Entrust®
Securing Digital Identities & Information

WHITE PAPER

X.509 CERTIFICATION PATH VALIDATION

Date: 27 January 2004

Author: Sharon Boeyen

TABLE OF CONTENTS

1. ABSTRACT 1

2. INTRODUCTION 2

2.1 PKI ENTITIES 2

2.2 RELYING PARTY TRUST 3

2.3 BUSINESS CONTROLS 4

2.4 CERTIFICATE EXTENSIONS 4

3. PATH VALIDATION 6

3.1 PATH VALIDATION INPUTS 6

3.1.1 Trust Anchor 6

3.1.2 Certification Path 7

3.1.3 Revocation Status Information 7

3.1.4 Process Constraints 7

3.2 THE PATH VALIDATION PROCESS 8

3.2.1 Path Structure Processes 8

3.2.2 Certificate Validity Processes 8

3.2.3 Business Controls Processes 9

3.3 PROCESS OUTPUTS 11

4. CONFORMANCE 12

5. CURRENT STATUS 13

6. SUMMARY/CONCLUSIONS 14

7. REFERENCES 16

8. ABOUT ENTRUST 17

1. ABSTRACT

Public-key infrastructure enables authentication in federated environments. Management of that federation needs to be considered when planning PKI deployment, to ensure users and assets are adequately protected. [X.509] defines a standard set of business controls that can be imposed to manage the extent of federation. Certification path validation is the set of standard processes that users/applications relying on public-key certificates execute to determine their trustworthiness and acceptability. This includes ensuring all relevant business controls are respected. This paper describes the path validation process and how the business controls can be used to manage the extent of federation.

2. INTRODUCTION

[X.509] defines an authentication infrastructure based on public-key techniques. It performs implicit authentication using data encryption and explicit authentication using digital signature. [X.509] has been developed for use in federated environments. For this reason it defines sophisticated controls for managing the extent of federation. Path validation ensures that the controls are enforced and the extent of federation appropriately managed.

2.1 PKI ENTITIES

The basic entities involved in the enablement of secure and trusted transactions in a Public Key Infrastructure (PKI) are:

a) Certification Authority (CA)

b) Subscriber

c) Relying party

d) Policy Authority (PA)

The CA issues X.509 public-key certificates to subscribers, binding their identities to the public keys. A CA may also manage the life-cycle of those certificates including their expiration, renewal and revocation. The subscriber is the holder of keys/certificates for use as electronic credentials. The relying party is the entity that uses subscriber certificates to establish trust and security for electronic transactions. The PA is the individual/group that establishes and manages the domain security policies under which all entities in the domain operate. The PA sets policy and procedures under which the CA operates. This includes the certificate policy definitions, the Certification Practice Statement (CPS) for the CA's management of certificates and subscriber agreements. The subscriber agreement specifies the responsibilities and procedures for subscribers with respect to protection of their private keys, reporting lost/compromised keys as well any restrictions on the use of their keys and certificates. The relying party policy establishes the rules for acceptance of certificates. Together this set of policies establishes control over federation for the PKI administrative. A PKI administrative domain is a set of PKI entities operating under uniform domain security policy as established by its PA. Automation of some PA functions can be managed centrally through the same system that operates the CA, however logically these are two distinct entities. Figure 1 illustrates these relationships among entities in a PKI administrative domain.

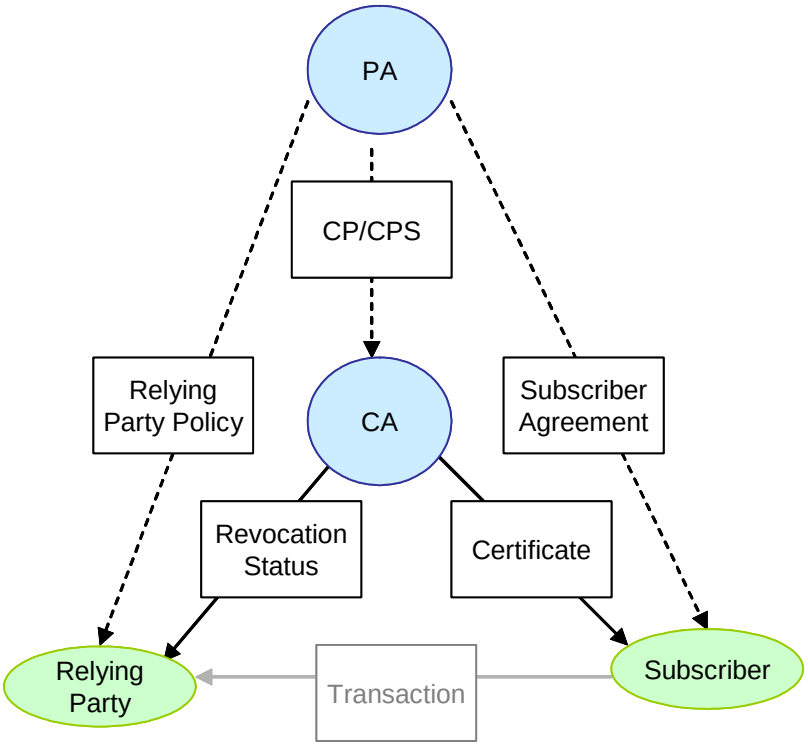

Figure 1: PKI Entities

2.2 RELYING PARTY TRUST

The primary focus for an organization deploying PKI is the protection of its own assets and protection of its users as both subscribers and relying parties. As relying parties, users make use of certificates to authenticate subscribers to some service, to accept a business commitment such as a purchase order, to ensure transmitted data is authentic and has not been altered and to enable privacy through encryption. Relying parties may be human users or application systems.

