Internet X.509 Public Key Infrastructure: Roadmap

Status of this Memo

   This document is an Internet-Draft and is in full conformance with
   all provisions of Section 10 of [RFC2026].

   This document is an Internet-Draft. Internet-Drafts are working
   documents of the Internet Engineering Task Force (IETF), its areas,
   and its working groups. Note that other groups may also distribute
   working documents as Internet-Drafts.

   Internet-Drafts are draft documents valid for a maximum of six
   months and may be updated, replaced, or obsoleted by other documents
   at any time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This draft is being discussed on the 'ietf-pkix' mailing list. To
   subscribe, send a message to ietf-pkix-request@imc.org with the
   single word subscribe in the body of the message. There is a Web
   site for the mailing list at <http://www.imc.org/ietf-pkix/>.

Abstract

   This document provides an overview or "roadmap" of the work done by
   the IETF PKIX working group. It describes some of the terminology
   used in the working group's documents, and the theory behind an
   X.509-based Public Key Infrastructure, Privilege Management
   Infrastructure (PMI), and Time Stamping and Data Certification
   Infrastructures. It identifies each document developed by the PKIX
   working group, and describes the relationships among the various
   documents. It also provides advice to would-be PKIX implementors
   about some of the issues discussed at length during PKIX development,
   in hopes of making it easier to build implementations that will
   actually interoperate.

[1](#) Introduction

[1.1](#) This Document

   This document is an informational Internet-Draft that provides a
   "roadmap" to the documents produced by the PKIX working group. It is
   intended to provide information; there are no requirements or
   specifications in this document.

   [Section 1.2](#) of this document defines key terms used in this document.
   [Section 1.3](#) covers some of the basic history behind the PKIX working
   group. [Section 2](#) covers Public Key Infrastructure (PKI) theory and
   functions. [Section 3](#) covers Privilege Management Infrastructure (PMI)
   theory and functions. [Section 4](#) provides an overview of the various
   PKIX documents. It identifies which documents address which areas,
   and describes the relationships among the various documents. [Section
   5](#) contains "Advice to implementors." Its primary purpose is to
   capture some of the major issues discussed by the PKIX working group,
   as a way of explaining why some of the requirements and
   specifications say what they say. This explaination should cut down
   on the number of misinterpretations of the documents, and help
   developers build interoperable implementations. [Section 6](#) contains a
   list of contributors we wish to thank. [Section 7](#) provides a list
   references. [Section 8](#) discusses security considerations, and [Section](#)

[9](#) provides contact information for the editors.

1.2 Terminology

There are a number of terms used and misused throughout PKI-related, PMI-related, and Time Stamp and Data Certification literature. To limit confusion caused by some of those terms, used throughout this document, we will use the following terms in the following ways:

   - Attribute Authority (AA) - An authority trusted by one or more users to create and sign attribute certificates. It is important to note that the AA is responsible for the attribute certificates during their whole lifetime, not just for issuing them.

   - Attribute Certificate (AC) - A data structure containing a set of attributes for an end-entity and some other information, which is digitally signed with the private key of the AA which issued it.

   - Certificate - Can refer to either an AC or a public key certificate. Where there is no distinction made the context should be assumed that the term could apply to both an AC or a public key certificate.

   - Certification Authority (CA) - An authority trusted by one or more users to create and assign public key certificates. Optionally the CA may create the user's keys. It is important to note that the CA is responsible for the public key certificates during their whole lifetime, not just for issuing them.

   - Certificate Policy (CP) - A named set of rules that indicates the applicability of a public key certificate to a particular community or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of public key certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range.

   - Certification Practice Statement (CPS) - A statement of the practices which a CA employs in issuing public key certificates.

   - End-entity - A subject of a certificate who is not a CA in the PKI or an AA in the PMI. (An EE from the PKI can be an AA in the PMI.)

- Public Key Certificate (PKC) - A data structure containing the
  public key of an end-entity and some other information, which is
  digitally signed with the private key of the CA which issued it.

- Public Key Infrastructure (PKI) - The set of hardware, software,
  people, policies and procedures needed to create, manage, store,
  distribute, and revoke PKCs based on public-key cryptography.

- Privilege Management Infrastructure (PMI) - A collection of ACs,
  with their issuing AA's, subjects, relying parties, and
  repositories, is referred to as a Privilege Management
  Infrastructure.

- Registration Authority (RA) - An optional entity given
  responsibility for performing some of the administrative tasks
  necessary in the registration of subjects, such as: confirming
  the subject's identity; validating that the subject is entitled
  to have the values requested in a PKC; and verifying that the
  subject has possession of the private key associated with the
  public key requested for a PKC.

- Relying party - A user or agent (e.g., a client or server) who
  relies on the data in a certificate in making decisions.

