JKST-IWG

# Certificate Path Processing Testing Guideline

## Ver 1.0

Mar 07, 2003

- CHANGES -

| Version | Date | Comments | Detail |
|---|---|---|---|
| 1.0 | 20030228 | Published | First Edition |
| | | | |

- Table of Contents -

# 1 Introduction

## 1.1 Background

The Interoperability Working Group (IWG), formed by Japan, Korea, and Singapore members, completed the multi PKI domains interoperability experiment[1]. In the experiment, the IWG established a CA-CA model with the Certificate and CRL and LDAP schema profile[2] to be interoperable each other.

Even though the different policies and trust models exist in each nation, the IWG successfully finished the interoperability tests and obtained some levels of confidence that an emerging framework could be possible. Trust models could be absorbed and/or coexist if a certificate and its chains are processed at the agreeable ways.

One of the lessons learnt from the project was that there are few frameworks, criteria, and even guidelines that all parties could be able to agree upon in terms of path processing test suites to evaluate the results each other. This difficulty stems largely from the fact that different PKI vendors have different testing methods and different PKI domains have different requirements in their own trust models.

In the multi PKI domain interoperability (especially different vendors in different countries involved), when no levels of conformance are guaranteed in terms of path processing, it would be difficult to ensure a Relying Party application in one country will validate the certificate and its path in the same way that the other does in other countries, and it would be hard to achieve the reliable infrastructure where secure business transactions are conducted.

Therefore, common agreeable test suites and the guideline should be created as criteria to check and verify the path processing logic in applications for the PKI environments, where the multiple CA topology and trust models could

---

[1] Achieving PKI Interoperability
Results of the JKS-IWG Interoperability project
http://www.japanpkiforum.jp/shiryou/IPA/final.pdf

[2] Achieving PKI Interoperability
Results of the JKS-IWG Interoperability project
Recommendations on Technical Certificate Profile
http://www.japanpkiforum.jp/shiryou/IPA/final_2pdf.pdf

coexist.

### 1.2 Objectives

The objective of this document is to test the path validation processing logic in the Relying Party (RP) application and certificate-issuance capabilities in the Certification Authority (CA) application. With this guideline, potential PKI users and service providers can evaluate applications, especially the RP application in the path processing logic function, which is crucial and critical to the trustworthiness of the PKI operations in business environments. By developing this document, the IWG will facilitate the CA-CA interoperability in multiple domains so as to ensure that each relying party can validate the certificates in the same fashion each other.

### 1.3 Intended Audience

This guideline is developed for the application vendors, PKI users, and service provides who actually uses the PKI applications for their businesses to ensure that the targeted applications can validate the certificates followed by the requirements derived from the IWG certificate and CRL profile.

## 2 Path Processing Test

### 2.1 Test Framework

#### 2.1.1 Test Design Fundamental

This document is developed based on the path processing logic of RFC3280[3] specification, a subset of X.509[4] standard, test reference 'Conformance Testing of Relying Party Client Certificate Path Processing Logic'[5], and the requirements derived from the standards and IWG Certificate and CRL Profile.

---

[3] RFC3280
Internet X.509 Public Key Infrastructure: Certificate and CRL Profile
http://www.ietf.org/rfc/rfc3280.txt

[4] ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8:
"INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION
- THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS"

[5] Conformance Testing of Relying Party Client Certificate Path Processing Logic, 2001 v1.07
http://csrc.nist.gov/pki/testing/x509paths.html

The specifications and requirements are used as a basis for test items necessary to evaluate the RP applications for targeted PKI architectures and services.

With the specifications and requirements, this guideline has two features:

1) Test is categorized more from the PKI user side.
2) Test can be used combining several test models.

The test is categorized based on the users'and service providers'perspectives rather than application developers'perspectives. The test is structured, more considering the PKI service environment for the users/service provider to evaluate the application easier. When the user/service provides plan to use the PKI, they are typically required to design the certification authority structure in **interconnection model**. They also need to decide the **service model** such as integrity and authentication services. In addition, they need to consider **revocation model** that checks the certificate status information. So that the testing framework should be categorized followed by the models where the users and service providers will be based on.

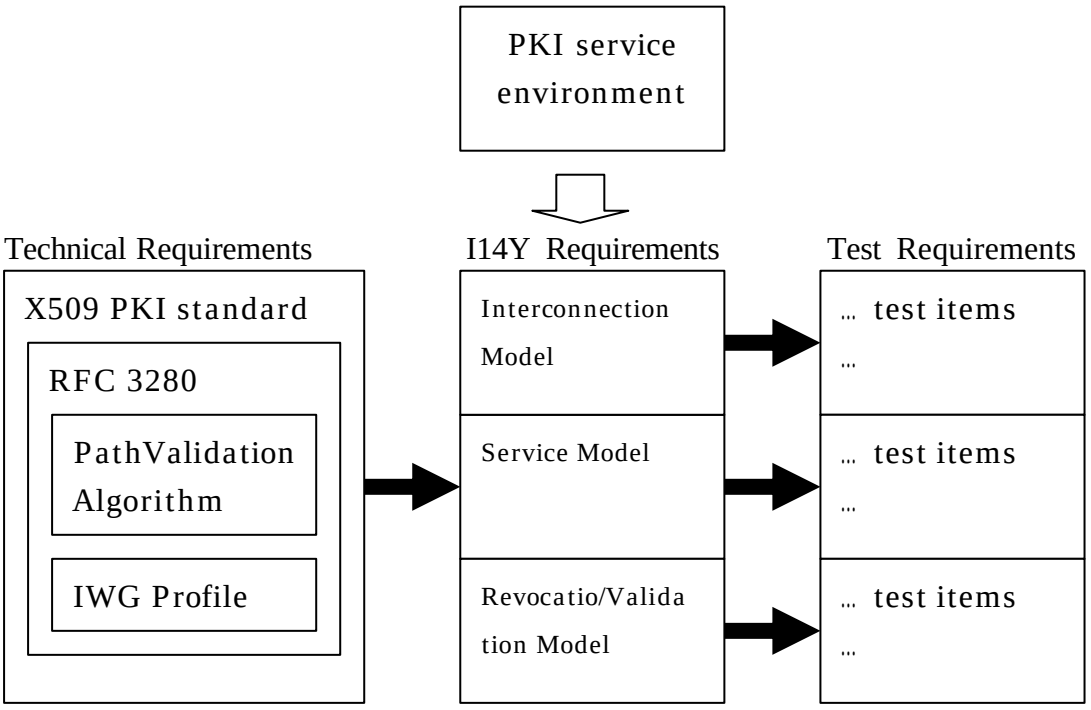



Figure 2.1 Extracting Test Items

Figure 2.1 shows the overall picture of the test designs and workflow. First, the technical requirements are identified in the path processing logic in the

RFC3280 and IWG Certificate and CRL Profile. Then the technical requirements are mapped to the three models, the interconnection model, the service model, and the revocation/validation model, generating a minimum set of the I14Y requirements. Finally each model generates test scenarios and test items to satisfy them.

A test item is an individual test case with a collection of inputs that cause one execution of an application. A set of test items is designed to cover an individual test requirement and can be divided into the success cases and failure cases. NOTE: The case generation depends on the ASN.1 structures of the certificate fields and the requirement of the specification and IWG Certificate and CRL Profile.

The test is conducted using the black box-based testing method, which means that there are several test values used for testing. The test case value is the essential part of testing. As the prefix values to trigger each test, several certificates, CRL/ARL and several initial parameters are provided. In the data, each test case contains verifiable value(s), which are to be evaluated by comparing the output of the application with the expected value or/and test scenario (success or failure) in the document.

In testing, the test planners can combine the models among the interconnection, service, and revocation/validation models to meet their specific requirements in the PKI environment currently concerned.

### 2.1.2 Test Scope

The test scope includes testing based on the following models:

Table 2.1 Test Models

| MODEL | DETAILS |
|---|---|
| Base | Base |
| Interconnection Model | Strict Hierarchy |
| | Cross Certification |
| | Cross Recognition |
| | Bridge CA |
| | Mesh |
| | Certificate Trust Lists |

| | |
|---|---|
| | Accreditation Certificate |
| Service Model | Signing |
| | Encryption |
| | Authentication |
| | Notary |
| Revocation & Validation Model | CRL |
| | OCSP |
| | Delegated Path Discovery/Validation |

For the base model, it is for the general test cases. For the interconnection model, there are several CA-CA architectures, strict hierarchy, Cross Certification (CC), Cross Recognition (CR), Bridge CA, Mesh, Certificate Trust Lists, and Accreditation Certificate are assumed. For the service models, signing, encryption, authentication, and notary are assumed. For the revocation & validation model, the CRL, OCSP and Delegated Path Discovery/Validation(DPD/DPV)[6] models are included. All the details in the assumptions of the models will be described in clause 2.3.

The scope also includes the testing whether the certificate and CRL have been generated in accordance with the IWG profile. This guideline specifies the requirements of the CA applications and test items.

The scope, however, excludes testing of the Relying Party application to parse ASN.1 structure correctly. This guideline does not include testing of the low level of crypto operations either.

### 2.1.3 Assumptions

1. The encryption, authentication, and notary services are currently out of scope in this document.

2. The bridge CA model is currently out of scope in this document. However, there are several test items to check the path length in the cross certification model via an anchor CA. In the model, the anchor

[6] RFC3379
Delegated Path Validation and Delegated Path Discovery Protocol Requirements
http://www.ietf.org/rfc/rfc3379.txt

CA can be treated as a Bridge CA to be connected with the other CAs.

