International Civil Aviation Organization

TAG/MRTD/22-WP/7
16/04/14
English Only

WORKING PAPER

TECHNICAL ADVISORY GROUP ON MACHINE READABLE TRAVEL DOCUMENTS (TAG/MRTD)

TWENTY-SECOND MEETING

Montréal, 21 to 23 May 2014

Agenda Item 2: Activities of the NTWG

UPDATED TECHNICAL REPORT LDS AND PKI MAINTENANCE

(Presented by the New Technologies Working Group)

1. INTRODUCTION

1.1 In its twentieth meeting in September 2011 the TAG/MRTD endorsed Working Paper 7 presenting the "Technical Report LDS and PKI Maintenance" version 1.0.

1.2 This Technical Report resulted from an evaluation of Doc 9303 and contains revised and extended specifications on:

a) LDS version information;

b) Certificate profiles;

c) Chip access control;

d) Active Authentication.

1.3 This Working Paper at hand presents an updated version of the Technical Report "LDS and PKI maintenance".

2. BACKGROUND

2.1 Doc 9303 specifies the certificate profiles for Country Signing CA certificates and Document Signer certificates. In practice the assignments of qualifications such as mandatory and

optional to the various certificate extensions appeared to be too strict in some cases, as well as not strict enough in others. This resulted in interoperability issues at the exchange of certificates. Therefore the specified profiles were revised in version 1.0 of the Technical Report.

2.2 Version 1.0 of the Technical Report also specifies profiles for Master List signer certificates and Communications certificates.

2.3 In practice it proves that with respect to the certificate profiles more guidance is desired on the interpretation of various fields and extensions in the certificates.

3. CURRENT STATUS

3.1 The updated Technical Report “LDS and PKI maintenance” version 2.0 provides additional guidance on the certificate profiles specified therein.

3.2 As such, version 2.0 of the Technical Report is to be interpreted as a logical follow up on the first evaluation of the electronic specifications in Doc 9303.

3.3 Version 2.0 of the Technical Report replaces version 1.0. The contents on LDS version information, chip access control and Active Authentication have remained unchanged.

4. ACTION BY THE TAG/MRTD

4.1 The TAG/MRTD is invited to:

a) reconfirm the necessity of a regular evaluation of the specifications in Doc 9303 to preserve the appropriate level of accuracy and security; and

b) endorse the updated Technical Report “LDS and PKI Maintenance” version 2.0; and

c) approve inclusion of this updated Technical Report into the seventh edition of Doc 9303.

— END —

MACHINE READABLE TRAVEL DOCUMENTS

Final TECHNICAL REPORT

LDS and PKI Maintenance

Version – **2.0**
Date – April 21, 2014

Release Control

Release	Date	Description
0.1	Jan 7, 2013	First draft for TF5
0.2	Jan 8, 2013	Updated draft
0.3	Jan 21, 2013	Update after TF5 Singapore
0.4	June 6, 2013	Final draft after TF5 London
0.5	July 31, 2013	Update to address comments received during review of 0.4
0.6	Sept 13 2013	Added note on Subject Alternative Names extension (see 3.2.1 Note 4) Added note on CRLDP extension for PKD (see 3.2.1 Note 5)
0.7	Oct 4, 2013	Added note clarifying CSCA must not re-use serial number (see 3.2.2) Added option to include IssuerAltName in CRL after CSCA name change (see 3.2.3 extensions table and Note 1) Corrected note on CRLDP extension for PKD (see 3.2.1 Note 5) Corrected keyUsage bits settings for communication certificates (see 3.2.1 extensions table)
0.8	Oct 10, 2013	Added a note in the comments column of 3.2.1 extensions table clarifying why digital signature is optional for communication certificates
2.0	April 21, 2014	Final clarifications including: NameChange extension also optional in CSCA self-signed Root certificates Text reorganized (not changed) for UTCTime and GeneralizedTime in profiles Replaced "…" with "CountryCode" for PKD CRL URL example Restructured part of Note 5 into a bulleted list

Table of contents

1. Introduction

The specifications for the electronic part of Machine Readable Travel Documents have been in place since 2004. In this fast moving world it is necessary to evaluate these specifications from time to time to stay up to date, especially with respect to the cryptographic security features and PKI.
Therefore an evaluation work plan has been developed, addressing the various aspects that may need to be updated.
This Technical Report results from this evaluation, and provides updated specifications for relevant subjects.

1.1 Assumptions

It has been assumed that the reader is familiar with the concepts and mechanisms offered by public key cryptography and public key infrastructures.

It has been assumed that the reader is familiar with the contents of *ICAO Doc 9303-Machine Readable Travel Documents, part 1-Machine Readable Passports, Volume 2-Specifications for Electronically Enabled Passports with Biometric Identification Capability, sixth edition-2006*, e.g. *ICAO Doc 9303-Machine Readable Travel Documents, part 3-Machine Readable Official Travel Documents, Volume 2-Specifications for Electronically Enabled MRtds with Biometric Identification Capability, third edition-2008*, as well as *ICAO Supplement to Doc 9303, latest release*.

1.2 Terminology

1.2.1 Technical report terminology

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in RFC 2119, *S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, March 1997.*

In case OPTIONAL features are implemented, they MUST be implemented as described in this Technical Report.

1.2.2 Abbreviations

Abbreviation	
C_{DS}	Document Signer Certificate
DER	Distinguished Encoding Rule
ICAO	International Civil Aviation Organization
LDS	Logical Data Structure
SO_D	Document Security Object

1.3 Reference documentation

The following documentation served as reference for Doc 9303, Technical Reports and the Supplement:

ANSI X9.62:2005, *"Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", 7 January 1999.*

FIPS 180-2, *Federal Information Processing Standards Publication (FIPS PUB) 180-2, Secure Hash Standard, August 2002*.

FIPS 186-2 or 186-3, *Federal Information Processing Standards Publication (FIPS PUB) 186-2 (+ Change Notice), Digital Signature Standard, 27 January 2000 (Supersedes FIPS PUB 186-1 dated 15 December 1998)*).

ISO 1073-2: 1976, *Alphanumeric character sets for optical recognition — Part 2: Character set OCR-B — Shapes and dimensions of the printed image*

ISO 1831: 1980, *Printing specifications for optical character recognition*

ISO 3166-1: 2006, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 3166-2: 2007, *Codes for representation of names of countries and their subdivisions — Part 2: Country subdivision code*

ISO/IEC 7810: 1995, *Identification cards — Physical characteristics*

ISO/IEC 7816-2: 2007, *Identification cards - Integrated circuit cards - Part 2: Cards with contacts - Dimensions and location of the contacts.*

ISO/IEC 7816-4: 2005, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*

ISO/IEC 7816-5: 2004, *Identification cards — Integrated circuit cards — Part 5: Registration of application providers*

ISO/IEC 7816-6: 2004, *Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange (Defect report included)*

ISO/IEC 7816-11: 2004, *Identification cards — Integrated circuit cards — Part 11: Personal verification through biometric methods*

ISO 8601:2000, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 8825-1:2002, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO/IEC 9796-2: 2002, *Information Technology — Security Techniques — Digital Signature Schemes giving message recovery — Part 2: Integer factorization based mechanisms*.

