Network Working Group
Request for Comments: 5217
Category: Informational

M. Shimaoka, Ed.
SECOM
N. Hastings
NIST
R. Nielsen
Booz Allen Hamilton
July 2008

Memorandum for Multi-Domain Public Key Infrastructure Interoperability

Status of This Memo

Abstract

The objective of this document is to establish a terminology framework and to suggest the operational requirements of Public Key Infrastructure (PKI) domain for interoperability of multi-domain Public Key Infrastructure, where each PKI domain is operated under a distinct policy. This document describes the relationships between Certification Authorities (CAs), provides the definition and requirements for PKI domains, and discusses typical models of multi-domain PKI.

Table of Contents

1. Introduction

1.1. Objective

The objective of this document is to establish a terminology framework and to provide the operational requirements, which can be used by different Public Key Infrastructure (PKI) authorities who are considering establishing trust relationships with each other. The document defines different types of possible trust relationships, identifies design and implementation considerations that PKIs should implement to facilitate trust relationships across PKIs, and identifies issues that should be considered when implementing trust relationships. This document defines terminology and interoperability requirements for multi-domain PKIs from one perspective. A PKI domain can achieve multi-domain PKI interoperability by complying with the requirements in this document. However, there are other ways to define and realize multi-domain PKI interoperability.

1.2. Document Outline

Section 2 introduces the PKI basics, which provide a background for multi-domain PKI. Section 3 provides the definitions and requirements of 'PKI domain' and describes the typical models of multi-domain PKI. Section 4 considers the Trust List Models depending on relying party-CA relationships (not CA-CA trust relationships, as they are not a focus of this document). Section 5 identifies abbreviations used in the document.

2. Public Key Infrastructure (PKI) Basics

2.1. Basic Terms

The following terms are used throughout this document. Where possible, definitions found in RFC 4949 [RFC4949] have been used.

Certificate: A digitally signed data structure that attests to the binding of a system entity's identity to a public key value (based on the definition of public key certificate in RFC 4949 [RFC4949]).

Certificate Policy: A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements (X.509 [CCITT.X509.2000]). Note that to avoid confusion, this document uses the terminology "Certificate Policy Document" to refer to the document that defines the rules and "Policy Object Identifier (OID)" to specify a particular rule set.

Certificate Policy Document: A document that defines the rules for the issuance and management of certificates and identifies Policy Object Identifiers (OIDs) for these rules. A Certificate Policy Document may define more than one Policy OID.

Policy Object Identifier (Policy OID): An identifier applied to a set of rules governing the issuance and management of certificates. Policy OIDs are defined in the Certificate Policy Documents.

Certification Authority (CA): An entity that issues certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate (RFC 4949 [RFC4949]).

End Entity (EE): A system entity that is the subject of a certificate and that is using, or is permitted and able to use, the matching private key only for a purpose or purposes other than signing a certificate; i.e., an entity that is not a CA (RFC 4949 [RFC4949]).

Relying party: A system entity that depends on the validity of information (such as another entity's public key value) provided by a certificate (from the RFC 4949 [RFC4949] definition of certificate user).

2.2. Relationships between Certification Authorities

CAs establish trust relationships by issuing certificates to other CAs. CA relationships are divided into 'certification hierarchy' [RFC4949] and 'cross-certification' [RFC4949].

In a certification hierarchy, there are two types of CAs: 'superior CA' and 'subordinate CA', as described in RFC 4949 [RFC4949].

Superior CA: A CA that is an issuer of a subordinate CA certificate.

A cross-certification can be either unilateral or bilateral.

Unilateral cross-certification: Cross-certification of one CA (CA1) by another CA (CA2) but no cross-certification of CA2 by CA1.

Bilateral cross-certification: Cross-certification of one CA (CA1) by another CA (CA2) and cross-certification of CA2 by CA1.

2.2.1. Hierarchical CA Relationships

In a hierarchical relationship, as shown in Figure 1, one CA assumes a parent relationship to the other CA.

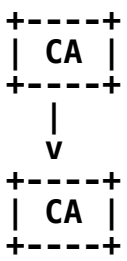

Figure 1: Hierarchical CA Relationship

There are two types of hierarchical relationships, depending on whether a subordinate CA certificate or a unilateral cross-certificate is used. In the case where one (superior) CA issues a subordinate CA certificate to another, the CA at the top of the hierarchy, which must itself have a self-signed certificate, is called a root CA. In the case where one CA issues unilateral cross-certificates to other CAs, the CA issuing unilateral cross-certificates is called a Unifying CA. Unifying CAs use only unilateral cross-certificates.

NOTE: In this document, the definition of root CA is according to the second definition (context for hierarchical PKI) of 'root CA' in RFC 4949 [RFC4949]. This document uses the terminology 'trust anchor CA' for the first definition (context for PKI) of 'root CA' in RFC 4949.

Root CA: A CA that is at the top of a hierarchy, and itself should not issue certificates to end entities (except those required for its own operation) but issues subordinate CA certificates to one or more CAs.

Subordinate CA: A CA whose public key certificate is issued by another superior CA, and itself must not be used as a trust anchor CA.

Unifying CA: A CA that is at the top of a hierarchy, and itself should not issue certificates to end entities (except those required for its own operation) but establishes unilateral cross-certification with other CAs. A Unifying CA must permit CAs to which it issues cross-certificates to have self-signed certificates.

2.2.2. Peer-to-Peer CA Relationships

In a peer relationship, no parent-child relationship is created. To establish peer relationships, only cross-certificates are used. Peer relationships can be either unilateral or bilateral, as shown in Figure 2.

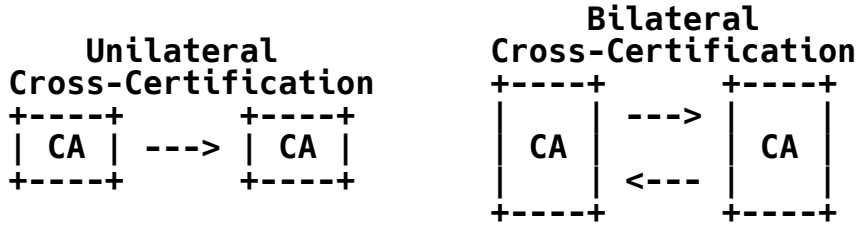

Figure 2: Peer-to-Peer CA Relationships

In the case where a CA exists only to manage cross-certificates, that CA is called a Bridge CA. CAs can establish unilateral or bilateral cross-certification with a Bridge CA, as shown in Figure 3.

Bridge CA: A CA that, itself, does not issue certificates to end entities (except those required for its own operation) but establishes unilateral or bilateral cross-certification with other CAs.