Relying parties should not blindly trust any and all issued certificates however. Before placing trust in a certificate, relying parties must create and validate a certification path. A certification path is a sequence of certificates beginning with a certificate issued by a known CA whose public key is already trusted by the relying party (trust anchor) and ending with a certificate issued to the subscriber. A valid certification path conforms to domain policy. If the subscriber's certificate was issued by the CA that is the relying party's trust anchor, the certification path comprises only that single certificate. Otherwise, the subscriber's certificate is preceded in the path by a sequence of CA certificates (certificates that have CAs as both the issuer and subject).

In federated environments PKI administrative domains establish relationships that enable relying parties in one domain to trust subscriber certificates issued in a different domain. In these situations, the certification path includes a CA certificate issued by a CA in the relying party domain to a CA in a federated domain. There may be additional CA certificates, issued by federated CAs to other federated CAs, if for example the subscriber is in a domain that is only indirectly federated with the relying party domain.

Certification path validation is the set of processes that ensures the path is properly structured, each certificate in the path is valid, and the subscriber's certificate is acceptable and appropriate for the intended use. Use of the subscriber's certificate must be done in compliance with the relevant policies of the relying party's domain as well as in compliance with the policies imposed on use of their certificates by PAs in all federated domains in the certification path.

2.3 BUSINESS CONTROLS

The PA may determine that relying parties in their community are permitted to trust only a subset of the certificates in the federation. This occurs most frequently in federated environments where remote CAs may issue certificates under policies deemed unacceptable by the local PA. In such cases, business controls can be used to place boundaries around the set of acceptable certificates, thereby controlling the extent of federation. These business controls are realized as extensions included in CA certificates issued by CAs in each domain. [X.509] and [PKIX] define a standard set of extensions specifically for this purpose. The use of certificate extensions to represent business controls in CA certificates is referred to as “infrastructure constraints” in this paper. In a federated environment, for example, infrastructure constraints can be used to limit the set of acceptable certificates issued in a remote domain to only those issued under a specific certificate policy or to a specific set of subjects. Since a certification path can include multiple CA certificates that span several federated CAs, several sets of infrastructure constraints may appear in a single certification path. Each CA in the path can impose a set of infrastructure constraints that limit the acceptability of subsequent certificates in the path in accordance with their own policy as established by their PA.

If all relying parties in a domain operate under identical policy, the policy can be enforced by infrastructure constraints in certificates issued by the local CA. However this is not always the case. For example, in a corporate setting, it may be that employees in the legal department can only accept digital signatures on contracts if the signer’s certificate was issued under a “high assurance” policy, but all employees can accept digitally signed email content if the signer’s certificate was issued under a high or medium assurance level policy. Infrastructure constraints cannot completely address this requirement. The PA needs the ability to partition the relying party community to impose additional constraints, over and above those imposed by the more generic infrastructure constraints, on some instances of path validation. [X.509] specifies a set of configurable initialization parameters for the path validation process. The tailoring of these parameters to satisfy the needs of a partitioned relying party community is referred to in this paper as “process constraints”.

The PA must be able to adapt its policy to changes in the environment and modify the conditions under which it’s local relying parties conduct path validation as the set of relationships with external CAs changes and as the nature of applications and transactions used by local relying parties changes. In order to maintain authority over its domain, process constraints should be managed by the authority through automated means thereby ensuring that individual relying parties do not deliberately or inadvertently circumvent the local security policy. While local policy may be managed centrally through the same system that operates the CA, the PA function is logically a distinct function.

Together, infrastructure constraints and process constraints enable standards compliant path validation processes to ensure that the relevant business controls imposed in the relying party domain, as well as those imposed in federated domains are respected. As a result only certificates that are compliant with the policies of each of those domains will be accepted by relying parties.

2.4 CERTIFICATE EXTENSIONS

There is a set of standard certificate extensions defined in [X.509] and [PKIX]. The majority of these play a role in the path validation process. Most are used to convey infrastructure constraints. Others are used in the processes that ensure a certification path is properly structured or to ensure that a specific certificate is valid in its own right. Figure 2 provides an outline of this set of certificate extensions and the role they play in path validation. Each of these is described in more detail in Section 3.

