ICAO

Doc 9303

Machine Readable Travel Documents

Seventh Edition, 2015

Part 12: Public Key Infrastructure for MRTDs

Approved by the Secretary General and published under his authority

INTERNATIONAL CIVIL AVIATION ORGANIZATION

ICAO

Doc 9303

Machine Readable Travel Documents

Seventh Edition, 2015

Part 12: Public Key Infrastructure for MRTDs

Approved by the Secretary General and published under his authority

INTERNATIONAL CIVIL AVIATION ORGANIZATION

Published in separate English, Arabic, Chinese, French, Russian
and Spanish editions by the
INTERNATIONAL CIVIL AVIATION ORGANIZATION
999 Robert-Bourassa Boulevard, Montréal, Quebec, Canada H3C 5H7

Downloads and additional information are available at www.icao.int/security/mrtd

Doc 9303, *Machine Readable Travel Documents*
Part 12 — *Public Key Infrastructure for MRTDs*
ISBN 978-92-9249-800-9

AMENDMENTS

Amendments are announced in the supplements to the *Products and Services Catalogue;* the Catalogue and its supplements are available on the ICAO website at www.icao.int. The space below is provided to keep a record of such amendments.

RECORD OF AMENDMENTS AND CORRIGENDA

AMENDMENTS		
No.	Date	Entered by

CORRIGENDA		
No.	Date	Entered by

TABLE OF CONTENTS

Page

1. SCOPE

The Seventh Edition of Doc 9303 represents a restructuring of the ICAO specifications for Machine Readable Travel Documents. Without incorporating substantial modifications to the specifications, in this new edition Doc 9303 has been reformatted into a set of specifications for Size 1 Machine Readable Official Travel Documents (TD1), Size 2 Machine Readable Official Travel Documents (TD2), and Size 3 Machine Readable Travel Documents (TD3) size documents, as well as visas. This set of specifications consists of various separate documents in which general (applicable to all MRTDs) as well as MRTD form factor specific specifications are grouped.

This Part 12 of Doc 9303 is based on the Sixth Edition of Doc 9303, Part 1, *Machine Readable Passports*, Volume 2 *Specifications for Electronically Enabled Passports with Biometric Identification Capability* (2006) and the Third Edition of Doc 9303, Part 3, *Machine Readable Official Travel Documents*, Volume 2, *Specifications for Electronically Enabled MRTDs with Biometric Identification Capability* (2008).

Part 12 defines the Public Key Infrastructure (PKI) for the eMRTD application. Requirements for issuing States or organizations are specified, including operation of a Certification Authority (CA) that issues certificates and CRLs. Requirements for receiving States and their Inspection Systems validating those certificates and CRLs are also specified.

Doc 9303-12 should be read in conjunction with:

- Doc 9303-10 — *Logical Data Structure (LDS) for Storage of Biometrics and other data in the Contactless Integrated Circuit (IC)*; and
- Doc 9303-11 — *Security Mechanisms for MRTDs*.

2. OVERVIEW OF THE PUBLIC KEY INFRASTRUCTURE

The eMRTD Public Key Infrastructure (PKI) enables the creation and subsequent verification of digital signatures on eMRTD objects, including the Document Security Object (SO_D) to ensure the signed data is authentic and has not been modified. Revocation of a certificate, failure of the certification path validation procedure or failure of digital signature verification does not on its own cause an eMRTD to be considered invalid. Such a failure means that the electronic verification of the integrity and authenticity of the LDS data has failed and other non-electronic mechanisms could then be used to make that determination as part of the overall inspection of the eMRTD.

The eMRTD PKI is much simpler than more generic multi-application PKIs such as the Internet PKI defined in [RFC 5280]. In the eMRTD PKI, each issuing State/Authority establishes a single Certification Authority (CA) that issues all certificates directly to end-entities, including Document Signers. These CAs are referred to as Country Signing Certification Authorities (CSCAs). There are no other CAs in the infrastructure. Receiving States establish trust directly in the keys/certificates of each issuing State or organization's CSCA.

The eMRTD PKI is based on generic PKI standards including [X.509] and [RFC 5280]. Those base PKI standards define a large set of optional features and complex trust relationships among CAs that are not relevant to the eMRTD application. A profile of those standards, tailored to the eMRTD application, is specified in this Part of Doc 9303. Some of the unique aspects of the eMRTD application include:

- there is precisely one CSCA per issuing State;
- certification paths include precisely one certificate (e.g. Document Signer);

- signature verification must be possible 5-10 years after creation;
- CSCA name change is supported; and
- CSCA Link certificates are not processed as intermediate certificates in a certification path.

For the most part, the eMRTD PKI infrastructure is compliant with [RFC 5280]. However, the fact that CSCAs can undergo a name change imposes unique requirements on the eMRTD PKI that are incompatible with some of the CRL validation procedures defined in [RFC 5280]. These differences have been kept to a minimum and are clearly identified.

This Part 12 of Doc 9303 specifies the eMRTD PKI profile including:

- roles and responsibilities of entities in the infrastructure;
- cryptographic algorithms and key management;
- certificate and CRL content;
- certificate and CRL distribution mechanisms; and
- certification path validation.

3. ROLES AND RESPONSIBILITIES

The authenticity and integrity of data stored on eMRTDs is protected by Passive Authentication. This security mechanism is based on digital signatures and consists of the following PKI entities:

- **Country Signing CA (CSCA):** Each issuing State/Authority establishes a single CSCA as its national trust point in the context of eMRTDs. The CSCA issues public key certificates for one or more (national) Document Signers and optionally for other end-entities such as Master List Signers and Deviation List Signers. The CSCA also issues periodic Certificate Revocation Lists (CRL) indicating whether any of the issued certificates have been revoked.
- **Document Signers (DS):** A Document Signer digitally signs data to be stored on eMRTDs; this signature is stored on the eMRTD in a Document Security Object.
- **Inspection Systems (IS):** An Inspection System verifies the digital signature, including certification path validation to verify the authenticity and integrity of the electronic data stored on the eMRTD as part of Passive Authentication.
- **Master List Signers:** A Master List Signer is an optional entity that digitally signs a list of CSCA certificates (domestic and foreign) in support of the bilateral distribution mechanism for CSCA certificates.
- **Document List Signers** are defined in Doc 9303-3.

The secure facilities to generate key pairs SHALL be under the control of the issuing State or organization. Each key pair includes a “private” key and a “public” key. The private keys and associated systems or facilities SHALL be well protected from any outside or unauthorized access through inherent design and hardware security facilities.

While the CSCA certificate remains relatively static, a large number of Document Signer certificates will be created over time.

The CSCA of each issuing State or organization acts as the trust point for the receiving State. The issuing State or organization distributes its own CSCA public key to receiving States in the form of a certificate. The receiving State establishes that this certificate (and certified key) are "trusted" through out-of-band means, and stores a "Trust Anchor" for that trusted key/certificate. These CSCA certificates SHALL be self-signed certificates issued directly by the CSCA. CSCA certificates MUST NOT be subordinate or cross certificates in a larger PKI infrastructure. CSCA self-issued link certificates may also be issued to help the receiving State in establishing trust in a new CSCA key/certificate following a key-rollover.

Note.— In some States there is a requirement that a centralized Controller of Certification Authority (CCA) be the supreme authority to publish self-signed certificates for all applications. In these cases, a possible solution is for the CSCA to create a self-signed certificate (satisfying the ICAO Doc 9303 requirements) and have that certificate countersigned by the CCA (satisfying the State's own CCA requirement). However, these countersigned certificates are not part of the eMRTD PKI and would not be distributed to receiving States.