- Root CA - A CA that is directly trusted by an EE; that is,
  securely acquiring the value of a Root CA public key requires
  some out-of-band step(s). This term is not meant to imply that a
  Root CA is necessarily at the top of any hierarchy, simply that
  the CA in question is trusted directly. Note that the term
  'trust anchor' is commonly used with the same meaning as 'root
  CA' in this document.

- Subordinate CA - A "subordinate CA" is one that is not a Root CA
  for the EE in question. Often, a subordinate CA will not be a
  Root CA for any entity but this is not mandatory.

- Subject - A subject is the entity (AA, CA, or EE) named in a
  certificate, either a PKC or AC. Subjects can be human users,
  computers (as represented by Domain Name Service (DNS) names or
  Internet Protocol (IP) addresses), or even software agents.

- Time Stamp Authority (TSA) - A TSA is a trusted Third Party who
  provides a "proof-of-existence" for a particular datum prior to
  an instant in time.

- Top CA - A CA that is at the top of a PKI hierarchy. Note: This
  is often also called a "Root CA," since in data structures terms

and in graph theory, the node at the top of a tree is the
"root." However, to minimize confusion in this document, we
elect to call this node a "Top CA," and reserve "Root CA" where
there is a single CA directly trusted by the EE. Readers new to
PKIX should be aware that these terms are not used consistently
throughout the PKIX documents, as the Internet PKI profile
[2459bis] uses "Root CA" to refer to what this and other
documents call a "Top CA," and "most-trusted CA" to refer to
what this and other documents call a "Root CA."

1.3 History

The PKIX working group was formed in October of 1995 to develop
Internet standards necessary to support PKIs. The first work item was
a profile of the ITU-T Recommendation X.509 PKC [FORMAT]. X.509,
which is a widely accepted basis for a PKI, including data formats
and procedures related to distribution of public keys via PKCs
digitally signed by CAs. X.509 does not however include a profile to
specify the support requirements for many of the PKC data structure's
sub- fields, for any of the extensions, nor for certain data values.
The Internet PKI profile [FORMAT] went through many draft versions
before becoming an RFC. Other profiles have been developed in PKIX
for particular algorithms to make use of the Internet PKI Profile
[FORMAT]. There has been no sense of conflict between the authors
that developed these profiles as they are seen as complimentary. The
Internet PKI profile has been a draft standard for more than six

months and is currently going through an update process to clarify
any inconsistencies and to bolster certain sections, see [2459bis].

In parallel with the profile development, work was undertaken to
develop the protocols necessary to manage PKI-related information
was. The first developed was the Certificate Management Protocol
(CMP). It defines a message protocol to initialize, certify, update,
and revoke PKI entities [CMP]. The demand for an enrollment protocol
and the desire to use PKCS-10 message format as the certificate
request syntax lead to the development of two different documents in
two different groups. The Certificate Request Syntax (CRS) draft was
developed in the SMIME WG which used PKCS-10 [PKCS10] as the
certification request message format. Certificate Request Message
Format [CRMF] draft was also developed but in the PKIX WG. It was to
define a simple enrollment protocol that would subsume both the CMP
and CRS enrollment protocols, but it did not use PKCS-10 as the
certificate request message format. Then the certificate management
message format document, was developed to define an extended set of
management messages that flow between the components of the Internet

PKI. Certificate Management Messages over CMS (CMC) was developed to allow the use of an existing protocol (S/MIME) as a PKI management protocol, without requiring the development of an entirely new protocol such as CMP [CMC]. It also included [PKCS10] as the certificate request syntax, which caused work on the CRS draft to stop. Information from the certificate management message format document was moved into [CMP] and [CMC] so work on the certificate management message format document was discontinued. After some operational experience with [CMP], two drafts, one for using HTTP as the transport protocol and one for Transmission Control Protocol (TCP), were written to solve problems encountered by implementors. These drafts were merged into one draft Transport Protocols for CMP [TPCMP]. [CMP] has been a draft standard for more than six months and is currently undergoing revisions to document. The transport section has been removed and will remain in [TPCMP].

Another long debated topic in the WG dealt with certificate revocation. Numerous drafts have been developed to address different issues related certificate revocations. CMP supports revocation request, response, revocation announcement, and requests for CRL messages. CMC defines revocation request, revocation response, and requests for CRL messages, but uses CMS as the encapsulating protocol. [OCSP] was developed to address concerns that not all relying parties want to go through the process checking CRLs from every CA in the certification path. It defines an on-line mechanism to determine the status of a given certificate, which may provide more timely revocation information than is possible with CRLs. The Simple Certification Verification Protocol (SCVP) was produced to allow relying parties to off-load all of their certification verification to another entity [SCVP]. The WG was arguably split over whether such a function should be supported and whether it should be its own protocol or included in OCSP. In response, a draft defining OCSP Extensions was produced to include the functions of SCVP. [OCSP] has been a draft standard for more than six months and is in the

process of being revised [OCSPv2]. To capture the work from the OCSP Extensions, two drafts were developed: Delegated Path Validation [DPV] and Delegated Path Discovery [DPD]. The WG recognizes an eed to address online delegated path validation and delegated path discovery. At least three candidates currently exist. There are: OCSPv2, SCVP, and DVCS. Given this multiplicity, the WG undertook to produce [DPREQ] in order to factilate selection from among these or possibly others.

