matters should be defined in the certificate policy.

Additional information, to help affected entities understand the provisions of the certificate policy, may be included in the certificate policies extension in the form of policy qualifiers.

9.1.2 Cross-certificates and policy handling

The warranties and obligations shared by the subject CA, the intermediate CA and the relying party are defined by the certificate policy identified in the cross-certificate, in accordance with which the subject CA may act as, or on behalf of, an end entity. The warranties and obligations shared by the public-key certificate subject, the subject CA and the

© ISO/IEC 2020 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

intermediate CA are defined by the certificate policy identified in the end-entity public-key certificate, in accordance with which the intermediate CA may act as, or on behalf of, a relying party.

A certification path is said to be valid under the set of certificate policies that are common to all public-key certificates in the path.

In addition to the situation described above, there are two special cases to be considered:

a) the CA does not use the certificate policies extension to convey its policy requirements to relying parties; and

b) the relying party or intermediate CA delegates the job of controlling policy to the next CA in the path.

In the first case, the public-key certificate should not contain a certificate policies extension at all. As a result, the set of policies under which the path is valid will be null. But, the path may be valid nonetheless. Relying parties shall still ensure that they are using the public-key certificate in conformance with the policies of the CAs in the path.

In the second case, the relying party or intermediate CA should include the special value *any-policy* in the *initial-policy-set* or cross-certificate. Where a public-key certificate includes the special value *any-policy*, it should not include any other certificate policy identifiers. The identifier *any-policy* should not have any associated policy qualifiers.

The relying party can ensure that all its obligations are conveyed by setting the *initial-explicit-policy* indicator. In this way, only authorities that use the certificate policies extension as their way of achieving binding are accepted in the path, and relying parties have no additional obligations. Because CAs also attract obligations when they act as, or on behalf of, a relying party, they can ensure that all their obligations are conveyed by setting the **`requireExplicitPolicy`** component of the **`policyConstraints`** extension in the cross-certificate.

9.1.3 Policy mapping

Some certification paths may cross boundaries between policy domains. The warranties and obligations according to which the cross-certificate is issued may be materially equivalent to some or all of the warranties and obligations according to which the subject CA issues end-entity public-key certificates, even though the policy authorities under which the two CAs operate may have selected different object identifiers for these materially equivalent policies. In this case, the intermediate CA may include a policy mappings extension in the cross-certificate. In the policy mappings extension, the intermediate CA assures the relying party that it will continue to enjoy the familiar warranties, and that it should continue to fulfil its familiar obligations, even though subsequent entities in the certification path operate in a different policy domain. The intermediate CA should include one or more mappings for each of a subset of the policies under which it issued the cross-certificate, and it should not include mappings for any other policies. If one or more of the certificate policies according to which the subject CA operates is identical to those according to which the intermediate CA operates (i.e., it has the same unique identifier), then these identifiers should be excluded from the policy mapping extension, but included in the certificate policies extension.

Policy mapping has the effect of converting all policy identifiers in public-key certificates further down the certification path to the identifier of the equivalent policy, as recognized by the relying party.

Policies shall not be mapped either to or from the special value *any-policy*.

Relying parties may determine that public-key certificates issued in a policy domain other than its own should not be relied upon, even though a trusted intermediate CA may determine its policy to be materially equivalent to its own. It can do this by setting the *initial-policy-mapping-inhibit input* to the path validation procedure. Additionally, an intermediate CA may make a similar determination on behalf of its relying parties. In order to ensure that relying parties correctly enforce this requirement, it can set **`inhibitPolicyMapping`** in a **`policyConstraints`** extension.

9.1.4 Certification path processing

The relying party faces a choice between two strategies:

a) it can require that the certification path be valid under at least one of a set of policies pre-determined by the relying party; or

b) it can ask the path validation module to report the set of policies for which the certification path is valid.

The first strategy may be most appropriate when the relying party knows, *a priori*, the set of policies that are acceptable for its intended use.

The second strategy may be the most appropriate when the relying party does not know, *a priori*, the set of policies that are acceptable for its intended use.

In the first instance, the certification path validation procedure will indicate the path to be valid only if it is valid under one or more of the policies specified in the *initial-policy-set*, and it will return the sub-set of the *initial-policy-set* under which the path is valid. In the second instance, the certification path validation procedure may indicate that the path is

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

invalid under the *initial-policy-set*, but valid under a disjoint set: the *authorities-constrained-policy-set*. Then the relying party shall determine whether its intended use of the public-key certificate is consistent with one or more of the certificate policies under which the path is valid. By setting the *initial-policy-set* to *any-policy*, the relying party can cause the procedure to return a valid result if the path is valid under any (unspecified) policy.