ISO/IEC 9797-1:1999, *Information technology —Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher.*

ISO/IEC 10373-6:2011, *Identification cards – Test methods – Part 6: Proximity cards*

ISO/IEC 10373-6:2001/Amd 7:2010, *Identification cards – Test methods – Part 6: Proximity cards – Test methods for ePassports and ePassport Readers*

ISO/IEC 10646:2003, *Information technology — Universal Multiple-Octet Coded Character Set (UCS).*

ISO/IEC 10918, *Information technology — Digital compression and coding of continuous-tone still images.*

ISO 11568-2:2005, *Banking — Key management (retail) — Part 2: Symmetric ciphers, their key management and life cycle.*

ISO/IEC 11770-2:1996, *Information technology Security techniques Key management Part 2: Mechanisms using symmetric techniques.*

ISO/IEC 14443-1:2008, *Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 1: Physical Characteristics*

ISO/IEC 14443-2:2010, *Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 2: Radio Frequency Power and Signal Interface*

ISO/IEC 14443-3:2011, *Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 3: Initialization and Anticollision*

ISO/IEC 14443-4:2008, *Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 4: Transmission protocol*

ISO/IEC15444, *Information Technology - JPEG 2000 image coding system*

ISO/IEC 15946: 2002, *Information technology Security techniques Cryptographic techniques based on elliptic curves.*

ISO/IEC 19794-4, *Information technology — Biometric data interchange formats — Part 4: Finger image data*

ISO/IEC 19794-5, *Information technology — Biometric data interchange formats — Part 5: Facial image data*

ISO/IEC 19794-6, *Information technology — Biometric data interchange formats — Part 6: Iris image data*

RFC 2119, *S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, March 1997.*

RFC 3279, *W. Polk, R. Housley, L. Bassham, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", April 2002.*

RFC 3280, *R. Housley, W. Polk, W. Ford, D. Solo, "X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", April 2002.*

RFC 5280, *D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", May 2008.*

RFC 3369, *R. Housley, Cryptographic Message Syntax (CMS), August 2002.*

RFC 3447, *J. Jonsson, B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", February 2003.*

TR-03111, *Bundesamt für Sicherheit in der Informationstechnik, „Technical Guideline - Elliptic Curve Cryptography - Version 1.11", April 2009.*

Unicode 4.0.0, *The Unicode Consortium. The Unicode Standard, Version 4.0.0, defined by: The Unicode Standard, Version 4.0 (Boston, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1) (Consistent with ISO/IEC 10646-1)*

2. LDS Versioning

2.1 Present specification

ICAO Doc 9303 specifies in Volume 2, Section III, the LDS version 1.7. LDS version numbering and Unicode version numbering are specifies in the EF.COM, as follows:

Header. The Header contains the following information, which enables a receiving State or approved receiving organization locate and decode the various Data Groups and Data Elements contained within the block of data recorded by the issuing State or organization.

APPLICATION IDENTIFIER (AID)
LDS VERSION NUMBER
UNICODE VERSION NUMBER

LDS Version Number. The LDS Version Number defines the format version of the LDS. The exact format to be used for storing this value will be defined in the technology mapping annexes. Standardized format for an LDS Version Number is "aabb", where,

- "aa" = number (01 –99) identifying the Version of the LDS (i.e., Significant additions to the LDS)
- "bb" = number (01-99) identifying the Update of the LDS

Future upgrades to the standardized organization of the LDS have been anticipated and will be addressed through publication of Amendments to the specifications by ICAO. A Version Number will be assigned to each upgrade to ensure that receiving States and approved receiving organizations will be able to accurately decode all versions of the LDS.

Unicode Version Number. The Unicode Version Number identifies the coding method used when recording alpha, numeric and special characters, including national characters. The standardized format for a Unicode Version Number is “aabbcc”, where, The exact format to be used for storing this value will be defined in the technology mapping annexes.

- “aa” = number identifying the **Major version** of the Unicode Standard (i.e. Significant additions to the standard, published as a book);
- “bb” = number identifying the **Minor version** of the Unicode Standard (i.e. Character additions or more significant normative changes, published as a Technical Report); and
- “cc” = number identifying the **Update version** of the Unicode Standard (i.e. Any other changes to normative or important informative portions of the Standard that could change program behavior. These changes are reflected in new Unicode Character Database files and an update page).

Although future upgrades to the standardized organization of the LDS have been anticipated and the decoding of future LDS versions is supported by the LDS version number, this construction has a drawback.
The EF.COM file is not signed. Therefore undetected manipulation of its contents is possible. This may be of interest for an attacker, who aims at masking the presence of new security features in a specific LDS version.
Therefore it is desirable that the LDS version number is part of the signed information, and as such protected by Passive Authentication.

2.2 Revised specification

The Document Security Object has been extended with a signed attribute, containing the LDS and Unicode version information:

```
LDSVersionInfo ::= SEQUENCE {
      ldsVersion       PRINTABLE STRING
      unicodeVersion   PRINTABLE STRING }
```

The version number of the Document Security Object has been incremented from V0 to V1.

Specification of the Security Object V1 is as follows:
The Document Security object is implemented as a SignedData Type, as specified in RFC 3369, *R. Housley, Cryptographic Message Syntax (CMS), August 2002*. All security objects MUST be produced in Distinguished Encoding Rule (DER) format to preserve the integrity of the signatures within them.

Signed Data Type
The processing rules in RFC3369 apply.

m mandatory – the field MUST be present
x do not use – the field SHOULD NOT be populated
o optional – the field MAY be present
c choice – the field contents is a choice from alternatives

Value		Comments
SignedData		
version	m	Value = v3
digestAlgorithms	m	
encapContentInfo	m	
eContentType	m	id-icao-mrtd-security-ldsSecurityObject
eContent	m	The encoded contents of an ldsSecurityObject
certificates	m	Nations SHALL include the Document Signer Certificate (C_{DS}) which can be used to verify the signature in the signerInfos field.
Crls	x	It is recommended that States do not use this field
signerInfos	m	It is recommended that states only provide 1 signerinfo within this field.
SignerInfo	m	
version	m	The value of this field is dictated by the sid field. See RFC3369 Section 5.3 for rules regarding this field
Sid	m	
issuerandSerialNumber	c	It is recommended that nations support this field over subjectKeyIdentifier.
subjectKeyIdentifier	c	
digestAlgorithm	m	The algorithm identifier of the algorithm used to produce the hash value over encapsulatedContent and SignedAttrs.
signedAttrs	m	Producing nations may wish to include additional attributes for inclusion in the signature, however these do not have to be processed by receiving nations except to verify the signature value.
signatureAlgorithm	m	The algorithm identifier of the algorithm used to produce the signature value, and any associated parameters.

Value		Comments
signature	m	The result of the signature generation process.
unsignedAttrs	o	Producing States may wish to use this field, but it is not recommended and receiving nations may choose to ignore them.