3. The OCSP model is currently out of scope in this document.

4. The Cross certification model assumes that the root CA (in the hierarchy) is cross-certifying the other CAs and vice versa. No subordinate CAs are cross-certifying the other CAs.

5. The path processing logic used in this document is derived from RFC3280.

6. The Trust Anchor CA is not used in the certification path. The trust anchor information is used as only input values specified in the RFC 3280.

7. The certificates and corresponding CRLs are signed with the same Certification Authority with the same key.

8. No values are tested in the following extensions…
   - privateKeyUsagePeriod
   - subjectAltName
   - issuerAltName
   - subjectDirectoryAttributes
   - extendedKeyUsage
   - inhibitAnyPolicy
   - freshestCRL
   - authorityInfoAccess
   - subjectInfoAccess

9. No test cases for criticality to save labor, but only critical extensions which defined locally in IWG profile, has test case for criticality.

### 2.1.4 Test Levels

The testing level indicates how much the application will be interoperable and secure.

The level 0 assumes that the CA application must issue certificates and CRL/ARL, which contains the mandatory fields in IWG profile, and the RP application must validate the components. The tester must run this test and

pass the test. Note that the document typically presumes that the RP application already test this level in the software development stage.

The level 1 assumes that the CA application must issue certificates and CRL/ARL, which contains the optional fields in IWG profile and the RP application must validate the components if necessary. The tester should run this test.

The level 2 assumes that the CA application will specify the multiple values and constraint-related fields in the certificates (such as Policy Constraints and Name Constraints) and the RP application will validate the components. The tester may run this test. Table 2.2 summarizes the test levels.

Table 2.2 Definition of Test Level

| Level | Criteria | Description |
|---|---|---|
| 0 | Certificate Issuance | Specify mandatory fields in IWG profile |
| | Validation Requirements | MUST run this test<br>(or MUST pass in the system test before this guideline is applied) |
| 1 | Certificate Issuance | Specify optional fields in IWG profile |
| | Validation Requirements | SHOULD run this test |
| 2 | Certificate Issuance | Specify multiple values and constraints-related values in IWG profile |
| | Validation Requirements | MAY run this test |

### 2.1.5 Document Conventions

Each test items is specified using the following convention. The interconnection model (**Int**), service model (**Srv**), and revocation model (**Rvk**) contain the categories such as Cross Certification (**CC**) in the interconnection model and Signing (**DS**) in the service model. In the categories, there are test items for Relying Party (**RP**) and Certification Authority (**CA**) applications. Each test item has the number with the test level. The examples are shown below.

- Int.CC.RP.22.Level 0

- Srv.DS.RP.22.Level 1
- Rvk.CRL.RP.25.Level 0

### 2.1.6 Usage of This Guideline

#### (1) Outline of this guideline

Specification of path validation, especially in multi-domain PKI, is complex much. Therefore, various PKI applications cannot always implement full path validation function. In multi-domain PKI, it is essential that skilled understanding about path validation whether a PKI application has enough function, because "**What kind of path validation function is required**" and "**How to evaluate**" are vary by each multi-domain PKI. Here issues are what they must need to assess below rightly, and what is difficult.

- Analysis of multi-domain PKI
- Required path validation function
- Expected and right validation result

This clears away such difficulty, and provides a framework for evaluating the path validation function of PKI application in multi-domain PKI without skilled understanding.

Therefore, at first, this defines typical PKI model to analyze multi-domain PKI easily. The guideline users understand easily which model matches each domain, by reference to typical PKI model.

At second, the guideline defines the testing requirements which is necessary for each domain, by extracting from the standards (e.g., ISO/ITU-T and IETF/PKIX and etc.), and set the right expected results. From this, users can learn required testing items and evaluation method easily for their domain.

##### (a) Definition of PKI model

If some PKI domains, which are operated by each unique security policy, interconnect mutually and provide a service astride both domains, this guideline is as reference for the PKI domains.

This guideline defines the typical PKI model as single PKI domain and a kind of interconnecting. Therefore, these models may correspond to many existing PKI domains. The guideline users can make use of these models as a material for analysis when they interconnect each other.

- What kind of model is my PKI domain?
- What kind of model is a destination PKI domain?

- What is an appropriate model for interconnecting both?
- Etc.

Furthermore, they can also make use of this as material for analysis when they provide a service astride interconnected domains.

- What requirement should they satisfy?
- What information should they process?
- Etc.

This guideline is classified from three viewpoints below to refer easily.

(a) Interconnection method
(b) Service pattern
(c) Path validation method

The users can refer to a necessary model by each viewpoint (a) to (c).

(i) Interconnection model

Users can make use of this as reference when they understand what kind of model their domain or destination domain is, or what is an appropriate model to interconnect mutually or unilaterally. This defined the requirements to establish for each model.

(ii) Revocation and Validation model

Users can make use of this as reference when they understand what kind of revocation information other domain provides, or what kind of revocation information they must process. And this may define what kind of protocol they need to access with the validation server, when it comes onstage in the future.

(iii) Service model

Users can make use of this as reference to learn what kind of service is feasible or what a requirement for the service is.

Specifically, they become able to consider easily the questions shown below.

- What is an appropriate model for interconnecting mutually or unilaterally?
- What service can we provide by interconnection?
- What is a necessary method for certificate/path validation of the destination domain?

(b) Definition and application of the testing criteria

This defines the test criteria which are based on IWG recommended profile. Users can confirm easily and systematically whether their profile is based on IWG recommended profile by meeting these criteria. Also Users can understand easily how compliant with IWG recommended profile, because test level is setting to each test-case.

This is criteria for IWG recommendation profile, but almost is the requirement for standard (X.509 or RFC3280 and so on). So this can apply as criteria of another framework by readjusting test level slightly, not only IWG. For example, when it is made as GPKI criteria, they assign the GPKI minimal requirement as level 0, and assign the other requirement as level 1 or higher. Then they can make the guideline for GPKI.

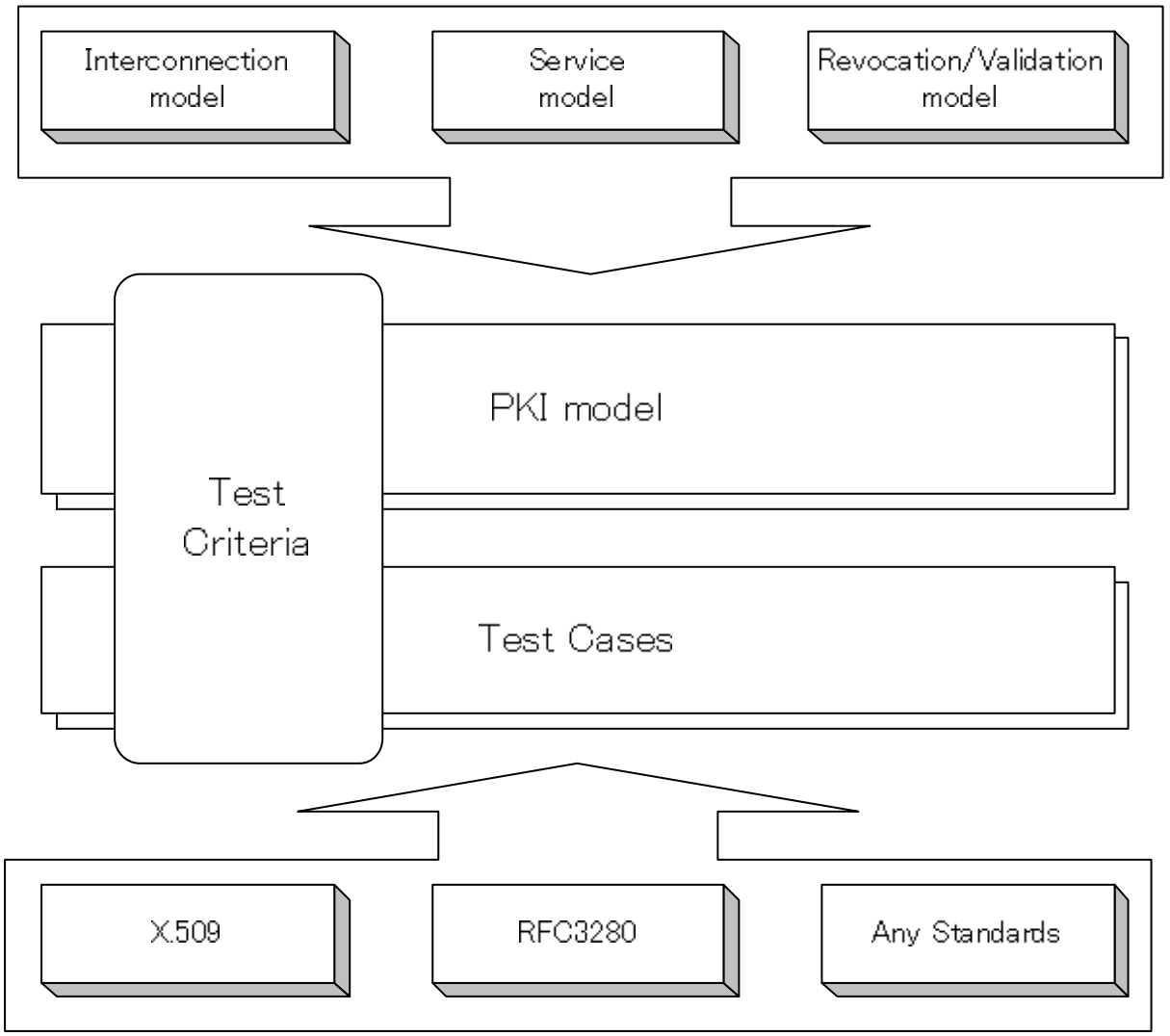


Figure 2.2 Applicablity for other criteria

(2) Who read this guideline?