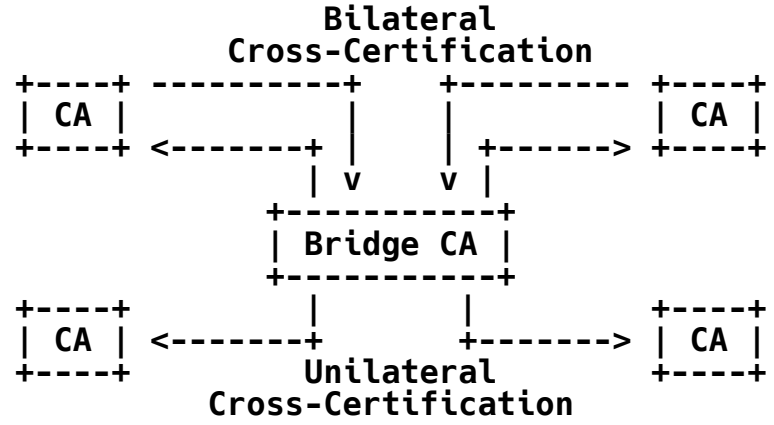

Figure 3: Bridge CA

2.3. Public Key Infrastructure (PKI) Architectures

Public Key Infrastructure (PKI): A system of CAs that perform some set of certificate management, archive management, key management, and token management functions for a community of users in an application of asymmetric cryptography and share trust relationships, operate under the same Certificate Policy Document specifying a shared set of Policy OID(s), and are either operated by a single organization or under the direction of a single organization.

In addition, a PKI that intends to enter into trust relationships with other PKIs must designate a Principal CA (PCA) that will manage all trust relationships. This Principal CA should also be the trust anchor CA for relying parties of that PKI.

Principal CA (PCA): A CA that should have a self-signed certificate is designated as the CA that will issue cross-certificates to Principal CAs in other PKIs, and may be the subject of cross-certificates issued by Principal CAs in other PKIs.

In discussing different possible architectures for PKI, the concept of a certification path is necessary. A certification path is built based on trust relationships between CAs.

Certification Path: An ordered sequence of certificates where the subject of each certificate in the path is the issuer of the next certificate in the path. A certification path begins with a trust anchor certificate and ends with an end entity certificate.

2.3.1. Single CA Architecture

Definition: A simple PKI consists of a single CA with a self-signed certificate that issues certificates to End Entities (EEs), as shown in Figure 4.

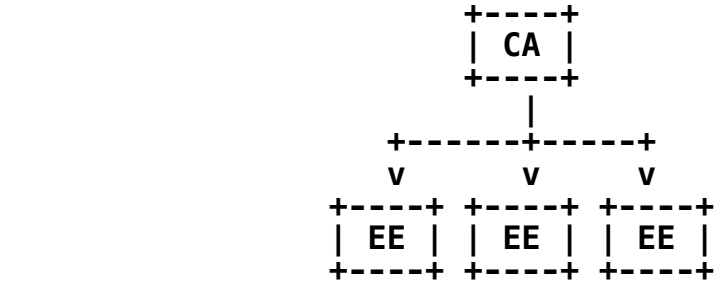

Figure 4: Simple PKI Architecture

Trust anchor CA: The trust anchor CA must be the CA that has a self-signed certificate.

Principal CA: Since this PKI architecture has one CA, the Principal CA must be that CA.

2.3.2. Multiple CA Architectures

2.3.2.1. Hierarchical PKI Architecture

Definition: A hierarchical PKI consists of a single root CA and one or more subordinate CAs that issue certificates to EEs. A hierarchical PKI may have intermediate CAs, which are subordinate CAs that themselves have subordinate CAs. The root CA must distribute a trust anchor (public key and associated data), but the format and protocol are irrelevant for this specification. And all subordinate CAs must have subordinate CA certificates, as shown in Figure 5.

Trust anchor CA: The trust anchor CA must be the root CA.

Principal CA: The Principal CA must be the root CA.

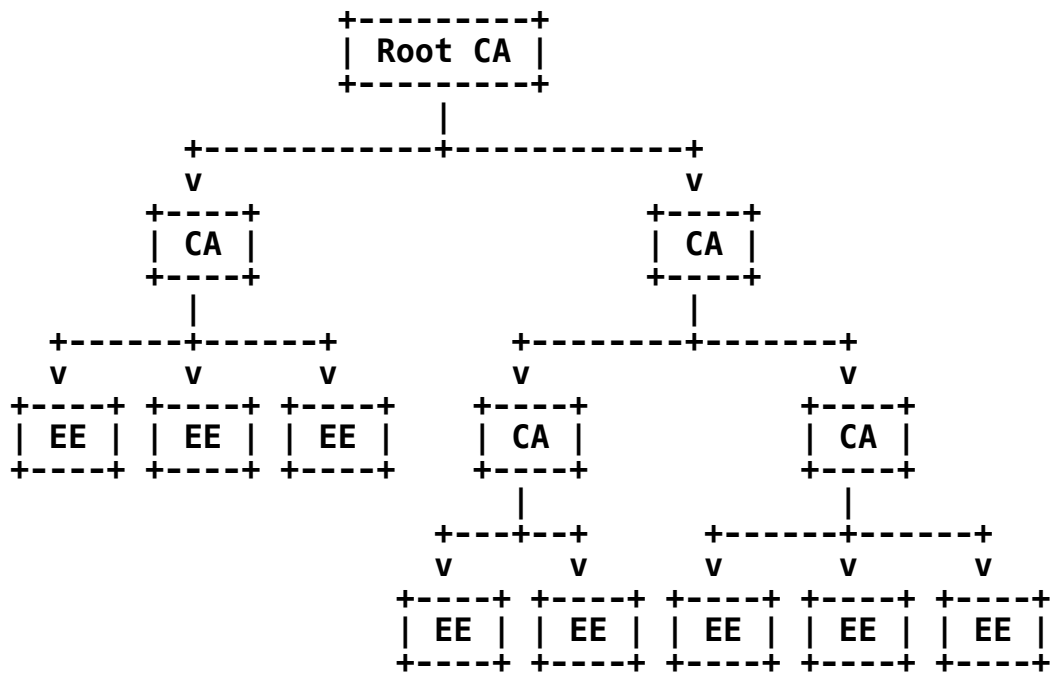

Figure 5: Hierarchical PKI Architecture

2.3.2.2. Mesh PKI Architectures

Definition: A mesh PKI consists of multiple CAs with self-signed certificates that issue certificates to EEs and issue cross-certificates to each other. A mesh PKI may be a full mesh, where all CAs issue cross-certificates to all other CAs, as shown in Figure 6. A mesh PKI may also be a partial mesh, where all CAs do not issue cross-certificates to all other CAs. In a partial mesh PKI, certification paths may not exist from all CAs to all other CAs, as shown in Figure 7.