One other certificate status draft called Open CRL Distribution Point (OCDP) was produced which documented two extensions: one to support an alternative CRL partitioning mechanism to the CRL Distribution

Point mechanism documented in the Internet PKI Profile [FORMAT] and one to support identifying other revocation sources available to certificate-users. The work from this draft was subsumed by an ITU-T | ISO/IEC Amendment to X.509, hence work on this draft was halted.

Development of the operational protocols has been slightly more straightforward. Four documents for the Light Weight Directory Access Protocol (LDAP) have been developed one for defining LDAPv2 as an access protocol to repositories [PKI-LDAPv2]; two for storing PKI information in an directory [SCHEMA] and [ADDSCHEMA]; and one for LDAPv3 requirements for PKI [PKI-LDAPv3]. Using the File Transfer Protocol (FTP) and the Hyper Text Transmission Protocol (HTTP) to retrieve PKCs and CRLs from PKI repositories was documented in [FTPHTTP]. Recognizing that LDAP directories are not the only repository service, the working group draft a Repository Locator Service [RLS] to make use of DNS SRV records to locate where and how PKI information can be retrieved from a repository.

In late 1998 the PKIX charter was revised to include protocols for time stamping and data certification services. [TSP] was developed to define protocols required to interact with a Time Stamp Authority (TSA) who asserts that a datum existed priot to a given time. [DVCS] allows to verify and assert the validity of all signatures attached to the signed document using all appropriate status information and PKCs or to verify and assert the validity of one or more PKCs at the specified time. Both [DVCS] and [TSP] use [CMS] as an encapsulating mechanism (though in [TSP] request for a time stamp are not required to use [CMS]). A draft for extending trust in tokens in time was developed to use [DCVS] to maintain the trust in a token issued by a non- repudiation Trusted Third Party (NR TTP) after the key initially used to establish trust in the token expires; however, this draft has expired. The [TRNRS] draft was developed to describe those features of a service which processes signed documents that must be present in order for that service to constitute a "technical non- repudiation" service.

Around the same time, a work item for ACs, defined in [X.509], was added. ACs are similar to PKCs, but they do not bind public keys to identities rather they bind attributes to identities. The attributes bound to the identity can represent anything, but are mostly used to support rule-based and role-based access control decisions. Two

drafts have since been developed: the Internet Attribute Certificates Profile for Authorizations [AC] and the Limited Attribute Certificate Acquisition Protocol [LAAP]. The first profiles the fields and extensions of the AC and the second provides a deliberately limited protocol to access a repository when LDAP is not appropriate.

Other drafts have been produced to address specific issues. [DHPOP]
was developed to define two mechanisms by which a signature can
produced using a Diffie-Hellman pair. This draft provides a mechanism
to use Diffie-Hellam key pairs to authenticate a PKCS-10
certification request. [REP] was developed during the revision to the
Internet PKI Profile [FORMAT] to separate the definitions of the
object identifiers and encoding rules for keys and digital signatures
in PKCs. The Qualified Certificates [QC] and Permanent Identifier
[PI] drafts were developed to address naming issues.

From the alphabet soup above, it is clear why this roadmap is
required.

2 PKI

2.1 Theory

At the heart of recent efforts to improve Internet security are a
group of security protocols such as Secure Multipurpose Internet Mail
Extensions (S/MIME), Transport Layer Security (TLS), and Internet
Protocol Security (IPSec). All of these protocols rely on public-key
cryptography to provide services such as confidentiality, data
integrity, data origin authentication, and non-repudiation. The
purpose of a PKI is to provide trusted and efficient key and public
key certificate management, thus enabling the use of authentication,
non-repudiation, and confidentiality.

Users of public key-based systems must be confident that, any time
they rely on a public key, the subject that they are communicating
with owns the associated private key, this applies whether an
encryption or digital signature mechanism is used. This confidence is
obtained through the use of PKCs, which are data structures that bind
public key values to subjects. The binding is achieved by having a
trusted CA verify the subject's identity and digitally sign each PKC.

A PKC has a limited valid lifetime, which is indicated in its signed
contents. Because a PKC's signature and timeliness can be
independently checked by a certificate-using client, PKCs can be
distributed via untrusted communications and server systems, and can
be cached in unsecured storage in certificate-using systems.