Extension	Path Structure	Certificate Validity	Business Controls
Authority key identifier	Key chaining		
Subject key identifier	Key chaining		
Basic constraints	Entity type		Path length control
Key usage		Appropriate key check	

Certificate policies			Policy controls
Policy mappings			Policy controls
Policy constraints			Policy controls
Inhibit any policy			Policy controls
Name constraints			Name controls
Subject alternative name			Name controls

Figure 2: Path Validation Extensions

The remaining standard certificate extensions serve other purposes, such as guidance for path development or support in locating revocation information. Those extensions are therefore outside the scope of this paper. However, for completeness they are listed below. Note that additional extensions can be defined by other groups for other purposes but those extensions would not have a role in standard path validation.

- Authority information access
- CRL distribution points
- Extended key usage
- Freshest CRL
- Issuer alternative name
- Private key usage period
- Subject directory attributes
- Subject information access

There is also a standard set of extensions that appear in CRLs rather than certificates. These are not listed here but can be found in [X.509] and [PKIX].

3. PATH VALIDATION

Within this paper, path validation is defined as the set of processes undertaken by the relying party to verify the integrity, trustworthiness and appropriateness of all certificates in a certification path, including the certificate issued to the subscriber. [X.509] describes all of these processes as necessary for the relying party, however only a subset are included in the "path processing procedures" portion of the standard. This paper combines the complete set of processes under the umbrella of "path validation" to provide a complete view of the processing necessary before a relying party trusts a subscriber certificate.

Path validation includes a set of process inputs, the processes themselves, and a set of process outputs. The processes themselves include checks to ensure that the certification path is a properly built path according to the [X.509] requirements, basic certificate checks that are performed against each certificate in the path, and a set of business controls checks to ensure that relying on the subscriber's certificate would adhere to the constraints put upon certificates in the path by the local PA as well as by all authorities that issued certificates included in the path.

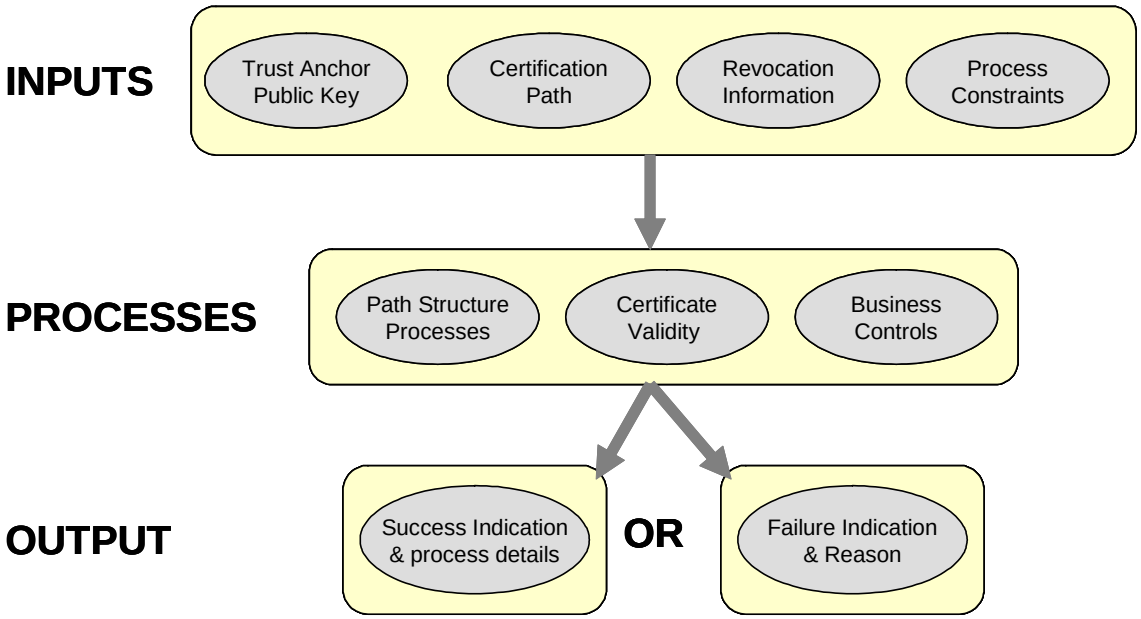

Figure 3: Path Validation Outline

3.1 PATH VALIDATION INPUTS

In addition to the certification path itself, the path validation process requires the relying party's trust anchor public key, revocation status information for each certificate in the path, and any process constraints relevant to the particular instance of path validation to tailor it to the policy environment in which the transaction is being executed (see Section 3.1.4).

3.1.1 Trust Anchor

A trust anchor is a CA that is trusted by the relying party and its public key is the basis upon which other trust decisions are made. There are a number of out-of-band schemes that can be used to convey trust anchor public keys to relying parties. These schemes need to ensure that relying parties are securely provided with the appropriate trust anchor public keys and that relying parties do not make use of CA public keys that may be conveyed through unapproved mechanisms. For example some PAs may restrict their relying party community to using a single trust anchor, the local CA, for all instances of path validation. In other environments, the PA may approve use of some/all trust anchors whose root keys are

pre-installed in popular browsers. There are additional schemes for conveying approved trust anchors and their public keys but specific schemes are outside the scope of this paper.