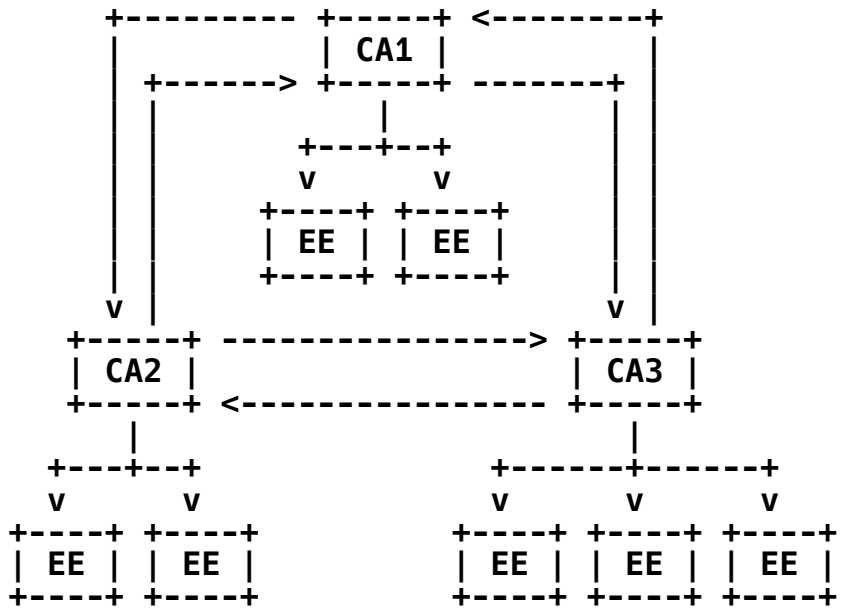

Figure 6: Full Mesh PKI Architecture

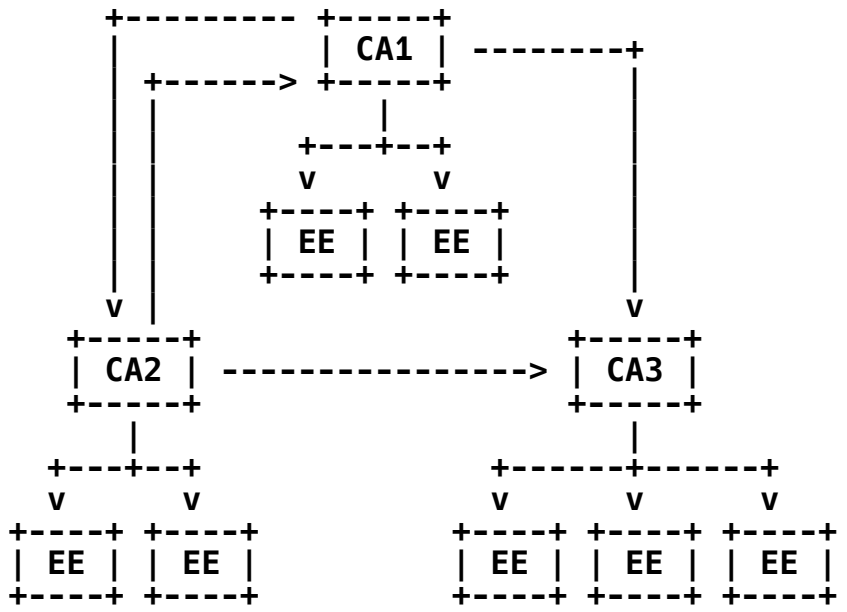

Figure 7: Partial Mesh PKI Architecture

Trust anchor CA: The trust anchor CA for an end entity is usually the CA that issued the end entity's certificate. The trust anchor CA for an end entity that is not issued a certificate from the mesh PKI may be any CA in the PKI. In a partial mesh, selection of the trust anchor may result in no certification path from the trust anchor to one or more CAs in the mesh. For example, in Figure 7 above, the selection of CA1 or CA2 as the trust anchor CA will result in paths from all end entities in the figure. However, the selection of CA3 as the trust anchor CA will result in certification paths only for those EEs whose certificates were issued by CA3. No certification path exists to CA1 or CA2.

Principal CA: The Principal CA may be any CA within the mesh PKI. However, the mesh PKI must have only one Principal CA, and a certification path should exist from the Principal CA to all other CAs within the mesh PKI.

Considerations: This model should be used sparingly, especially the partial mesh model, because of the complexity of determining trust anchors and building certification paths. A full mesh PKI may be useful for certification path building because paths of length one exist from all CAs to all other CAs in the mesh.

2.3.2.3. Hybrid PKI Architectures

Definition: A hybrid PKI is a PKI that uses a combination of the pure hierarchical model using subordinate CA certificates and the pure mesh model using cross-certificates.

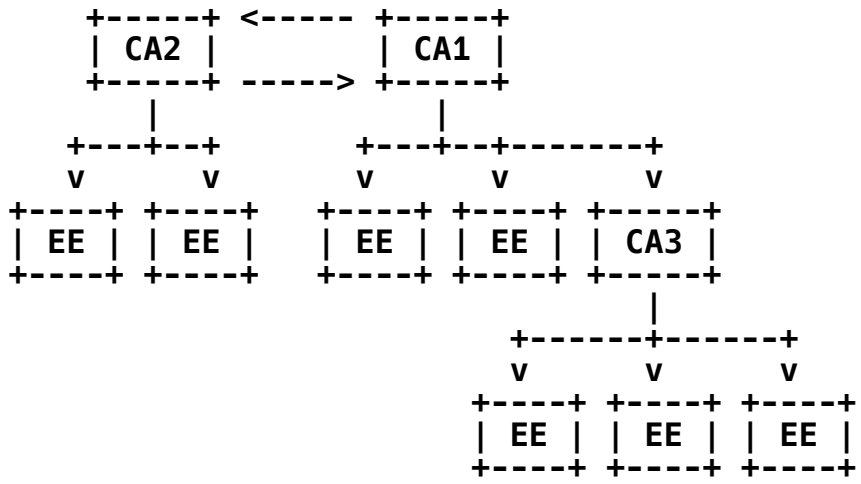

Figure 8: Hybrid PKI Architecture

Trust anchor CA: The trust anchor CA for a hybrid PKI may be any CA with self-issued certificates in the hybrid PKI. However, because of the potential complexity of a hybrid PKI, the PKI should provide guidance regarding the selection of the trust anchor to relying parties because a relying party may fail to build an appropriate certification path to a subscriber if they choose an inappropriate trust anchor.

Principal CA: The Principal CA may be any CA within the hybrid PKI and should have a self-signed certificate for cross-certification with other PKI domains. However, the hybrid PKI must have only one Principal CA and a certification path must exist from the Principal CA to every CA within the PKI.

Considerations: This model should be used sparingly because of the complexity of determining trust anchors and building certification paths. However, hybrid PKIs may occur as a result of the evolution of a PKI over time, such as CAs within an organization joining together to become a single PKI.

2.4. Relationships between PKIs and Relying Parties

Relying Parties establish trust relationships by trust anchor to a PKI. Relying Parties may use a Trust List for establishing trust relationships to one or more PKIs. A Trust List is a set of one or more trust anchors for trusting one or more PKIs.