PKCs are used in the process of validating signed data. Specifics
vary according to which algorithm is used, but the general process
works as follows (Note: there is no specific order in which the
checks listed below must be made; implementors are free to implement
them in the most efficient way for their systems):

- The recipient of signed data verifies that the claimed identity
     of the user is in accordance with the identity contained in the
     PKC;

   - The recipient validates that no PKC in the path is revoked (e.g.,
     by retrieving a suitably-current Certificate Revocation List
     (CRL) or querying an on-line certificate status responder), and
     that all PKCs are within their validity periods at the time the
     data was signed;

   - The recipient verifies that the data are not claimed to have any
     values for which the PKC indicates that the signer is not
     authorized;

   - The recipient verifies that the data have not been altered since
     signing, by using the public key in the PKC.

   - If all of these checks pass, the recipient can accept that the
     data was signed by the purported signer. The process for keys
     used for encryption is similar.

Note: It is of course possible that the data was signed by someone
very different from the signer, if for example the purported signer's
private key was compromised. Security depends on all parts of the
certificate-using system, including but not limited to: physical
security of the place the computer resides; personnel security (i.e.,
the trustworthiness of the people who actually develop, install, run,
and maintain the system); the security provided by the operating
system on which the private key is used; and the security provided
the CA. A failure in any one of these areas can cause the entire
system security to fail. PKIX is limited in scope, however, and only
directly addresses issues related to the operation of the PKI
subsystem. For guidance in many of the other areas, see [POLPROC].

2.2 Architecture Model

A PKI is defined as:

The set of hardware, software, people, policies and procedures needed
to create, manage, store, distribute, and revoke PKCs based on
public-key cryptography.

A PKI consists of five types of components [MISPC]:

   - Certification Authorities (CAs) that issue and revoke PKCs;

   - Organizational Registration Authorities (ORAs) that vouch for the
     binding between public keys and certificate holder identities
     and other attributes;

    - PKC holders are issued certificates and can sign digital
      documents and decrypt documents using private keys;

    - Clients that validate digital signatures and their certification
      paths from a known public key of a trusted CA and that encrypt
      document using public key from certificates of PKC holders;

    - Repositories that store and make available PKCs and Certificate
      Revocation Lists (CRLs).

   Figure 1 is a simplified view of the architectural model assumed by
   the PKIX Working Group.

```
    +---+      cert. publish          +------------+
    |   |  <---------------------  | End Entity | <-------
    | C |                             +------------+      "out-of-band"
    |   |                                | ^                  loading
    | e |                                | |        initial
    | r |                                | |         registration/
    | t |                                | |          certification
    |   |                                | |         key pair recovery
    | / |                                | |         key pair update
    |   |                                | |         certificate update
    | C |   PKI "USERS"              V |         revocation request
    | R | ------------------+-+-----+-+------+-+-------------------
    | L |   PKI MANAGEMENT     | ^                  | ^
    |   |      ENTITIES        | |                  | |
    |   |                      V |                  | |
    | R |              +------+              | |
    | e |   <------------ | RA    | <-----+       | |
    | p |       cert.     |       | ----+ |       | |
    | o |        publish   +------+     | |       | |
    | s |                               | |    | |
    | i |                              V |    V |
    | t |                        +-----------+
    | o |   <----------------------|     CA    |------->
    | r |                          +-----------+  "out-of-band"
    | y |      cert. publish            | ^          publication
    |   |      CRL publish              | |
    +---+                               | |     cross-certification
                                        | |     cross-certificate
                                        | |        update
                                        | |
                                        V |
```

```
                        +------+
                        | CA-2 |
                        +------+


                   Figure 1 - PKI Entities
```

2.3 Public Key Certificates

   ITU-T X.509 (formerly CCITT X.509) or ISO|IEC/ITU 9594-8, which was
   first published in 1988 as part of the X.500 Directory
   recommendations, defines a standard PKC format [X.509]. The PKC
   format in the 1988 standard is called the version 1 (v1) format.

   When X.500 was revised in 1993, two more fields,
   subjectUniqueIdentifier and issuerUniqueIdentifier were added,
   resulting in the version 2 (v2) format. These two fields may be used
   to support directory access control.

   The Internet Privacy Enhanced Mail (PEM) RFCs, published in 1993,
   include specifications for a public key infrastructure based on X.509
   v1 public key certificates [PEM]. The experience gained in attempts
   to deploy [PEM] made it clear that the v1 and v2 public key
   certificate formats are deficient in several respects. Most
   importantly, more fields were needed to carry information which PEM
   design and implementation experience has proven necessary. In
   response to these new requirements, ISO|IEC, ITU, and ANSI X9
   developed the X.509 version 3 (v3) PKC format. The v3 format extends
   the v2 format by adding provision for additional extension fields.
   Particular extension field types may be specified in standards or may
   be defined and registered by any organization or community. In June
   1996, standardization of the basic v3 format was completed [X.509].