This is a framework to evaluate a PKI application in multi-domain PKI. Therefore, it should become useful for evaluators of PKI application. Specifically, they may be service providers selecting the PKI application and introducing it into the service, or may be application certifiers accrediting that a certain PKI application satisfies a fixed function.

And a designer of Principal CA, who develops the interconnection with other domain, may make use of the information for the PKI model defined in this guideline as a reference when interconnecting.

(a) Participant of principal CA
  (i) Designing for certificate profile
  (ii) Choice of interconnection model
  (iii) How to provide the revocation information
(b) PKI Service Provider
  (i) Definition of path validation requirement for PKI application
  (ii) Definition of testing requirement for PKI application
  (iii) Requirement for what kind of service provide
(c) Accreditation organization
  (i) Criteria for CA accreditation
  (ii) Criteria for PKI application accreditation
  (iii) Criteria for interconnecting to other PKI domain

This guideline defines CA requirements that CA should clear and relying-party requirements that PKI service provider should clear. Authorization organization can make use of the extracted test items in this document as criteria for each party.

(3) How use this guideline

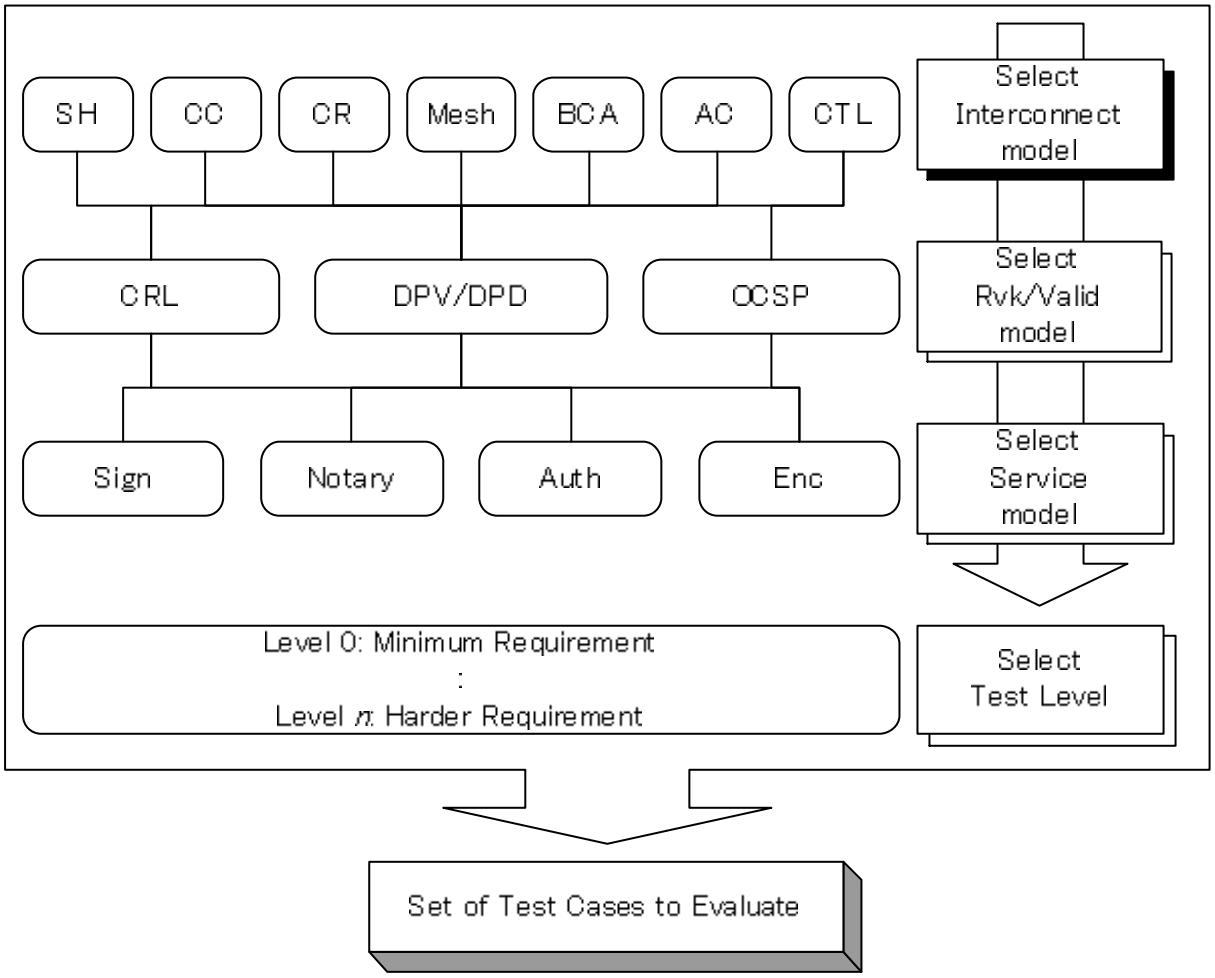
This guideline defines a lot of test cases to be compliant with IWG recommendation profile and various standards such as ITU-T/X.509 and PKIX/RFC3280. The guideline users can extract just appropriate test cases from this guideline, and then use the test cases to evaluate whether the applications

are compliant with the criteria.

The users must obtain the information below for extracting the appropriate test cases.

- Type of Interconnection to destination domain
- Type of Revocation / Validation at destination domain
- Type of Service between each domains
- Level of compliance to Criteria

The users can extract a set of required and appropriate test cases, according to these information and flow below.



## 2.2 Testing Models and Testing Requirements

### 2.2.1 Analysis of Various PKI domain

This section analyzes and categorizes the various PKI domains from the three viewpoints, CA topology, service model, and revocation/validation model.

#### (1) Definition of CA topology

This section analyzes and categorizes various CA topologies in the multi domain PKI. Especially "CA-CA Interoperability" published by PKI Forum[8] is referred.

(a) StrictHierarchy

(i) Definition

- Only Root CA issues self-signed certificate.
- Subordinate CAs don't issue self-signed certificate, only superior CA issues CA certificates to them.
- Subordinate CAs are not allowed to have multi superior CAs.

(ii) Usage

Basically, this model is used in single domain PKI. Many domains may operate CAs in their hierarchic structures with a single policy, and include no certificatePolicies extensions in certificates. This is useful for a vertical organization (e.g., an enterprise) that is applicable easily to the hierarchic structure.

(iii) Advantage and disadvantage

- Applicable to existing applications based on SSL.
- There are many applications, but only a few applications support the path processing.
- A lack of extended ability.
- Subordinate CAs are not allowed to cross-certify other CAs directly.

---

[7] CA-CA Interoperability
http://www.pkiforum.org/pdfs/ca-ca_interop.pdf

[8] PKI Forum
http://www.oasis-open.org/committees/pki/

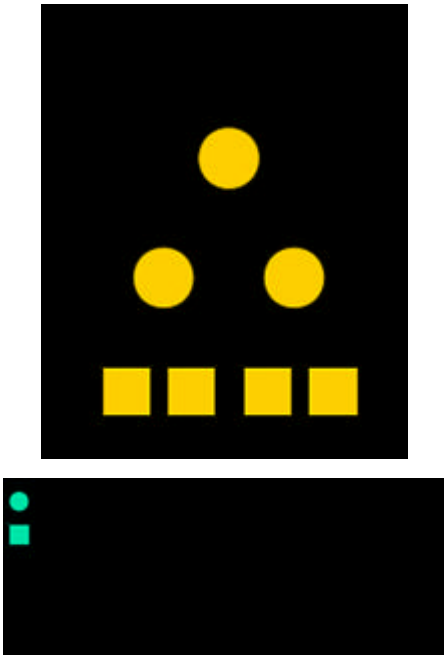


Figure 2.4 Strict Hierarchy model

(b) CrossCertification

(i) Definition

- The model in which CAs issue a cross-certificate to other CAs..

<CITE FROM X.509 4th>

> CAs issue certificates to other CAs either as a mechanism to authorize the subject CA's existence (e.g. in a strict hierarchy) or to recognize the existence of the subject CA (e.g. in a distributed trust model).
> The crosscertificate structure is used for both of these.

- There are two methods in cross-certification.
  - Mutual-certification: each CA issues the cross-certificate one another.
  - Unilateral-certification: only one CA issues the cross-certificate to another CA.
- CAs store cross-certificate by crossCertificatePair format.

(ii) Usage

Topologically speaking, cross-certification merely means issuing a CA certificate except a self-signed certificate. It means a trust relationship between CAs.

This is an original concept of Mesh model, BCA model, accreditation certificate model, and maybe hierarchy model. In a wide sense, this includes

also strict hierarchy model. In a narrow sense, this is used as core techniques of multi domain PKI to build a trust relationship with another domain.

(iii) Advantage and disadvantage

All CA products cannot generate and process the crossCertificatePair. Because this can issue the trust relationship precisely, this is suitable for notary service. Even if CAs revoke a cross-certificate, each subject CA can exist.