There are two types of maintenance models of Trust List, Local Trust List Model and Trust Authority Model. The two models are described in detail in Section 4.1.

3. PKI Domain

Two or more PKIs may choose to enter into trust relationships with each other. For these relationships, each PKI retains its own set of Certificate Policy OIDs and its own Principal CA. In addition to making a business decision to consider a trust relationship, each PKI determines the level of trust of each external PKI by reviewing external PKI Certificate Policy Document(s) and any other PKI governance documentation through a process known as policy mapping. Trust relationships are technically formalized through the issuance of cross-certificates. Such a collection of two or more PKIs is known as a PKI domain.

PKI domain: A set of two or more PKIs that have chosen to enter into trust relationships with each other through the use of cross-certificates. Each PKI that has entered into the PKI domain is considered a member of that PKI domain.

NOTE: This definition specifies a PKI domain recursively in terms of its constituent domains and associated trust relationships; this is different to the definition in RFC 4949 [RFC4949] that gives PKI domain as a synonym for CA domain and defines it in terms of a CA and its subject entities.

Domain Policy Object Identifier: A domain Policy Object Identifier (OID) is a Policy OID that is shared across a PKI domain. Each CA in the PKI domain must be operated under the domain Policy OID. Each CA may also have its own Policy OID(s) in addition to the domain Policy OID. In such a case, the CA must comply with both policies. The domain Policy OID is used to identify the PKI domain.

Policy Mapping: A process by which members of a PKI domain evaluate the Certificate Policies (CPs) and other governance documentation of other potential PKI domain members to determine the level of trust that each PKI in the PKI domain places on certificates issued by each other PKI in the PKI domain.

3.1. PKI Domain Properties

- A PKI domain may operate a Bridge CA or a Unifying CA that defines members of the domain by issuing cross-certificates to those members.
- A single PKI may simultaneously belong to two or more PKI domains.
- A PKI domain may contain PKI domains within its own membership.
- Two or more PKI domains may enter into a trust relationship with each other, creating a new PKI domain. They may choose to retain the existing PKI domains in addition to the new PKI domain or collapse the existing PKI domains into the new PKI domain.
- A member of a PKI domain may choose to participate in the PKI domain but restrict or deny trust in one or more other member PKIs of that same PKI domain.

3.2. Requirements for Establishing and Participating in PKI Domains

The establishment of trust relationships has a direct impact on the trust model of relying parties. As a result, consideration must be taken in the creation and maintenance of PKI domains to prevent creating inadvertent trust relationships.

3.2.1. PKI Requirements

In order for a PKI to participate in one or more PKI domains, that PKI must have the following:

- A Certificate Policy Document documenting the requirements for operation of that PKI. The Certificate Policy Document should be in RFC 3647 [RFC3647] format.
- One or more Policy OIDs defined in the Certificate Policy Document that are also asserted in all certificates issued by that PKI.
- A defined Principal CA.

PKI domains may also impose additional technical, documentation, or policy requirements for membership in the PKI domain.

When participating in a PKI domain, the domain Policy OID(s) must be asserted at least in cross-certificates issued by a participating PKI. After the participation, the PKI can assert the domain Policy OID(s) in certificates issued by that PKI, or may map the domain

Policy OID(s) to the Policy OID(s) asserted in certificates issued by that PKI.

3.2.2. PKI Domain Documentation

PKI domains must be formally defined and documented. This documentation may vary greatly depending on the PKI domain. However, it must:

- Establish the existence of the PKI domain;
- Define the authority for maintaining the PKI domain;

  Examples of PKI domain Authorities are (1) Representatives from two PKIs that agree to form a simple PKI domain, (2) A single entity that may or may not be related to any of the PKIs in the PKI domain, (3) A governance board made up of representatives from each PKI domain member.

- Define how the PKI domain is governed;
- Define the purpose and community of interest of the PKI domain; and

  Examples of PKI domain intents are (1) allow relying parties of one PKI to trust certificates issued by another PKI, (2) allow PKIs that support similar subscriber communities of interest to interact with each other, and (3) allow relying parties to trust certificates issued by a number of PKIs that all meet a set of requirements.

- Unless the PKI domain has a predetermined membership, describe the requirements and methods for joining the PKI domain, such as FPKIMETHOD [FPKIMETHOD].

Examples of governance documents that PKI domains may choose to use are:

- Statement of intent between two or more parties;
- Memorandum of Agreement between two or more parties;
- Certificate Policy Document for the PKI domain;
- Charter for the PKI domain; or
- Methodology for PKI domain membership.

3.2.3. PKI Domain Membership Notification

A cross-certificate from the Principal CA of one PKI to the Principal CA of another PKI indicates a mapping between one or more policies of the first PKI and one or more policies of the second PKI. When a relying party is determining if a certificate can be validated, it builds a certification path from the certificate being presented to a trust anchor. To prevent creating inadvertent trust relationships across PKI domains when a single PKI is a member of two or more disparate PKI domains, each PKI domain must be cognizant of what PKI domains in which its member PKIs participate. Figure 9 illustrates this concept.

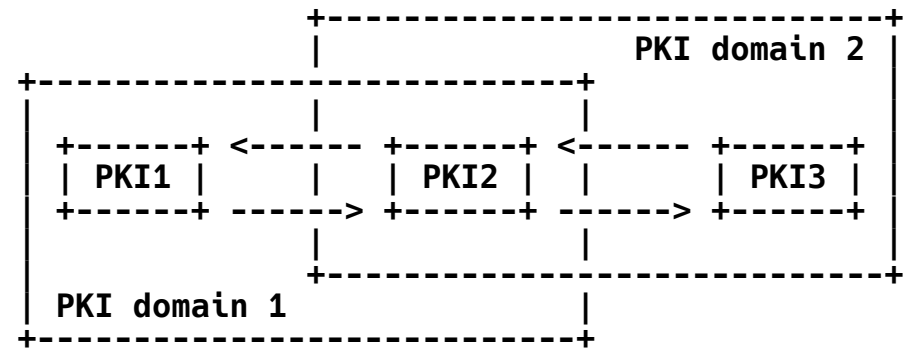

Figure 9: Participation in Multiple PKI Domains

As shown in Figure 9, PKI2 is a member of both PKI domain 1 and PKI domain 2. Since a certification path exists from PKI1 to PKI2, and from PKI2 to PKI3, a certification path also exists from PKI1 to PKI3. However, PKI1 does not share domain membership with PKI3, so the certification path validation from PKI1 to PKI3 with a validation policy for PKI domain 1 must not succeed. To ensure correct certification path validation and policy mapping, the cross-certificates issued by both PKI1 and PKI3 to PKI2 must contain constraints such as policy mapping or name constraints disallowing the validation of certification paths outside their respective domains.