3.1 Country Signing Certification Authority

It is RECOMMENDED that CSCA key pairs (KPu_{CSCA}, KPr_{CSCA}) be generated and stored in a highly protected, off-line CA infrastructure.

The CSCA private key (KPr_{CSCA}) is used to sign Document Signer certificates (C_{DS}), other certificates and CRLs.

Country Signing Certification Authority certificates (C_{CSCA}) are used to validate Document Signer certificates, Master List Signer certificates, Deviation List Signer certificates, CRLs and other certificates issued by the CSCA.

All certificates and CRLs MUST comply with the profiles specified in Section 7 and MUST be distributed using the distribution mechanisms as specified in Section 5.

For PKD participants, each CSCA certificate (C_{CSCA}) MUST also be forwarded to the PKD (for the purpose of validation of Document Signer certificates (C_{DS})).

CRLs MUST be issued on a periodic basis as specified in Section 4.

3.2 Document Signer

It is RECOMMENDED that Document Signer key pairs (KPu_{DS}, KPr_{DS}) be generated and stored in a highly protected infrastructure.

The Document Signer private key (KPr_{DS}) is used to sign Document Security Objects (SO_D).

Document Signer certificates (C_{DS}) are used to validate Document Security Objects (SO_D).

Each Document Signer certificate (C_{DS}) MUST comply with the certificate profile defined in Section 7 and MUST be stored in the contactless IC of each eMRTD that was signed with the corresponding DS private key (see Doc 9303-10 for details). This ensures that the receiving State has access to the Document Signer certificate relevant to each eMRTD.

Document Signer certificates of PKD participants should also be forwarded to ICAO for publication in the ICAO Public Key Directory (PKD).

3.3 Inspection System

Inspection Systems perform Passive Authentication to ensure the integrity and authenticity of the data stored on the eMRTD contactless IC. As part of that process, Inspection Systems MUST perform certification path validation as indicated in Section 6.

3.4 Master List Signer

The Master List Signer private key is used to sign CSCA Master Lists.

Master List Signer certificates are used to validate CSCA Master Lists.

3.5 Deviation List Signer

The Deviation List Signer private key is used to sign Deviation Lists.

Deviation List Signer certificates are used to validate Deviation Lists.

4. KEY MANAGEMENT

Issuing States or organizations SHALL have at least two key pair types:

- Country Signing CA key pair; and
- Document Signer key pair.

Issuing States or organizations MAY have additional key pair types:

- Master List Signer key pair; and
- Deviation List Signer key pair.

The Country Signing CA, Document Signer, Master List Signer, and Deviation List Signer public keys are issued using [X.509] certificates. The public keys contained in CSCA certificates are used to verify the CSCA signature on issued certificates (Document Signer, Master List Signer, Deviation List Signer and CSCA) and on issued CRLs. The public keys contained in Document Signer certificates are used to verify digital signatures created with the corresponding private key by the subject Document Signer on Document Security Objects (SO_D). The public keys contained in Master List Signer certificates are used to verify the digital signature on Master Lists. The public keys contained in Deviation List Signer certificates are used to verify the digital signature on Deviation Lists (defined in Doc 9303-3).

For Master List Signer, Deviation List Signer and Communications keys and certificates, the private key lifetime and the certificate validity period are left to the discretion of the issuing State or organization.

Both the CSCA certificates and Document Signer certificates are associated with a private key usage and a public key validity period as outlined in Table 1.

Table 1. Key Usage and Validity

	Use of Private Key	***Public Key Validity (assuming 10-year valid passports)***
Country Signing CA	3-5 years	13-15 years
Document Signer	Up to 3 months[1]	approx. 10 years
Master List Signer	Discretion of issuing State or organization	Discretion of issuing State or organization
Deviation List Signer	Discretion of issuing State or organization	Discretion of issuing State or organization
Communication	Discretion of issuing State or organization	Discretion of issuing State or organization

4.1 Document Signer Keys and certificates

The usage period of a Document Signer private key is much shorter than the validity period of the DS certificate for the corresponding public key.

4.1.1 Document Signer Public Key validity

The lifetime, i.e. the certificate validity period, of the Document Signer public key is determined by concatenating the following two periods:

- the length of time the corresponding private key will be used to issue eMRTDs, with;
- the longest validity period of any eMRTD issued under that key[2].

The Document Signer certificate (C_{DS}) SHALL be valid for this total period to enable the authenticity of eMRTDs to be verified. However the corresponding private key SHOULD only be used to issue documents for a limited period; once the last document it was used to issue has expired, the public key is no longer required.

4.1.2 Document Signer Private Key issuing period

When deploying their systems, issuing States or organizations may wish to take into account the number of documents that will be signed by any one individual Document Signer private key.

An issuing State or organization may deploy one or more Document Signers, each with its own unique key pair, that are active at any given time.

1 Note the corresponding `privateKeyUsage` extension in DS certificate might be slightly longer to allow for overlap or production requirements.

2 Some issuing States or organizations may issue eMRTDs before they become valid, for instance on a change of name upon marriage. In these situations, the “longest validity period of any eMRTD” includes the actual validity of the eMRTD (e.g. 10 years) plus the maximum time between when the eMRTD is issued and the time it becomes valid.

In order to minimize business continuity costs in the event of a Document Signer certificate being revoked, an issuing State or organization that issues a large number of eMRTDs per day may wish to:

- use a very short private key usage period; and/or
- deploy several concurrent Document Signers that are active at the same time, each with its own unique private key and public key certificate.

An issuing State or organization that issues a small number of eMRTDs per day may choose to deploy a single Document Signer and may also be comfortable with a slightly longer private key usage period.

Regardless of the number of eMRTDs issued per day, or number of Document Signers active at the same time, it is RECOMMENDED that the maximum period any Document Signer private key is used to sign eMRTDs be three months.

Once the last document signed with a given private key has been produced, it is RECOMMENDED that issuing States or organizations erase the private key in an auditable and accountable manner.

4.2 CSCA Keys and Certificates

The usage period of a CSCA private key is much shorter than the validity period of the CSCA certificate for the corresponding public key.

4.2.1 Country Signing CA Public Key validity

The lifetime, i.e. the certificate validity, of the CSCA public key is determined by concatenating the following periods:

- the length of time the corresponding CSCA private key will be used to sign Document Signer certificates (C_{DS}); and,
- the key lifetime of Document Signer public key certificates (See 4.1.1).

4.2.2 Country Signing CA Private Key issuing period

The usage period for the CSCA private key to sign certificates and CRLs is a delicate balance among the following factors:

- In the unlikely event of an issuing State or organization Country Signing Private CA Key being compromised, then the validity of all eMRTDs issued using Document Signer Keys whose certificates were signed by the compromised CSCA private key is called into doubt. Consequently issuing States or organizations MAY wish to keep the issuing period quite short;
- Keeping the issuing period very short, however, leads to having a very large number of CSCA public keys valid at any one time. This can lead to more complex certificate management within the border processing systems.

It is therefore RECOMMENDED that an issuing State or organization’s CSCA key pair be replaced every three to five years.

4.2.3 Country Signing CA Re-key

CSCA keys provide the trust points in the whole system and without these the system would collapse. Therefore issuing States or organizations SHOULD plan the replacement of their CSCA key pair carefully. Once the issuance period for the initial CSCA private signing key has elapsed, an issuing State or organization will always have at least two CSCA certificates (C_{CSCA}) valid at any one time.