   ISO|IEC, ITU, and ANSI X9 have also developed standard extensions for
   use in the v3 extensions field [X.509][X9.55]. These extensions can
   convey such data as additional subject identification information,
   key attribute information, policy information, and certification path
   constraints. However, the ISO/IEC/ITU and ANSI X9 standard extensions
   are very broad in their applicability. In order to develop
   interoperable implementations of X.509 v3 systems for Internet use,
   it is necessary to specify a profile for use of the X.509 v3
   extensions tailored for the Internet. It is one goal of PKIX to
   specify a profile for Internet, electronic mail, and IPSec
   applications, etc. Environments with additional requirements may
   build on this profile or may replace it.

2.4 **Functions of a PKI**

This section describes the major functions of a PKI. In some cases, PKIs may provide extra functions.

2.4.1 **Registration**

This is the process whereby a subject first makes itself known to a CA (directly, or through an RA), prior to that CA issuing a PKC or PKCs for that subject. Registration involves the subject providing its name (e.g., common name, fully-qualified domain name, IP address), and other attributes to be put in the PKC, followed by the

Arsenault, Turner   11

CA (possibly with help from the RA) verifying in accordance with its Certification Practice Statement (CPS) that the name and other attributes are correct.

2.4.2 **Initialization**

Initialization is when the subject (e.g., the user or client system) gets the values needed to begin communicating with the PKI. For example, initialization can involve providing the client system with the public key or PKC of a CA, or generating the client system's own public-private key pair.

2.4.3 **Certification**

This is the process in which a CA issues a PKC for a subject's public key, and returns that PKC to the subject or posts that PKC in a repository.

2.4.4 **Key Pair Recovery**

In some implementations, key exchange or encryption keys will be required by local policy to be "backed up," or recoverable in case the key is lost and access to previously-encrypted information is needed. Such implementations can include those where the private key exchange key is stored on a hardware token that can be lost or broken, or when a private key file is protected by a password which can be forgotten. Often, a company is concerned about being able to read mail encrypted by or for a particular employee when that employee is no longer available because she is ill or no longer works

for the company.

In these cases, the user's private key can be backed up by a CA or by
a separate key backup system. If a user or her employer needs to
recover these backed up key materials, the PKI must provide a system
that permits the recovery without providing an unacceptable risk of
compromise of the private key.

2.4.5 Key Generation

Depending on the CA's policy, the private-public key pair can either
be generated by the user in his local environment, or generated by
the CA. In the latter case, the key material may be distributed to
the user in an encrypted file or on a physical token (e.g., a smart
card or PC card).

2.4.6 Key Update

All key pairs need to be updated regularly (i.e., replaced with a new
key pair) and new PKCs issued. This will happen in two cases:
normally, when a key has passed its maximum usable lifetime; and
exceptionally, when a key has been compromised and must be replaced.

2.4.6.1 Key Expiry

In the normal case, a PKI needs to provide a facility to gracefully
transition from a PKC with an existing key to a new PKC with a new
key. This is particularly true when the key to be updated is that of
a CA. Users will know in advance that the key will expire on a
certain date; the PKI, working together with PKC-using applications,
should allow for appropriate keys to work before and after the
transition. There are a number of ways to do this; see [CMP] for an
example of one.

2.4.6.2 Key Compromise

In the case of a key compromise, the transition will not be
"graceful" in that there will be an unplanned switch of PKCs and
keys; users will not have known in advance what was about to happen.
Still, the PKI must support the ability to declare that the previous

PKC is now invalid and shall not be used, and to announce the
validity and availability of the new PKC.

Note: compromise of a private key associated with a Root CA is
catastrophic for users relying on that Root CA. If a Root CA's
private key is compromised, that CA's PKC must be revoked and all
PKCs subordinate to it must also be revoked. Until such time as the
Root CA has been issued a new PKC and the Root CA issues PKCs to
users relying upon it, users relying on that Root CA are cut off from
the rest of the system, as there is no way to develop a valid
certification path back to a trusted node.

Further, users will likely have to be notified by out-of-band
mechanisms about the change in CA keys. If the old key is
compromised, any "update" message telling subordinates to switch to a
new key could have come from an attacker in possession of the old
key, and could point to a new public key for which the attacker
already has the private key. It is possible to have anticipated this
event, and "pre-placed" replacement Root CA keys with all relying
parties, but some secure, out-of-band mechanism will have to be used
to tell users to make the switch, and this will only help if the
replacement key has not been compromised.