ASN.1 Profile LDS Security Object

```
LDSSecurityObject {iso(2) identified-organization(23) icao(136)
mrtd(1) security(1) ldsSecurityObject(1)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Imports from RFC 3280 [PROFILE], Appendix A.1
AlgorithmIdentifier FROM
PKIX1Explicit88 { iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-pkix1-explicit(18) }

-- Constants

ub-DataGroups INTEGER ::= 16

-- Object Identifiers

id-icao                 OBJECT IDENTIFIER ::= {2.23.136}
id-icao-mrtd            OBJECT IDENTIFIER ::= {id-icao 1}
id-icao-mrtd-security   OBJECT IDENTIFIER ::= {id-icao-mrtd 1}
id-icao-mrtd-security-ldsSecurityObject OBJECT IDENTIFIER ::= {id-
icao-mrtd-security 1}

-- LDS Security Object

LDSSecurityObjectVersion ::= INTEGER {V0(0), V1(1)
-- If LDSSecurityObjectVersion is V1, ldsVersionInfo MUST be present }

DigestAlgorithmIdentifier ::= AlgorithmIdentifier

LDSSecurityObject ::= SEQUENCE {
     version               LDSSecurityObjectVersion,
     hashAlgorithm         DigestAlgorithmIdentifier,
     dataGroupHashValues   SEQUENCE SIZE (2..ub-DataGroups) OF
     DataGroupHash
     ldsVersionInfo        LDSVersionInfo OPTIONAL
     -- If present, version MUST be V1 }

DataGroupHash ::= SEQUENCE {
     dataGroupNumber       DataGroupNumber,
     dataGroupHashValue    OCTET STRING }

DataGroupNumber ::= INTEGER {
     dataGroup1        (1),
     dataGroup2        (2),
     dataGroup3        (3),
     dataGroup4        (4),
     dataGroup5        (5),
```

```
      dataGroup6        (6),
      dataGroup7        (7),
      dataGroup8        (8),
      dataGroup9        (9),
      dataGroup10      (10),
      dataGroup11      (11),
      dataGroup12      (12),
      dataGroup13      (13),
      dataGroup14      (14),
      dataGroup15      (15),
      dataGroup16      (16)}

LDSVersionInfo ::= SEQUENCE {
      ldsVersion       PRINTABLE STRING
      unicodeVersion   PRINTABLE STRING }

END
```

Note:
The field `dataGroupValue` contains the calculated hash over the complete contents of the Data group EF, specified by `dataGroupNumber`.

2.3 Backwards compatibility

The change will be implemented in the revised LDS specifications V1.8.
The change has an impact on inspection systems. These systems will need to be able to parse the SO_D V1 structure. When the EF.COM is not present, version information (both the LDS version as well as the SO_D version) can only be retrieved from the SO_D.

2.4 Implementation strategy

With this change all information, present in the EF.COM, has been duplicated in the SO_D. This means that the EF.COM will be removed from the specifications from the next LDS version after V1.8.
It is RECOMMENDED that inspection systems that rely on the EF.COM will be modified to use the SO_D instead as soon as possible.

2.5 Documentation

Present documentation, affected by this change, is:

ICAO Doc 9303-Machine Readable Travel Documents, part 1-Machine Readable Passports, Volume 2-Specifications for Electronically Enabled Passports with Biometric Identification Capability, sixth edition-2006

- Normative Appendix 3 to Section IV - “Document Security Object”

ICAO Doc 9303-Machine Readable Travel Documents, part 3-Machine Readable Official Travel Documents, Volume 2-Specifications for Electronically Enabled MRtds with Biometric Identification Capability, third edition-2008

- Normative Appendix 3 to Section IV - “Document Security Object”

ICAO Supplement to Doc 9303, latest release

- R1-p1_v2_sIV_0006

3. Certificate and CRL Profiles

3.1 Present specification

ICAO Doc 9303 specifies in Volume 2, Section IV, Normative Appendix 1 the certificate profiles for Country Signing CA certificates and Document Signer certificates. These profiles are based on the requirement that each issuing State or entity SHALL create a single CSCA for the purpose of signing all Doc 9303 compliant MRTDs/MRtds.

In addition to these original certificate and CRL profiles, ICAO published "Machine Readable Travel Documents – Guidance Document – PKI for Machine Readable Travel Documents" in 2011. This document outlines detailed specific requirements inherited from the base documents referenced by the ICAO profiles, such as RFC 3280.

In practice the assignments of the qualifications **m**, **x**, **o**, and **c** to the various extensions appeared to be too strict in some cases, as well as not strict enough in others. This resulted in interoperability issues at the exchange of certificates. Therefore the specified profiles were reviewed and the first edition of this Technical Report was published in 2011 containing a revised set of certificate profiles.

The updated profiles in the original edition of this TR also contained two additional certificate profiles that were defined since the original profiles were published in ICAO 9303:

- for the CSCA Master List Signer
- for Communications, even though it is not strictly needed today. This is a future proofing step, for certificates that may be used for access to the PKD or for LDAP/EMAIL/ HTTP communications between countries. It is recommended to position this under the CSCA.

For these two new certificate types (Master List Signer and Communications), the private key lifetime and the certificate validity period are left to the discretion of the issuer.

The CRL profile was not modified at that time.

3.2 Revised specification

In 2011, ICAO published "Machine Readable Travel Documents – Guidance Document – PKI for Machine Readable Travel Documents". That document outlined detailed technical requirements inherited from the base specification (RFC 3280) for the original profiles published in ICAO 9303.

To date, a revision of the Guidance Document that aligns with the updated profiles from the original edition of this TR (and the associated updated base referenced specification – RFC 5280) has not been published.

The certificate and CRL profiles in this edition of the Technical Report on LDS and PKI Maintenance integrate the inherited detailed technical requirements from RFC 5280 directly. This approach should make it easier for implementors as all the requirements are presented together, rather than providing a separate update of the earlier Guidance Document.

The updated certificate profiles are in 3.2.1 and the updated CRL profile is in 3.2.3.

The Guidance Document also included explanatory text that was duplicated from RFC 3280. This updated TR performs the same service for the updated profiles by duplicating the relevant text from RFC 5280. Because RFC 5280 is the definitive specification for these requirements, this duplication is provided only as an informative convenience and the original referenced document is the definitive specification. The duplicated text can be found in Informative Appendix A of this Technical Report.

3.2.1 Certificate Profiles

The following tables outline requirements for eMRTD certificates and CRLs issued by CSCAs.

The first table lists the components of a certificate. For each component, the "Presence" column indicates whether the component MUST, MAY or MUST NOT be present. The "Comments" column indicates specific requirements associated with populating that component.

The second table lists a set of certificate extensions and, for each certificate type, indicates whether that extension MUST, MAY or MUST NOT be included. If a certificate extension MUST or MAY be included in certificates of a given type, an indication of whether that extension must be included with its criticality flag set to "critical" or "non-critical" is also provided. The "comments" column indicates specific requirements associated with populating that particular certificate extension. Most of these extensions are defined in other standard documents, such as RFC 5280. However there are two extensions that are specific to the eMRTD application. Their specification is in section 3.2.2 of this document.

The profile uses the following terminology for presence requirements of each of the components/extensions certificates:

m mandatory – the field MUST be present
x do not use – the field MUST NOT be populated
optional – the field MAY be present

The profile uses the following terminology for criticality requirements of extensions that may/must be included in certificates:

c critical – the extension is marked critical, receiving applications MUST be able to process this extension.
nc - the extensions is marked non-critical, receiving applications that do not understand this extension may ignore it.

Some of the requirements identified in these profiles are inherited from the referenced base profiles (such as RFC 5280). For convenience, the relevant text from the base profile that covers the specific requirement is duplicated in a table in Informative Appendix A. Note that the material in Appendix A is only a subset of the referenced specification and is not updated on a regular basis. Therefore the referenced document itself should be consulted for the full and current specification of the requirements.