Issuing States or organizations MUST notify receiving States that a CSCA key rollover is planned. This notification MUST be provided 90 days in advance of the key rollover. Once the key rollover has occurred the new CSCA certificate (certifying the new CSCA public key) is distributed to receiving States.

If the CSCA certificate is a new self-signed certificate, authentication of that certificate should be done using an out-of-band method.

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. These CSCA Link certificates need not be verified using an out-of-band method as the signature on the CSCA Link certificate is verified using an already trusted public key for that CSCA. Master Lists can also be used to distribute CSCA Link and CSCA self-signed root certificates.

Issuing States or organizations should refrain from using their new CSCA private key for the first two days after the CSCA key rollover, to ensure the corresponding new CSCA public key certificate has been distributed successfully.

Issuing States or organizations MUST use the newest CSCA private key for signing certificates, including Document Signer certificates, and for signing CRLs.

4.3 Certificate Revocation

Issuing States or organizations may need to revoke certificates in case of an incident (like a key compromise).

All CSCAs MUST produce periodic revocation information in the form of a Certificate Revocation List (CRL).

CSCAs MUST issue at least one CRL every 90 days, even if no certificates have been revoked since the previous CRL was issued. CRLs MAY be issued more frequently than every 90 days but not more frequently than every 48 hours.

If a certificate is revoked, a CRL indicating that revocation MUST be distributed within 48 hours.

Only certificates can be revoked, not Document Security Objects. The use of CRLs is limited to notifications of revoked certificates that had been issued by the CSCA that issued the CRL (including revocation notices for CSCA certificates, DS certificates, Master List Signer certificates, Deviation List Signer certificates and any other certificate types issued by that CA).

Partitioned CRLs are not used in the eMRTD application. All certificates revoked by a CSCA, including DS certificates, CSCA certificates, Master List Signer certificates and Deviation List Signer certificates are listed on the same CRL. Although the CRL is always signed with the newest (current) CSCA private signing key, the CRL includes revocation notices for certificates signed with that same private key as well as certificates signed with earlier CSCA private signing keys.

4.3.1 Revocation of CSCA Certificates

Revocation of a CSCA certificate is both extreme and difficult. Upon informing a receiving State that a CSCA certificate has been revoked, all other certificates signed using the corresponding CSCA private key are effectively revoked.

Where a CSCA Link certificate has been signed using an old CSCA private key to certify a new CSCA public key (see "Country Signing Re-key" in 4.2), revoking the old CSCA certificate SHALL also revoke the new CSCA certificate.

If a CSCA certificate needs to be revoked, the CSCA may issue a CRL signed with the private key that corresponds to the public key being revoked, as this is the only key users of the CRL will be able to verify at that time. The CSCA public key should be considered valid only for the purpose of verifying that CRL signature. Once a CRL user has verified the CRL signature, the CSCA private signing key is considered compromised and the certificate revoked for all future verifications.

To issue new documents, the issuing State or organization MUST revert to bootstrapping its authentication process from the beginning, by issuing a new CSCA Root certificate, distributing that certificate to receiving States, and supporting out-of-band confirmation that the certificate received by each receiving State is in fact the current authentic CSCA certificate.

4.3.2 Revocation of other Certificates

When an issuing State or organization wishes to revoke a Document Signer, Master List Signer, Deviation List Signer or communication certificate, it does not need to wait until the `nextUpdate` period in the current CRL is due to issue a new CRL. It is RECOMMENDED that a new CRL be issued within a 48-hour period of revocation notification.

4.4 Cryptographic Algorithms

An issuing State or organization MUST support the same algorithm for use in their CSCA and Document Signing keys, although different key sizes may be required depending on the algorithm selected.

Issuing States or organizations SHALL choose appropriate key lengths offering protection against attacks. Suitable cryptographic catalogues SHOULD be taken into account.

Receiving States MUST support all algorithms at points where they wish to validate the signature on eMRTDs.

For use in their CSCA, Document Signing keys and, where applicable, Document Security Objects, issuing States or organizations SHALL support one of the algorithms below.

4.4.1 RSA

Those issuing States or organizations implementing the RSA algorithm for signature generation and verification of certificates and the Document Security Object (SO_D) SHALL use [RFC 4055]. [RFC 4055] specifies two signature mechanisms, RSASSA-PSS and RSASSA-PKCS1_v15. It is RECOMMENDED that issuing States or organizations generate signatures according to RSASSA-PSS, but receiving States MUST also be prepared to verify signatures according to RSASSA-PKCS1_v15.

4.4.2 Digital Signature Algorithm (DSA)

Those issuing States or organizations implementing DSA for signature generation or verification SHALL use [FIPS 186-4].

4.4.3 Elliptic Curve DSA (ECDSA)

Those issuing States or organizations implementing ECDSA for signature generation or verification SHALL use [X9.62] or [ISO/IEC 15946]. The elliptic curve domain parameters used to generate the ECDSA key pair MUST be described explicitly in the parameters of the public key, i.e. parameters MUST be of type ECParameters (no named curves, no implicit parameters) and MUST include the optional co-factor. ECPoints MUST be in uncompressed format.

It is RECOMMENDED that the guideline [TR 03111] be followed.

4.4.4 Hashing Algorithms

SHA-224, SHA-256, SHA-384 and SHA-512, are the only permitted hashing algorithms. See [FIPS 180-2].

5. DISTRIBUTION MECHANISMS

PKI objects need to be distributed to the receiving States. A number of different distribution mechanisms are used, depending on the type of object and operational requirements. It is important to note that distribution of these objects does NOT establish trust in those objects, or the private/public keys associated with them. Mechanisms for establishing trust are specified in Section 6.

The objects that need to be distributed from issuing States or organizations to receiving States include:

- CSCA certificates;
- Document Signer certificates;
- CRLs (null and non-null);
- Master List Signer certificates; Master Lists; and
- Deviation List Signer certificates.

The distribution mechanisms used in the eMRTD PKI include:

- PKD;
- Bilateral exchange;
- Master Lists;
- Deviation Lists; and
- eMRTD contactless IC.

A primary and secondary (where relevant) distribution mechanism is specified for each object as outlined in Table 2.

Table 2. Primary and Secondary Distribution

	CSCA Certificates	***Document Signer Certificates***	***CRLs (Null & Non-null)***	***Master List Signer Certificates***	***Master Lists***	***Deviation List Signer Certificates***
Primary	Bilateral	eMRTD contactless IC	Bilateral	Master Lists	PKD/ Bilateral	Deviation Lists
Secondary	Master Lists	PKD	PKD			

Operationally, receiving States are not obliged to use both the primary and secondary source. In the daily operation of an Inspection System, it is at the inspecting authority's discretion whether to use the primary or the secondary source. If the authority of the receiving State uses the secondary source for a certificate or CRL in its daily operations, it should be prepared to support the primary source as well.

Issuing States or organizations need to plan their key pair rollover strategies for both CSCA keys and Document Signer keys in order to enable propagation of certificates and CRLs into receiving States' border control systems in a timely manner. Ideally propagation will occur within 48 hours, but some receiving States may have remote and poorly connected border outposts to which it may take more time for certificates and CRLs to propagate out. Receiving States SHOULD make every effort to distribute these certificates and CRLs to all border stations within 48 hours.

Issuing States or organizations should expect that CSCA certificates (C_{CSCA}) will be propagated by receiving States within 48 hours.

Issuing States or organizations ensure the timely propagation of Document Signer certificates (C_{DS}) by including the Document Signer certificate (C_{DS}) within the Document Security Object (SO_D). They should expect that Document Signer certificates (C_{DS}) published in the PKD will also be propagated to border stations within 48 hours.