3.1.2 Certification Path

A certification path is a sequence of one or more certificates, with each certificate forming a link in a chain from the relying party's trust anchor to the subscriber's public key. In the special case where the relying party's trust anchor also issued the subscriber's certificate, the certification path consists of a single certificate, the subscriber's certificate. In some applications, such as S/MIME, a certification path or a portion of a certification path may be provided to the relying party by the subscriber. In other environments, the relying party must locate and retrieve the necessary certificates from some repositories. Certification path development techniques are outside the scope of this paper.

3.1.3 Revocation Status Information

In many environments, the CA is responsible for managing the life-cycle of the certificates it issues, including revoking those certificates if they are no longer to be relied upon, perhaps because the subscriber no longer meets the requirements for membership of the domain. If a certificate is revoked, the certificate issuing CA generally makes this status change available on a Certificate Revocation List (CRL). Revocation status can be communicated to relying parties through mechanisms other than CRLs but specific schemes are outside the scope of this paper. CRLs are updated on a regular basis enabling relying parties to determine the status at the start of the current window. CRLs may be used directly by relying parties or indirectly through servers that provide responses to revocation status queries. In other environments CAs may not perform revocation of certificates, but must make that information available to relying parties so they have that revocation status for certificates. Current revocation status information is required for each of the certificates in the certification path. Specific techniques for relying parties to obtain revocation status information are outside the scope of this paper.

3.1.4 Process Constraints

Process constraints are used to partition the relying party community to support differing policies among them. This partitioning may be done on the basis of a set of relying parties, on the basis of the application they use or even on the basis of specific transactions, depending on the requirements of the policy. Process constraints further restrict the acceptable set of certificates from that constrained by infrastructure constraints.

While the most obvious environment for use of process constraints is in a federated environment where the goal is to restrict relying parties to trusting only a subset of the entities in the federation, there are other uses that can be made of process constraints. Process constraints can be used, for instance to enable a simple authorization service. If role information is included in subscriber certificates, relying parties can authenticate the subscriber as well as make basic authorization decisions based upon the subscriber's authenticated identity and/or associated role information. Path validation, with an appropriate set of business controls, automatically provides this service. For example, assume that ABC Corp order entry system has a requirement to accept transactions from only XYZ Corp subscribers in the accounts payable area. The role of XYZ Corp employees may be indicated in their certificates explicitly through a certificate policy identifier. Process constraints can be used to restrict access to the order entry system to only those subscribers authenticating with an appropriate certificate.

Process constraints can also play a role within a single CA, non-federated, PKI. In this situation, there is no opportunity to impose infrastructure constraints so process constraints are the only way to impose restrictions on path validation. For example, assume a single CA PKI where the CA issues authentication certificates to users for use with a number of web-based electronic services. Some users, by design, may have credentials that can only be used to authenticate to a single specific service, while other users may have credentials that can be used for authenticating to all such services. In this example the service applications are relying parties when they perform certification path validation to ensure users are properly authenticated. Each such service may use process constraints to ensure that only certificates issued under specific certificate policies will be trusted.

3.2 THE PATH VALIDATION PROCESS

The path validation process consists of a number of individual processes that fall into three basic categories: path structure, certificate validity and business controls. There is no specific order in which the individual processes must be executed. However, because many of the business controls imposed through infrastructure constraints affect subsequent certificates in the path, there is a requirement to process certificates in the exact order they appear in the path beginning with the certificate that was signed by the relying party's trust anchor and ending with the subscriber certificate. [X.509] and [PKIX] both include an outline of a procedure for path validation in their standards. While standard conformance demands that an implementation be functionally equivalent to the external behavior resulting from those outlines, it is not mandatory for implementations to use either of those algorithms directly.

3.2.1 Path Structure Processes

Path structure processing consists of four tests. These tests ensure that the certification path is properly constructed. If all certificates in the path pass these tests, then the certification path is a proper path in that the intermediate entities are all CAs and the path forms an unbroken chain of certificates to the subscriber public key. If any of these individual processes fails, then the path validation process fails.

3.2.1.1 Name Chaining

Name chaining involves comparing the value of the subject field in one certificate with the issuer field in the subsequent certificate. For each pair of adjacent certificates in the path, these values must match. This check ensures that the certification path is truly an unbroken chain of entities relating directly to one another and that it has no missing links.

3.2.1.2 Key Chaining

Key chaining involves checking that the key certified in each certificate verifies the digital signature on the subsequent certificate. This check establishes cryptographic trust from the relying party's trust anchor to the subscriber's public key. Ensuring that the value of the **subjectKeyIdentifier** extension in one certificate matches the value of the **issuerKeyIdentifier** extension in the subsequent certificate contributes to the key chaining process.