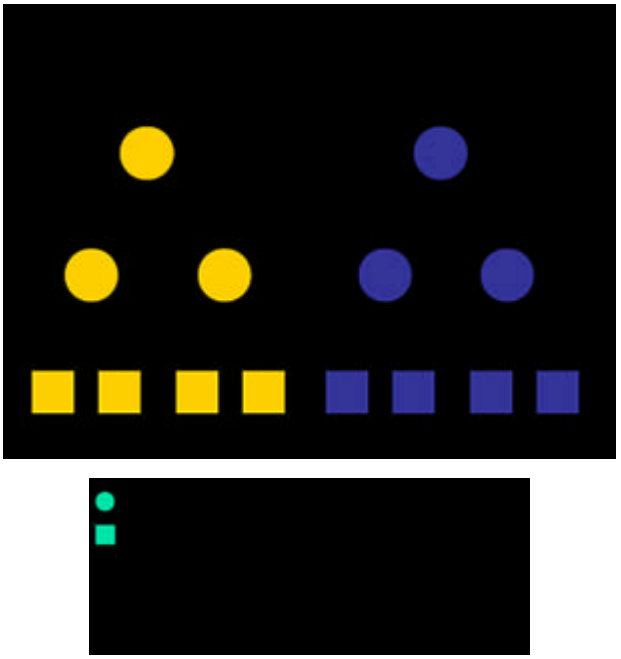


Figure 2.5 Cross Certification model

(c) CrossRecognition

(i) Definition

- The model in which each EE is allowed to specify multiple trust anchors.

(ii) Usage

This is suitable when a strict hierarchy model builds a trust relationship with another one.

(iii) Advantage and disadvantage

Most existing SSL-based applications are grow to be suitable for this by just a little modifying. Because this cannot represent a trust relationship, this model is not suitable to auditing, notary and non-repudiation.

The entity controlling the trust relationship is EE, but not CA.

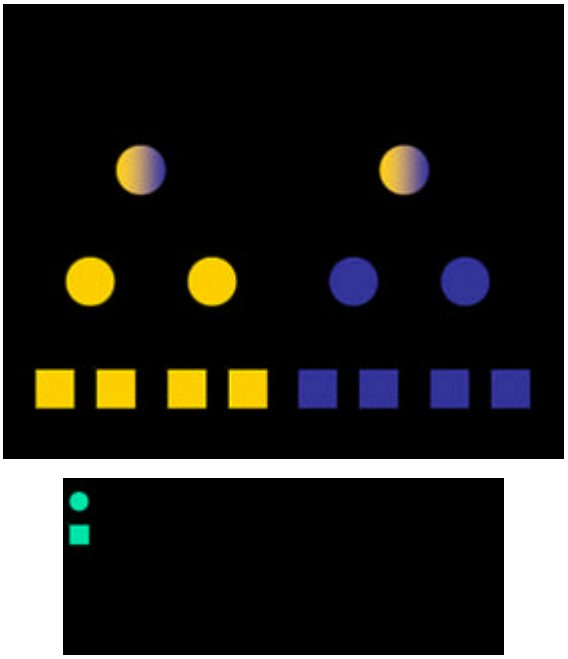


Figure 2.6 Cross Recognition model

(d) Mesh

(i) Definition

- The model in which plural CAs cross-certify at least one other CA.

(ii) Usage

This model is not a CA topology, which is intended to solve certain requirements. Mesh model is merely a result of many cross-certifications.

(iii) Advantage and disadvantage

If each CAs hold their self-signed certificate, they are not effected by the key compromise in other CAs.

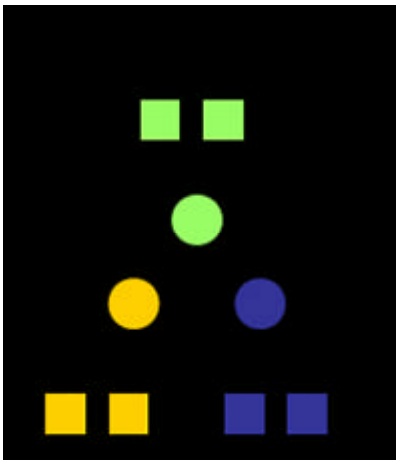


Figure 2.7 Mesh model

(e) BridgeCA

(i) Definition

- The model in which Bridge CA that have self-signed certificate cross-certifies the other plural CAs.

(ii) Usage

This is useful to reduce the complexity of cross-certification. The Bridge CA should be a Trusted Third Party.

(iii) Advantage and disadvantage

- The limited number of cross-certification
- The burden on a Bridge CA operation unit is heavy.
- High skills for path processing are required.

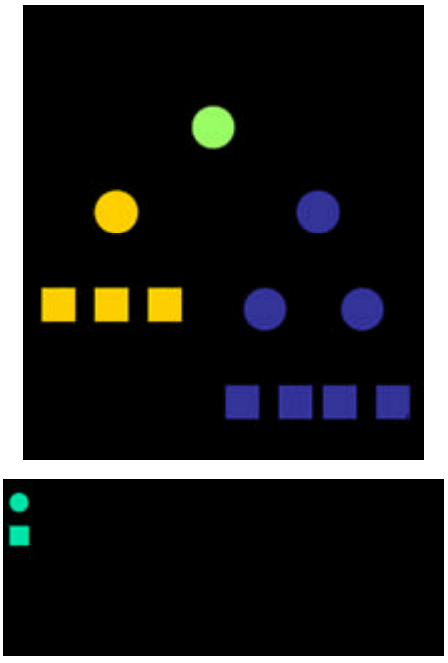


Figure 2.8 Bridge CA model

(f) AccreditationCertificate

(i) Definition

- The model in which only certain CA is allowed to certify plural CAs

that have a self-signed certificate.

(ii) Usage

In the case that only the strict hierarchy is supported by the applications, and a CA operation independent from a superior CA is desirable, this model is useful.

(iii) Advantage and disadvantage

- Each CA is able to operate independently from superior CA.
  - *Superior CA compromise, Superior CA key rollover, Exchange of a superior CA, etc...*
- All applications are not necessary to support the path processing because they can process the path as merely strict hierarchy model. This cannot restrict complex constraints in the certification path.
- Subordinate CAs are forbidden to cross-certify other CAs directly, and the accreditation from Accreditation CA is necessary.

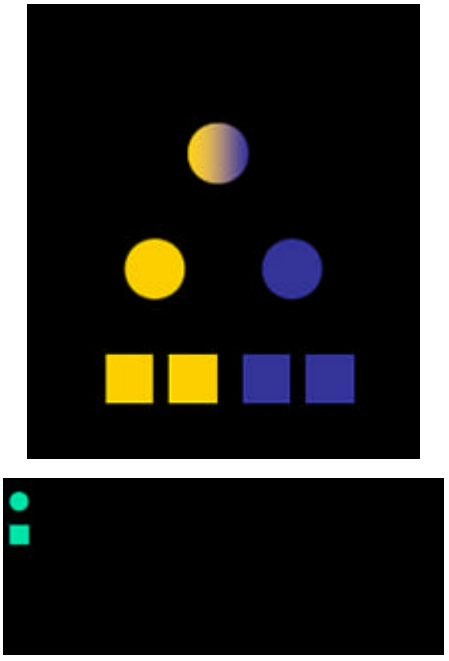


Figure 2.9 Accreditation Certificate model

(g) CertificateTrustLists

(i) Definition

- The trust anchors of each domain issue the certificate trust lists that are lists of trust anchor certificates of the subject domain.
- EEs are allowed to specify other trust anchor certificates in only their CTL when validating the certification path.

(ii) Usage

- When PKI system cannot process or issue the cross-certificate, this model is suitable like Cross-Recognition.
- Especially for a PKI system needing strict audit of interconnection, this model is more suitable than Cross-Recognition.

(iii) Advantage and disadvantage

- In this model, CAs can manage EEs' multiple trust anchors, but EEs cannot manage it.
- CAs do not need to issue a cross-certificate, and applications do not need to process the cross-certificates.
- CAs must issue a certificate trust lists formatted by PKCS#7, and applications must process it.

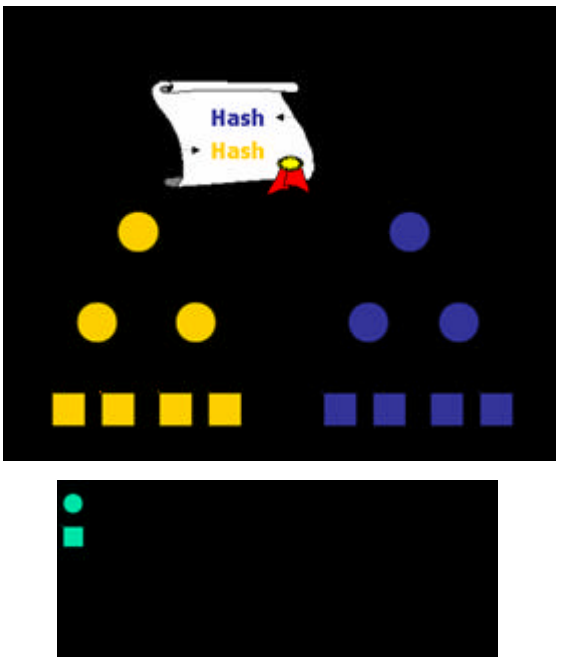



Figure 2.10 Certificate Trust Lists model

(2) Definition of PKI Service model

This section defines the principal service models adopted in the international PKI.

