Entrust®, Inc.

Cross-Certification and PKI Policy Networking

Author: Jim Turnbull

Date: August 2000
Version: 1.0

Entrust®
Securing the Internet

© Entrust, Inc., 2000

Entrust is a registered trademark of Entrust Technologies Inc. in the United States and certain other countries. In Canada, Entrust is a registered trademark of Entrust Technologies Limited. All other Entrust Technologies product names and service names are trademarks of Entrust Technologies. All other company and product names are trademarks or registered trademarks of their respective owners.

The material provided in this document is for information purposes only. It is not intended to be advice. You should not act or abstain from acting based upon such information without first consulting a professional. ENTRUST TECHNOLOGIES DOES NOT WARRANT THE QUALITY, ACCURACY OR COMPLETENESS OF THE INFORMATION CONTAINED IN THIS ARTICLE. SUCH INFORMATION IS PROVIDED "AS IS" WITHOUT REPRESENTATION, WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, BY USAGE OF TRADE, OR OTHERWISE, AND ENTRUST TECHNOLOGIES SPECIFICALLY DISCLAIMS ANY AND ALL REPRESENTATIONS, AND WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, OR FITNESS FOR A SPECIFIC PURPOSE.

1. Introduction

This paper will explain the two methods used by Entrust/PKI to extend trust between Certification Authorities (CAs): peer-to-peer cross-certification and hierarchical cross-certification. The benefits of each method will be described and example architectures of each method will be provided. The paper will also explain *PKI policy networking*, which provides a way to limit the trust relationship between CAs to best suit an organization's needs.

2. Cross-certification

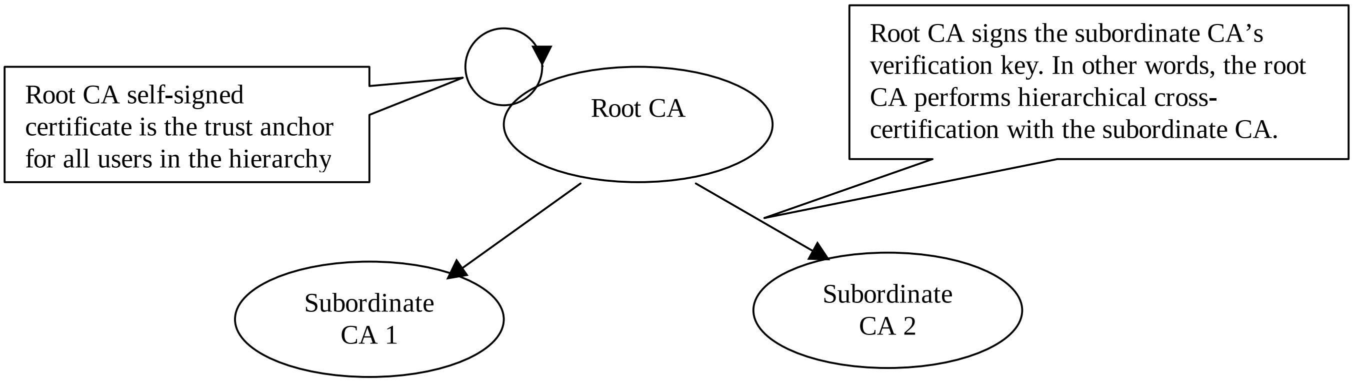
The term cross-certification refers to two operations:

- The first operation, which is generally executed infrequently, is the establishment of a trust relationship between two CAs through the signing of another CA's public key in a certificate referred to as a "cross-certificate".

- The second operation, executed frequently by the client application, involves verifying the trustworthiness of a user's certificate signed by a CA within your PKI network. The operation is often referred to as "walking a chain of trust". The "chain" refers to a list of cross-certificate validations that are "walked" (or traced) from the root CA key or "trust anchor" of the verifying user to the CA key required to validate the other user's certificate.