3.2.1.3 Duplicate Certificates

The duplicate certificate process involves checking to ensure that each certificate appears only once in the path. If the path includes two certificates that have matching issuer names and identical serial numbers, then they are considered duplicates. This check ensures that a loop does not adversely alter the effect of business controls, changing the outcome of proper path validation. For example, a duplicate certificate could impact the path length controls (see Section 3.2.3.3).

3.2.1.4 Subject Entity Type

CAs are the only legitimate issuers of certificates. Therefore, each certificate in the path, except the subscriber's certificate, needs to be checked to ensure that its subject is also a CA. This check is performed by ensuring that the **basicConstraints** extension is present in the certificate, with the **cA** element set to **TRUE**. For some transactions, such as verifying the digital signature on a CRL, the subscriber of interest is itself a CA while in most other transactions the subscriber of interest is not a CA. Therefore this check is not required on the subscriber's certificate.

3.2.2 Certificate Validity Processes

Certificate validity processing ensures that each certificate is a valid X.509 public-key certificate and that the basic requirements for use of the public key are satisfied. If any of these processes results in failure for any certificate in the path, the path validation process fails.

3.2.2.1 Syntax Check

The syntax for certificates is defined in [X.509]. This includes a set of base certificate fields and an extension capability that enables additional information to be conveyed. Each certificate extension defines its own syntax. A standard set of extensions is specified in [X.509] and [PKIX]. The syntax of all base certificate fields as well as all known extensions must be checked to ensure the syntax of each conforms

to the standards. If there are unknown extensions present, their syntax need not (and cannot) be checked.

3.2.2.2 Integrity Check

The integrity of each certificate must be verified to ensure that certificate content has not been altered from that signed by the CA. This check involves verifying the digital signature on the certificate. If the signature on any certificate in the path fails to verify, then that certificate has been corrupted and cannot be trusted.

3.2.2.3 Validity Period Check

Each certificate includes a validity period indicating the window of time during which the certified public key is considered valid for use. Generally, the certificate validity period is compared against the current time. However, there are some circumstances when the certificate validity period is compared to a time in the past.

3.2.2.4 Revocation Check

Even if all certificates are within their validity period, it is possible that one or more may have been revoked by its issuer. Current revocation status information must be checked for each certificate in the path. For example, this process could ensure a relying party did not accept the signature of a recently dismissed disgruntled employee on a business transaction.

3.2.2.5 Criticality Check

All certificate extensions include an indication of whether their processing is critical to the acceptance of that certificate for use. The criticality flag provides a backward/forward compatibility mechanism that enables CAs to state what should happen when a relying party that does not yet support the new extension encounters it. If an extension is flagged critical, but the relying party path validation process does not support that extension, then the certificate cannot be used. Unknown non-critical extensions can safely be ignored and other processing continued. This process prevents relying parties from trusting a certificate under conditions that its issuer did not intend. Often the policy under which a certificate was issued will free its issuer from liability if the certificate is not used according to the issuer's specific policy, including processing all critical extensions in the certificate.

3.2.2.6 Key Usage Check

The **keyUsage** certificate extension is the mechanism used by the issuing CA to indicate the general use for the public key. For the subscriber's certificate this may be set to any one of a number of values. For example, if the key can be used for client authentication, the **keyUsage** extension would have the "**digitalSignature**" indicator set. If this is the only indicator set, such a certificate would not be appropriate for encrypting data for that subscriber. All other certificates in the path must have the **keyCertSign** indicator set in their **keyUsage** extension, indicating that these certificates can be used to verify the subject CA's signature on subsequent certificates.

3.2.3 Business Controls Processes

Business controls processes ensure that all certificates in the path adhere to business and policy constraints imposed on the path validation process through infrastructure constraints and/or process constraints. Process constraints are introduced at the beginning of the path validation process through initialization of a set of process variables. As such, process constraints automatically apply to the complete set of certificates in the path. There is no criticality flag associated with process constraints. All process constraints are treated as if flagged critical and must be adhered to. Most infrastructure constraints apply only to subsequent certificates, but some also impact the complete set of certificates in the path. Some infrastructure constraints can be flagged non-critical allowing unknown extensions to be ignored by relying party path validation processes that do not support them. Generally, if infrastructure constraints are flagged non-critical it is because the CA is introducing a new extension and anticipates a large community of relying parties that cannot process the extension. In this case the CA is stating that relying parties that do not yet support processing of the new extension can ignore its presence and proceed with the remaining processing.

The set of standard business controls can be broadly categorized as constraints imposed through naming, policy and length of acceptable certification paths. All of the specific controls listed below for each of these categories can be imposed as infrastructure constraints. A subset of these can also be imposed as process constraints. Infrastructure constraints can only be included in CA certificates. Using both techniques together, a CA can issue certificates to remote CAs that impose basic business controls uniformly applicable to the complete set of relying parties and use process constraints to further tailor the set of acceptable external certificates to the specific needs of relying parties, applications and transactions. Process constraints further limit the set of acceptable certificates to a subset of those acceptable under infrastructure constraints. If any of these individual constraints is present and supported but processing fails, then path validation fails. Path validation also fails if any individual constraint is present, flagged critical, but its processing is not supported.