(a) Signing

A typical case is that "a relying-party in Y country validates a signed-data by using a valid certificate in X country." The digital signature will be effective

in the international e-commerce. The typical implementations of this model are PKCS7/CMS singed-data or XMLsignature. In this document, a legal effectiveness is out of scope.

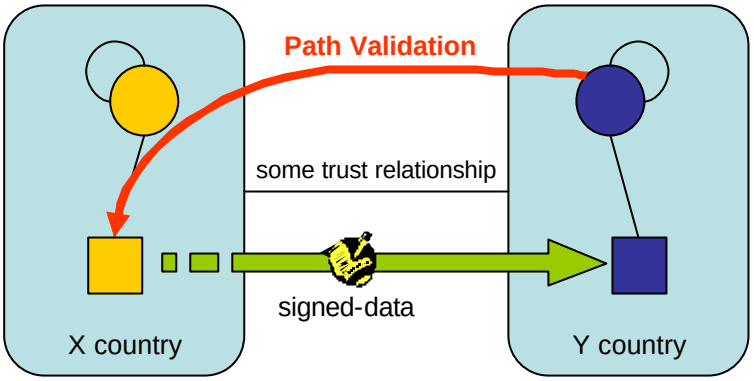



Figure 2.11 Signing model

(b) Notary Service (Long term signature)

A typical case is that "a relying-party in Y country validates a signed-data that existed prior to a particular time in X country." In international mediation, an ability to establish the existence of data prior to the specified times will be necessary. This model may require a TimeStampAuthority or Notary.

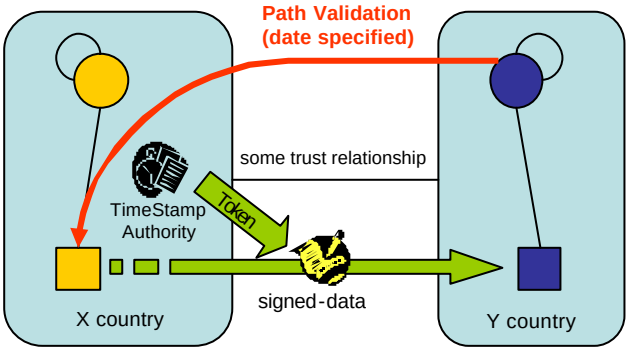



Figure 2.12 Notary model

(c) Authentication

A typical case is that "a client in Y country connects to server in X country by authentication." In the international community, a strict authentication

may be used for distinguishing an individual. A typical implementation of this model is TLS client authentication.

(d) Encryption

A typical case is that "a subscriber in X country encrypts a piece of data by using a certificate of a relying-party in Y country." In the international e-commerce, the encryption is used for the exchange of confidential information. The typical implementation of this model is PKCS7/CMS Encryption-data or XMLsecurity.

(3) Definition of Revocation / Validation model

This section defines the validation models in the international PKI.

(a) CRL

A typical case is that "a relying-party of Y country requires obtaining a CRL for a path processing, which is issued by the issuer of subscriber certificate in X country."

(b) OCSP

A typical case is that "a relying-party of Y country requires a response from an OCSP Responder in X country for validating a subscriber certificate of X country."

The case that "a relying-party of Y country requests to an OCSP Responder in Y country to validate a subscriber certificate of X country" is regarded as a delegated path validation.

(c) Delegated Path Discovery/Validation

A typical case is that "a relying-party of Y country needs a VA (validation authority) of Y country to validate a certificate path between the relying party and a subscriber in X country."

This subsection will be revised after RFC of DPD/DPV is published.

2.2.2 Requirements for Path Processing

This section defines the requirements to confirm the path processing about each model categorized in section 2.2.1. The requirements below are almost derived from ITU-T/X.509, IETF/PKIX RFC3280, and IWG recommended profile.

(1) Base Requirements

CA.01: CAs should issue a certificate that directoryName in its issuer DN and subject DN are encoded by UTF8String except for a country attribute.

**[IWG profile]**

CA.02: CAs should generate all keyIdentifier by the 160bit SHA-1 hash in all certificates they issue. This is derived from the method defined in paragraph (1) of Section 4.2.1.2 Subject Key Identifier in RFC 3280.

**[IWG profile, RFC3280 4.2.1.1 & 4.2.1.2]**

CA.03: CAs should generate consistently all keyIdentifiers in all certificates.

**[IWG Profile, RFC3280 4.2.1.1 & 4.2.1.2]**

CA.04: CAs should issue a certificate including a consistent format of authorityKeyIdentifier in all certificates they issue.

**[IWG profile, RFC3280 4.2.1.1]**

CA.05: CAs should issue a self-signed certificate which has the basicConstraints present and critical with cA flag asserted.

**[IWG profile]**

CA.06: CAs should issue a certificate whose validity is encoded by UTCTime.

**[X.509 7]**

RP.07: The application should validate successfully the correct certification path.

RP.08-11: The application should ensure that the issuer distinguishedName of a certain certificate and the subject distinguishedName of its issuer certificate should be identical about each certificate in the certification path.

**[X.509 10.5.1]**

RP.12: The application should trace the certification chain by keyIdentifier in authorityKeyIdentifier and subjectKeyIdentifier of each certificate in the certification path.

**[RFC3280 4.2.1.2]**

RP.13-16: The application should ensure that the validity of each certificate in the certification path should include the current time.

**[X.509 10.5.1]**

RP.17-18: The application should treat a validity set as UTCTime with a year of 50 about each certificate in the certification path.

**[X.509 7]**

RP.19: The application should verify each certificate in the certification path by its issuer certificate.

**[X.509 10.5.1]**

RP.20: The application should ensure whether the subscriber certificate is revoked or not.

**[X.509 10.5.1]**

RP.21: The application should process a certification path which contains a certificate which has unrecognized extensions.

**[X.509 7]**

(2) Interconnection requirements

(a) Strict Hierarchy

CA.01: CAs should issue a CA certificate including cA flag set to TRUE in critical basicConstraints extension, except for self-signed certificate.

**[X.509 8.4.2.1]**

CA.02: CAs should issue a CA certificate including keyCertSign in critical keyUsage extension, except for self-signed certificate.

**[X.509 8.2.2.3]**

CA.03: CAs should issue a CA certificate including pathLenConstraints in critical basicConstraints extension, except for self-signed certificate.

**[X.509 8.4.2.1]**

CA.04: CAs should issue CA certificates including a policyIdentifier in critical certificatePolicies extension, except for self-signed certificate.

**[X.509 8.2.2.6]**

CA.05: CAs should issue CA certificates including plural policyIdentifier in critical certificatePolicies extension, except for self-signed certificate.

**[X.509 8.2.2.6]**

CA.06: CAs should issue CA certificates including a policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate.

**[X.509 8.2.2.6]**

CA.07: CAs should issue CA certificates including plural policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate.

**[X.509 8.2.2.6]**

RP.08: The application should validate successfully correct certification path.

RP.09-10: The application should validate a certification path including a subordinate CA certificate.

**[X.509 10.5.1]**

RP.11-13: The application should ensure whether all CA certificate in the certification path have cA flag set to TRUE in critical basicConstraints extension.

**[X.509 10.5.1]**

RP.14: The application should ensure whether the certification path length is shorter than pathLenConstraints or not in any CA certificate.

**[X.509 10.5.1]**

RP.15-17: The application should ensure whether all CA certificate in the certification path have keyCertSign in critical keyUsage extension.

**[IWG profile]**

RP.18-21: The application should process certificatePolicy in all certificates for validating the certification path.

**[X.509 8.1.1]**

RP.22: The application should ensure whether all CA certificate in certification path is revoked or not.

**[X.509 10.5.1]**

RP.23: The application should verify all CA certificates in certification path by its issuer certificate.

**[X.509 10.5.1]**

(b) Cross Certification

CA.01: CAs should issue a cross-certification request including a subjectKeyIdentifier extension in extensionRequest, and its value should be identical with subjectKeyIdentifier in their self-signed certificate.

**[IWG profile]**

CA.02: CAs should issue a cross-certificate including SubjectKeyIdentifier, which should be the same as SubjectKeyIdentifier in corresponding cross-certification request.

**[IWG profile]**

CA.03: CAs should issue a cross-certificate including a policyIdentifier in critical certificatePolicies extension, except for self-signed certificate. This assertion is the same as Int.SH.CA.04 requirement.

**[X.509 8.2.2.6]**

CA.04: CAs should issue a cross-certificate including plural policyIdentifier in critical certificatePolicies extension, except for self-signed certificate. This assertion is the same as Int.SH.CA.05 requirement.

**[X.509 8.2.2.6]**

CA.05: CAs should issue a cross-certificate including a policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate. This assertion is the same as Int.SH.CA.06 requirement.

**[X.509 8.2.2.6]**

CA.06: CAs should issue a cross-certificate including plural policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate. This assertion is the same as Int.SH.CA.07 requirement.

**[X.509 8.2.2.6]**

CA.07: CAs should issue a cross-certificate including a policyMapping extension.

**[X.509 8.1.3]**

CA.08: CAs should issue a cross-certificate including plural policyMapping extension.

**[X.509 8.1.3]**

CA.09: CAs should issue a cross-certificate including cA flag set to TRUE in critical basicConstraints extension, except for self-signed certificate. This assertion is the same as Int.SH.CA.01 requirement.

**[X.509 8.4.2.1]**

CA.10: CAs should issue a cross-certificate including keyCertSign in critical keyUsage extension, except for self-signed certificate.