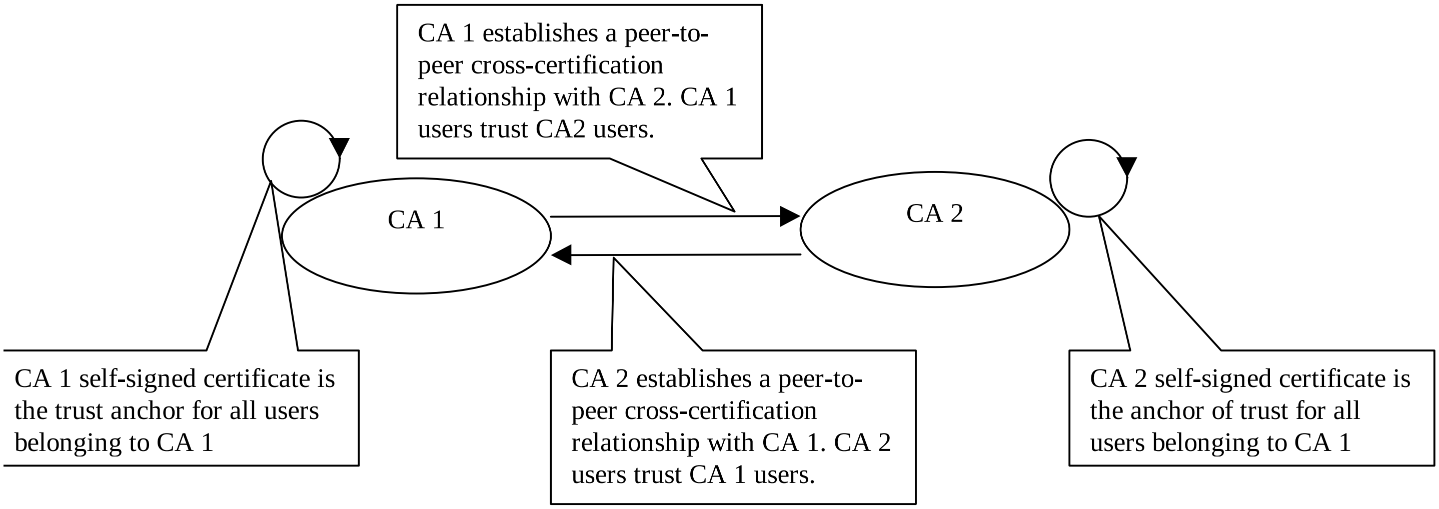
A "trust anchor" is the CA verification key used by the client application as the starting point for all certificate validation. Hierarchical cross-certification is distinguished from peer-to-peer cross-certification by the location of the user's trust anchor vis-à-vis the user.

- If the user's trust anchor is *not* the user's local CA, then the user's local CA is a subordinate CA in a hierarchy of CAs. The user's trust anchor is the public key of the root CA of the hierarchy. Subordinate CAs cannot perform peer-to-peer cross-certification with other CAs but may, if permitted by policy, add subordinate CAs to the hierarchy below itself. All certificate validation by clients within a hierarchy starts with the root CA's public key. The following is a basic hierarchical cross-certification architecture.


Root CA self-signed certificate is the trust anchor for all users in the hierarchy
Root CA
Root CA signs the subordinate CA's verification key. In other words, the root CA performs hierarchical cross-certification with the subordinate CA.
Subordinate CA 1
Subordinate CA 2

Hierarchical cross-certification between a root (autonomous) CA an subordinate (non-autonomous) CAs.

- If the user’s trust anchor is the user’s local CA, then the user’s local CA is an *autonomous* CA. Autonomy refers to the fact that the CA doesn’t rely on a superior CA in a hierarchy. An autonomous CA can perform peer-to peer cross-certification with other autonomous CAs, and can act as the root CA in a hierarchy of CAs. All certificate validation for clients within an autonomous CA starts with the local CA’s self-signed certificate.


CA 1 establishes a peer-to-peer cross-certification relationship with CA 2. CA 1 users trust CA2 users.
CA 1
CA 2
CA 1 self-signed certificate is the trust anchor for all users belonging to CA 1
CA 2 establishes a peer-to-peer cross-certification relationship with CA 1. CA 2 users trust CA 1 users.
CA 2 self-signed certificate is the anchor of trust for all users belonging to CA 1

2.1. Benefits of Hierarchical Cross-certification

Hierarchical cross-certification is ideal *within* organizations where multiple CAs are needed and where the organization requires maximum control over all CAs in the hierarchy.

Entrust/PKI 5.0 satisfies the requirement for hierarchical cross-certification with the following critical features and benefits:

- The root CA can control the policy of subordinate CAs including whether additional CAs can be added to the hierarchy by subordinate CAs.