9.1.5 Self-issued certificates

A CA may issue a certificate to itself under three circumstances:

a) as a convenient way of encoding the public key associated with the private key used to sign the public-key certificate, so that it can be communicated to and stored as trust anchor information by relying parties;

b) for certifying additional public keys of the CA used for purposes other than those covered by category a) (such as OCSP and possibly CRL signing); and

c) for replacing its own expired CA certificates.

These types of CA certificates are called self-issued certificates, and they can be recognized by the fact that the issuer and subject names present in them are identical. For the purposes of path validation, self-issued certificates of category a) are self-signed certificates and are therefore verified with the public key contained in them, and if they are encountered in the path, they shall be ignored.

Self-issued certificates of type b) may only appear as end-entity public-key certificates in a path, and shall be processed as end-entity public-key certificates.

Self-issued certificates of type c) (also known as self-issued intermediate certificates) may appear as intermediate certificates in a path. As a matter of good practice, when replacing a key that is on the point of expiration, a CA should request the issuance of any in-bound cross-certificates that it requires for its replacement public key before using the key. Nevertheless, if self-issued certificates of this category are encountered in the path, they shall be processed as intermediate certificates, with the following exception: they do not contribute to the path length for the purposes of processing the **`pathLenConstraint`** component of the **`basicConstraints`** extension and the *skip-certificates* values associated with the *policy-mapping-inhibit-pending* and *explicit-policy-pending indicators*.

If a CA uses the same key to sign public-key certificates and CRLs, a single self-issued certificate of category a) shall be used. If a CA uses a different key to sign CRLs than that used to sign public-key certificates, the CA may choose to issue two self-issued public-key certificates of category a), one for each of the keys. In this situation, relying parties would need access to both self-issued public-key certificates to establish separate trust anchors for public-key certificates and CRLs signed by that CA. Alternatively, a CA may issue one self-issued public-key certificate of category a) for public-key certificate signing and one self-issued public-key certificate of category b) for CRL signing. In this situation, relying parties use the key certified in the public-key certificate of category a) as their single trust anchor for both public-key certificates and CRLs signed by that CA. In this case, if the self-issued public-key certificate of category b) were to be used to verify signatures on CRLs, there is no means defined in this Specification to check the validity of that public-key certificate.

If self-issued certificates of category b) are encountered within a path, they shall be ignored.

NOTE – Other mechanisms for distributing CA public keys are outside the scope of this Specification.

9.2 Key and policy information extensions

9.2.1 Requirements

The following requirements relate to key and policy information:

a) CA key pair updating can occur at regular intervals or in special circumstances. There is a need for a certificate extension to convey an identifier of the public key to be used to verify the public-key certificate signature. A relying party can use such identifiers in finding the correct CA certificate for validating the public-key certificate issuer's public key.

b) In general, a public-key certificate subject has different public keys and, correspondingly, different public-key certificates for different purposes, e.g., digital signature and encipherment key agreement. A public-key certificate extension is needed to assist a relying party in selecting the correct public-key certificate for a given subject for a particular purpose or to allow a CA to stipulate that a certified key may only be used for a particular purpose.

c) Subject key pair updating can occur at regular intervals or in special circumstances. There is a need for a public-key certificate extension to convey an identifier to distinguish between different public keys for the same subject used at different points in time. A relying party can use such identifiers in finding the correct public-key certificate.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

d) The private key corresponding to a certified public key is typically used over a different period from the validity of the public key. With a key pair used for creating and validating signatures, the usage period for the private key used for signing is typically shorter than that for the public key used to verify the signature. The validity period of the public-key certificate indicates a period for which the public key may be used, which is not necessarily the same as the usage period of the private key. In the event of a private key compromise, the period of exposure can be limited if the relying party knows the legitimate use period for the private key. There is therefore a requirement to be able to indicate the usage period of the private key in a public-key certificate.

e) Because public-key certificates may be used in environments where multiple certificate policies apply, provision needs to be made for including certificate policy information in public-key certificates.

f) When cross-certifying from one organization to another, it can sometimes be agreed that certain of the two organizations' policies can be considered equivalent. A CA certificate needs to allow the issuing CA to indicate that one of its own certificate policies is equivalent to another certificate policy in the subject CA's domain. This is known as policy mapping.

g) A user of an encipherment or digital signature system which uses public-key certificates defined in this Specification needs to be able to determine in advance the algorithms supported by other users.

h) There is a need to specify that an end-entity public-key certificate shall be validated using an authorization and validation list.