Receiving States SHOULD make every attempt whether electronically or by other means to act upon CRLs, including those CRLs issued under exceptional circumstances.

Timely propagation of Master List Signer certificates (C_{DS}) is ensured by including them within each Master List.

5.1 PKD Distribution Mechanism

ICAO provides a Public Key Directory (PKD) service. This service SHALL accept PKI objects, including certificates, CRLs and Master Lists, from PKD participants, store them in a directory, and make them accessible to all receiving States.

CSCA certificates (C_{CSCA}) are not stored individually as part of the ICAO PKD service. However, they may be present in the PKD if they are contained on Master Lists.

Each Document Signer certificate (C_{DS}) remains in the PKD until its certificate validity period has expired, regardless of whether or not the corresponding private key is still in use.

Certificates, CRLs and Master Lists stored in the PKD by all PKD participants SHALL be made available to all parties (including non-PKD participants) that need this information for validating the authenticity and integrity of digitally stored eMRTD data.

5.1.1 PKD upload

Only PKD participants MAY upload certificates, CRLs and Master Lists to the PKD. All certificates and CRLs MUST comply with the profiles in Section 7. All Master Lists MUST comply with the specification in Section 8.

The PKD consists of a “Write Directory” and a “Read Directory”. PKD participants SHALL use the Lightweight Directory Access Protocol (LDAP) to upload their objects to the Write Directory. Once the digital signature has been verified on an object, and other due diligence checks completed, the object is published in the Read Directory.

5.1.2 PKD download

Read access to all certificates, CRLs and Master Lists published in the PKD SHALL be available to PKD participants and non-participants. Access control SHALL NOT be implemented for PKD read access.

It is the receiving State’s responsibility to distribute objects downloaded from the PKD to its Inspection Systems and to maintain a current CRL cache along with the certificates necessary to verify the signatures on eMRTD data.

5.2 Bilateral Exchange Distribution Mechanism

For CRLs and CSCA certificates (C_{CSCA}), the primary distribution channel is bilateral exchange between issuing States or organizations and receiving States. Bilateral exchange can also be used to distribute Master Lists.

The specific technology used for that bilateral exchange may vary depending on the policies of each issuing State or organization that has a need to distribute its certificates, CRLs and Master Lists, as well as the policies of each receiving State that needs access to those objects. Some examples of technologies that may be used in bilateral exchange include:

- diplomatic courier/pouch;
- email exchange;
- download from a website associated with the issuing CSCA; and
- download from an LDAP server associated with the issuing CSCA.

This is not an exhaustive list and other technologies may also be used.

5.3 Master List Distribution Mechanism

Master Lists are a supporting technology for the bilateral distribution scheme. As such, distribution of CSCA certificates via Master Lists is a subset of the bilateral distribution scheme.

A Master List is a digitally signed list of the CSCA certificates that are “trusted” by the receiving State that issued the Master List. CSCA self-signed Root certificates and CSCA Link certificates may be included in a Master List. The

structure and format of a Master List is defined in Section 8. Publication of a Master List enables other receiving States to obtain a set of CSCA certificates from a single source (the Master List issuer) rather than establish a direct bilateral exchange agreement with each of the issuing authorities or organizations represented on that list.

A Master List Signer is authorized by a CSCA to compile, digitally sign, and issue Master Lists. Master Lists MUST NOT be signed and issued directly by a CSCA itself. Master List Signer certificates MUST comply with the certificate profile defined in Section 7.

Before issuing a Master List the issuing Master List Signer SHOULD extensively validate the CSCA certificates to be countersigned, including ensuring that the certificates indeed belong to the identified CSCAs. The procedures used for this out-of-band validation SHOULD be reflected in the published certificate policies of the CSCA that issued the Master List Signer certificate.

Each Master List MUST include the Master List Signer's certificate that will be used to verify the signature on that Master List as well as the CSCA certificates of the CSCA that issued that Master List Signer certificate.

If new CSCA certificates have been received by a receiving State, and its validation procedures have been completed, it is RECOMMENDED that a new Master List be compiled and issued.

Use of a Master List does enable more efficient distribution of CSCA certificates for some receiving States. However a receiving State making use of Master Lists MUST still determine its own policies for establishing trust in the certificates contained on that list (see Section 6 for details).

6. PKI TRUST AND VALIDATION

In the eMRTD PKI environment, the Inspection Systems in receiving States act in the role of PKI relying parties. Successful verification of the digital signature on the Document Security Object of an eMRTD ensures the authenticity and integrity of the data stored on the contactless IC of that eMRTD. That signature verification process requires that the relying party establish that the Document Signer public key used to verify the signature is itself "trusted".

The various distribution mechanisms defined in Section 5 allow receiving States to gain access to the certificates and CRLs that they need to verify digital signatures in question. However, these distribution schemes do not establish trust in those certificates, CRLs or the public keys that will be used to verify signatures on those certificates and CRLs.

The public keys contained in CSCA certificates (C_{CSCA}) are used to verify the digital signature on certificates (including Document Signer Master List Signer and Deviation List Signer certificates) and CRLs. Therefore, to accept an eMRTD from another issuing State, the receiving State MUST already have placed into some form of trust store, accessible by their border control system, a trusted copy of the issuing State or organization CSCA certificate (C_{CSCA}), or other form of Trust Anchor information for that CSCA public key as derived from the certificate.

It is a receiving State's responsibility to establish trust in the CSCA certificates (C_{CSCA}) and store the certificates (or information from the certificates) as Trust Anchors, in a secure way for use by their border inspection systems.

6.1 Trust Anchor Management

As specified in [RFC 5280] a Trust Anchor must be established that can be used to anchor the validation procedure for a given Document Signer, Master List Signer, Deviation List Signer or other type of certificate.

Each Trust Anchor is comprised of a trusted public key and associated metadata. Trust Anchors MUST include, at a minimum:

- the trusted public key and any associated key parameters;
- the public key algorithm;
- the name of the key owner; and
- the value of the `SubjectAltName` extension of the CSCA certificate containing the ICAO assigned three-letter code of the issuing authority or organization. Although this is not used in the certification path or CRL validation procedures, it is used in Passive Authentication defined in Doc 9303-11.

In the eMRTD application, a separate Trust Anchor is established for each public key of a given CSCA. For the initial public key obtained from a CSCA, trust MUST be established through an out-of-band mechanism. For example, if a CSCA certificate was downloaded from a server associated with the CSCA, out-of-band communication (e.g. phone or email) could be used to verify that the downloaded certificate is in fact the authentic certificate for that CSCA. Also, the relying party might analyse the policies, procedures and practices of the issuing CSCA to determine whether they are secure enough to satisfy the local requirements for use of certificates. Once an initial Trust Anchor is established for a given CSCA, the process could be simplified for subsequent keys for that same CSCA. If the CSCA issues a CSCA Link certificate, then out-of-band communication with the CSCA to verify the authenticity of the new certificate could be skipped because the already trusted public key for that same CSCA is used to verify the signature on that CSCA Link certificate.

Trust Anchor information may be stored as a trusted copy of the CSCA certificate itself, or in some other trusted format.

Because signatures on certificates issued by CSCAs need to be verifiable long after that CSCA has updated its key pair, a receiving State will typically have more than one Trust Anchor for the same CSCA at any one time. If a CSCA has undergone a name change, some of these Trust Anchors will contain the old CSCA name and others will contain the new name.