- The root CA can revoke subordinate CAs if required.

- The root CA controls peer-to-peer cross-certification relationships with other autonomous CAs.

- Since the root CA is the anchor of trust for all users and CAs within the hierarchy, maximum physical security policies and practices are only required for the root CA, rather than for all CAs within the hierarchy.

- Only using the root CA to certify and issue policy to subordinate CAs can enhance the security of the root CA. By not using the root CA to support users within the root CA domain, the CA will be less exposed to operators and can be physically secured more tightly than otherwise.

2.2. Benefits of Peer-to-Peer Cross-certification

Peer-to-peer cross-certification is ideal *between* organizations where each organization wants maximum control over it’s own organization. Peer-to-peer cross-certification must occur between autonomous CAs, where an autonomous CA can be either the root CA in a hierarchy of CAs, or else a stand-alone CA.

Entrust/PKI 5.0 satisfies the requirement for peer to peer cross-certification with the following critical features and benefits:

- Autonomous CAs can establish or revoke peer-to-peer cross-certification relationships with other *existing* autonomous CAs as business needs dictate. This provides greater flexibility over hierarchical cross-certification since a hierarchy of CAs must be created by first creating the root CA, then creating subordinate CAs, and then creating subordinate CAs below the subordinate CAs.

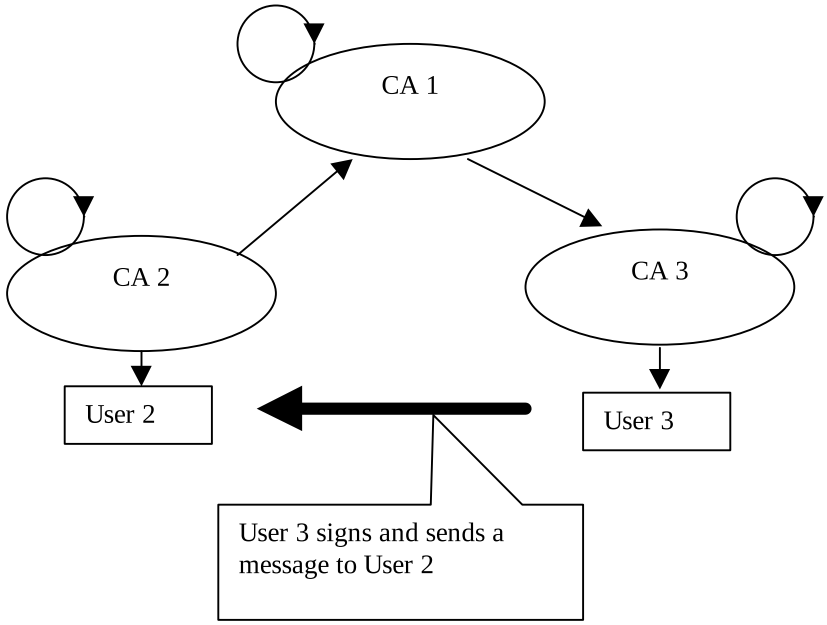
- An autonomous CA does not rely on another CA for its anchor of. This is more appropriate than a hierarchy for business relationships between distinct and separate organizations.

2.3. Cross-Certification Examples

Suppose peer-to-peer cross-certification is in place where CA 2 has unilaterally cross-certified with CA 1 and CA 1 has unilaterally cross-certified with CA 3 (see diagram below). CA 2’s self-signed certificate is the trust anchor for User 2 and CA 3’s self-signed certificate is the trust anchor for User 3. The trust anchors are depicted as circles with an arrow This is meant to illustrate that the CA’s verification public key is signed by the corresponding signing private key. In other words, the CA verification certificate is a self-signed CA certificate.

Suppose User 2 receives a signed message from User 3 and User 2 attempts to verify the signature. Assuming all certificates are valid, the signature will verify successfully because User 2’s CA trust anchor, namely CA 2, signed CA 1’s verification public key, creating a cross-certificate; CA 1 signed CA 3’s verification public key, creating a cross-certificate; and CA 3 signed User 3’s verification public key, creating User 3’s verification certificate.