Additionally, once the Root CA is brought back up with a new key, it
will likely be necessary to re-issue PKCs, signed with the new key,
to all subordinate users, since their current PKC would be signed
with a now-revoked key.

Arsenault, Turner   13

2.4.7 **Cross-certification**

A CA certificate is a certificate in a hierarchy that is neither a
self-signed certificate, nor an end-entity certificate. [2459bis]
does not make a difference between a CA certificate and a cross
certificate since it defines a cross-certificate as "a certificate
issued by one CA to another CA". Some people in the WG believe that
a cross certificate is a special kind of CA certificate. A cross
certificate is issued by a CA under one Top CA for another CA under
a different Top CA. CAs in the same hierarchy have part of their
names imposed by the Top CA or by the CAs under that Top CAS. When a
cross certificate is issued, there is no relationship between the
names of the CAs.

Typically, a cross-certificate is used to allow client systems or
end entities in one administrative domain to communicate securely
with client systems or end users in another administrative domain.
Use of a cross-certificate issued from CA_1 to CA_2 allows user

Alice, who trusts CA_1, to accept a PKC used by Bob, which was
issued by CA_2. Cross-certificates can also be issued from one CA to
another CA in the same administrative domain, if required.

Cross-certificates can be issued in only one direction, or in both
directions, between two CA's. That is, just because CA_1 issues a
cross-certificate for CA_2, CA_2 does not have to issue a cross-
certificate for CA_1.

2.4.8 Revocation

When a PKC is issued, it is expected to be in use for its entire
validity period. However, various circumstances may cause a PKC to
become invalid prior to the expiration of the validity period. Such
circumstances include change of name, change of association between
subject and CA (e.g., an employee terminates employment with an
organization), and compromise or suspected compromise of the
corresponding private key. Under such circumstances, the CA needs to
revoke the PKC.

X.509 defines one method of PKC revocation. This method involves each
CA periodically issuing a signed data structure called a certificate
revocation list (CRL). A CRL is a list that identifies the
references of revoked PKCs. This list contains a date of issue and
is signed by a CA and made freely available in a public repository.
Each revoked PKC is identified in a CRL by its PKC serial number.
When a certificate-using system uses a PKC, that system not only
checks the PKC signature and validity but also acquires a suitably
recent CRL and checks that the PKC serial number is not on that CRL.
The meaning of "suitably recent" may vary with local policy, but it
usually means the most recently issued CRL. A CA issues a new CRL on
a regular periodic basis (e.g., hourly, daily, or weekly). CA's may

also issue CRLs aperiodically. For example, if an important key is
deemed compromised, the CA may issue a new CRL to expedite
notification of that fact, even if the next CRL does not have to be
issued for some time. (A problem of aperiodic CRL issuance is that
end-entities may not know that a new CRL has been issued, and thus
may not retrieve it from a repository.)

An entry is added to the CRL as part of the next update following
notification of revocation. An entry may be removed from the CRL
after appearing on one regularly scheduled CRL issued beyond the
revoked PKC's validity period. Leaving the revoked PKC on the CRL for
this extra period allows for PKCs that are revoked prior to issuing a
new CRL and whose invalidity date falls before the CRL issuing time

to be accounted for. If the revoked PKC is not retained on the CRL
for this extra period then the possibility arises that a revoked PKC
may never appear on a CRL.

An advantage of the CRL revocation method is that CRLs may be
distributed by exactly the same means as PKCs themselves, namely, via
untrusted communications and server systems.

One limitation of the CRL revocation method, using untrusted
communications and servers, is that the time granularity of
revocation is limited to the CRL issue period. For example, if a
revocation is reported now, that revocation will not be reliably
notified to certificate-using systems until the next CRL is issued,
which may be up to one hour, one day, or one week depending on the
frequency that the CA issues CRLs.

As with the X.509 v3 PKC format, in order to facilitate interoperable
implementations from multiple vendors, the X.509 v2 CRL format needed
to be profiled for Internet use. This was done as part of the
Internet PKI Profile [FORMAT]. However, PKIX does not require CAs to
issue CRLs. On-line methods of revocation notification may be
applicable in some environments as an alternative to the X.509 CRL.
PKIX defines a few protocols that support on-line checking. [OCSP],
[DVCS], and [SCVP] all support on-line checking of the status of
PKCs.

On-line revocation checking may significantly reduce the latency
between a revocation report and the distribution of the information
to relying parties. Once the CA accepts the report as authentic and
valid, any query to the on-line service will correctly reflect the
PKC validation impacts of the revocation. However, these methods
impose new security requirements; the PKC validator must trust the
on-line validation service while the repository does not need to be
trusted.