9.2.2 Public-key certificate and CRL extensions

The following extensions are defined:

a) authority key identifier;

b) subject key identifier;

c) key usage;

d) extended key usage;

e) private key usage period;

f) certificate policies;

g) policy mappings

h) authorization and validation.

These extensions shall be used only as public-key certificate extensions, except for the **authorityKeyIdentifier** extension 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.

9.2.2.1 Authority key identifier extension

This extension, which may be used as either a public-key certificate extension or CRL extension, identifies the public key to be used to verify the signature on this public-key certificate or CRL. It enables distinct keys used by the same CA to be distinguished (e.g., as key updating occurs) and it enables distinct keys used by the same CRL issuer to be distinguished. This extension is defined as follows:

```
authorityKeyIdentifier EXTENSION ::= {
  SYNTAX         AuthorityKeyIdentifier
  IDENTIFIED BY  id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
  keyIdentifier              [0]  KeyIdentifier OPTIONAL,
  authorityCertIssuer        [1]  GeneralNames OPTIONAL,
  authorityCertSerialNumber  [2]  CertificateSerialNumber OPTIONAL,
  ... }
  (WITH COMPONENTS {..., authorityCertIssuer        PRESENT,
                         authorityCertSerialNumber  PRESENT } |
   WITH COMPONENTS {..., authorityCertIssuer        ABSENT,
                         authorityCertSerialNumber  ABSENT } )

KeyIdentifier ::= OCTET STRING
```

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 then the public-key certificate

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

or CRL issuer shall ensure they are consistent. A key identifier shall be unique with respect to all key identifiers for the issuing authority for the public-key certificate or CRL 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** data type).

CAs shall assign public-key certificate serial numbers such that every (issuer, public-key certificate serial number) pair uniquely identifies a single public-key certificate. The **keyIdentifier** component can be used to select CA certificates during path construction. The **authorityCertIssuer**, **authoritySerialNumber** pair can only be used to provide preference to one public-key certificate over others during path construction.

This extension shall always be flagged as non-critical.

9.2.2.2 Subject key identifier extension

This extension identifies the public key being certified. It enables distinct keys used by the same subject to be differentiated (e.g., as key updating occurs). This extension is defined as follows:

```
subjectKeyIdentifier EXTENSION ::= {
  SYNTAX          SubjectKeyIdentifier
  IDENTIFIED BY   id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier
```

A key identifier shall be unique with respect to all key identifiers for the subject with which it is used.

This extension shall always be flagged as non-critical.

9.2.2.3 Key usage extension

This extension identifies the intended usage for which the public-key certificate has been issued. The intended usage may be further constrained by policy. This policy may be stated in a certificate policy definition, a contract, or other specification. However, a policy shall not override the constraint indicated by a **KeyUsage** bit, e.g., a certificate policy could not allow a public-key certificate to be used for digital signature if **KeyUsage** indicated that it could only be used for key agreement.

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

Setting a specific value of **KeyUsage** in a public-key certificate does not in itself signal for an instance of communication that the communicating parties are acting in accordance with this setting, e.g., when signing a document. The definition of methods by which parties may signal their intent for a specific instance of communication (e.g., commitment to content for that specific instance) is outside the scope of this Specification, but it is anticipated that multiple methods will exist. Although not recommended, it is possible to use the content of the public-key certificate, e.g., certificate policy, to signal the intent of the signing. However, since that signal was made when the public-key certificate was issued by the CA, such use may not meet the requirement that declaring the intent is made at the time of signing by the signer.

More than one bit may be set in an instance of the **keyUsage** extension. The setting of multiple bits shall not change the meaning of each individual bit but shall indicate that the public-key certificate may be used for all of the purposes indicated by the set bits. There may be risks incurred when setting multiple bits. A review of those risks is documented in Annex K.

This extension is defined as follows:

```
keyUsage EXTENSION ::= {
  SYNTAX          KeyUsage
  IDENTIFIED BY   id-ce-keyUsage }

KeyUsage ::= BIT STRING {
  digitalSignature  (0),
  contentCommitment (1),
  keyEncipherment   (2),
  dataEncipherment  (3),
  keyAgreement      (4),
  keyCertSign       (5),
  cRLSign           (6),
  encipherOnly      (7),
  decipherOnly      (8) }
```

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Bits in the **KeyUsage** type are as follows:

a) **digitalSignature**: for verifying digital signatures that are used with an entity authentication service, a data origin authentication service and/or an integrity service;

b) **contentCommitment**: for verifying digital signatures which are intended to signal that the signer is committing to the content being signed. The type of commitment the public-key certificate can be used to support may be further constrained by the CA, e.g., through a certificate policy. The precise type of commitment of the signer e.g., "reviewed and approved" or "with the intent to be bound", may be signalled by the content being signed, e.g., the signed document itself or some additional signed information.