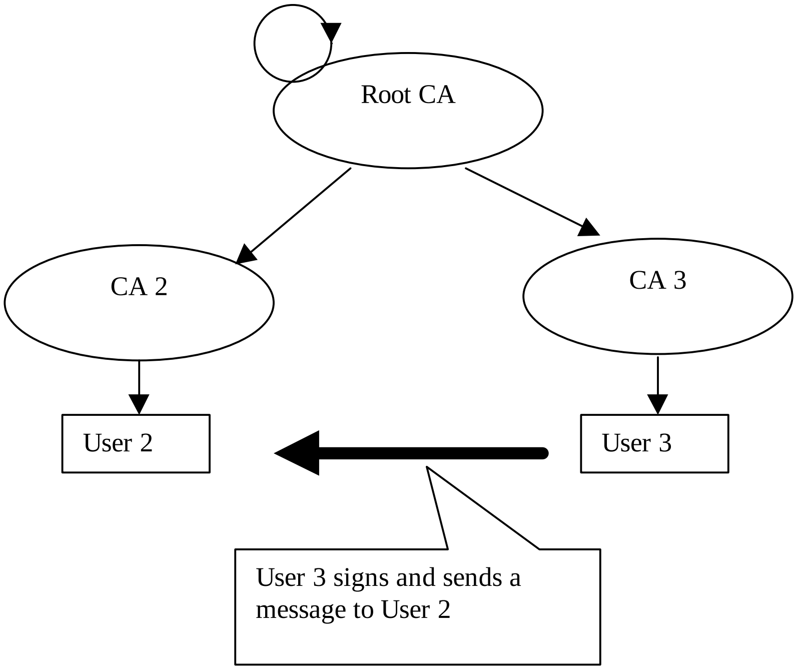
The important point is that for User 2 to trust User 3, a chain of trust must exist from the CA trust anchor, namely CA2, to User 3's verification certificate. This chain of trust is formed with CA 2's trust anchor, two cross-certificates and User 2's verification certificate.



A hierarchical cross-certification structure consists of a root CA and a hierarchy of CAs branching out below the root as shown in the diagram below. This hierarchy can be as arbitrarily broad and deep. Only CAs with a self-signed CA verification public key can act as the root CA in hierarchical cross-certification. Arrows represent trust relationships where the root CA signs the CA verification public keys of all CAs immediately below the root. These CAs in turn can sign the CA verification public keys of all CAs immediately below each of them.

The main point that distinguishes hierarchical cross-certification from peer-to-peer cross-certification is the location of the CA trust anchor. Notice in the diagram below that all arrows point from the root CA to all subordinate CAs, namely CA 1 and CA 2.

The reason for this is that the CA trust anchor for all subordinate CAs and users is the root CA verification public key. This is the key trait that describes hierarchical cross-certification from peer-to-peer cross-certification.



In hierarchical cross-certification, when registering with the PKI, the user receives the root CA verification public key as his CA trust anchor, which is securely stored in the user's profile. For example, when User 2 registers with CA 2, he will securely download the root CA verification public key and the CA 2 verification certificate signed by the root CA.

For certificate validation, using the same hierarchy example above, suppose User 2 receives a signed message from User 3 and User 2 attempts to verify the signature. Assuming all certificates are valid, the signature will verify successfully because User 2's CA trust anchor, namely the root CA verification public key, signed CA 3's verification public key, creating CA 3's subordinate CA certificate; and CA 3 signed User 3's verification public key, creating User 3's verification certificate. Notice that even when verifying certificates of users from your own local CA, certificate validation still starts from the root CA trust anchor.

Peer-to-peer cross-certification can better match the business relationships between organizations than hierarchical cross-certification. For example, peer-to-peer cross-certification allows a local CA to establish a cross-certification relationship with a CA which just applies to the local CA. In hierarchical cross-certification, the root CA can only extend trust to an external CA for the entire hierarchy.