2.4.9 **Certificate & Revocation Notice Distribution & Publication**

As alluded to in sections 2.1 and 2.5.8 above, the PKI is responsible
for the distribution of PKCs and PKC revocation notices (whether in

CRL form or in some other form) in the system. "Distribution" of PKCs
includes transmission of the PKC to its owner, and may also include
publication of the PKC in a repository. "Distribution" of revocation
notices may involve posting CRLs in a repository, transmitting them
to end-entities, or forwarding them to on-line responders.

[3](#) PMI

[3.1](#) Theory

Many systems use the PKC to perform identity based access control decisions (i.e., the identity may be used to support identity-based access control decisions after the client proves that it has access to the private key that corresponds to the public key contained in the PKC). For many systems this is sufficient, but increasingly systems are beginning to find that rule-based and role-based access control is required. These forms of access control decisions require additional information that is normally not included in a PKC, because the lifetime of the information is much shorter than the lifetime of the public-private key pair. To support binding this information to a PKC the Attribute Certificate (AC) was defined in ANSI and later incorporated into ITU-T Recommendation X.509. The AC format allows any additional information to be bound to a PKC by including, in a digitally signed data structure, a reference back to one specific PKC or to multiple PKCs, useful when the subject has the same identity in multiple PKCs. Additionally, the AC can be constructed in such a way that it is only useful at one or more particular targets (e.g., web server, mail host).

Users of a PMI must be confident that the identity purporting to posses an attribute has the right to possess that attribute. This confidence may be obtained through the use of PKCs or it may be configured in the AC-using system. If PKCs are used the party making the access control decision can determine "if the AC issuer is trusted to issue ACs containing this attribute."

ACs are complicated by the fact that they can point to an identity which may be in more than one PKC. If the RP has multiple certification chains to chose from then it has to make the determination as to which certification path to trust. Regardless, before the RP uses the AC it must make sure that a path from the AC back to its trust point is valid.

[3.2](#) Architectural Model

A Privilege Management Infrastructure, or PMI, is defined as:

The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke ACs.

A PMI consists of five types of components [[AC](#)]:

- Attribute Authorities (AAs), or Attribute Certificate Issuer,
  that issue and revoke ACs;

Note: AAs may implicitly revoke ACs by using very short validity
periods.

- Attribute Certificate Users that parses or processes an AC;

- Attribute Certificate Verifiers that check the validity of an AC
  and then makes use of the result;

- Clients that request an action for which authorization checks are
  to be made;

- Repositories that store and make available certificates and
  Certificate Revocation Lists (CRLs).

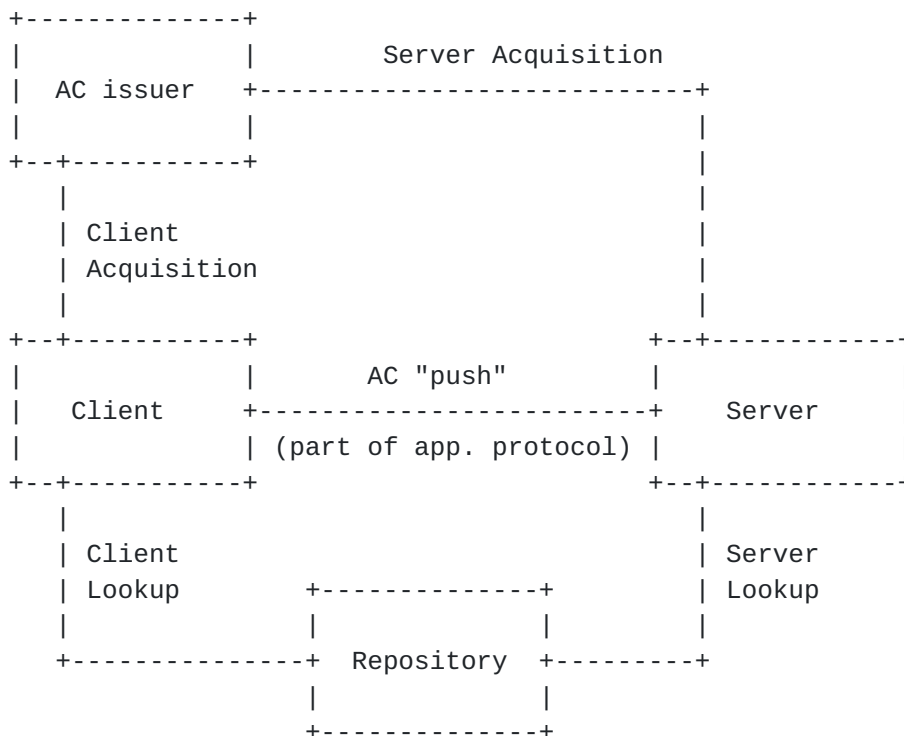Figure 2 is an example of the exchanges that may involve ACs.

```
    +--------------+
    |              |              Server Acquisition
    |   AC issuer  +----------------------------+
    |              |                            |
    +--+-----------+                            |
       |                                        |
       | Client                                 |
       | Acquisition                            |
       |                                        |
    +--+-----------+              +--+------------+
    |              |   AC "push"  |  |            |
    |    Client    +--------------------------+   Server    |
    |              | (part of app. protocol) |             |
    +--+-----------+              +--+------------+
       |                             |
       | Client                      | Server
       | Lookup       +--------------+   | Lookup
       |              |              |   |
    +---------------+  Repository  +---------+
                   |              |
                   +--------------+
```