**[X.509 8.2.2.3]**

CA.11: CAs should issue a cross-certificate including pathLenConstraints in critical basicConstraints extension, except for self-signed certificate. This assertion is the same as Int.SH.CA.02 requirement.

**[X.509 8.4.2.1]**

CA.12: CAs should issue a cross-certificate including a critical policyConstraints extension.

**[X.509 10.5.2, 10.5.3]**

CA.13: CAs should issue a cross-certificate including a critical nameConstraints extension.

**[X.509 10.5.2]**

CA.14: CAs should issue a cross-certificate including a critical inhibitAnyPolicy extension.

**[X.509 10.5.2]**

CA.15-18: CAs should issue a certificate that anybody can find out the revocation information.

**[IWG profile]**

RP.19: The application should validate successfully correct certification path.

RP.20-21: The application should validate a certification path including a cross-certificate.

**[X.509 8.1.2]**

RP.22-25: The application should process certificatePolicy in all certificates for validating certification path.

**[X.509 8.1.1]**

RP.26-28: The application should ensure whether all cross-certificates in the certification path have cA flag set to TRUE in critical basicConstraints extension.

**[X.509 10.5.1]**

RP.29: The application should ensure whether the certification path length is shorter than pathLenConstraints or not in any cross-certificate.

**[X.509 10.5.1]**

RP.30-32: The application should ensure whether all cross-certificates have keyCertSign in critical keyUsage extension.

**[IWG profile]**

RP.33-34: The application should process policyConstraints extension in all cross-certificates for validating certification path.

**[X.509 10.5.2, 10.5.3]**

RP.35-37: The application should process nameConstraints extension in all cross-certificates for validating certification path.

**[X.509 10.5.2, 10.5.3]**

RP.38: The application should ensure whether all certificates in certification path are revoked or not.

**[X.509 10.5.1]**

RP.39: The application should verify all cross-certificates in certification path by its issuer certificate.

**[X.509 10.5.1]**

(c) Cross Recognition

CA.01: CAs should issue CA certificates including a policyIdentifier in critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.4 requirement.

**[X.509 8.2.2.6]**

CA.02: CAs should issue CA certificates including plural policyIdentifier in critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.5 requirement.

**[X.509 8.2.2.6]**

CA.03: CAs should issue CA certificates including a policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.6 requirement.

**[X.509 8.2.2.6]**

CA.04: CAs should issue CA certificates including plural policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.7 requirement.

**[X.509 8.2.2.6]**

RP.05: The application should validate successfully correct certification path.

RP.06-08: The application should validate a certification path including other PKI domain certificates from its trust list.

**[IWG profile]**

RP.09: The application should verify whether trust anchor certificate in certification path was altered or not.

**[X.509 10.5.1]**

RP.10-13: The application should process certificatePolicy in all certificates for validating certification path.

**[X.509 8.1.1]**

(d) Mesh (in the future)

TBD in the future.

(e) Bridge CA (in the future)

TBD in the future.

(f) Accreditation Certificate (in the future)

TBD in the future.

(g) Certificate Trust Lists (in the future)

TBD in the future.

(3) Service requirements

(a) Signing

CA.01: CAs should issue an EE certificate including digitalSignature in critical keyUsage extension.

**[IWG profile]**

CA.02: CAs should issue a CA certificates including a policyIdentifier in critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.4 requirement.

**[X.509 8.2.2.6]**

CA.03: CAs should issue a CA certificates including plural policyIdentifier in critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.5 requirement.

**[X.509 8.2.2.6]**

CA.04: CAs should issue a CA certificates including a policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.6 requirement.

**[X.509 8.2.2.6]**

CA.05: CAs should issue a CA certificates including plural policyIdentifier in non-critical certificatePolicies extension, except for self-signed certificate. This assertion is tested by testing Int.SH.CA.7 requirement.

**[X.509 8.2.2.6]**

RP.06: The application should validate successfully correct certification path.

RP.07: The application should ensure whether the subscriber certificate has an appropriate usage in critical keyUsage extension.

**[IWG consideration]**

RP.08-11: The application should process certificatePolicy in all certificates for validating certification path.

**[X.509 8.1.1]**

(b) Notary (in the future)

TBD in the future.

*May require TSA*

*Cross-Recognition cannot validate a signed-data that existed before particular time, because trust relationship is established by no signature..*

*Require signingTime in the signed-data.*

*Provide the necessary information to validate a certificate (e.g., CRLDP) as to refer from other domains.*

(c) Authentication (in the future)

TBD in the future.

*Require setting digitalSignature to keyUsage.*

*Require setting the attribute (e.g., e-mail, iPAddress or dNSName) of entity to subjectAltName.*

*MAY Require extendedKeyUsage*

(d) Encryption (in the future)

TBD in the future.

*Obtain a certificate via trustworthy way.*

*Obtaining a certificate from out-of-band is not trusted in multi domain PKI.*

(4) Revocation/Validation requirements

(a) CRL

*Be able to obtain appropriate CRL even if other domain EE.*

*If each CRL is different in revocation information, it should be recognized by other domain EE.*

CA.01: CAs should issue a CA (CRL issuer) certificate including CRLSign in critical keyUsage extension.

**[IWG profile]**

CA.02: CAs should issue a revocation list including a critical issuingDistributionPoints extension.

**[IWG profile]**

CA.03: CAs should issue a CRL including an onlyContainsUserCerts flag set to TRUE in a critical issuingDistributionPoints extension.

**[X.509 8.6.2.2, RFC3280 5.2.5]**

CA.04: CAs should issue an ARL including an onlyContainsCACerts flag set to TRUE in a critical issuingDistributionPoints extension.

**[X.509 8.6.2.2, RFC3280 5.2.5]**

CA.05: CAs should issue a certificate including distributionPoint, when it is not CA entry, in cRLDistributionPoints extension.

**[X.509 8.6.2.2, RFC3280 5.2.5]**

CA.06: CAs should issue a revocation list including distributionPoint, which is consistent with CRLDistributionPoints extension of the certificate they issue, in issuingDistributionPoint extension.

**[RFC3280 5.2.5]**

CA.07: CAs should issue a revocation list including keyIdentifier in authorityKeyIdentifier extension.

**[IWG profile]**

RP.08: The application should validate successfully correct certification path.

RP.09-10: The application should associate a CRL with a certificate to verify.

**[X.509 10.5.1]**

RP.11: The application should ensure whether the revocationDate of the

certificate is valid or not.

**[IWG consideration]**

RP.12: The application should verify a revocation list by the revocation list issuer certificate.

**[RFC3280 6.3.3 (b)]**

RP.13: The application should ensure whether the revocation list issuer certificate has CRLSign in critical keyUsage extension.

**[RFC3280 6.3.3 (f)]**

RP.14: The application should verify whether revocation list was altered or not.

**[X.509 10.5.1, RFC3280 6.3.3 (g)]**

RP.15-16: The application should process appropriately a revocation list including an unknown/well-known CRL entry extension if it is critical or not.

**[X.509 8]**

RP.17-18: The application should process appropriately a revocation list including an unknown/well-known CRL extension if it is critical or not.

**[X.509 8]**

RP.19-20: The application should process appropriately a certificate when using a revocation list including an onlyContainsUserCerts flag set to TRUE in critical issuingDistributionPoint extension. The certificate has no basicConstraints extension.

**[RFC3280 6.3.3 (b)]**

RP.21-22: The application should process appropriately a certificate when using a revocation list including an onlyContainsCACerts flag set to TRUE in critical issuingDistributionPoint extension. The certificate has cA flag set to TRUE in critical basicConstraints extension.

**[RFC3280 6.3.3 (b)]**

RP.23-24: The application should process appropriately a certificate when using a revocation list including an onlyContainsUserCerts flag set to TRUE in critical issuingDistributionPoint extension. The certificate has no basicConstraints extension.

**[RFC3280 6.3.3 (b)]**

RP.25-26: The application should process appropriately a certificate when using a revocation list including an onlyContainsUserCerts flag set to TRUE in critical issuingDistributionPoint extension. The certificate has cA flag set to TRUE in critical basicConstraints extension.

**[RFC3280 6.3.3 (b)]**

RP.27-31: The application should ensure whether each distributionPoint are

consistent between a critical issuingDistributionPoint extension in the revocation list and a cRLDistributionPoints extension in the certificate.

**[RFC3280 5.2.5]**

(b) OCSP (in the future)

TBD in the future.

*Trustworthiness to OCSP Responder --- require a validation method for foreign OCSP Responder.*

*Reachability to OCSP Responder --- URI in AIA should be internet URI.*

*Reduce the overhead of network transaction.*

(c) Delegated Path Discovery/Validation (in the future)

TBD in the future.

## 2.3 Testing Assumptions

### 2.3.1 Base model

(a) Entity

Root CA: the only CA which has its self-signed certificate

Subscriber: the end entity whose certificate has been signed by RootCA

Relying Party: the end entity who validates the data signed by subscriber.

(b) Base profile

The followings are only profiles as a summary of certificate in the experiment.