3. PKI Policy Networking

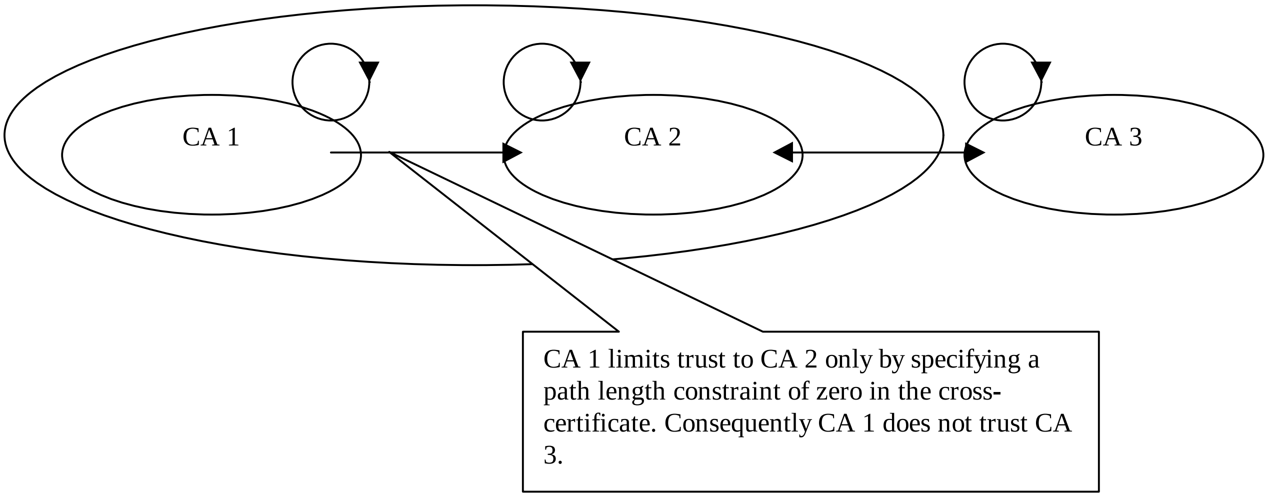
Cross-certification is used to extend trust to another CA. By extending trust, users in one CA will trust the user certificates belonging to the cross-certified CA. For large organizations that have implemented multiple CAs for scalability reasons, full trust between cross-certified CAs is often appropriate. However, for disparate organizations that have a specific and limited business relationship, establishing full trust between CAs is often not appropriate.

Entrust/PKI provides fine-grained control to limit and control the level of trust between CAs. This control can be implemented within a hierarchical or peer-to-peer PKI network. With this level of control, organizations will be able to design trust relationships between their CAs that closely match their business relationships.

There are three basic ways of constraining the trust between CAs: path length, name and policy. The cross-certificate between two CAs (peer-to-peer cross-certification) or the subordinate CA certificate (hierarchical cross-certification) is used to convey the constraint, and the client application automatically enforces the specified constraint when validating certificates.