6.2 Certificate/CRL Validation and Revocation Checking

As part of the process of verifying the authenticity and integrity of data objects in the eMRTD application (e.g. Document Security Objects, Master Lists, Deviation Lists, etc.), a Receiving State:

- validates the certificate used to verify the signature on the data object (e.g. Document Signer Certificate, Master List Signer certificate, Deviation List Signer certificate);
- validates the CRL that is used to check the revocation status of the certificate in question; and
- processes the CRL to verify the revocation status of the certificate in question.

Sample algorithms for these processes are available, such as those specified in [RFC 5280]. Receiving States need not implement the specific algorithm defined in RFC 5280, but MUST provide functionality equivalent to the external behaviour resulting from this procedure. Any algorithm may be used by a particular implementation as long as it derives the correct result.

Appendix D provides guidance for receiving States that choose to base their algorithm on that specified in [RFC 5280].

7. CERTIFICATE AND CRL PROFILES

Issuing States or organizations MUST issue certificates and CRLs that conform to the profiles specified below. All certificates and CRLs MUST be produced in Distinguished Encoding Rule (DER) format to preserve the integrity of the signatures within them. The profiles for CSCA and DS certificates that were included in the sixth edition of this specification differ in some areas from the current profiles. Inspection Systems MUST be capable of handling certificates that were issued in accordance with those earlier profiles (see Appendix C) as well as the current profiles.

These profiles are based on the requirement that each issuing State or organization or entity SHALL create a single CSCA for the purpose of signing all Doc 9303 compliant eMRTDs.

Certificate profiles are defined in Section 7.1 for the following certificate types:

- Country Signing CA;
- Document Signer;
- CSCA Master List Signer;
- Deviation List Signer; and
- Communications – even though it is not strictly needed today. This is a future proofing step. These certificates may be used for access to the PKD or for LDAP/EMAIL/HTTP communications between States. It is recommended that these certificates be issued by the CSCA.

The Country Signing CA, Document Signer and CSCA Master List Signer objects are defined in Section 3. The Deviation List Signer object is defined in Doc 9303-3.

The CRL profile is defined in Section 7.2.

The profiles use the following terminology for presence requirements of each of the components/extensions:

m mandatory — the field MUST be present;
x do not use — the field MUST NOT be present;
o optional — the field MAY be present.

The profiles use the following terminology for criticality requirements of extensions that may/must be included:

c critical — receiving applications MUST be able to process this extension;
nc 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 (e.g. RFC 5280). For convenience, the relevant text from the base profile that covers the specific requirement is duplicated in a table in Appendix B.

7.1 Certificate Profiles

Table 3 defines the certificate profile requirements for the fields of the certificate body. Table 4 defines the requirements for certificate extensions.

Table 3. Certificate Fields Profile

Certificate Component	*Presence*	*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	`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 See 7.1.1 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

Certificate Component	*Presence*	*Comments*
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 7.1.1 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

Table 4. Certificate Extensions Profile

Extension name	***CSCA Self-Signed Root***		***CSCA Link***		***Document Signer***		***Master List Signer and Deviation 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 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		

Extension name	*CSCA Self-Signed Root*		*CSCA Link*		*Document Signer*		*Master List Signer and Deviation List Signer*		*Communication*		*Comments*
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 7.1.2
IssuerAltName	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	See 7.1.2
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 7.1.3
CRLDistributionPoints	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**o**	**nc**	
distributionPoint	m		m		m		m		m		MUST be ldap, http or https See 7.1.4
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 2
privateInternetExtensions	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	See Note 3
NameChange	**o**	**nc**	**o**	**nc**	**x**		**x**		**x**		See 7.1.5
DocumentType	**x**		**x**		**m**	**nc**	**x**		**x**		See 7.1.6
Netscape Certificate Type	**x**		**x**		**x**		**x**		**x**		See Note 4
other private extensions	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	

Note 1.— 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.— The freshest CRL extension is used to point to a delta CRL. Delta CRLs are not supported in the eMRTD PKI. Therefore this extension is prohibited.

Note 3.— There are two Private Internet Extensions (Authority Information Access and Subject Information Access) defined in RFC 5280 that are used to point to information about the issuer or subject of a certificate. These extensions 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 4.— 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.

7.1.1 Issuer and Subject Field requirements

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]
- `commonName`. MUST be present.

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

7.1.2 Issuer and Subject Alternative Name requirements

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.

In the eMRTD application, alternative names serve the following two functions.

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

- `rfc822Name`;
- `dNSName`; or
- `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; and
- if this country code does not uniquely define the issuing State or organization, the attribute `stateOrProvinceName` SHALL be used to indicate the ICAO assigned three-letter code for the issuing State or organization.
- Other attributes are not permitted.

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.

7.1.3 Extended key usage extension requirements

The Object Identifier (OID) that must be included in the `extendedKeyUsage` extension for Master List Signer certificates is `2.23.136.1.1.3`.

The Object Identifier (OID) that must be included in the `extendedKeyUsage` extension for Deviation List Signer certificates is `2.23.136.1.1.8`.

For communication certificates the value of this extension depends on the communication protocol used (see RFC 5280, section 4.2.1.12).

7.1.4 CRL distribution points extension requirements

CSCAs may publish their CRL in several places including the PKD, their own website, etc.

For CRLs that are published in locations other than the PKD (e.g. website or local LDAP server), the values that are to be included in this extension are under the control of the CSCA issuing the certificates and the CRL in question.

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 three-letter code). If this country code does not uniquely identify the issuing State or organization, the entry will be created 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 its 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 its website as a source) even if it also submits its 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

7.1.5 Name change extension

When a CSCA key rollover occurs, a certificate MUST be issued that links the old public key to the new public 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 Distinguished Name (DN) unnecessarily as there is an adverse impact on relying parties (they must retain both the old and new names as valid CSCAs for the same issuing State or organization until all eMRPs 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` field contains the old name and the `subject` field contains the new name. 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. 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}
```

7.1.6 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. 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}
```

7.2 CRL Profile

Table 5 defines the CRL profile requirements for the fields of the CRL body. Table 6 defines the CRL profile requirements for CRL and CRL Entry extensions.

Table 5. CRL Fields Profile

Certificate List Component	***CSCA 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

Certificate List Component	***CSCA CRL***	***Comments***
`issuer`	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
`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
`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

Table 6. CRL and CRL Entry Extensions Profile

Extension Name	*CSCA 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. Given that ICAO Doc 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.

8. CSCA MASTER LIST STRUCTURE

Master Lists are implemented as instances of the `ContentInfo` Type, as specified in [RFC 5652]. The `ContentInfo` MUST contain a single instance of the `SignedData` Type as profiled below. No other data types are included in the `ContentInfo`. All Master Lists MUST be produced in DER format to preserve the integrity of the signatures within them.

8.1 SignedData Type

The processing rules in [RFC 5652] apply.

The specification of Master List structure uses the following terminology for presence requirements of each field.

- m mandatory — the field MUST be present
- r recommended — the field SHOULD be present
- x do not use — the field MUST NOT be present
- o optional — the field MAY be present.

Table 7. Master List

Value		*Comments*
`SignedData`		
`Version`	m	Value = `v3`
`digestAlgorithms`	m	
`encapContentInfo`	m	
`eContentType`	m	`id-icao-cscaMasterList`
`eContent`	m	The encoded contents of an `cscaMasterList`
`certificates`	m	The Master List Signer certificate MUST be included and the CSCA certificate, which can be used to verify the signature in the `signerInfos` field SHOULD be included.
`crls`	x	
`signerInfos`	m	It is RECOMMENDED that States only provide 1 `signerinfo` within this field.