3.2.3.1 Name Controls

[X.509] is based on the premise that all authorities that are "reachable" from a trust anchor operate a consistent naming scheme that results in no particular name being assigned to more than one entity. In order to ensure that certificate subject names are unambiguous, they are generally assigned within a hierarchical structure. Several nameforms, such as distinguished names, rfc822 names, DNS names, etc, are part of natural hierarchies given their fundamental syntax and structure. Name constraint business controls enable constraints of the following types to be specified:

- Any subject names of a specified nameform must be within nominated permitted subtrees in the hierarchy for that nameform;
- Any subject names of a specified nameform must not be within nominated "excluded subtrees" in the hierarchy for that nameform; and
- Only certificates that include names of a specified nameform are acceptable (note that this is a recent addition to the standards and not yet widely supported).

All three types of name constraint can be combined. At present, the standards only allow name controls to be specified as infrastructure constraints by setting the relevant fields of the **nameConstraints** extension. Work is currently underway to extend the standards to enable name controls as process constraints.

3.2.3.2 Policy Controls

Certificate policy is the standardized mechanism used by a CA to indicate the applicability of a certificate to a particular community and/or class of application with common security requirements, such as the authentication of a user to a home-banking application or the encryption of email messages among federal government employees. Each certificate policy has a unique identifier. Certificates may be issued under one or more policies as indicated in the **certificatePolicies** extension of that certificate. If two PKI administrative domains issue certificates under equivalent certificate policies, but use different identifiers for those policies, then they can equate those identifiers through the **policyMapping** extension in the cross-certificates issued between those CAs. The **certificatePolicies** extension can be included in both subscriber and CA certificates. The **policyMapping** extension can only be included in CA certificates. Unless otherwise constrained, a remote policy, mapped to an equivalent local policy, would be considered a match for its equivalent local policy. There is also a special identifier for "**anyPolicy**" that, can be included in certificates. Unless policy mapping has been inhibited by a related business control described below, this identifier is considered a match for any specific policy identifier. These two certificate extensions are the basic tools with which a number of policy-related constraints can be built.

As with other extensions, any policy related controls implemented as non-critical infrastructure constraints can be ignored by relying party path validation systems that do not support those extensions. The descriptions below assume that the extensions are supported by the relying party system.

One of the policy controls that can be imposed on path validation is that all certificates be issued under at least one common policy, without necessarily specifying a particular policy. This control can be imposed as an infrastructure constraint by setting the **requireExplicitPolicy** field of the **policyConstraints** extension and/or as a process constraint by setting the explicit-policy-indicator variable.

An additional control can be imposed limiting acceptable certificate policies to those issued under a specific set of identified policies. This control can be imposed as an infrastructure constraint by including a specific set of policy identifiers in the **certificatePolicies** extension and/or as a process constraint by initializing the initial-policy-set to the policies of interest.

Policy controls also enable the inhibiting of policy mappings. If this control is set, and if policy is required as discussed above, all certificates in the path must contain at least one identical policy identifier. Mapped policy identifiers are no longer considered acceptable matches in this case. This control can be imposed as an infrastructure constraint by setting the **inhibitPolicyMapping** field of the **policyConstraints** extension. In this case it impacts subsequent certificates only. This control can also be imposed as a process constraint by setting the policy-mapping-inhibit-indicator, in which case it impacts all certificates in the path.

As mentioned earlier, there is a special identifier for "**anyPolicy**" that is considered a match for any specific policy identifier. Note, however, that policy mappings are never permitted to/from this identifier. While this "wildcard" identifier does serve a purpose in some environments, there are many environments where it is considered harmful, in which case specific identifiers must be used. The final policy-related control outlaws the **anyPolicy** identifier. This control can be imposed as an infrastructure constraint by including the **inhibitAnyPolicy** extension. In this case it impacts subsequent certificates only. This control can also be imposed as a process constraint by setting the inhibit-any-policy-indicator, in which case it impacts all certificates in the path.

3.2.3.3 Path Length Controls

The final standardized business control enables CAs to constrain the length of a certification path. If one CA imposes a constraint on the acceptable length of a certification path, subsequent certificates in the path can only further reduce the acceptable length. Path length constraints in subsequent certificates that have a higher value must be ignored. At present, the base standards only enable this control as an infrastructure constraint through the inclusion of a **basicConstraints** extension with the **pathLenConstraint** field present. There is no corresponding process constraint.

3.3 PROCESS OUTPUTS

The path validation process results in either success or failure.