To fully prevent inadvertent trust, any PKI that is a member of one or more PKI domains must inform all those PKI domains of its membership in all other PKI domains. In addition, that PKI must inform all those PKI domains of which it is a member, any time its membership status changes with regards to any other PKI domain. If a PKI domain is informed of the change in status of one of its member PKIs with regards to other PKI domains, that PKI domain must review the constraints in any cross-certificate issued to that PKI. If the change in membership would result in a change to the allowed or

disallowed certification paths, the PKI domain must ensure that all such cross-certificates are revoked and re-issued with correct constraints.

3.2.4. Considerations for PKIs and PKI Domains with Multiple Policies

In some cases, a single PKI may issue certificates at more than one assurance level. If so, the Certificate Policy Document must define separate Policy OIDs for each assurance level, and must define the differences between certificates of different assurance levels.

A PKI domain may also support more than one assurance level. If so, the PKI domain must also define separate Policy OIDs for each assurance level, and must define the differences in requirements for each level.

When PKIs and PKI domains choose to establish trust relationships, these trust relationships may exist for only one defined assurance level, may have a one-to-one relationship between PKI assurance levels and PKI domain assurance levels, or may have many-to-one or one-to-many relationships between assurance levels. These relationships must be defined in cross-certificates issued between PKIs in the PKI domain.

3.3. PKI Domain Models

Two or more PKI domains may choose to enter into trust relationships with each other. In that case, they may form a larger PKI domain by establishing a new Unifying or Bridge CA or by issuing cross-certificates between their Principal CAs.

3.3.1. Unifying Trust Point (Unifying Domain) Model

In the Unifying Trust Point Model, a PKI domain is created by establishing a joint, superior CA that issues unilateral cross-certificates to each PKI domain, as shown in Figure 10. Such a joint, superior CA is defined as a Unifying CA, and the Principal CAs in each PKI domain have the hierarchical CA relationship with that Unifying CA. In this model, any relying party from any of the PKI domains must specify the Unifying CA as its trust anchor CA in order to validate a subscriber in the other PKI domains. If the relying party does not desire to validate subscribers in other PKI domains, the relying party may continue to use the Principal CA from the old PKI domain as its trust anchor CA.

This model may be used for merging multiple PKI domains into a single PKI domain with less change to existing PKI domains, or may be used to combine multiple PKI domains into one PKI domain for relying

parties. The unilateral cross-certificate issued by the Unifying CA to the Principal CAs in each PKI domain may include any policy mapping.

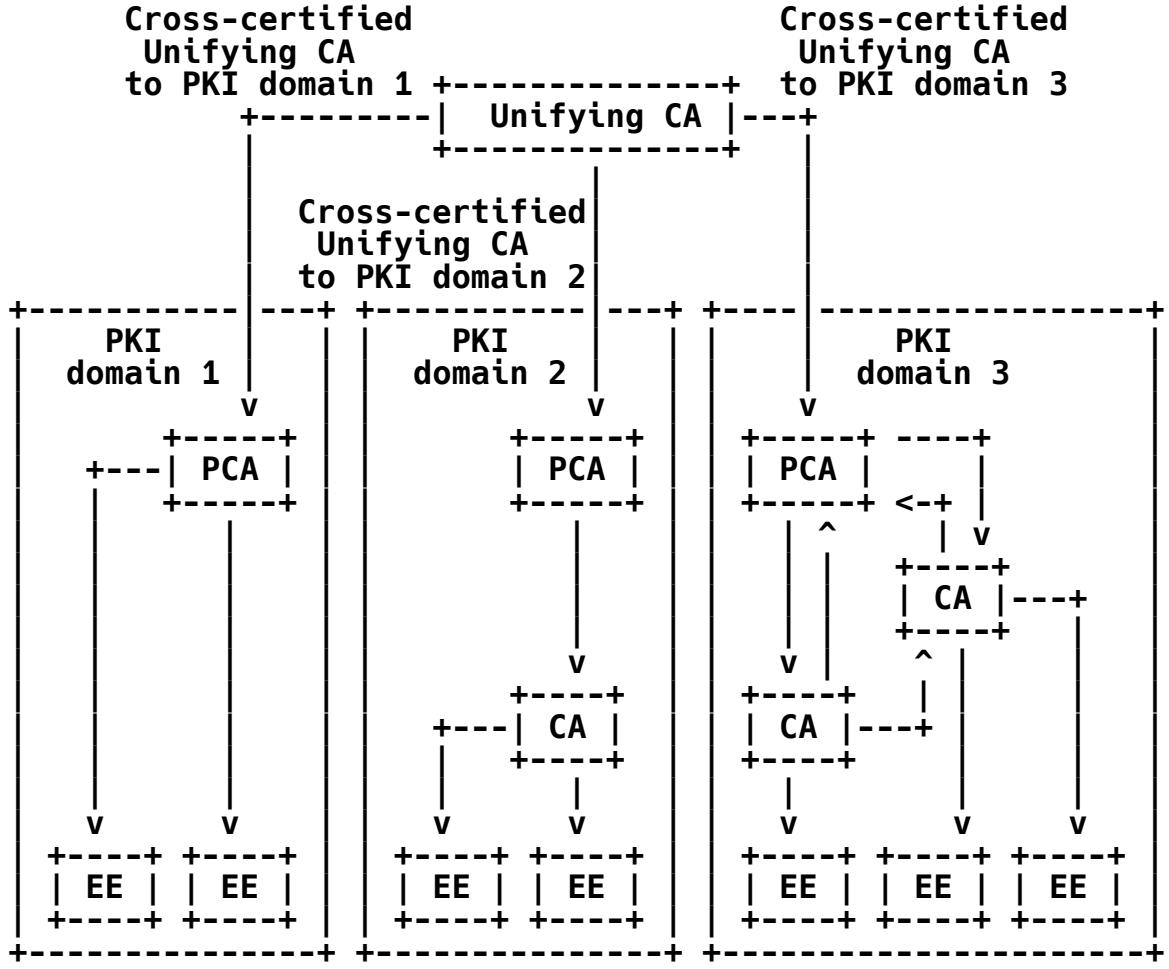

Figure 10: Unifying Trust Point (Unifying Domain) Model

3.3.2. Independent Trust Point Models

In Independent Trust Point Models, relying parties continue to use only the trust anchor of their PKI domain. A relying party in the individual trust point model can continue to use the trust anchor of its PKI domain.

3.3.2.1. Direct Cross-Certification Model

In this model, each PKI domain trusts each other by issuing a cross-certificate directly between each Principal CA, as shown in

Figure 11. This model may be used for shortening a certification path or establishing a trust relationship expeditiously.

Considerations: A PKI domain in this model needs to take into account that the other PKI domain may cross-certify with any other PKI domains. If a PKI domain wants to restrict a certification path, the PKI domain should not rely on the validation policy of the relying party, but should include the constraints in the cross-certificate explicitly. A PKI domain that relies on the validation policy of the relying party about such constraints cannot guarantee that the constraints will be recognized and followed.