Value		*Comments*
SignerInfo	m	
version	m	The value of this field is dictated by the sid field. See [RFC 5652] for rules regarding this field
sid	m	
subjectKeyIdentifier	r	It is RECOMMENDED that this field be supported rather than issuerandSerialNumber.
digestAlgorithm	m	The algorithm identifier of the algorithm used to produce the hash value over encapsulatedContent and SignedAttrs. See Note 1.
signedAttrs	m	Additional attributes may be included. However these do not have to be processed by Receiving States except to verify the signature value. signedAttrs MUST include signing time (see [PKCS #9]).
signatureAlgorithm	m	The algorithm identifier of the algorithm used to produce the signature value, and any associated parameters. See Note 1.
signature	m	The result of the signature generation process.
unsignedAttrs	o	Although this field MAY be included, Receiving States may choose to ignore it.

Note 1.— DigestAlgorithmIdentifiers MUST omit "NULL" parameters, while the SignatureAlgorithmIdentifier (as defined in RFC 3447) MUST include NULL as the parameter if no parameters are present, even when using SHA2 Algorithms in accordance with RFC 5754. Implementations MUST accept DigestAlgorithmIdentifiers with both conditions, absent parameters or with NULL parameters.

8.2 ASN.1 Master List Specification

```
CscaMasterList
{ iso-itu-t(2) international-organization(23) icao(136) mrtd(1)
security(1) masterlist(2)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS

 -- Imports from RFC 5280 [PROFILE], Appendix A.1
    Certificate
      FROM PKIX1Explicit88
```

```
          { iso(1) identified-organization(3) dod(6)
            internet(1) security(5) mechanisms(5) pkix(7)
              mod(0) pkix1-explicit(18) };
-- CSCA Master List

CscaMasterListVersion ::= INTEGER {v0(0)}

CscaMasterList ::= SEQUENCE {
  version            CscaMasterListVersion,
  certList           SET OF Certificate }

-- Object Identifiers

id-icao-cscaMasterList OBJECT IDENTIFIER ::=
                                  {id-icao-mrtd-security 2}
id-icao-cscaMasterListSigningKey OBJECT IDENTIFIER ::=
                                  {id-icao-mrtd-security 3}
END
```

9. REFERENCES (NORMATIVE)

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

FIPS 186-4 — FIPS 186-4, Federal Information Processing Standards Publication (FIPS PUB) 186-4, *Digital Signature Standard (DSS)*, July 2013 (Supersedes FIPS PUB 186-3 dated June 2009).

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

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

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

RFC 4055 — RFC 4055, J. Schaad, B. Kaliski, R. Housley, Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, June 2005.

RFC 5652 — RFC 5652, R. Housley, Cryptographic Message Syntax, September 2009.

RFC 5280 — 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.

TR 03111 — BSI TR-03111: Elliptic Curve Cryptography v 2.0, 2012.

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

X.509 ITU-T X.509 | ISO/IEC 9594-8, 2008: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.

X.690 ITU-T X.690 2008: Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

— — — — — — — —

Appendix A to Part 12

LIFETIMES (INFORMATIVE)

The following examples illustrate calculation of private key usage periods and public key certificate validity for various scenarios as described in Section 4.

A.1 EXAMPLE 1

The first example illustrates a scenario where eMRTDs are valid for five years. Because a relatively large number of eMRTDs are issued per day, the policy is to keep private key usage periods and public key certificate validity to a minimum. For this example, the minimum private key usage period for Document Signer certificates is one month.

Item	***Usage/Validity Period***
eMRTD validity	5 years
Document Signer private key usage period	1 month
Document Signer certificate validity	5 years + 1 month
CSCA private key usage period	3 years
CSCA certificate validity	8 years + 1 month

The consequences of this example are that by the time the first CSCA certificate becomes invalid at least 36 Document Signer certificates will have been issued (one corresponding to each private key that has a one-month usage period). In the last few months before the first CSCA certificate becomes invalid, there will be at least two additional CSCA certificates issued (one corresponding to each private key that has a three-year usage period).

A.2 EXAMPLE 2

The second example illustrates a scenario where eMRTDs are valid for ten years. The policy is to keep private key usage periods and public key certificate validity to an average length.

Item	***Usage/Validity Period***
eMRTD validity	10 years
Document Signer private key usage period	2 months
Document Signer certificate validity	10 years + 2 months

Item	*Usage/Validity Period*
CSCA private key usage period	4 years
CSCA certificate validity	14 years + 2 months

The consequences of this example are by the time the first CSCA certificate becomes invalid at least 24 Document Signer certificates will have been issued (one corresponding to each private key that has a two-month usage period). In the last few months before the first CSCA certificate becomes invalid, there will be at least three additional CSCA certificates issued (one corresponding to each private key that has a four-year usage period).

A.3 EXAMPLE 3

The final example illustrates a scenario where eMRTDs are valid for ten years, and the policy is to use the maximum private key usage periods and public key certificate validity.

Item	*Usage/Validity Period*
eMRTD validity	10 years
Document Signer private key usage period	3 months
Document Signer certificate validity	10 years + 3 months
CSCA private key usage period	5 years
CSCA certificate validity	15 years + 3 months

The consequences of this example are by the time the first CSCA certificate becomes invalid at least 20 Document Signer certificates will have been issued (one corresponding to each private key that has a three-month usage period). In the last few months before the first CSCA certificate becomes invalid, there will be at least three additional CSCA certificates issued (one corresponding to each private key that has a five-year usage period).

— — — — — — — —

Appendix B to Part 12

CERTIFICATE AND CRL PROFILE REFERENCE TEXT (INFORMATIVE)

The certificate and CRL profiles defined in Section 7 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 and CRL profiles. 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.

Table 8. 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.

Component / Extension	*Reference*	*Relevant Excerpts*
	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.

Component / Extension	***Reference***	***Relevant Excerpts***
	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`		

Component / Extension	*Reference*	*Relevant Excerpts*
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		

Component / Extension	***Reference***	***Relevant Excerpts***
`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.

Component / Extension	*Reference*	*Relevant Excerpts*
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		

Table 9. 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.

Component / Extension	***Reference***	***Relevant Excerpts***
`issuer`	RFC 5280 – Appendix A.1	`X520countryName ::= PrintableString (SIZE (2))` `X520SerialNumber ::= PrintableString (SIZE 1..ub-serial-number))`
	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 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.
`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.

Component / Extension	*Reference*	*Relevant Excerpts*
	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, 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.

Component / Extension	***Reference***	***Relevant Excerpts***
	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	

— — — — — — — —

Appendix C to Part 12

EARLIER CERTIFICATE PROFILES (INFORMATIVE)

The certificate profiles in this Appendix were specified in the Sixth Edition of ICAO Doc 9303. Although CSCAs MUST issue certificates that comply with the current profiles as specified in Section 7, the earlier profiles are included here for information only as certificates that were issued in compliance with the earlier profiles will be in circulation, and processed by Inspection Systems for several years.