Table 2.1 Base model Certificate Profile

| Field | critical flag | Root CA | Sub scriber | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| serialNumber | - | x | x | |
| signature | - | x | x | 2 |
| validity | - | x | x | 3 |
| issuer | - | x | x | 4 |

| | | | | |
|---|---|---|---|---|
| subject | - | x | x | 4 |
| subjectPublicKeyInfo | - | x | x | 5 |
| issuerUniqueID | - | - | - | |
| subjectUniqueID | - | - | - | |
| authorityKeyIdentifier | n | - | x | |
| keyIdentifier | - | - | x | 6 |
| subjectKeyIdentifier | n | x | x | 6 |
| keyUsage | c | - | x | 7 |
| certificatePolicies | c | - | - | |
| policyMappings | n | - | - | |
| subjectAltName | n | - | - | |
| basicConstraints | c | - | - | |
| policyConstraints | c | - | - | |
| cRLDistributionPoints | n | - | x | |
| distributionPoint | - | - | x | |
| fullName | - | - | x | 8 |
| 1 v3(2) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTCTime | | | | |
| 4 UTF8String | | | | |
| 5 rsaEncryption (1 2 840 113549 1 1 1) | | | | |
| 6 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 7 only digitalSignature | | | | |
| 8 directoryName or URI | | | | |

Table 2.2 Base model CRL Profile

| Field | critical flag | CRL | ARL | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| signature | - | x | x | 2 |
| issuer | - | x | x | 3 |
| thisUpdate | - | x | x | 4 |
| nextUpdate | - | x | x | 4 |
| RevockedCertificates | - | x | x | |

| | | | | |
|---|---|---|---|---|
| userCertificate | - | x | x | |
| revocationDate | - | x | x | 4 |
| crlEntryExtensions | | - | - | |
| authorityKeyIdentifier | n | x | x | |
| keyIdentifier | - | x | x | 5 |
| cRLNumber | n | - | - | |
| issuingDistributionPoint | c | x | x | |
| distributionPoint | - | x | x | |
| fullName | - | x | x | 6 |
| onlyContainsUserCerts | - | x | - | |
| onlyContainsCACerts | - | - | x | |
| 1 v2(1) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTF8String | | | | |
| 4 UTCTime | | | | |
| 5 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 6 directoryName or URI | | | | |

(c) Inputs for validation

user-initial-policy-set: any-policy

trustAnchorInfo: Root CA

initial-explicit-policy: false

2.3.2 Interconnection model

(1) Strict Hierarchy

(a) Entity

RootCA: the only CA which has self-signed certificate

SubCA-1: the CA which has had its certificate signed by RootCA

Subscriber-1: the end entity whose certificate has been signed by SubCA-1

SubCA-2: the CA which has had its certificate signed by SubCA-1

Subscriber-2: the end entity whose certificate has been signed by SubCA-2

(b) Base profile

The followings are only profiles as a summary of certificates in the experiment.

Table 2.3 Strict Hierarchy Base Certificate Profile

| Field | critical flag | Root CA | Sub CA | Sub scriber | note |
|---|---|---|---|---|---|
| version | - | x | x | x | 1 |
| serialNumber | - | x | x | x | |
| signature | - | x | x | x | 2 |
| validity | - | x | x | x | 3 |
| issuer | - | x | x | x | 4 |
| subject | - | x | x | x | 4 |
| subjectPublicKeyInfo | - | x | x | x | 5 |
| issuerUniqueID | - | - | - | - | |
| subjectUniqueID | - | - | - | - | |
| authorityKeyIdentifier | n | - | x | x | |
| keyIdentifier | - | - | x | x | 6 |
| subjectKeyIdentifier | n | x | x | x | 6 |
| keyUsage | c | - | - | x | 7 |
| certificatePolicies | c | - | x | x | |
| policyIdentifier | - | - | x | x | 8 |
| policyQualifiers | - | - | - | - | |
| policyMappings | n | - | - | - | |
| subjectAltName | n | - | - | - | |
| basicConstraints | c | - | x | - | |
| cA | - | - | x | x | |
| pathLenConstraint | - | - | - | - | |
| policyConstraints | c | - | - | - | |
| cRLDistributionPoints | n | - | - | x | |
| distributionPoint | - | - | - | x | |
| fullName | - | - | - | x | 9 |
| 1 v3(2) | | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | | |
| 3 UTCTime | | | | | |
| 4 UTF8String | | | | | |
| 5 rsaEncryption (1 2 840 113549 1 1 1) | | | | | |
| 6 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | | |
| 7 only digitalSignature | | | | | |

| |
|---|
| 8 consistent policyIdentifier |
| 9 directoryName or URI |

Table 2.4 Strict Hierarchy Base CRL Profile

| Field | critical flag | CRL | ARL | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| signature | - | x | x | 2 |
| issuer | - | x | x | 3 |
| thisUpdate | - | x | x | 4 |
| nextUpdate | - | x | x | 4 |
| RevockedCertificates | - | x | x | |
| userCertificate | - | x | x | |
| revocationDate | - | x | x | 4 |
| crlEntryExtensions | | - | - | |
| authorityKeyIdentifier | n | x | x | |
| keyIdentifier | - | x | x | 5 |
| cRLNumber | n | - | - | |
| issuingDistributionPoint | c | x | x | |
| distributionPoint | - | x | x | |
| fullName | - | x | x | 6 |
| onlyContainsUserCerts | - | x | - | |
| onlyContainsCACerts | - | - | x | |
| 1 v2(1) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTF8String | | | | |
| 4 UTCTime | | | | |
| 5 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 6 directoryName or URI | | | | |

(c) Inputs for validation

user-initial-policy-set: policy-A

trustAnchorInfo: Root CA

initial-explicit-policy: true

(2) Cross Certification

(a) Entity

RootCA-X: the CA which has its self-signed certificate

RootCA-Y: the CA which has achieved Cross-Certification relationship with RootCA-X

Subscriber-Y: the end entity whose certificate has been signed by RootCA-Y

RootCA-Z: the CA which has achieved Cross-Certification relationship with RootCA-Y

Subscriber-Z: the end entity whose certificate has been signed by RootCA-Z

(b) Base profile

The followings are only profiles as a summary of certificates in the experiment. .

Table 2.5 Cross Certification Base Certificate Profile

| Field | critical flag | Root CA | Cross Cert | Sub scriber | note |
|---|---|---|---|---|---|
| version | - | x | x | x | 1 |
| serialNumber | - | x | x | x | |
| signature | - | x | x | x | 2 |
| validity | - | x | x | x | 3 |
| issuer | - | x | x | x | 4 |
| subject | - | x | x | x | 4 |
| subjectPublicKeyInfo | - | x | x | x | 5 |
| issuerUniqueID | - | - | - | - | |
| subjectUniqueID | - | - | - | - | |
| authorityKeyIdentifier | n | - | x | x | |
| keyIdentifier | - | - | x | x | 6 |
| subjectKeyIdentifier | n | x | x | x | 6 |
| keyUsage | c | - | x | x | 7 |
| certificatePolicies | c | - | x | x | |
| policyMappings | n | - | x | - | |
| subjectAltName | n | - | - | - | |
| basicConstraints | c | - | x | - | |
| cA | - | - | x | - | |
| pathLenConstraint | - | - | - | - | |
| policyConstraints | c | - | - | - | |

| | | | | | |
|---|---|---|---|---|---|
| cRLDistributionPoints | n | - | x | x | |
| distributionPoint | - | x | x | x | |
| fullName | - | x | x | x | 8 |
| 1 v3(2) | | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | | |
| 3 UTCTime | | | | | |
| 4 UTF8String | | | | | |
| 5 rsaEncryption (1 2 840 113549 1 1 1) | | | | | |
| 6 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | | |
| 7 only digitalSignature | | | | | |
| 8 directoryName or URI | | | | | |

Table 2.6 Cross Certification Base CRL Profile

| Field | critical flag | CRL | ARL | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| signature | - | x | x | 2 |
| issuer | - | x | x | 3 |
| thisUpdate | - | x | x | 4 |
| nextUpdate | - | x | x | 4 |
| RevockedCertificates | - | x | x | |
| userCertificate | - | x | x | |
| revocationDate | - | x | x | 4 |
| crlEntryExtensions | | - | - | |
| authorityKeyIdentifier | n | x | x | |
| keyIdentifier | - | x | x | 5 |
| cRLNumber | n | - | - | |
| issuingDistributionPoint | c | x | x | |
| distributionPoint | - | x | x | |
| fullName | - | x | x | 6 |
| onlyContainsUserCerts | - | x | - | |
| onlyContainsCACerts | - | - | x | |
| 1 v2(1) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |

| |
|---|
| 3 UTF8String |
| 4 UTCTime |
| 5 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) |
| 6 directoryName or URI |

(c) Inputs for validation

user-initial-policy-set: policy-X

trustAnchorInfo: Root CA-X

initial-explicit-policy: true

(3) Cross Recognition

(a) Entity

RootCA-X: the CA which has self-signed certificate

RootCA-Y: the CA which has achieved Cross-Recognition relationship with RootCA-X

Subscriber-Y: the end entity whose certificate has been signed by RootCA-Y

(b) Base profile

The followings are only profiles as a summary of certificates in the experiment. .