Certificate Body

Certificate Component	Presence in Certificates	Comments
Certificate	m	
TBSCertificate	m	see next part of the table
signatureAlgorithm	m	value inserted here dependent on algorithm selected
signatureValue	m	value inserted here dependent on algorithm selected
TBSCertificate		
version	m	MUST be v3
serialNumber	m	MUST be positive integer and maximum 20 Octets MUST use 2's complement encoding and be represented in the smallest number of octets
signature	m	value inserted here MUST be the same as that in signatureAlgorithm component of Certificate sequence
issuer	m	country Name and serialNumber, if present, MUST be PrintableString

Certificate Component	Presence in Certificates	Comments
		Other attributes that have DirectoryString syntax MUST be either PrintableString or UTF8String countryName MUST be Upper Case See Note 3 for naming conventions
validity	m	MUST terminate with Zulu (Z) Seconds element MUST be present Dates through 2049 MUST be in UTCTime UTCTime MUST be represented as YYMMDDHHMMSSZ Dates in 2050 and beyond MUST be in GeneralizedTime. GeneralizedTime MUST NOT have fractional seconds GeneralizedTime MUST be represented as YYYYMMDDHHMMSSZ
subject	m	countryName and serialNumber, if present, MUST be PrintableString Other attributes that have DirectoryString syntax MUST be either PrintableString or UTF8String countryName MUST be Upper Case countryName in issuer and subject fields MUST match See Note 3 for naming conventions
subjectPublicKeyInfo	m	
issuerUniqueID	x	
subjectUniqueID	x	
extensions	m	See next table on which extensions should be present Default values for extensions MUST NOT be encoded

Extensions

Extension name	CSCA Self-Signed Root		CSCA Link		Document Signer		Master List Signer		Communication		Comments
	Presence	**Criticality**	**Presence**	**Criticality**	**Presence**	**Criticality**	**Presence**	**Criticality**	**Presence**	**Criticality**	
AuthorityKeyIdentifier	**o**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	
keyIdentifier	m		m		m		m		m		
authorityCertIssuer	o		o		o		o		o		
authorityCertSerialNumber	o		o		o		o		o		
SubjectKeyIdentifier	**m**	**nc**	**m**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	
subjectKeyIdentifier	m		m		m		m		m		
KeyUsage	**m**	**c**	**m**	**c**	**m**	**c**	**m**	**c**	**m**	**c**	
digitalSignature	x		x		m		m		o		Some communication certificates (e.g. TLS certificates) require that the keyUsage bits

Extension name	CSCA Self-Signed Root		CSCA Link		Document Signer		Master List Signer		Communication		Comments
											be set in accordance with the particular cipher suite used. Some cipher suites do, and some do not require the digitalSignature bit to be set.
nonRepudiation	x		x		x		x		x		
keyEncipherment	x		x		x		x		o		
dataEncipherment	x		x		x		x		x		
keyAgreement	x		x		x		x		o		
keyCertSign	m		m		x		x		x		
cRLSign	m		m		x		x		x		
encipherOnly	x		x		x		x		x		
decipherOnly	x		x		x		x		x		
PrivateKeyUsagePeriod	**m**	**nc**	**m**	**nc**	**m**	**nc**	**o**	**nc**	**o**	**nc**	
notBefore	o		o		o		o		o		At least one of notBefore or notAfter MUST be present MUST be encoded as generalizedTime
notAfter	o		o		o		o		o		
CertificatePolicies	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	
PolicyInformation	m		m		m		m		m		
policyIdentifier	m		m		m		m		m		
policyQualifiers	o		o		o		o		o		
PolicyMappings	**x**		**x**		**x**		**x**		**x**		See Note 1
SubjectAltName	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	See Note 4
IssuerAltName	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	See Note 4
SubjectDirectoryAttributes	**x**		**x**		**x**		**x**		**x**		
Basic Constraints	**m**	**c**	**m**	**c**	**x**		**x**		**x**		
cA	m		m		x		x		x		
PathLenConstraint	m		m		x		x		x		MUST always be ‘0’
NameConstraints	**x**		**x**		**x**		**x**		**x**		See Note 1
PolicyConstraints	**x**		**x**		**x**		**x**		**x**		See Note 1
ExtKeyUsage	**x**		**x**		**x**		**m**	**c**	**m**	**c**	See Note 2
CRLDistributionPoints	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**o**	**nc**	
distributionPoint	m		m		m		m		m		MUST be ldap, http or https See Note 5
reasons	x		x		x		x		x		
cRLIssuer	x		x		x		x		x		
InhibitAnyPolicy	**x**		**x**		**x**		**x**		**x**		See Note1
FreshestCRL	**x**		**x**		**x**		**x**		**x**		See Note 6
privateInternetExtensions	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	See Note 7
NameChange	**o**	**nc**	**o**	**nc**	**x**		**x**		**x**		See 3.2.2
DocumentType	**x**		**x**		**m**	**nc**	**x**		**x**		See 3.2.2
Netscape Certificate Type	**x**		**x**		**x**		**x**		**x**		See Note 8

Extension name	CSCA Self-Signed Root		CSCA Link		Document Signer		Master List Signer		Communication		Comments
other private extensions	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	

Note 1 - Extensions for Intermediate Certificates:
The extension, by definition, can only appear in intermediate CA certificates (certificates issued by one CA to another CA). Intermediate CA certificates are not used in the eMRTD PKI. Therefore this extension is prohibited from eMRTD certificates.

Note 2 – Extended Key Usage Values:
As specified in the Master List TR, the OID that must be included in the extended key usage extension for Master List certificates is 2.23.136.1.1.3. For communication certificates the value of this extension depends on the communication protocol used (see RFC 5280, section 4.2.1.12).

Note 3 – Certificate and Naming conventions
The following naming and addressing conventions for Issuer and Subject fields are REQUIRED.

- countryName. MUST be present. The value contains a country code that MUST follow the format of two letter country codes, specified in ISO 3166-1: 2006, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*)
- commonName. MUST be present.

Other attributes MAY also be included at the discretion of the issuing State.

Note 4 – Alternative Names:
Because the functions served by alternative names in the eMRTD application are specific to this application, and different from those defined for the Internet PKI in [RFC 5280], values in the Subject Alternative Name extension of eMRTD certificates do not generally unambiguously identify the certificate subject.

Alternative Names serves two functions.

The first function is to provide contact information of the subject and/or issuer of the certificate. For that purpose it SHOULD include at least one of the following:

- rfc822Name;
- dNSName;
- uniformResourceIdentifier

The second function is to provide a directory string made of ICAO assigned country codes. For this purpose certificates issued using this profile MUST additionally include a directory name that is constructed as follows:

- localityName that contains the ICAO country code as it appears in the MRZ;
- if this country code does not uniquely define the issuing State or entity, the attribute stateOrProvinceName SHALL be used to indicate the ICAO assigned three letter code for the issuing State or entity.

Other attributes are disallowed.

In CSCA Self-signed Root certificates, the IssuerAltName and SubjectAltName extensions MUST be identical. In CSCA Link certificates, the values MAY be different. For example, if a change has occurred

with the rfc822Name of the CSCA immediately prior to issuance of a CSCA Link certificate, the IssuerAltName extension would contain the old rfc822Name and the SubjectAltName extension would contain the new rfc822Name. Any subsequent CSCA Link certificates would contain the new rfc822Name in both extensions.