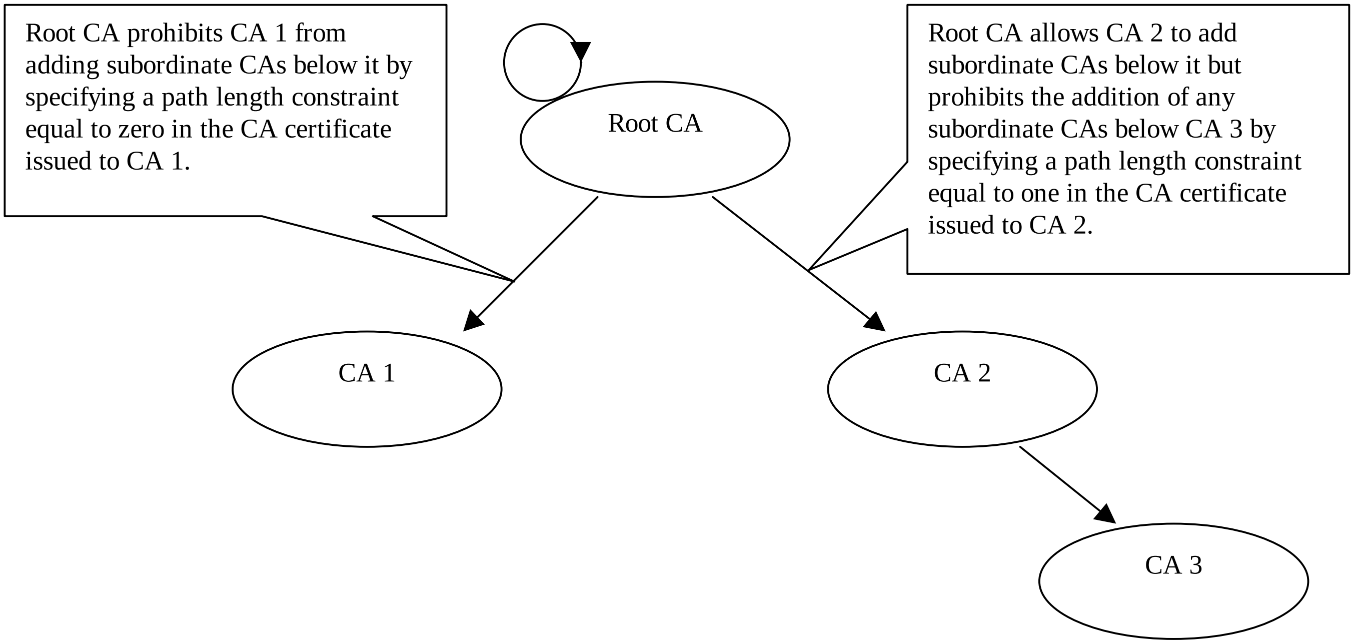
3.1. Path Length Constraints

In conjunction with peer-to-peer cross-certification, path length constraints can be used to control transitive trust. That is, you can control whether your CA should trust any cross-certification relationships that have been established by CAs with whom you have cross-certified. For example, in the diagram below, CA 1 has unilaterally cross-certified with CA 2 and CA 2 has cross-certified with CA 3.



CA 1 limits trust to CA 2 only by specifying a path length constraint of zero in the cross-certificate. Consequently CA 1 does not trust CA 3.

In hierarchical cross-certification, path length constraints can be used to control the addition of subordinate CAs. This control is important in hierarchical cross-certification since all members of the hierarchy trust each other.

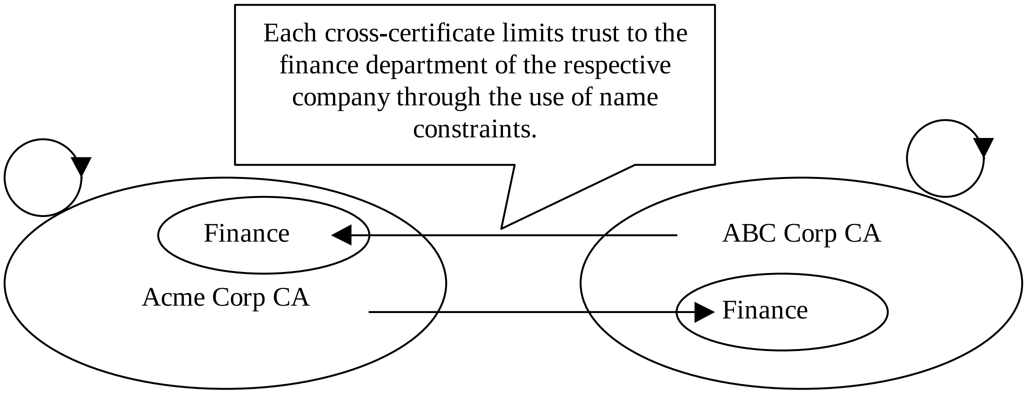


Root CA prohibits CA 1 from adding subordinate CAs below it by specifying a path length constraint equal to zero in the CA certificate issued to CA 1.

Root CA allows CA 2 to add subordinate CAs below it but prohibits the addition of any subordinate CAs below CA 3 by specifying a path length constraint equal to one in the CA certificate issued to CA 2.

3.2. Name Constraints

In peer-to-peer cross-certification, name constraints can be used to limit trust to a sub-group of cross-certified CAs based on their distinguished name (DN). For example, suppose all employees in Acme Corp. are organized within organizational units such that each user's DN includes their organizational unit. Users in the finance department have DNs like 'cn=John Smith, *ou=Finance*, o=ABC, c=US' while users in sales have DNs like 'cn=Alice Jones, *ou=Sales*, o=ABC, c=US'. So if ABC Corp establishes a cross-certified relationship with Acme Corp and vice-versa, it can be limited such that only the finance groups within each CA trust each other (see diagram below).