   Since a content commitment signing is considered a digitally signed transaction, the **digitalSignature** bit need not be set in the certificate. If it is set, it does not affect the level of commitment the signer has endowed in the signed content.

   It is not incorrect to refer to this **keyUsage** bit using the identifier **nonRepudiation**. However, the use of this identifier has been deprecated. Regardless of the identifier used, the semantics of this bit are as specified in this Specification;

c) **keyEncipherment**: for enciphering keys or other security information, e.g., for key transport;

d) **dataEncipherment**: for enciphering user data, but not keys or other security information as in c);

e) **keyAgreement**: for use as a public key agreement key, such an when an asymmetric Diffie-Hellman key pair is used for key management.

f) **keyCertSign**: for verifying a CA's signature on public-key certificates.

   Since public-key certificate signing is considered a commitment to the content of the public-key certificate by the CA, neither the **digitalSignature** bit nor the **contentCommitment** bit need be set in the CA certificate. If either (or both) is set, it does not affect the level of commitment the signer has endowed in the signed public-key certificate;

g) **cRLSign**: for verifying an authority's signature on CRLs.

   Since CRL signing is considered to be commitment to the content of the CRL by the CRL issuer, neither the **digitalSignature** bit nor the **contentCommitment** bit need be set in the public-key certificate. If either (or both) is set, it does not affect the level of commitment the signer has endowed in the signed CRL;

h) **encipherOnly**: public key agreement key for use only in enciphering data when used with **keyAgreement** bit also set (meaning with other key usage bit set is undefined);

i) **decipherOnly**: public key agreement key for use only in deciphering data when used with **keyAgreement** bit also set (meaning with other key usage bit set is undefined).

Application specifications should indicate which of the **digitalSignature** or **contentCommitment** bits are appropriate for their use. If a signing application has no knowledge of the signer's intent regarding commitment to content, the application shall sign and support that signing with a public-key certificate that has the **digitalSignature** bit set in that public-key certificate's **keyUsage** extension.

Even though a digital signature was verified using a public-key certificate that has only the **digitalSignature** bit set, other factors external to the verification of the digital signature may also play a role in determining the intent of the signing. Conversely, even though a digital signature was verified using a public-key certificate that has only the **contentCommitment** bit set, external factors may be used by the signer to disclaim commitment to the signed content.

The bit **keyCertSign** is for use in CA certificates only. If **KeyUsage** is set to **keyCertSign**, the value of the **cA** component of the **basicConstraints** extension shall be set to **TRUE**. CAs may also use other defined key usage bits in **KeyUsage**, e.g., **digitalSignature** for providing authentication and integrity of online administration transactions.

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

If the extension is flagged as critical or if the extension is flagged as non-critical but the relying party recognizes it, then the public-key certificate shall be used only for a purpose for which the corresponding key usage bit is set to one. If the extension is flagged as non-critical and the relying party does not recognize it, then this extension shall be ignored.

A bit set to zero indicates that the key is not intended for that purpose. If the extension is present with all bits set to zero, the key is intended for a purpose other than those listed above.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

9.2.2.4 Extended key usage extension

This extension 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. This extension is defined as follows:

```
extKeyUsage EXTENSION ::= {
  SYNTAX         SEQUENCE SIZE (1..MAX) OF KeyPurposeId
  IDENTIFIED BY  id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER
```

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

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 relying 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 flagged either as critical or as non-critical.

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

If the extension is flagged as non-critical, then it indicates the intended purpose or purposes of the key, and may be used in finding the correct key/public-key certificate of an entity that has multiple keys/public-key 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. (A relying party may nevertheless require that a particular purpose be indicated in order for the public-key certificate to be acceptable.)