Note 5- CRL Distribution Point:
CSCAs may publish their CRL in several places including the PKD, their own website etc. Values that point to copies published directly by the CSCA (e.g. on their own website) are under the control of the CSCA.

For CRLs submitted to the PKD, PKD participants MAY include two URL values for their CRL using the following template (replace "CountryCode" with the Issuing State or organization ICAO assigned 3 letter code). If this country code does not uniquely identify the Issuing State or organization, the entry will be creating by appending the symbol "_" to the three letter country code in the MRZ, and then the ICAO assigned three letter code for the Issuing State or organization which uniquely identifies the Issuing State or organization:

https://pkddownload1.icao.int/CRLs/CountryCode.crl
https://pkddownload2.icao.int/CRLs/CountryCode.crl

This is a mandatory extension and revocation status checks are a mandatory part of the validation procedure. Therefore at least one value MUST be populated

- The PKD values may be the only values in the extension;
- There may be additional values (e.g. a CSCA may also choose to publish their CRL on a website and include a pointer to that source); or
- A CSCA may also choose to include only a single value (e.g. a pointer to their website as a source) even if they also submit their CRL to the PKD.

The following examples illustrate the PKD values that would be populated in certificates issued by the Issuing Authority for Singapore and for Hong Kong:

Singapore PKD example:
https://pkddownload1.icao.int/CRLs/SGP.crl
https://pkddownload2.icao.int/CRLs/SGP.crl

Hong Kong example:
https://pkddownload1.icao.int/CRLs/CHN_HKG.crl
https://pkddownload2.icao.int/CRLs/CHN_HKG.crl

Note 6- Freshest CRL:
This extension is used to point to a delta CRL. Delta CRLs are not supported in the eMRTD PKI. Therefore this extension is prohibited.

Note 7- Private Internet Extensions:
These are two extensions defined in RFC 5280 that are used to point to information about the issuer or subject of a certificate. These extensions can be used to point to a variety of types of information and a few specific types are also defined in RFC 5280. The extensions (Authority Information Access and Subject Information Access) are not required in the eMRTD PKI. However, as they do not impact interoperability and are non-critical, they may optionally be included in eMRTD certificates.

Note 8- Netscape Certificate Type:
The Netscape Certificate Type extension can be used to limit the purposes for which a certificate can be used. The extKeyUsage and basicConstraints extensions are now the standard extensions for those purposes

and are used in the eMRTD application. Because of the potential conflict between values in the standard extensions and in the Netscape proprietary extension, the Netscape extension is prohibited.

3.2.2 eMRTD-Specific Certificate Extensions

Two certificate extensions listed in the table above are specific to the eMRTD application and are defined below.

Name Change Extension

When a CSCA key rollover occurs a certificate MUST be issued that links the new key to the old key to provide a secure transition for relying parties. Generally this is achieved through the issuance of a self-issued certificate where the issuer and subject fields are identical but the key used to verify the signature represents the old key pair and the certified public key represents the new key pair.

It is RECOMMENDED that CSCAs do not change their DN unnecessarily as there is an adverse impact on relying parties (other states must retain both the old and new names as valid CSCAs for the same state until all ePassports signed under the old name have expired). However, if a name change is necessary this MUST be conveyed to relying parties through the issuance of a CSCA Link certificate where the issuer is the old DN and the subject is the new DN. This CSCA Link certificate also conveys a key rollover where the key used to verify the signature represents the old key pair and the certified public key represents the new key pair. Certificates that convey both a CSCA name change and a key rollover for that CSCA MUST include the NameChange extension to identify the certificate as such. The NameChange extension MUST be set to non-critical. This has no effect on PathLengthConstraint; it remains ‘0’.

In addition, the NameChange extension MAY also be included in the new CSCA self signed certificate created upon the change of the CSCA DN. In such a self-signed CSCA Root certificate both the issuer and subject fields contain the new DN. Unlike the CSCA self-issued link certificate, containing both the old and new DN for the CSCA, inclusion of the NameChange extension in a CSCA self-signed Root certificate simply indicates that a name change has occurred and does not link the old DN to the new one.

A CSCA MUST NOT re-use certificate serial numbers. Each certificate issued by a CSCA, regardless of whether that CSCA has undergone a name change or not, MUST be unique.

ASN.1 for Name Change extension:

```
nameChange  EXTENSION  ::=  {
    SYNTAX           NULL
    IDENTIFIED BY    id-icao-mrtd-security-extensions-nameChange}

id-icao-mrtd-security-extensions OBJECT IDENTIFIER ::= {id-icao-mrtd-
security 6}
id-icao-mrtd-security-extensions-nameChange OBJECT IDENTIFIER ::= {id-
icao-
mrtd-security-extensions 1}
```

Document Type Extension

The DocumentType extension MUST be used to indicate the document types as they appear in the MRZ that the corresponding Document Signer is allowed to produce. The extension is identified below. This extension MUST always be set to non-critical.

ASN.1 for Document Type List extension:

```
documentTypeList  EXTENSION  ::=  {
    SYNTAX           DocumentTypeListSyntax
    IDENTIFIED BY    id-icao-mrtd-security-extensions-documentTypeList}

DocumentTypeListSyntax ::= SEQUENCE {
    version  DocumentTypeListVersion,
    docTypeList SET OF DocumentType }

DocumentTypeListVersion ::= INTEGER {v0(0)}

-- Document Type as contained in MRZ, e.g. "P" or "ID" where a
-- single letter denotes all document types starting with that letter
DocumentType ::= PrintableString(1..2)

id-icao-mrtd-security-extensions-documentTypeList OBJECT IDENTIFIER ::=
{id-icao-mrtd-security-extensions 2}
```

3.2.3 CRL Profile

Certificate List Component	**Country Signing CA CRL**	Comments
CertificateList	m	
tBSCertList	m	See next part of the table
signatureAlgorithm	m	Value inserted here dependent on algorithm selected
signatureValue	m	Value inserted here dependent on algorithm selected
tBSCertList		
version	m	MUST be v2
signature	m	value inserted here MUST be the same as that in signatureAlgorithm component of CertificateList sequence
issuer	m	countryName and serial Number, if present, MUST be PrintableString Other attributes that have DirectoryString syntax MUST be either PrintableString or UTF8String countryName MUST be Upper Case
thisUpdate	m	MUST terminate with Zulu (Z) Seconds element MUST be present Dates through 2049 MUST be in UTCTime UTCTime MUST be represented as YYMMDDHHMMSSZ Dates in 2050 and beyond MUST be in GeneralizedTime. GeneralizedTime MUST NOT have fractional seconds GeneralizedTime MUST be represented as YYYYMMDDHHMMSSZ

Certificate List Component	Country Signing CA CRL	Comments
nextUpdate	m	MUST terminate with Zulu (Z) Seconds element MUST be present Dates through 2049 MUST be in UTCTime UTCTime MUST be represented as YYMMDDHHMMSSZ Dates in 2050 and beyond MUST be in GeneralizedTime. GeneralizedTime MUST NOT have fractional seconds GeneralizedTime MUST be represented as YYYYMMDDHHMMSSZ
revokedCertificates	m	If present, MUST NOT be empty
crlExtensions	m	See next table on which extensions should be present Default values for extensions MUST NOT be encoded

Extension Name	Country Signing CA CRL	Criticality	Comments
CRL Extensions			
authorityKeyIdentifier	**m**	**nc**	This MUST be the same value as the subjectKeyIdentifier field in the CRL Issuer's certificate.
keyIdentifier	m		
authorityCertIssuer	o		
authorityCertSerialNumber	o		
issuerAlternativeName	**o**	**nc**	See Note 1
cRLNumber	**m**	**nc**	MUST be non-negative integer and maximum 20 Octets MUST use 2's complement encoding and be represented in the smallest number of octets
deltaCRLIndicator	**x**		
issuingDistributionPoint	**x**		
freshestCRL	**x**		
CRL Entry Extensions			
reasonCode	**x**		
holdInstructionCode	**x**		
invalidityDate	**x**		
certificateIssuer	**x**		

Note 1 – If a CSCA has undergone a name change, this extension MAY be included in CRLs issued following the CSCA name change. If present, the value(s) in this extension MUST be identical to the issuer field of certificates issued by the CSCA under that previous name. Once all certificates issued under a previous CSCA name have expired, that CSCA name can be excluded from subsequent CRLs.