Success indicates that the subscriber's certificate, based on the inputs to the process, is acceptable for the intended use. Specifically:

- The path was properly constructed and all tests in section 3.2.1 passed;
- All certificates in the path are considered valid and all tests in section 3.2.2 passed; and
- All business controls imposed as process constraints as well as in infrastructure constraints have been satisfied. The only exception to this is where infrastructure constraints are included as non-critical extensions that are not supported by the relying party. In this case path validation still succeeds but some non-essential business controls have been ignored.

If path validation succeeds, then the output includes a list of certificate policy identifiers (if any) that were common to all certificates, an indication of any policy mappings that occurred during the process and an indication of whether the presence of policy identifiers in all certificates was a business control that had been imposed on the process for this specific path, either as a process constraint or an infrastructure constraint.

Path validation failure results if any one of the tests described in 3.2.1, 3.2.2 or 3.2.3 fails. If path validation fails, then the output includes an indication of the reason for failure. A subscriber key for which no valid path can be found should not be used by the relying party.

4. CONFORMANCE

Path validation is a critical element of protecting the interests of relying parties and ensuring that relying parties use only certificates that comply with the security policy of their local PA as well as that of all PAs of domains represented in the certification path. In addition to ensuring that relying party path validation systems being deployed satisfy the functional requirements of the local domain, another critical factor is ensuring that the systems being deployed are actually performing these processes in compliance with the definitions for path validation standardized in [X.509] and [PKIX]. The path validation process is a relatively complex process and until recently there has not been a comprehensive test suite available to assess conformance. In 2003 the National Institute of Standards and Technology (NIST), together with the National Security Agency (NSA) and DigitalNet, produced a test suite that tests path validation implementations to ensure that they perform elements of path validation in conformance with those base standards. The test suite, known as Public Key Interoperability Test Suite (PKITS) provides test descriptions as well as test data for all aspects of certification path validation. NIST has also produced a set of PKI client protection profiles that group together functional sets of path validation checks.

The basic profile includes all path structure and certificate validity checks described in 3.2.1 and 3.2.2 of this paper as well as a small number of business control checks described in 3.2.3.

The bridge-enabled profile adds requirements to support the process constraints and most of the infrastructure constraints described in 3.2.3.

Other profiles focus on enhanced CRL processing.

Each functional area specifies the set of tests that a relying party path validation system must pass. The PKITS test suite has been thoroughly reviewed by X.509 and [PKIX] experts and some vendors have already performed many of these tests privately as part of the review of the test suite. After a full year of review and scrutiny the PKITS test suite is now very stable and can be considered the definitive interpretation of the base standards. It is anticipated that within the early part of 2004 necessary approvals of both the test suite and the client protection profiles will be obtained and formal test laboratories will be established to perform independent testing against the PKITS test suite and accreditation of path validation implementations to specific protection profiles. Given the increasing popularity of the bridge trust model, the protection profile that enables implementations to claim they are “bridge-enabled” is expected to be particularly important, especially in government deployments.

5. CURRENT STATUS

Path validation is a requirement of all relying parties making use of public-key certificates, even if the path consists of a single subscriber certificate. Relying party path validation systems are currently provided by PKI-specific toolkit vendors, application vendors and web browser vendors. In the early days of PKI deployment, many of these systems were basic and performed only a small number of the checks described in this paper, while others were quite comprehensive. Presently, the vast majority of path validation systems perform at least the basic checks associated with the path structure and certificate validity described in 3.2.1 and 3.2.2 and many perform at least some of the checks associated with business controls described in 3.2.3. However, only a small minority of these perform all the checks described in this paper. The aspects that are currently supported only by the minority of path validation systems include all of the process constraints as well as those infrastructure constraints that are more recent additions to the standards, such as the **inhibitAnyPolicy** extension. Some business controls, such as a process constraint to support relying party partitioning based on name controls are still not yet standardized and are therefore not available in current relying party path validation systems. Once these new features are standardized, and PKITS is enhanced to include tests for conformance to them, these can reliably be added to the PKI deployment planning process.

6. SUMMARY/CONCLUSIONS

Path validation is a key element in ensuring that relying parties are protected from potential vulnerabilities when using public-key certificates. Path validation is the primary tool used by the relying party to ensure that certificates they are planning to use are authentic, issued by a trusted authority, valid, and appropriate for the intended use. Only after all these factors have been validated is the relying party afforded the assurances and protection of the issuing authorities in accordance with their stated policy and practices.

In addition to the basic checks to ensure a properly structured path and valid set of certificates, path validation includes processing a number of business controls conveyed as infrastructure constraints and as process constraints. Figure 4 illustrates how business controls can be used to manage the extent of federation, as determined by policy. Conformant path validation ensures these controls are enforced.