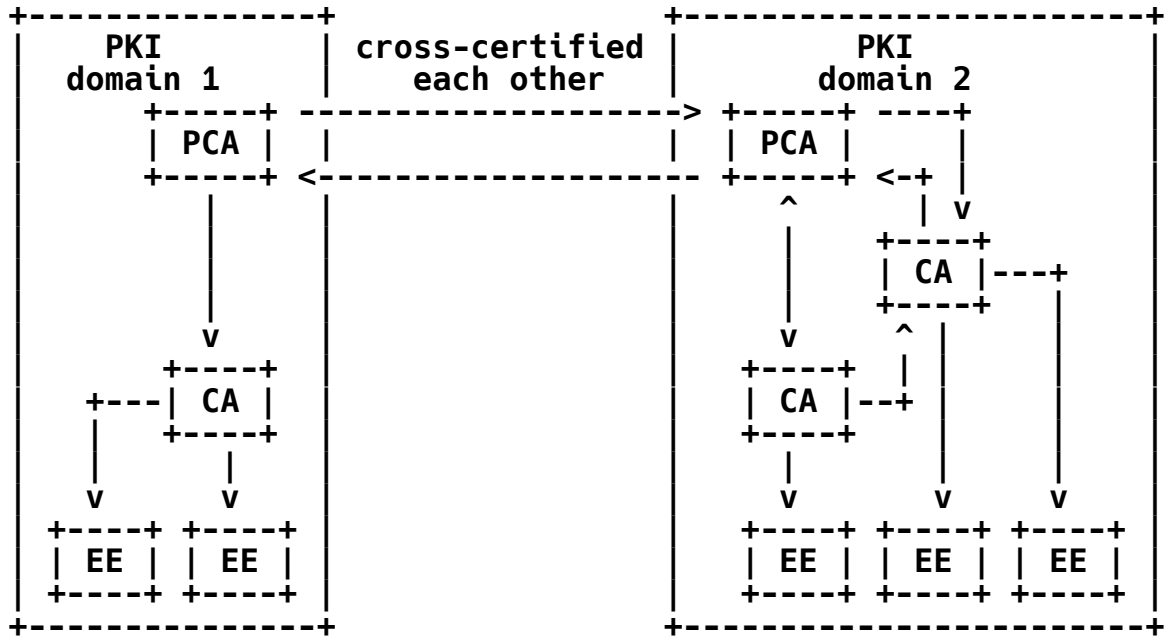

Figure 11: Direct Cross-Certification Model

3.3.2.2. Bridge Model

In this model, every PKI domain trusts each other through a Bridge CA by cross-certification, as shown in Figure 12. The trust relationship is not established between a subscriber domain and a relying party domain directly, but established from the Principal CA of the relying party's PKI domain via a Bridge CA. This model is useful in reducing the number of cross-certifications required for a PKI domain to interoperate with other PKI domains.

Requirements for Bridge model:

- The Bridge CA must not be used as the trust anchor CA in any PKI domain.
- The Bridge CA should issue cross-certificates with other PKI domains mutually or may issue cross-certificates unilaterally.
- The Bridge CA must not issue End Entity (EE) certificates except when it is necessary for the CA's operation.
- The Bridge CA must use its own domain Policy OID, not other PKI domain Policy OID(s), for the policy mapping.
- The Bridge CA should be a neutral position to all PKI domains, which trust through the Bridge CA. For example, in Figure 12, in the case that a relying party who trusts the PCA of PKI domain 1 as its trust anchor CA builds the certification path to a subscriber in PKI domain 3:

  Cross-Certificate from PKI domain 1 to the Bridge CA:

  issuerDomainPolicy ::= domain Policy OID of PKI domain 1

  subjectDomainPolicy := domain Policy OID of the Bridge CA

  Cross-Certificate from the Bridge CA to PKI domain 3:

  issuerDomainPolicy ::= domain Policy OID of the Bridge CA

  subjectDomainPolicy ::= domain Policy OID of PKI domain 3

- Cross-certificates issued by the Bridge CA and cross-certificate issued to the Bridge CA should include the requireExplicitPolicy with a value that is greater than zero in the policyConstraints extension because a relying party may not set the initial-explicit-policy to TRUE.
- PKI domains cross-certified with the Bridge CA should not cross-certify directly to other PKI domains cross-certified with the same Bridge CA.
- The Bridge CA should clarify the method for the policy mapping of cross-certification to keep its transparency.

Considerations: The Bridge CA should be operated by an independent third party agreed upon by the PKI domains or a consortium consisting of representatives from the PKI domain members. The Bridge CA should do policy mapping in a well-documented and agreed-upon manner with all PKI domains. When applying the name constraints, the Bridge CA needs to avoid creating conflicts between the name spaces of the cross-certified PKI domains. The PKI domains that perform cross-certification with the Bridge CA should confirm the following:

* Does the Bridge CA perform the policy mapping via its own domain Policy OID?

* Does the Bridge CA clarify the method of policy mapping in the cross-certification?

* Is the Bridge CA able to accept the domain policy that the PKI domain desires?

  + If the domain policy is mapped to one with a lower security level, the PKI domain should not accept it. Otherwise, the PKI domain must carefully consider the risks involved with accepting certificates with a lower security level.