*Inspection Systems are not required to process this extension. As stated in Doc 9303, "*Given that ICAO 9303 dictates a single CSCA per country, the countryName component of the issuer field is sufficient to uniquely identify the CSCA. The latest Public Key of that CSCA is used to verify the signature of the CRL. Since a CSCA issues a single CRL, this CRL covers all certificates issued with that countryName." In addition to that mandatory check, an optional check that the issuer field of the certificate is equal to the issuer field of the CRL or one of the values of the issuerAltName extension in the CRL MAY also be done.

Note 2.— It is possible that the CRL contains other revocation information, for example concerning system operator or registration authority certificates.

3.3 Backwards compatibility

These certificate profiles impose new requirements to the certificate issuers. From an interoperability point of view relying parties SHOULD be capable of accepting certificates that conform to the previous profile as well as the profiles specified in this chapter.

3.4 Implementation strategy

Issuers are RECOMMENDED to start issuing certificates conforming this new profile starting at their next CSCA roll-over.

3.5 Documentation

Present documentation, affected by this change, is:

ICAO Doc 9303-Machine Readable Travel Documents, part 1-Machine Readable Passports, Volume 2-Specifications for Electronically Enabled Passports with Biometric Identification Capability, sixth edition-2006

- Normative Appendix 1 to Section IV - "Certificate Profile"

ICAO Doc 9303-Machine Readable Travel Documents, part 3-Machine Readable Official Travel Documents, Volume 2-Specifications for Electronically Enabled MRtds with Biometric Identification Capability, third edition-2008

- Normative Appendix 1 to Section IV - "Certificate Profile"

ICAO Supplement to Doc 9303, latest release

- R3-p1_v2_sIV_0038

4. Access Control

4.1 Present specification

The present access control mechanism for eMRTDs is Basic Access Control.

4.2 Revised specification

The revised access control mechanisms are described in the ICAO Technical Report `Supplemental Access Control for Machine Readable Travel Documents`.

5. Active Authentication

5.1 Present specification

ICAO Doc 9303 specifies in section IV, par. 8.1 with respect to Active Authentication that "For signature generation in the Active Authentication mechanism, States SHALL use ISO/IEC 9796-2 Digital Signature scheme 1 (ISO/IEC 9796-2, Information Technology — Security Techniques — Digital Signature Schemes giving message recovery — Part 2: Integer factorisation based mechanisms, 2002.)"

Doc9303 specifies in section IV, par. 8.4 with respect to the use of ECDSA that "Those States implementing the ECDSA algorithm for signature generation or verification SHALL use X 9.62 (X9.62, "Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", 7 January 1999).

ISO/IEC 9796 specifies that the hash value is incorporated in the signature format. X9.62 specifies that the hash value itself must be used as input for the signature algorithm. This is confusing, use of ECDSA conforming to X9.62 would violate the requirement in par. 8.1.

To prevent different implementations caused by this confusion the Supplement to Doc9303 Release 7 recommends the use of RSA for AA and not ECDSA (see issue **R7-p1_v2_sIV_0057**).

The specification in this chapter provides a specification of the use of ECDSA in Active Authentication, in which a choice is made between the alternative ways for implementation.

5.2 Revised specification

There are three issues that need clarification or additional specification:

- The signature type returned by AA.
- Way to specify the HASH algorithm used.
- When HASH algorithm output is longer than the length of the ECDSA key, there are different ways to form the result.

5.2.1 The signature type returned by AA

X9.62 and ISO/IEC 9796 propose different methods.
Within these ICAO specifications a **plain signature (r||s)** SHALL be returned by the eMRTD for AA when using ECDSA. With respect to the length of **r** and **s** please refer to BSI TR 03111, par 5.2.1.
Only prime curves with uncompressed points SHALL be used.

Justification
plain signature (r||s) is

- recommended in TR-03111
- also used with EAC specified by EU
- already implemented on various products

5.2.2 Way to specify the HASH algorithm used

Following the current specification one can only specify in DG15 whether RSA or ECDSA is used. This can be done in the OID field of SubjectPublicKeyInfo, using the OIDs defined in RFC 3279. For RSA the used HASH algorithm is defined within the signature, in accordance to the signature generation scheme of ISO/IEC 9796-2. In case ECDSA is used there is no possibility to include any supplementary information within the signature itself.

The ASN.1 data structure `SecurityInfos` SHALL be provided by the MRTD chip in DG14 to indicate supported security protocols. Specification of the selected HASH algorithm MUST be incorporated into `SecurityInfos` in DG14. The `SecurityInfos` data structure is specified as follows:

```
SecurityInfos ::= SET of SecurityInfo

SecurityInfo ::= SEQUENCE {
      protocol OBJECT IDENTIFIER,
      requiredData ANY DEFINED BY protocol,
      optionalData ANY DEFINED BY protocol OPTIONAL
}
```

The elements contained in a SecurityInfo data structure have the following meaning:

- The object identifier protocol identifies the supported protocol.
- The open type requiredData contains protocol specific mandatory data.
- The open type optionalData contains protocol specific optional data.

If ECDSA based signature algorithm is used for Active Authentication by the MRTD chip, the SecurityInfos MUST contain following SecurityInfo entry:

```
ActiveAuthenticationInfo ::= SEQUENCE {
      protocol id-icao-mrtd-security-aaProtocolObject,
      version INTEGER  -- MUST be 1
      signatureAlgorithm OBJECT IDENTIFIER
}

-- Object Identifiers
id-icao OBJECT IDENTIFIER ::= {2 23 136}
id-icao-mrtd OBJECT IDENTIFIER ::= {id-icao 1}
id-icao-mrtd-security OBJECT IDENTIFIER ::= {id-icao-mrtd 1}

id-icao-mrtd-security-aaProtocolObject OBJECT IDENTIFIER ::=
   {id-icao-mrtd-security 5}
```

The object identifiers for `signatureAlgorithm` are defined in chapter 5.2.1 “Plain Format” of TR-03111.

Note:
SecurityInfos MAY contain entries to other protocols than Active Authentication (like Basic Access Control, Chip Authentication, Terminal Authentication).

Justification
Using security info in DG14 allows the eMRTD to specify the exact algorithm without requiring changes to the DG15 structure which would introduce potential compatibility issues.
Implicit algorithm selection is not recommended due to being vague and prone to misinterpretations.