Table 2.7 Cross Recognition Base Certificate Profile

| Field | critical flag | Root CA | Sub scriber | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| serialNumber | - | x | x | |
| signature | - | x | x | 2 |
| validity | - | x | x | 3 |
| issuer | - | x | x | 4 |
| subject | - | x | x | 4 |
| subjectPublicKeyInfo | - | x | x | 5 |
| issuerUniqueID | - | - | - | |
| subjectUniqueID | - | - | - | |
| authorityKeyIdentifier | n | - | x | |
| keyIdentifier | - | - | x | 6 |

| | | | | |
|---|---|---|---|---|
| subjectKeyIdentifier | n | x | x | 6 |
| keyUsage | c | - | x | 7 |
| certificatePolicies | c | - | x | |
| policyIdentifier | - | - | x | 8 |
| policyQualifiers | - | - | - | |
| policyMappings | n | - | - | |
| subjectAltName | n | - | - | |
| basicConstraints | c | - | - | |
| policyConstraints | c | - | - | |
| cRLDistributionPoints | n | - | x | |
| distributionPoint | - | - | x | |
| fullName | - | - | x | 9 |
| 1 v3(2) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTCTime | | | | |
| 4 UTF8String | | | | |
| 5 rsaEncryption (1 2 840 113549 1 1 1) | | | | |
| 6 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 7 only digitalSignature | | | | |
| 8 consistent policyIdentifier | | | | |
| 9 directoryName or URI | | | | |

Table 2.8 Cross Recognition Base CRL Profile

| Field | critical flag | CRL | ARL | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| signature | - | x | x | 2 |
| issuer | - | x | x | 3 |
| thisUpdate | - | x | x | 4 |
| nextUpdate | - | x | x | 4 |
| RevockedCertificates | - | x | x | |
| userCertificate | - | x | x | |
| revocationDate | - | x | x | 4 |

| | | | | |
|---|---|---|---|---|
| crlEntryExtensions | | - | - | |
| authorityKeyIdentifier | n | x | x | |
| keyIdentifier | - | x | x | 5 |
| cRLNumber | n | - | - | |
| issuingDistributionPoint | c | x | x | |
| distributionPoint | - | x | x | |
| fullName | - | x | x | 6 |
| onlyContainsUserCerts | - | x | - | |
| onlyContainsCACerts | - | - | x | |
| 1 v2(1) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTF8String | | | | |
| 4 UTCTime | | | | |
| 5 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 6 directoryName or URI | | | | |

(c) Inputs for validation

user-initial-policy-set: policy-X, policy-Y

trustAnchorInfo: Root CA-X, RootCA-Y

initial-explicit-policy: true

(4) Mesh

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

(5) Bridge CA

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

(6) Accreditation Certificate

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

(7) Certificate Trust Lists

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

2.3.3 Service model

(1) Signing

(a) Entity

RootCA: the only CA which has self-signed certificate

Subscriber: the end entity whose certificate is issued by RootCA

(b) Base profile

The followings are only profiles as a summary of certificates in the experiment. .

Table 2.9 Signing Base Certificate Profile

| Field | critical flag | Root CA | Sub scriber | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| serialNumber | - | x | x | |
| signature | - | x | x | 2 |
| validity | - | x | x | 3 |
| issuer | - | x | x | 4 |
| subject | - | x | x | 4 |
| subjectPublicKeyInfo | - | x | x | 5 |
| issuerUniqueID | - | - | - | |
| subjectUniqueID | - | - | - | |
| authorityKeyIdentifier | n | - | x | |
| keyIdentifier | - | - | x | 6 |
| subjectKeyIdentifier | n | x | x | 6 |
| keyUsage | c | - | x | 7 |
| certificatePolicies | c | - | x | |
| policyIdentifier | - | - | x | 8 |

| | | | | |
|---|---|---|---|---|
| policyQualifiers | - | - | - | |
| policyMappings | n | - | - | |
| subjectAltName | n | - | - | |
| basicConstraints | c | - | - | |
| policyConstraints | c | - | - | |
| cRLDistributionPoints | n | - | x | |
| distributionPoint | - | - | x | |
| fullName | - | - | x | 9 |
| 1 v3(2) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTCTime | | | | |
| 4 UTF8String | | | | |
| 5 rsaEncryption (1 2 840 113549 1 1 1) | | | | |
| 6 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 7 only digitalSignature | | | | |
| 8 consistent policyIdentifier | | | | |
| 9 directoryName or URI | | | | |

Table 2.10 Signing Base CRL Profile

| Field | critical flag | CRL | ARL | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| signature | - | x | x | 2 |
| issuer | - | x | x | 3 |
| thisUpdate | - | x | x | 4 |
| nextUpdate | - | x | x | 4 |
| RevockedCertificates | - | x | x | |
| userCertificate | - | x | x | |
| revocationDate | - | x | x | 4 |
| crlEntryExtensions | | - | - | |
| authorityKeyIdentifier | n | x | x | |
| keyIdentifier | - | x | x | 5 |
| cRLNumber | n | - | - | |

| | | | | |
|---|---|---|---|---|
| issuingDistributionPoint | c | x | x | |
| distributionPoint | - | x | x | |
| fullName | - | x | x | 6 |
| onlyContainsUserCerts | - | x | - | |
| onlyContainsCACerts | - | - | x | |
| 1 v2(1) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTF8String | | | | |
| 4 UTCTime | | | | |
| 5 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 6 directoryName or URI | | | | |

(c) Inputs for validation

user-initial-policy-set: policy-A

trustAnchorInfo: Root CA

initial-explicit-policy: true

(2) Notary

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

(3) Authentication

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

(4) Encryption

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

### 2.3.4 Revocation/Validation model

(1) CRL

(a) Entity

RootCA-A: the only CA which has self-signed certificate

Subscriber-A: the end entity whose certificate is issued by RootCA-A

SubCA: the CA which has had is certificate issued by RootCA-A

Subscriber-SubCA: the end entity whose certificate has been signed by SubCA

(b) Base profile

The followings are only profiles as a summary of certificates in the experiment. .

Table 2.11 CRL Base Certificate Profile

| Field | critical flag | Root CA | Sub scriber | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| serialNumber | - | x | x | |
| signature | - | x | x | 2 |
| validity | - | x | x | 3 |
| issuer | - | x | x | 4 |
| subject | - | x | x | 4 |
| subjectPublicKeyInfo | - | x | x | 5 |
| issuerUniqueID | - | - | - | |
| subjectUniqueID | - | - | - | |
| authorityKeyIdentifier | n | - | x | |
| keyIdentifier | - | - | x | 6 |
| subjectKeyIdentifier | n | x | x | 6 |
| keyUsage | c | - | x | 7 |
| certificatePolicies | c | - | x | |
| policyIdentifier | - | - | x | 8 |
| policyQualifiers | - | - | - | |
| policyMappings | n | - | - | |

| | | | | |
|---|---|---|---|---|
| subjectAltName | n | - | - | |
| basicConstraints | c | - | - | |
| policyConstraints | c | - | - | |
| cRLDistributionPoints | n | - | x | |
| distributionPoint | - | - | x | |
| fullName | - | - | x | 9 |
| 1 v3(2) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTCTime | | | | |
| 4 UTF8String | | | | |
| 5 rsaEncryption (1 2 840 113549 1 1 1) | | | | |
| 6 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 7 only digitalSignature | | | | |
| 8 consistent policyIdentifier | | | | |
| 9 directoryName or URI | | | | |

Table 2.12 CRL Base CRL Profile

| Field | critical flag | CRL | ARL | note |
|---|---|---|---|---|
| version | - | x | x | 1 |
| signature | - | x | x | 2 |
| issuer | - | x | x | 3 |
| thisUpdate | - | x | x | 4 |
| nextUpdate | - | x | x | 4 |
| RevockedCertificates | - | x | x | |
| userCertificate | - | x | x | |
| revocationDate | - | x | x | 4 |
| crlEntryExtensions | | - | - | |
| authorityKeyIdentifier | n | x | x | |
| keyIdentifier | - | x | x | 5 |
| cRLNumber | n | - | - | |
| issuingDistributionPoint | c | x | x | |
| distributionPoint | - | x | x | |

| | | | | |
|---|---|---|---|---|
| fullName | - | x | x | 6 |
| onlyContainsUserCerts | - | x | - | |
| onlyContainsCACerts | - | - | x | |
| 1 v2(1) | | | | |
| 2 sha1withRSAEncryption (1 2 840 113549 1 1 5) | | | | |
| 3 UTF8String | | | | |
| 4 UTCTime | | | | |
| 5 160bit SHA-1 aka RFC3280 "4.2.1.2 Subject Key Identifier" (1) | | | | |
| 6 directoryName or URI | | | | |

(c) Inputs for validation

user-initial-policy-set: unspecified

trustAnchorInfo: Root CA-A

initial-explicit-policy: unspecified

(2) OCSP

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

(3) Delegated Path Discovery/Validation

TBD in the future.

(a) Entity

(b) Base profile

(c) Inputs for validation

## 2.4 Testing Items for Base model

Microsoft Excel
? ? ? ? ?

*See the detail at the end of this guideline.*

## 2.5 Testing Items for Interconnection model

### 2.5.1 Strict Hierarchy


*See at the end of this guideline.*

### 2.5.2 Cross Certification


*See at the end of this guideline.*

### 2.5.3 Cross Recognition


*See at the end of this guideline.*

### 2.5.4 Mesh

(TBD in the future)

### 2.5.5 Bridge CA

(TBD in the future)

### 2.5.6 Accreditation Certificate

(TBD in the future)

### 2.5.7 Certificate Trust Lists

(TBD in the future)

## 2.6 Testing Items for Service model

### 2.6.1 Signing


*See at the end of this guideline.*

### 2.6.2 Notary

(TBD in the future)

### 2.6.3 Authentication

(TBD in the future)

### 2.6.4 Encryption

(TBD in the future)

## 2.7 Testing Items for Revocation/Validation model

### 2.7.1 CRL

Microsoft Excel
??????

*See at the end of this guideline.*

### 2.7.2 OCSP

(TBD in the future)

### 2.7.3 Delegated Path Discovery/Validation

(TBD in the future)