Table 10. Certificate Body

Certificate Component	*Section in RFC 3280*	*Country Signing CA Certificate*	*Document Signer Certificate*	*Comments*
Certificate	4.1.1	m	m	
TBSCertificate	4.1.1.1	m	m	See next part of the table
SignatureAlgorithm	4.1.1.2	m	m	Value inserted here dependent on algorithm selected
SignatureValue	4.1.1.3	m	m	Value inserted here dependent on algorithm selected
TBSCertificate	4.1.2			
version	4.1.2.1	m	m	SHALL be v3
serialNumber	4.1.2.2	m	m	
signature	4.1.2.3	m	m	Value inserted here SHALL match the OID in signatureAlgorithm
issuer	4.1.2.4	m	m	See A1.5
validity	4.1.2.5	m	m	Implementations SHALL specify using UTC time until 2049 from then on using GeneralizedTime
subject	4.1.2.6	m	m	See A1.5
subjectPublicKeyInfo	4.1.2.7	m	m	

Certificate Component	*Section in RFC 3280*	*Country Signing CA Certificate*	*Document Signer Certificate*	*Comments*
issuerUniqueID	4.1.2.8	x	x	
subjectUniqueID	4.1.2.8	x	x	
extensions	4.1.2.9	m	m	See next table on which extensions SHOULD be present

Table 11. Extensions

Extension name	*Paragraph in RFC 3280*	*Country Signing CA Certificate*	*Document Signer Certificate*	*Comments*
AuthorityKeyIdentifier	4.2.1.1	o	m	Mandatory in all certificates except for self-signed CSCA certificates
SubjectKeyIdentifier	4.2.1.2	m	o	
KeyUsage	4.2.1.3	mc	mc	This extension SHALL be marked CRITICAL
PrivateKeyUsagePeriod	4.2.1.4	o	o	This would be the issuing period of the private key
CertificatePolicies	4.2.1.5	o	o	
PolicyMappings	4.2.1.6	x	x	
SubjectAltName	4.2.1.7	x	x	
IssuerAltName	4.2.1.8	x	x	
SubjectDirectoryAttributes	4.2.1.9	x	x	
BasicConstraints	4.2.1.10	mc	x	This extension SHALL be marked CRITICAL
NameConstraints	4.2.1.11	x	x	
PolicyConstraints	4.2.1.12	x	x	
ExtKeyUsage	4.2.1.13	x	x	

Extension name	***Paragraph in RFC 3280***	***Country Signing CA Certificate***	***Document Signer Certificate***	***Comments***
CRLDistributionPoints	4.2.1.14	o	o	If issuing States or organizations choose to use this extension they SHALL include the ICAO PKD as a distribution point. Implementations may also include relative CRL DPs for local purposes; these may be ignored by other receiving States.
InhibitAnyPolicy	4.2.1.15	x	x	
FreshestCRL	4.2.1.16	x	x	
privateInternetExtensions	4.2.2	x	x	
other private extensions	N/A	o	o	If any private extension is included for national purposes then it SHALL NOT be marked. Issuing States or organizations are discouraged from including any private extensions.
AuthorityKeyIdentifier	4.2.1.1			
keyIdentifier		m	m	If this extension is used this field SHALL be supported as a minimum
authorityCertIssuer		o	o	See A1.5
authorityCertSerialNumber		o	o	
SubjectKeyIdentifier	4.2.1.2			
subjectKeyIdentifier		m	m	
KeyUsage	4.2.1.3			
digitalSignature		x	m	
nonRepudiation		x	x	
keyEncipherment		x	x	
dataEncipherment		x	x	
keyAgreement		x	x	
keyCertSign		m	x	

Extension name	*Paragraph in RFC 3280*	*Country Signing CA Certificate*	*Document Signer Certificate*	*Comments*
cRLSign		m	x	
encipherOnly		x	x	
decipherOnly		x	x	
BasicConstraints	4.2.1.10			
cA		m	x	TRUE for CA certificates
PathLenConstraint		m	x	0 for New CSCA certificate, 1 for Linked CSCA certificate
CRLDistributionPoints	4.2.1.14			
distributionPoint		m	x	
reasons		m	x	
cRLIssuer		m	x	
CertificatePolicies	4.2.1.5			
PolicyInformation				
policyIdentifier		m	m	
policyQualifiers		o	o	

— — — — — — — —

Appendix D to Part 12

RFC 5280 VALIDATION COMPATIBILITY (INFORMATIVE)

This Appendix provides guidance to receiving States wishing to use systems that implement the [RFC 5280] certification path and CRL validation algorithms.

The eMRTD PKI trust model is a subset of that covered by the validation procedures defined in [RFC 5280]. Section D.1 identifies the subset of steps from the [RFC 5280] definition that are required for the eMRTD application and provides the necessary inputs and initialization values and processes for certification path validation, CRL validation and revocation checking.

Section D.2 covers the remaining steps from the [RFC 5280] definition that are not relevant to the eMRTD application. The inputs and initialization values for certification path validation and CRL validation are provided. The guidance in this section is for use in situations where the tools implement the full [RFC 5280] algorithms, rather than just the subset described in D.1.

Section D.3 provides guidance to support the extension of [RFC 5280] based CRL processing to cover revocation checking after a CSCA has undergone a name change.

D.1 STEPS RELEVANT TO eMRTD

The eMRTD certification path validation procedure defined here is based on the procedure described in [RFC 5280].The same terminology and process descriptions are used. The eMRTD certificate profiles restrict certification paths to a single certificate and prohibit use of many optional features that are used in other applications, such as the Internet PKI defined in [RFC 5280]. Path validation steps associated with these features are omitted from the eMRTD certification path validation procedure.

D.1.1 Certification Path Validation Procedure

D.1.1.1 Inputs

[RFC 5280] defines a set of nine inputs to the path validation algorithm. Only the following three are relevant to the eMRTD application:

- certification path: A single certificate (e.g. the Document Signer certificate);
- current date/time; and
- Trust Anchor information, including:
    - trusted issuer name: If the Trust Anchor is in the form of a CSCA certificate, the trusted issuer name is the value of the `subject` field of that certificate;

- trusted public key algorithm: If the Trust Anchor is in the form of a CSCA certificate, the trusted public key algorithm is taken from the `SubjectPublicKeyInfo` field of that certificate;
- trusted public key: If the Trust Anchor is in the form of a CSCA certificate, the trusted public key is taken from the `SubjectPublicKeyInfo` field of that certificate; and
- trusted public key parameters: This is an optional input that is included only if the trusted public key algorithm requires parameters. If the Trust Anchor is in the form of a CSCA certificate, these parameters are taken from the `SubjectPublicKeyInfo` field of that certificate.

If an implementation requires that the additional six inputs be supplied, recommendations for these are provided in D.2.

There could be several Trust Anchors for the CSCA that issued the certificate being validated. Of these Trust Anchors, the one that MUST be used is the one that contains the public key that matches the value of the Authority Key Identifier extension in the certificate being validated.

D.1.1.2 Initialization

There are eleven State variables defined in [RFC 5280]. Only the following five are relevant to the eMRTD application:

- application: max_path_length: Initialize to “0”;
- working_issuer_name: Initialize to the value of the trusted issuer name;
- working_public_key_algorithm: Initialize to the value of the trusted public key algorithm;
- working_public_key: Initialize to the value of the trusted public key; and
- working_public_key_parameters: Initialize to the value of the trusted public key parameters.

If an implementation requires that the additional six variables be initialized, recommendations for these are provided in D.2.