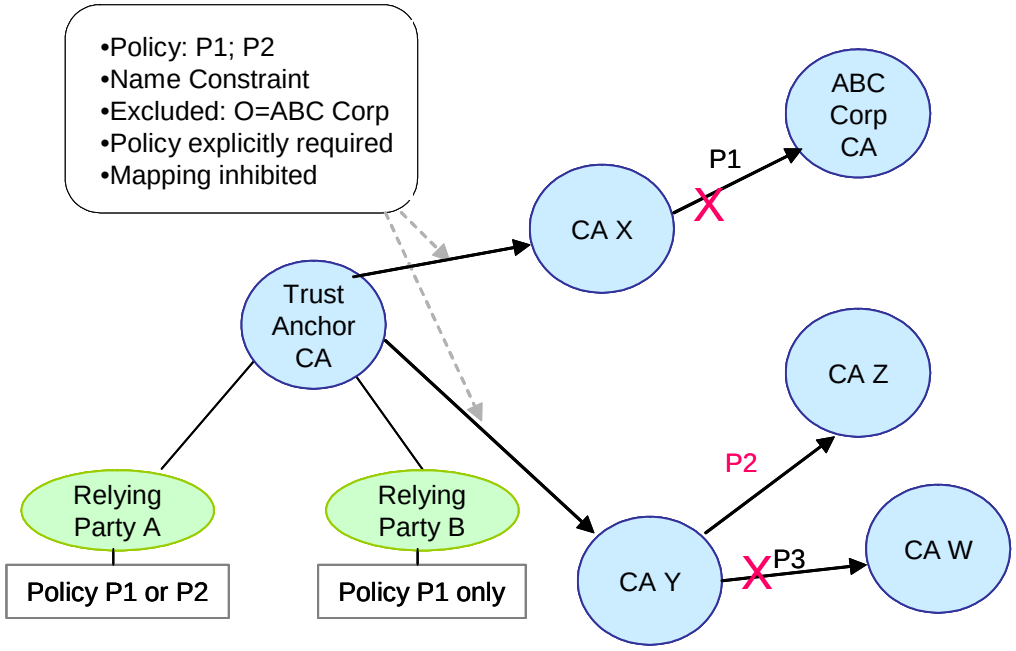

Figure 4: Managing Federation

In this example, the process constraint on relying party B prevents them from trusting certificates issued by CA Z, although those same certificates are trusted by relying party A. The policy controls imposed by their trust anchor CA prevent either relying party from trusting certificates issued by CA W. The name controls imposed by the trust anchor CA prevent either relying party from trusting the certificate issued to ABC Corp CA as well as any certificates issued by that CA.

As the federation evolves, the needs for infrastructure constraints and process constraints may change. It is important to be able to modify the constraints without disruption to the deployed users. For example, process constraints can be configured for subsets of relying parties and published in a repository from which they are automatically downloaded and initialize the variables in the path validation system. As changes are made to these configurations, the relying party systems can be automatically updated through the next download of settings from the repository. Similarly, if the infrastructure constraints need to be updated, it should be relatively easy to replace an existing CA certificate with one that includes updated infrastructure constraints and the old certificate can be revoked. Initial PKI deployment should be able to proceed with assurance that the environment can be updated as needs change, rather than holding up initial deployment until future external relationships are anticipated and planned in detail.

PKI deployment planning includes ensuring that products/services being deployed support the requirements to protect relying parties. Some functional aspects that need to be carefully considered include:

- CA support for inclusion of relevant infrastructure constraints;
- Central management of process constraints;
- Flexibility in configuring infrastructure and process constraints to suit specific and changing needs; and
- Relying party path validation support for processing initialization and infrastructure constraints.

Given the vital role path validation plays in PKI, it is critical that relying party systems that perform path validation implement the set of processes properly and in compliance with the requirements of the base standards. Otherwise, relying parties may unknowingly place trust in untrustworthy certificates, defeating the very purpose of the PKI. The PKITS test suite now provides an authoritative and comprehensive set of conformance tests for all aspects of path validation. These tests are freely available, along with the test data, from the NIST web site and can be run against a path validation system to ensure proper functioning. In addition to the PKITS test suite, the bridge-enabled protection profile provides a valuable companion specification describing the features that are required by a path validation implementation to claim support for the bridge-enabled environment. Relying party systems that support all these features and pass the corresponding PKITS tests provide a solid comprehensive path validation process.

7. REFERENCES

[X.509] ITU-T Recommendation X.509 (2000 E): Information Technology, Open systems interconnection - The Directory: Public-key and attribute certificate frameworks.

[PKIX] RFC 3280: Internet X.509 Public Key Infrastructure Certificate and CRL Profile

[PKITS]: Public Key Interoperability Test Suite http://csrc.nist.gov/pki/testing/x509paths.html

8. ABOUT ENTRUST

Entrust, Inc. [Nasdaq: ENTU] is a world leader in securing digital identities and information, enabling businesses and governments to transform the way they conduct online transactions and manage relationships with customers, partners and employees. Entrust's solutions promote a proactive approach to security that provides accountability and privacy to online transactions and information. Over 1,200 enterprises and government agencies in more than 50 countries use Entrust's portfolio of security software solutions that integrate into the broad range of applications organizations use today to leverage the Internet and enterprise networks. For more information, please visit **http://www.entrust.com.**