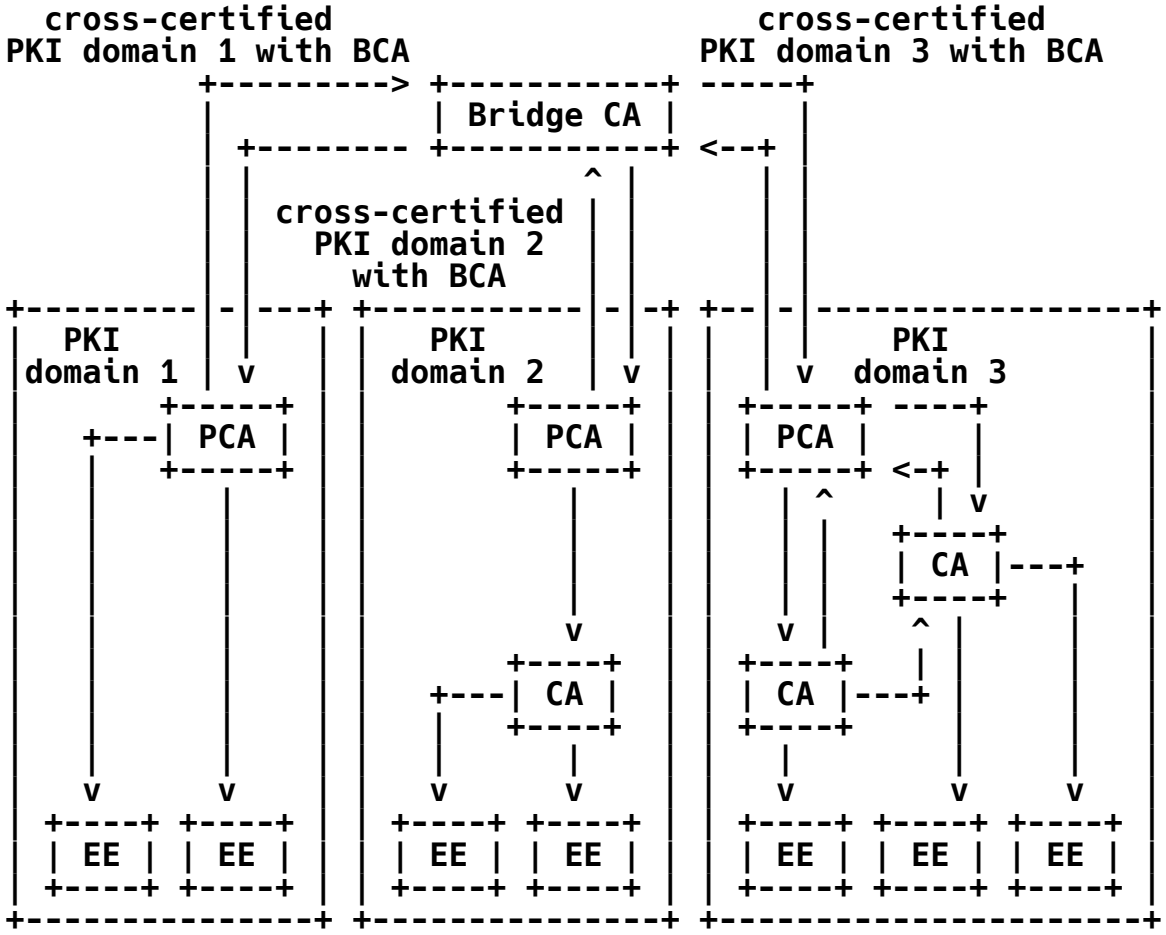

Figure 12: Bridge Model

3.4. Operational Considerations

Each PKI domain may use policy mapping for crossing different PKI domains. If a PKI domain wants to restrict a certification path, the PKI domain should not rely on the validation policy of the relying party, but should include the constraints in the cross-certificate explicitly.

For example, when each PKI domain wants to affect the constraints to a certification path, it should set the requireExplicitPolicy to zero in the policyConstraints extension of any cross-certificates. A PKI domain that relies on the validation policy of the relying party about such constraints cannot guarantee the constraints will be recognized and followed.

4. Trust Models External to PKI Relationships

As opposed to PKI domain trust relationships entered into by PKIs themselves, trust across multiple PKIs can be created by entities external to the PKIs through locally configured lists of trust anchors.

Trust List: A set of one or more trust anchors used by a relying party to explicitly trust one or more PKIs.

Note that Trust Lists are often created without the knowledge of the PKIs that are included in the list.

4.1. Trust List Models

4.1.1. Local Trust List Model

A Trust List can be created and maintained by a single relying party for its own use.

Local Trust List: A Trust List installed and maintained by a single relying party for its own use. NOTE: This definition is similar to "trust-file PKI" defined in RFC 4949 [RFC4949]. However, this document prefers the term "Local Trust List" contrasting with "Trust Authority" defined below.

Figure 13 illustrates a Local Trust List.

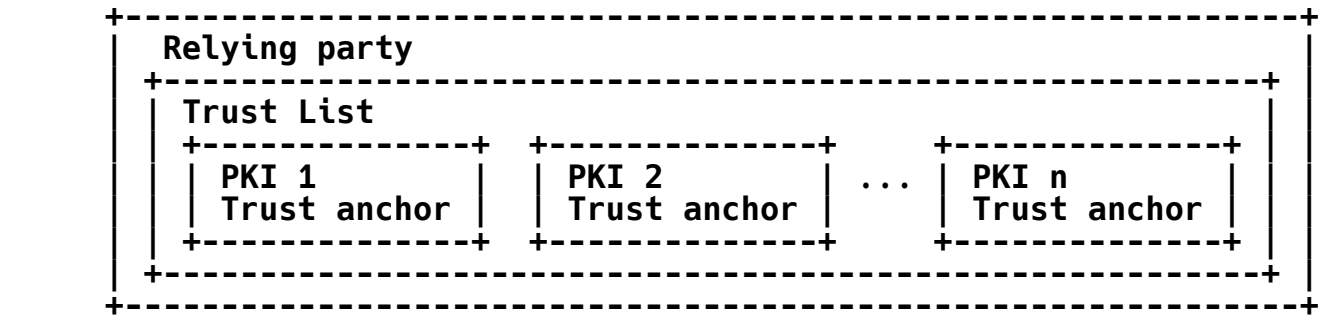

Figure 13: Relying Party Local Trust List Model

Creating a Local Trust List is the simplest method for relying parties to trust EE certificates. Using Local Trust Lists does not require cross-certification between the PKI that issued the relying party's own certificate and the PKI that issued the EE's certificate,nor does it require implementing mechanisms for processing complex certification paths, as all CAs in a path can be included in the Local Trust List. As a result, Local Trust Lists are

the most common model in use today. However, because Local Trust Lists are created and managed independently by each relying party, the use of Local Trust Lists can be difficult for an enterprise to manage.

4.1.2. Trust Authority Model

Alternatively, a Trust List can be created and maintained for using by multiple relying parties. In this case, the entity responsible for the Trust List is known as a Trust Authority.

Trust Authority: An entity that manages a Trust List for use by one or more relying parties.

Figure 14 illustrates a Trust Authority and how it is used by Relying Parties. Note that the Trust Authority replaces the PKI trust anchor(s) in the Local Trust List for each participating relying party.