D.1.1.3 Certificate processing

eMRTD certificate processing steps are a subset of those defined in [RFC 5280]. The result of processing an eMRTD certificate using this simplified process will be consistent with the result using the full RFC 5280 algorithm. If the additional inputs and State variables are configured as described in D.2:

a) Verify the basic certificate information. The certificate MUST satisfy each of the following:

- the signature on the certificate can be verified using working_public_key_algorithm, the working_public_key, and the working_public_key_parameters;
- the certificate validity period includes the current time;
- at the current time, the certificate is not revoked (see 6.3 for details); and
- the certificate issuer name is the working_issuer_name.

b) Assign the certificate `subjectPublicKey` to working_public_key.

c) If the `subjectPublicKeyInfo` field of the certificate contains an algorithm field with non-null parameters, assign the parameters to the working_public_key_parameters variable. If the `subjectPublicKeyInfo` field of the certificate contains an algorithm field with null parameters or parameters are omitted, compare the certificate `subjectPublicKey` algorithm to the working_public_key_algorithm. If the certificate `subjectPublicKey` algorithm and the working_public_key_algorithm are different, set the working_public_key_parameters to null.

d) Assign the certificate `subjectPublicKey` algorithm to the working_public_key_algorithm variable.

e) Recognize and process any other critical extensions present in the certificate.

f) Process any other recognized non-critical extensions present in the certificate.

If any of the checks in step a) fail or if there are any unrecognized critical extensions in the certificate that cannot be processed, the path validation procedure fails. Otherwise the procedure succeeds.

D.1.1.4 Outputs

If path validation succeeds, the procedure terminates, returning a success indication together with the working_public_key, the working_public_key_algorithm, and the working_public_key_parameters.

If path validation fails, the procedure terminates, returning a failure indication and an appropriate reason.

D.1.2 CRL Validation and Revocation Checking

The CRL validation algorithm in [REC 5280] covers various types of CRLs including delta CRLs, partitioned CRLs, indirect CRLs, etc. The CRL profile for the eMRTD application is very restrictive and prohibits use of any of these features. Use of the `issuingDistributionPoint` extension as well as all of the standardized CRL-entry extensions is also prohibited. As a result, CRL validation and revocation checking for the eMRTD application is relatively simple.

D.1.2.1 Inputs

[RFC 5280] defines two inputs to the CRL validation algorithm. Only the following one of these is relevant to the eMRTD application. If an implementation requires that the additional input be supplied, a recommendation for this is provided in D.2.

- certificate: certificate serial number and issuer name

D.1.2.2 Initialization

There are three State variables defined in [RFC 5280]. Only the following one of these is relevant to the eMRTD application. If an implementation requires that the additional two variables be initialized, recommendations for these are provided in D.2.

- cert_status : initialize to the value UNREVOKED.

D.1.2.3 CRL Processing

All CRLs in the eMRTD application are complete CRLs that cover all current certificates issued by the CSCA that issued the CRL. There are no partitioned, delta or indirect CRLs. The steps in the CRL processing algorithm for the eMRTD application are:

a) Obtain the current CRL for the CSCA that issued the certificate. If the CRL cannot be obtained, the cert_status variable is set to UNDETERMINED, and processing is stopped.

b) Verify that the CRL issuer is the same CSCA that issued the certificate in question. Because there is a single CSCA in each country, and the eMRTD application is a closed application with Inspection Systems retaining a cache of CRLs that is unique to this application, verifying that the country name is the same in the issuer field of the CRL and the issuer field of the certificate is sufficient.

- If the CSCA has not undergone a name change since the certificate was issued, the issuer field in the CRL and the issuer field in the certificate will be identical.

- If the CSCA has undergone a name change since the certificate was issued, the country attribute of the name in the issuer field of the certificate and in the issuer field of the CRL will be the same, but some other attributes may be different.

- If the relying party wishes to verify that substitution of some non eMRTD CRL has not happened, it may optionally verify that it has Trust Anchors for both CSCA names and that those Trust Anchors are for the same CSCA. If the CSCA has undergone a name change and has included the optional `issuerAltName` extension in the CRL, the relying party MAY optionally verify that the issuer field in the certificate is identical to one of the values in this extension.

If the CRL issuer is not the CSCA that issued the certificate, the cert_status variable is set to UNDETERMINED, and processing is stopped.

c) Validate the certification path for the issuer of the CRL. Note that in the eMRTD application all CRLs are issued by CSCAs that are the Trust Anchors for the respective paths. Unlike the algorithm in [RFC 5280], the eMRTD application does NOT require that the Trust Anchor used to validate the CRL certification path be the same Trust Anchor that was used to validate the target certificate. However, if the Trust Anchors are different, they MUST both be Trust Anchors for the same CSCA. Unlike [RFC 5280], the eMRTD application has multiple Trust Anchors for a given CSCA that are valid at the same time. If the certification path cannot be successfully validated, the cert_status variable is set to UNDETERMINED, and processing is stopped.

d) Verify the signature on the CRL. If the signature cannot be successfully verified, the cert_status variable is set to UNDETERMINED, and processing is stopped.

e) Search for the certificate on the CRL. If an entry is found that matches the certificate issuer and serial number, then the cert_status variable is set to UNSPECIFIED.

D.1.2.4 Output

Return the cert_status. If steps a), b), c) or d) failed, the status will be UNDETERMINED. If the certificate was listed as revoked on the CRL, the status will be UNSPECIFIED. If CRL validation succeeded, but the certificate was not listed on the CRL, the status will be UNREVOKED.

D.2 STEPS NOT REQUIRED BY eMRTD

D.2.1 Certification Path Validation

Settings for additional inputs that are not relevant to eMRTD validation include:

- initial-policy-mapping-inhibit: Set to inhibit policy mapping;
- initial-any-policy-inhibit: Set to inhibit processing of the any-policy value;
- initial-permitted-subtrees: Set to permit all subtrees;
- initial-excluded-subtrees: Set to exclude no subtrees;
- initial-explicit-policy: This should NOT be set; and
- user-initial-policy-set: Set to the special value “any-policy”.

Initialization of State variables that are not relevant to the eMRTD application include:

- permitted_subtrees: Initialize to permit all subtrees;
- excluded_subtrees: Initialize to exclude no subtrees;
- inhibit_any_policy: If initial-any-policy-inhibit is set, initialize to “0”. Otherwise, set to the value 1 or any value greater than that;
- policy_mapping: Initialize to “0”;
- explicit_policy: Initialize to “2”; and
- valid_policy_tree: Initialize the valid_policy element to “anyPolicy”, the qualifier_set element to empty and the expected_policy_set to “anyPolicy”.

D.2.2 CRL Validation

Settings for additional inputs that are not relevant to eMRTD validation include:

- use-deltas: Set to prohibit use of deltas.

Initialization of State variables that are not relevant to the eMRTD application include:

- reasons_mask: Initialize to an empty set; and
- Interim_reasons_mask: Initialize to the special value “all-reasons”.

D.3 MODIFICATIONS REQUIRED TO PROCESS CRLS

CRL validation systems that comply with the CRL validation procedure in [RFC 5280] are not intended to support environments where a CA has undergone a name change, such as the eMRTD application environment. Therefore these systems require some modification to handle this special case, as described below:

a) In clause 6.3.3, step a) of the [RFC 5280] CRL validation procedure, the name in the distribution point field of the CRL Distribution Points extension of the certificate in question is used to update the local cache with the relevant CRL(s). For the eMRTD application, this step would need to be modified and only the `countryName` attribute of the distribution point field should be used to identify and obtain the appropriate CRL.

b) In clause 6.3.3, step f) of the [RFC 5280] CRL validation procedure, there is a requirement that the same Trust Anchor be used to validate the certification path for the CRL issuer that was used to validate the target certificate. This is NOT a requirement for the eMRTD application because independent Trust Anchors are established for each public key of the CSCA.

The Trust Anchor used for validation of the CRL issuer will be the one for the CSCA's public key that corresponds to the private key used to sign the CRL. The Trust Anchor used to validate the certification path for the target certificate may be for an earlier CSCA key pair.

— END —

ISBN 978-92-9249-800-9