Each cross-certificate limits trust to the finance department of the respective company through the use of name constraints.
Finance
Acme Corp CA
ABC Corp CA
Finance

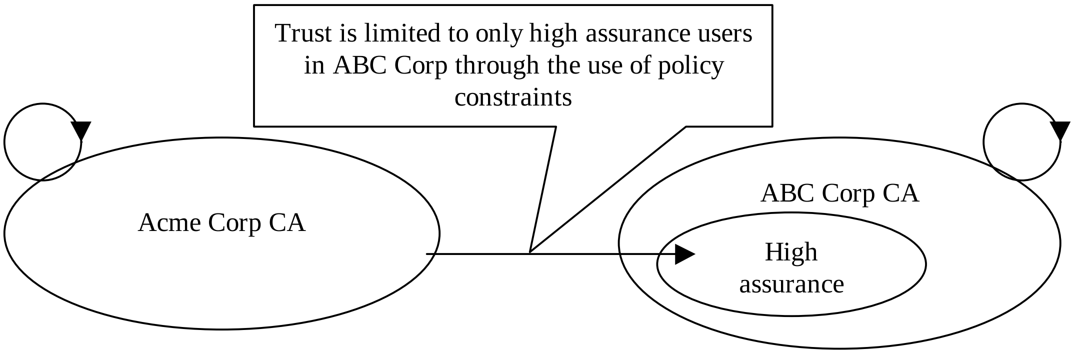
In hierarchical cross-certification, name constraints can be used to restrict the addition of subordinate CAs and their users based on restricted DNs. For example, the root CA of Acme Corp. can restrict subordinate CAs to DNs within the Acme corporation by mandating that all subordinate CAs and users with the subordinate CAs must have DNs within the name space 'o=Acme, c=US'.

3.3. Policy Constraints

Policy Constraints can be used to limit trust to only those users in another CA who have certain policy values within their certificates.

An example of its use is assurance levels. Assurance refers to the degree to which you are sure that a user really is who his certificates say he is. An organization may have different levels of assurance depending on the way in which the user is authenticated before issuing the user his certificates. A low assurance policy could be associated with a user requesting an activation code over the phone. A high assurance policy could be associated with a user requesting an activation code in person with proper identification. Depending on the organization's needs and policies, credentials may be issued to each user with one of these two assurance levels, depending on each user's authority and access control requirements.

In peer-to-peer cross-certification, suppose each user in ABC Corp's CA belongs to either the basic or high assurance group, and that each user is tagged as belonging to one or the other user group through the inclusion of a specific policy OID in the user's certificates. Next suppose Acme Corp would like to cross-certify with ABC Corp but would like to limit the trust relationship to only the high assurance users in ABC Corp. This can be accomplished through the use of policy constraints.


Trust is limited to only high assurance users in ABC Corp through the use of policy constraints
Acme Corp CA
ABC Corp CA
High assurance

Using policy constraints to limit trust to only those users with high assurance certificates can also be used in hierarchical cross-certification to prohibit subordinate CAs from adding low assurance users to the hierarchy. In this case the subordinate CA certificate issued to a subordinate CA could specifically prohibit the validation of user certificates that include low assurance policy OIDs.

Another peer-to-peer cross-certification example involves using policy constraints to limit trust to arbitrary sub-groups within another CA domain, independent of these users' DNs. For example, suppose a sub-group of employees within ABC Corp deal regularly with a sub-group of employees in Acme Corp., but these subgroups cross many functional boundaries and are not correlated to a particular DN structure. In this case policy OIDs could be placed in each employee's certificates to convey their need to deal with the other company and policy constraints could be used to limit the trust between the two organizations to just those employees with these particular policy OIDs.

3.4. Policy Mapping

Policy mapping is not a constraint but an interoperability feature. Policy mapping can be used to map one organization's policy with other organization's policy. For example, the policy OID which conveys high assurance could be different for each organization. Policy mapping information can be included in the cross-certificate or subordinate CA certificate to associate one policy OID with another. This allows organizations with different policies to still make use of policy constraints.

4. Summary

Hierarchical cross-certification is ideally suited *within* large organizations that want their root CA to have maximum control over all subordinate CAs within the organization's hierarchy. By contrast, peer-to-peer cross-certification is ideally suited *between* organizations where maximum flexibility is needed to form and revoke trust relationships with other organizations as changing business needs dictate. Through Entrust/PKI support for both hierarchical *and* peer-to-peer cross-certification, an organization has the flexibility and security to create the right PKI network to suit its needs.

PKI policy networking provides the flexibility and control needed to establish and enforce *limited* trust relationships that *mirror* the business relationships between or within organizations.

Entrust is a registered trademark of Entrust Technologies Limited.

All other product and company names are trademarks of their respective owners.