5.2.3 HASH calculation output versus ECDSA key length

Because the calculation of the hash value from the message to be signed is part of the ECDSA signature process, using a HASH algorithm that gives a longer result than the length of used ECDSA key, will force part of the HASH value to be discarded.
Therefore a HASH algorithm, whose output length is of the same length or shorter than the length of the ECDSA key in use, SHALL be used with AA.

5.3 Backwards compatibility

n/a

5.4 Implementation strategy

n/a

5.5 Documentation

Present documentation, affected by this change, is:

ICAO Doc 9303-Machine Readable Travel Documents, part 1-Machine Readable Passports, Volume 2-Specifications for Electronically Enabled Passports with Biometric Identification Capability, sixth edition-2006

- Section IV - par. 7.2.2
- Section IV - par. 8.1
- Section IV - Normative Appendix 4

ICAO Doc 9303-Machine Readable Travel Documents, part 3-Machine Readable Official Travel Documents, Volume 2-Specifications for Electronically Enabled MRtds with Biometric Identification Capability, third edition-2008

- Section IV - par. 7.2.2
- Section IV - par. 8.1
- Section IV - Normative Appendix 4

ICAO Supplement to Doc 9303, latest release

- R7-p1_v2_sIV_0057
- R7-p3_v2_sIV_0010

6. Extended Length

6.1 Present specification

ICAO Doc 9303 currently does not contain specifications for the use of Extended Length APDUs. Specifications with respect to using Extended Length are under development within ISO/IEC JTC1 SC17 WG4.

Results of these developments will be incorporated into, or referenced in, Doc 9303 once finalized.

Informative Appendix A – Certificate and CRL Profile Reference Text

The Certificate profiles specified in Section 3.2.1 and the CRL profile specified in Section 3.2.3 are based on definitions and base profile requirements specified in referenced documents. Brief excerpts of some relevant sections from these source documents (as of the time of writing) are replicated in the tables below. These excerpts are provided to assist the reader in understanding the background for some of the requirements specified in the eMRTD certificate profiles in this Technical Report. They are not intended to be relied on instead of the referenced documents. In all cases, to obtain the full specification of the referenced component/extension and to obtain the most current specification, the actual referenced documents MUST be used.

Certificate Fields and Extensions

Component / Extension	Reference	Relevant Excerpts
Certificate	RFC 5280 - 4.1.1	
TBSCertificate	RFC 5280 - 4.1.1.1	
signatureAlgorithm	RFC 5280 - 4.1.1.2	
signatureValue	RFC 5280 - 4.1.1.3	
TBSCertificate	RFC 5280 - 4.1.2	
version	RFC 5280 - 4.1.2.1	When extensions are used, as expected in this profile, version MUST be 3 (value is 2)
serialNumber	RFC 5280 - 4.1.2.2	The serial number MUST be a positive integer assigned by the CA to each certificate. It MUST be unique for each certificate issued by a given CA (i.e., the issuer name and serial number identify a unique certificate). CAs MUST force the serialNumber to be a non-negative integer. Given the uniqueness requirements above, serial numbers can be expected to contain long integers. Certificate users MUST be able to handle serialNumber values up to 20 octets. Conformant CAs MUST NOT use serialNumber values longer than 20 octets.
	X.690 - 8.3.2	If the contents octets of an integer value encoding consist of more than one octet, then the bits of the first octet and bit 8 of the second octet: a) shall not all be ones; and b) shall not all be zero. NOTE – These rules ensure that an integer value is always encoded in the smallest possible number of octets.
	X.690 - 8.3.3	The contents octets shall be a two's complement binary number equal to the integer value, and consisting of bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet in turn up to and including the last octet of the contents octets.
signature	RFC 5280 - 4.1.1.2	This field MUST contain the same algorithm identifier as the signatureAlgorithm field in the sequence Certificate
issuer	RFC 5280 – Appendix A.1	X520countryName ::= PrintableString (SIZE (2)) X520SerialNumber ::= PrintableString (SIZE

		(1..ub-serial-number))
	RFC 5280 – 4.1.2.4	CAs conforming to this profile MUST use either the PrintableString or UTF8String encoding of DirectoryString
	ISO 3166-1	
validity	RFC 5280 - 4.1.2.5	Both notBefore and notAfter may be encoded as UTCTime or GeneralizedTime. CAs conforming to this profile MUST always encode certificate validity dates through the year 2049 as UTCTime; certificate validity dates in 2050 or later MUST be encoded as GeneralizedTime.
(if encoded as UTCTime)	X.690 – 11.8.1	The encoding shall terminate with "Z", as described in the ITU-T X.680 | ISO/IEC 8824-1 clause on UTCTime.
	X.690 – 11.8.2	The seconds element shall always be present
(if encoded as GeneralizedTime)	X.690 – 11.7.1	The encoding shall terminate with a "Z", as described in the ITU-T Rec. X.680 | ISO/IEC 8824-1 clause on GeneralizedTime.
	X.690 – 11.7.2	The seconds element shall always be present
	RFC 5280 – 4.1.2.5.2	GeneralizedTime values MUST NOT include fractional seconds. For the purposes of this profile, GeneralizedTime values MUST be expressed Greenwich Mean Time (Zulu) and MUST include seconds (i.e., times are YYYYMMDDHHMMSSZ), even where the number of seconds is zero.
subject	RFC 5280 – Appendix A.1	X520countryName ::= PrintableString (SIZE (2)) X520SerialNumber ::= PrintableString (SIZE (1..ub-serial-number))
	RFC 5280 – 4.1.2.6	CAs conforming to this profile MUST use either the PrintableString or UTF8String encoding of DirectoryString
subjectPublicKeyInfo	RFC 5280 - 4.1.2.7	
issuerUniqueID	RFC 5280 - 4.1.2.8	CAs conforming to this profile MUST NOT generate certificates with unique identifiers
subjectUniqueID	RFC 5280 - 4.1.2.8	CAs conforming to this profile MUST NOT generate certificates with unique identifiers
extensions	X.690 – 11.5	The encoding of a set value or sequence value shall not include an encoding for any component value which is equal to its default value
AuthorityKeyIdentifier	RFC 5280 – 4.2.1.1	The keyIdentifier field of the authorityKeyIdentifier extension MUST be included in all certificates generated by conforming CAs to facilitate certification path construction. There is one exception; where a CA distributes its public key in the form of a "self-signed" certificate, the authority key identifier MAY be omitted.
keyIdentifier		
authorityCertIssuer		
authorityCertSerialNumber		