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

This Specification defines the following key purposes that can be included in the extended key usage extension:

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

id-avlSign  OBJECT IDENTIFIER ::= {id-kp 2}
```

The id-**avlSign** key purpose is used to indicate that the private key may be used for signing an AVL.

Other purposes that can also be included are defined in other specifications, such as IETF RFC 5280.

9.2.2.5 Private key usage period extension

This extension indicates the period of use of the private key corresponding to the certified public key. It is applicable only for private key used for creating digital signatures. This extension is defined as follows:

```
privateKeyUsagePeriod EXTENSION ::= {
  SYNTAX         PrivateKeyUsagePeriod
  IDENTIFIED BY  id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
  notBefore  [0]  GeneralizedTime OPTIONAL,
  notAfter   [1]  GeneralizedTime OPTIONAL,
  ... }
  (WITH COMPONENTS {..., notBefore  PRESENT } |
   WITH COMPONENTS {..., notAfter   PRESENT } )
```

The **notBefore** component indicates the earliest date and time at which the private key may be used for signing. If the **notBefore** component is not present, then no information is provided as to when the period of valid use of the private key commences beyond that specified in the validity component of the public-key certificate. The **notAfter** component indicates the latest date and time at which the private key may be used for signing. If the **notAfter** component is not

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

present then no information is provided as to when the period of valid use of the private key concludes beyond that specified in the validity component of the public-key certificate.

This extension shall always be flagged as non-critical.

NOTE 1 – The period of valid use of the private key may be different from the certified validity of the public key as indicated by the public-key certificate validity period. The usage period for the private key used for signing is typically shorter than that for the public key used for verifying the signature.

NOTE 2 – The period of use of the private key corresponding to a public key can only be enforced if both the private key and the corresponding public-key certificate are placed in a tamper-resistant hardware module that contains a reliable clock synchronized with UTC. When this is not the case, a signer may avoid using a signing private key up to the very end of the validity period of the public-key certificate. This is one possible use of this extension.

NOTE 3 – In general, this Specification does not associate any semantic with this extension. Any particular use of this extension will have to specify the semantic associated with that usage.

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

9.2.2.6 Certificate policies extension

This extension 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 12. 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 relying 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 extension is defined as follows:

```
certificatePolicies EXTENSION ::= {
  SYNTAX         CertificatePoliciesSyntax
  IDENTIFIED BY  id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
  policyIdentifier  CertPolicyId,
  policyQualifiers  SEQUENCE SIZE (1..MAX) OF PolicyQualifierInfo OPTIONAL,
  ... }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
  policyQualifierId  CERT-POLICY-QUALIFIER.&id({SupportedPolicyQualifiers}),
  qualifier          CERT-POLICY-QUALIFIER.&Qualifier
             ({SupportedPolicyQualifiers}{@policyQualifierId}) OPTIONAL,
  ... }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= {...}
```

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

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

If the extension is flagged as critical, it indicates that the public-key certificate shall only be used for the purpose, and in accordance with the rules implied by one of the indicated certificate policies. The rules of a particular policy may require the relying party to process the qualifier value in a particular way.

If the extension is flagged as non-critical, use of this extension does not necessarily constrain use of the public-key certificate to the policies listed. However, a relying party may require a particular policy to be present in order to use the public-key certificate (see clause 12). Policy qualifiers may, at the option of the relying party, be processed or ignored.

© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

Certificate policies and certificate policy qualifier types may be defined by any organization with a need. Object identifiers used to identify certificate policies and certificate policy qualifier types shall be assigned in accordance with Rec. ITU-T X.660 | ISO/IEC 9834-1. A CA may assert any-policy by using the **anyPolicy** object identifier in order to trust a public-key certificate for all possible policies. Because of the need for identification of this special value to apply regardless of the application or environment, that object identifier is assigned in this Specification. No object identifiers will be assigned in this Specification for specific certificate policies. That assignment is the responsibility of the entity that defines the certificate policy.

```
anyPolicy OBJECT IDENTIFIER ::= {id-ce-certificatePolicies 0}
```

The identifier **anyPolicy** should not have any associated policy qualifiers.

The following ASN.1 information object class is used in defining certificate policy qualifier types:

```
CERT-POLICY-QUALIFIER ::= CLASS {
  &id                    OBJECT IDENTIFIER UNIQUE,
  &Qualifier             OPTIONAL }
WITH SYNTAX {
  POLICY-QUALIFIER-ID &id
  [QUALIFIER-TYPE     &Qualifier] }
```

A definition of a policy qualifier type shall include:

- a statement of the semantics of the possible values; and
- an indication of whether the qualifier identifier may appear in a certificate policies extension without an accompanying value and, if so, the implied semantics in such a case.

NOTE – A qualifier may be specified as having any ASN.1 type. When the qualifier is anticipated to be used primarily with applications that do not have ASN.1 decoding functions, it is recommended that the type **OCTET STRING** be specified. The ASN.1 **OCTET STRING** value can then convey a qualifier value encoded according to any convention specified by the policy element defining organization.

9.2.2.7 Policy mappings extension

This extension, which shall be used in CA certificates only, allows an issuing CA to indicate that, for the purposes of the user of a certification path containing this CA certificate, one of the issuer's certificate policies can be considered equivalent to a different certificate policy used in the subject CA's domain. This extension is defined as follows:

```
policyMappings EXTENSION ::= {
  SYNTAX         PolicyMappingsSyntax
  IDENTIFIED BY  id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
  issuerDomainPolicy   CertPolicyId,
  subjectDomainPolicy  CertPolicyId,
  ... }
```

The **issuerDomainPolicy** component indicates a certificate policy that is recognized in the issuing CA's domain and that can be considered equivalent to the certificate policy indicated in the **subjectDomainPolicy** component that is recognized in the subject CA's domain.

Policies shall not be mapped to or from the special value **anyPolicy**.

This extension may, at the option of the issuing CA, be either flagged as critical or as non-critical. It is recommended that it be flagged as critical, otherwise a relying party may not correctly interpret the stipulation of the issuing CA.

NOTE 1 – An example of policy mapping is as follows. The U.S. government domain may have a policy called Canadian Trade and the Canadian government may have a policy called U.S. Trade. While the two policies are distinctly identified and defined, there may be an agreement between the two governments to accept certification paths extending cross-border within the rules implied by these policies for relevant purposes.

NOTE 2 – Policy mapping implies significant administrative overheads and the involvement of suitably diligent and authorized personnel in related decision-making. In general, it is preferable to agree upon a more global use of common policies than it is to apply policy mapping. In the above example, it would be preferable for the U.S., Canada and Mexico to agree upon a common policy for North American Trade.

NOTE 3 – It is anticipated that policy mapping will be practical only in limited environments in which policy statements are very simple.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020

9.2.2.8 Authorization and validation extension

This extension may only be present in end-entity public-key certificates. It indicates that this public-key certification shall not be accepted if it cannot be checked against a particular authorization validation list (AVL).

```
authorizationValidation EXTENSION ::= {
  SYNTAX          AvlId
  IDENTIFIED BY   id-ce-authorizationValidation }

AvlId ::= SEQUENCE {
  issuer         Name,
  serialNumber   AvlSerialNumber OPTIONAL,
  ... }
```

The **issuer** component shall identify the issuer of the AVL against which this end-entity public-key certificate shall be checked.

The **serialNumber** component shall be present if the AVL includes the **serialNumber** component. Otherwise, it shall be absent.

This extension shall always be flagged as critical.

9.3 Subject and issuer information extensions

9.3.1 Requirements

The following requirements relate to public-key certificate **subject** component and public-key certificate and CRL **issuer** component:

a) Public-key certificates need to be usable by applications that employ a variety of name forms, including Internet electronic mail names, Internet domain names, Rec. ITU-T X.400 originator/recipient addresses, and EDI party names. It is therefore necessary to be able securely to associate multiple names of a variety of name forms with a public-key certificate subject or a public-key certificate or CRL issuer.

b) There may be a need for convoying identity and/or privilege information in a public-key certificate. Such information is provided as directory attributes.

9.3.2 Certificate and CRL extensions

The following extensions are defined:

a) Subject alternative name;

b) Issuer alternative name;

c) Subject directory attributes.

These extensions shall be used only as public-key certificate extensions, except for issuer alternative name, which may also be used as a CRL extension. As public-key certificate extensions, they may be present in CA certificates and end-entity public-key certificates.

9.3.2.1 Subject alternative name extension

This extension contains one or more alternative names, using any of a variety of name forms, for the entity that is bound by the issuing CA to the certified public key. This extension is defined as follows:

```
subjectAltName EXTENSION ::= {
  SYNTAX          GeneralNames
  IDENTIFIED BY   id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
  otherName                  [0]  INSTANCE OF OTHER-NAME,
  rfc822Name                 [1]  IA5String,
  dNSName                    [2]  IA5String,
  x400Address                [3]  ORAddress,
  directoryName              [4]  Name,
  ediPartyName               [5]  EDIPartyName,
  uniformResourceIdentifier  [6]  IA5String,
  iPAddress                  [7]  OCTET STRING,
  registeredID               [8]  OBJECT IDENTIFIER,
```


© ISO/IEC 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 9594-8:2020