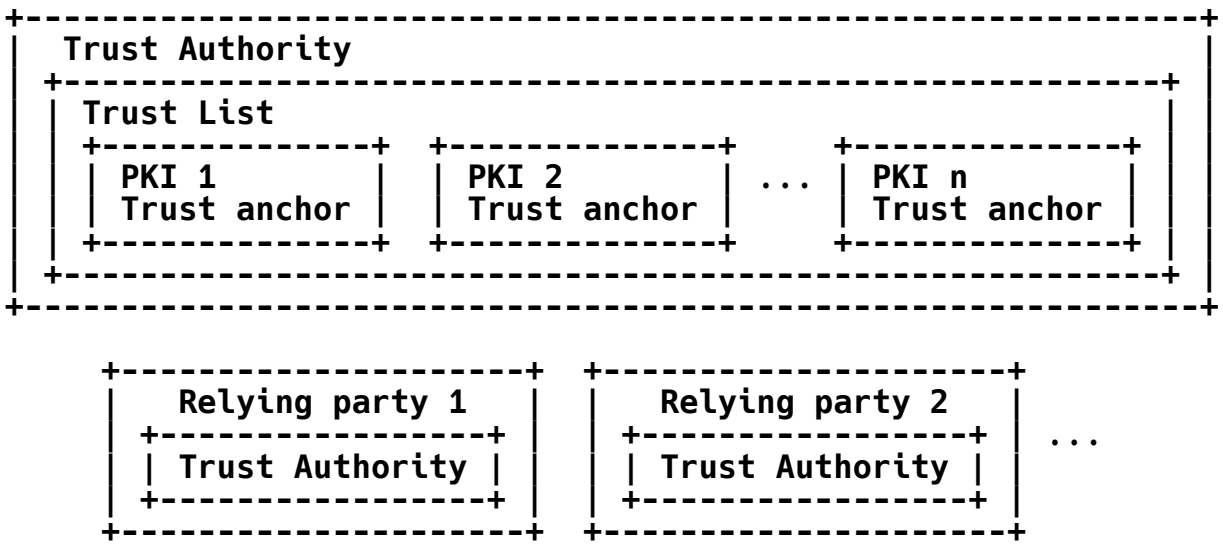

Figure 14: Trust Authority Model

A Trust Authority may be operated by a PKI, a collection of relying parties that share a common set of users, an enterprise on behalf of all of its relying parties, or an independent entity. Although PKIs generally establish trust relationships through cross-certificates, a PKI may choose to provide a Trust Authority to support relying parties that do not support processing of certification paths. A collection of relying parties that share a common set of users may choose to maintain a single Trust Authority to simplify the management of Trust Lists. An enterprise may choose to provide a

Trust Authority to implement enterprise policies and direct all Relying Parties within the enterprise to use its Trust Authority. Finally, an independent entity may choose to operate a Trust Authority as a managed service.

4.2. Trust List Considerations

4.2.1. Considerations for a PKI

A PKI should publish its Certificate Policy Document so that Relying Parties and Trust Authorities can determine what, if any, warranties are provided by the PKI regarding reliance on EE certificates.

A PKI should broadly publicize information regarding revocation or compromise of a trust anchor CA or Principal CA certificate through notice on a web page, press release, and/or other appropriate mechanisms so that Relying Parties and Trust Authorities can determine if a trust anchor CA or Principal CA certificate installed in a Trust List should be removed.

A PKI should publish Certificate Revocation Lists (CRLs) or other information regarding the revocation status of EE certificates to a repository that can be accessed by any party that desires to rely on the EE certificates.

4.2.2. Considerations for Relying Parties and Trust Authorities

Relying Parties and Trust Authorities are responsible for the following prior to including a PKI in the Trust List:

- Reviewing the Certificate Policy Document of each PKI to determine that the PKI is operated to an acceptable level of assurance;
- Reviewing the Certificate Policy Document of each PKI to ensure any requirements imposed on Relying Parties are met;
- Determining if the PKI provides any warranties regarding reliance on EE certificates, and if these warranties are acceptable for the intended reliance on the EE certificates. Reliance may be at the relying party's own risk; and
- Periodically reviewing information published by the PKI to its repository, including Certificate Policy Document updates or notice of CA revocation or compromise.

A PKI can choose to join or leave PKI domains in accordance with its Certificate Policy Document. If the relying party or Trust Authority does not wish to inherit trust in other members of these PKI domains,

it is the responsibility of the relying party or Trust Authority to inhibit policy mapping.

4.2.3. Additional Considerations for Trust Authorities

A Trust Authority should establish a Trust Authority Policy that identifies the following:

- The intended community of Relying Parties that will use the Trust Authority;
- The process by which trust anchors are added or removed from the Trust List;
- Any warranties provided by the Trust Authority for reliance on EE certificates. These warranties may be those provided by the PKIs themselves or may be additional warranties provided by the Trust Authority;
- Information regarding how the Trust Authority protects the integrity of its Trust List; and
- Information regarding how Relying Parties interact with the Trust Authority to obtain information as to whether an EE certificate is trusted.

5. Abbreviations

CA: Certification Authority

EE: End Entity

OID: Object Identifier

PCA: Principal Certification Authority

PKI: Public Key Infrastructure

6. Security Considerations

This section highlights security considerations related to establishing PKI domains.

6.1. PKI Domain Models

For all PKI domain models described in Section 3.3 created through the issuance of cross-certificates, standard threats including message insertion, modification, and man-in-the-middle are not

applicable because all information created by CAs, including policy mapping and constraints, is digitally signed by the CA generating the cross-certificate.

Verifying that a given certificate was issued by a member of a PKI domain may be a time-critical determination. If cross-certificates and revocation status information cannot be obtained in a timely manner, a denial of service may be experienced by the end entity. In situations where such verification is critical, caching of cross-certificates and revocation status information may be warranted.

An additional security consideration for PKI domains is creating inadvertent trust relationships, which can occur if a single PKI is a member of multiple PKI domains. See Section 3.2.3 for a discussion of creating inadvertent trust relationships and mechanisms to prevent it.

Finally, members of PKI domains must participate in domain governance, or at a minimum, be informed anytime a PKI joins or leaves the domain, so that domain members can make appropriate decisions for maintaining their own membership in the domain or choosing to restrict or deny trust in the new member PKI.

6.2. Trust List Models

In these models, many standard attacks are not applicable since certificates are digitally signed. Additional security considerations apply when trust is created through a Trust List.

A variation of the modification attack is possible in Trust List Models. If an attacker is able to add or remove CAs from the relying party or Trust Authority Trust List, the attacker can affect which certificates will or will not be accepted. To prevent this attack, access to Trust Lists must be adequately protected against unauthorized modification. This protection is especially important for trust anchors that are used by multiple applications, as it is a key vulnerability of this model. This attack may result in unauthorized usage if a CA is added to a Trust List, or denial of service if a CA is removed from a Trust List.

For Trust Authority models, a denial-of-service attack is also possible if the application cannot obtain timely information from the trust anchor. Applications should specify service-level agreements with Trust Authority. In addition, applications may choose to locally cache the list of CAs maintained by the Trust Authority as a backup in the event that the trust anchor's repository (e.g., Lightweight Directory Access Protocol (LDAP) directory) is not available.

7. References

7.1. Normative References

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

7.2. Informative References

[CCITT.X509.2000] International Telephone and Telegraph Consultative Committee, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", CCITT Recommendation X.509, March 2000.

[FPKIMETHOD] "US Government PKI Cross-Certification Criteria and Methodology", January 2006, <http://www.cio.gov/fpkia/documents/crosscert_method_criteria.pdf>.

[RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, November 2003.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

Authors' Addresses

Masaki Shimaoka (editor)
SECOM Co., Ltd. Intelligent System Laboratory
SECOM SC Center, 8-10-16 Shimorenjaku
Mitaka, Tokyo 181-8528
JP

EMail: m-shimaoka@secom.co.jp

Nelson Hastings
National Institute of Standard and Technology
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20899-8930
US

EMail: nelson.hastings@nist.gov

Rebecca Nielsen
Booz Allen Hamilton
8283 Greensboro Drive
McLean, VA 22102
US

EMail: nielsen_rebecca@bah.com

Full Copyright Statement

Intellectual Property