SubjectKeyIdentifier	RFC 5280 – 4.2.1.2	To facilitate certification path construction, this extension MUST appear in all conforming CA certificates, that is, all certificates including the basic constraints extension (section 4.2.1.9) where the value of cA is TRUE
subjectKeyIdentifier		
KeyUsage	RFC 5280 – 4.2.1.3	The usage restriction might be employed when a key that could be used for more than one operation is to be restricted
digitalSignature		The digitalSignature bit is asserted when the subject public key is used with a digital signature mechanism to support security services other than certificate signing (bit 5), or CRL signing (bit 6)...
nonRepudiation		
keyEncipherment		
dataEncipherment		
keyAgreement		
keyCertSign		The keyCertSign bit is asserted when the subject public key is used for verifying a signature on public key certificates
cRLSign		The cRLSign bit is asserted when the subject public key is used for verifying a signature on certificate revocation list (e.g., a CRL, delta CRL, or an ARL). This bit MUST be asserted in certificates that are used to verify signatures on CRLs
encipherOnly		
decipherOnly		
PrivateKeyUsagePeriod	RFC 3280 – 4.2.1.4	CAs conforming to this profile MUST NOT generate certificates with private key usage period extensions unless at least one of the two components is present and the extension is non-critical
notBefore		Where used, notBefore and notAfter are represented as GeneralizedTime and MUST be specified and interpreted as defined in section 4.1.2.5.2
notAfter		
CertificatePolicies	RFC 5280 – 4.2.1.4	If this extension is critical, the path validation software MUST be able to interpret this extension (including the optional qualifier), or MUST reject the certificate
PolicyInformation		
policyIdentifier		
policyQualifiers		
PolicyMappings	RFC 5280 – 4.2.1.5	
SubjectAltName	RFC 5280 – 4.2.1.6	
IssuerAltName	RFC 5280 – 4.2.1.7	
SubjectDirectoryAttributes	RFC 5280 – 4.2.1.8	
Basic Constraints	RFC 5280 – 4.2.1.9	The basic constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate. Conforming CAs MUST include this extension in all CA certificates that contain public keys used to validate digital signatures on

		certificates and MUST mark the extension as critical in such certificates
cA		The cA boolean indicates whether the certified public key belongs to a CA. If the cA boolean is not asserted, then the keyCertSign bit in the key usage extension MUST NOT be asserted
PathLenConstraint		
NameConstraints	RFC 5280 – 4.2.1.10	
PolicyConstraints	RFC 5280 – 4.2.1.11	
ExtKeyUsage	RFC 5280 – 4.2.1.12	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
CRLDistributionPoints	RFC 5280 – 4.2.1.13	
distributionPoint		
reasons		
cRLIssuer		
InhibitAnyPolicy	RFC 5280 – 4.2.1.14	
FreshestCRL	RFC 5280 – 4.2.1.15	
privateInternetExtensions	RFC 5280 – 4.2.2	
NameChange		
DocumentType		
Netscape Certificate Type		
other private extensions		

CRL Fields and Extensions

Component / Extension	Reference	Relevant Excerpts
CertificateList	RFC 5280 - 5.1.1	
tBSCertList	RFC 5280 - 5.1.1.1	
signatureAlgorithm	RFC 5280 - 5.1.1.2	
signatureValue	RFC 5280 - 5.1.1.3	
	RFC 5280 - 5.1.2	
version	RFC 5280 - 5.1.2.1	This optional field describes the version of the encoded CRL. When extensions are used, as required by this profile, this field MUST be present and MUST specify version 2 (the integer value is 1).
signature	RFC 5280 - 5.1.2.2	This field MUST contain the same algorithm identifier as the signature field in the sequence CertificateList
issuer	RFC 5280 - Appendix A.1	X520countryName ::= PrintableString (SIZE (2)) X520SerialNumber ::= PrintableString (SIZE

Component / Extension	Reference	Relevant Excerpts
	RFC 5280 5.1.2.3 and 4.1.2.4	CAs conforming to this profile MUST use either the PrintableString or UTF8String encoding of DirectoryString
thisUpdate	RFC 5280 5.1.2.4	CRL issuers conforming to this profile MUST encode thisUpdate as UTCTime for dates through the year 2049. CRL issuers conforming to this profile MUST encode thisUpdate as GeneralizedTime for dates in the year 2050 or later.
(if encoded at UTCTime)	X.690 – 11.8.1	The encoding shall terminate with "Z", as described in the ITU-T X.680 | ISO/IEC 8824-1 clause on UTCTime.
	X.690 – 11.8.2	The seconds element shall always be present
(if encoded at GeneralizedTime)	X.690 – 11.7.1	The encoding shall terminate with a "Z", as described in the ITU-T Rec. X.680 | ISO/IEC 8824-1 clause on GeneralizedTime.
	X.690 – 11.7.2	The seconds element shall always be present
	RFC 5280 – 4.1.2.5.2	GeneralizedTime values MUST NOT include fractional seconds. For the purposes of this profile, GeneralizedTime values MUST be expressed Greenwich Mean Time (Zulu) and MUST include seconds (i.e., times are YYYYMMDDHHMMSSZ), even where the number of seconds is zero.
nextUpdate	5.1.2.5	CRL issuers conforming to this profile MUST encode nextUpdate as UTCTime for dates through the year 2049. CRL issuers conforming to this profile MUST encode nextUpdate as GeneralizedTime for dates in the year 2050 or later.
(if encoded at UTCTime)	X.690 – 11.8.1	The encoding shall terminate with "Z", as described in the ITU-T X.680 | ISO/IEC 8824-1 clause on UTCTime.
	X.690 – 11.8.2	The seconds element shall always be present
(if encoded at GeneralizedTime)	X.690 – 11.7.1	The encoding shall terminate with a "Z", as described in the ITU-T Rec. X.680 | ISO/IEC 8824-1 clause on GeneralizedTime.
	X.690 – 11.7.2	The seconds element shall always be present
	RFC 5280 – 4.1.2.5.2	GeneralizedTime values MUST NOT include fractional seconds. For the purposes of this profile, GeneralizedTime values MUST be expressed Greenwich Mean Time (Zulu) and MUST include seconds (i.e., times are YYYYMMDDHHMMSSZ), even where the number of seconds is zero.
revokedCertificates	RFC 5280 - 5.1.2.6	When there are no revoked certificates, the revoked certificates list MUST be absent. Otherwise, revoked certificates are listed by their serial numbers.
crlExtensions	RFC 5280 - 5.2	Conforming CRL issuers are REQUIRED to include the authority key identifier (Section 5.2.1) and the CRL number (Section 5.2.3) extensions in all CRLs issued.
	X.690 – 11.5	The encoding of a set value or sequence value shall not include an encoding for any component value which is equal to its default value
authorityKeyIdentifier	RFC 5280 - 5.2.1	Conforming CRL issuers MUST use the key identifier method,

Component / Extension	Reference	Relevant Excerpts
		and MUST include this extension in all CRLs issued.
issuerAlternativeName	RFC 5280 - 5.2.2	
cRLNumber	RFC 5280 - 5.2.3	CRL issuers conforming to this profile MUST include this extension in all CRLs and MUST mark this extension as non-critical. CRLNumber ::= INTEGER (0..MAX) Given the requirements above, CRL numbers can be expected to contain long integers. CRL verifiers MUST be able to handle CRLNumber values up to 20 octets. Conforming CRL issuers MUST NOT use CRLNumber values longer than 20 octets.
	X.690 – 8.3.2	If the contents octets of an integer value encoding consist of more than one octet, then the bits of the first octet and bit 8 of the second octet: a) shall not all be ones; and b) shall not all be zero. NOTE – These rules ensure that an integer value is always encoded in the smallest possible number of octets.
	X.690 – 8.3.3	The contents octets shall be a two's complement binary number equal to the integer value, and consisting of bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet in turn up to and including the last octet of the contents octets.
deltaCRLIndicator	RFC 5280 - 5.2.4	
issuingDistributionPoint	RFC 5280 - 5.2.5	
freshestCRL	RFC 5280 - 5.2.6	
reasonCode	RFC 5280 - 5.3.1	
holdInstructionCode	RFC 5280 - 5.3.2	
invalidityDate	RFC 5280 - 5.3.3	
certificateIssuer	RFC 5280 - 5.3.4	