                  Figure 2: AC Exchanges

Attribute Certificates

ANSI X.9 first published the Attribute Certificate format. It defined
the standard version 1 (v1) AC format. They later created a version 2
(v2) AC by modifying the owner field to point to either an identity
or a specific PKC and including an extension mechanism. In 1997 ITU-T
included it in [X.509].

   ANSI, ITU-T, and IETF have developed standard extensions and
   attributes for use in the v2 ACs. Extensions can convey such
   information as an audit identity that can be used to create an audit
   trail, identity specific servers and services where the AC owner can
   use their AC, point to a specific issuer's key, and indicate where to
   get revocation information. The AC is generic enough to allow any
   attribute to be conveyed in the data structure. Without limiting the
   attributes and extensions that can be included in an AC it is very
   difficult to develop interoperable implementations for Internet use.
   It is the goal of PKIX to specify a profile for the Internet,
   electronic mail, IPSec applications, etc. Environments with
   additional requirements may build on this profile or replace it.

   The [AC] profile constrains many of the options allowed in X.509. For
   example, the AC chains, like their PKC brethren, are allowed by
   X.509, but the AC profile recommends that they not be supported in to
   simplify the implementation.

4 PKIX Documents

   This section identifies the five different areas in which the PKIX
   working group has developed documents. The first area involves
   profiles of the X.509 v3 PKC standards and the X.509 v2 CRL standards
   for the Internet. The second area involves operational protocols, in
   which relying parties can obtain information such as PKCs or PKC
   status. The third area covers management protocols, in which
   different entities in the system exchange information needed for
   proper management of the PKI. The fourth area provides information
   about certificate policies and certificate practice statements,
   covering the areas of PKI security not directly addressed in the rest
   of PKIX. The fifth area deals with providing time stamping and data
   certification services, which can be used to build such services as
   non-repudiation.

4.1 Profiles

   An X.509 v3 PKC is a very complex data structure. It consists of
   basic information fields, plus a number of optional extensions. Many
   of the fields and numerous extensions can take on a wide range of
   options. This provides an enormous degree of flexibility, which
   allows the X.509 v3 PKC format to be used with a wide range of
   applications in a wide range of environments. Unfortunately, this
   same flexibility makes it extremely difficult to produce independent
   implementations that will actually interoperate with one another. In
   order to build an Internet PKI based on X.509 v3 PKCs, the PKIX

working group had to develop a profile of the X.509 v3 PKC
specification.

A profile of the X.509 v3 PKC specification is a description of the
contents of the PKC and which extensions must be supported, which
extensions may be supported, and which extensions may not be

supported. The Internet PKI Profile [FORMAT] provides such a profile
of X.509 v3 PKC for the Internet PKI. In addition, the Internet PKI
Profile [FORMAT] suggests ranges of values for many of the
extensions.

The Internet PKI Profile [FORMAT] also provides a profile for Version
2 CRLs for use in the Internet PKI. CRLs, like PKCs, have a number of
optional extensions. In order to promote interoperability, it is
necessary to constrain the choices an implementor supports.

In addition to profiling the PKC and CRL formats, it is necessary to
define particular Object Identifiers (OIDs) for certain encryption
algorithms, because there are a variety of OIDs registered for some
algorithm suites. PKIX has produced two documents ([RPKDS] and [KEA])
which provide guidance on the proper implementation of specific
algorithms.

Some countries are in a process of updating their legal frameworks in
order to regulate and incorporate recognition of signatures in
electronic form. Many of these frameworks introduce certain basic
requirements on PKCs, often termed Qualified Certificates, supporting
these types of "legal" signatures. Partly as a result of this there
is a need for a specific PKC profile providing standardized support
for certain related issues such as a common structure for expressing
unambiguous identities of certified subjects (unmistakable identity).
In December 1998, PKIX adopted as a work item the development of a
refinement of [RFC2459] that further profiles PKIX PKC into qualified
certificates. This work is reflected in [QC].

Like the X.509 v3 PKC, the AC also a very complex data structure
consisting of basic information fields, a number of optional
extensions, and a virtually unlimited number of attributes. Again,
many of the fields, extensions, and attributes can take on a wide
range of options allowing an enormous degree of flexibility. In order
to build an Internet PMI based on ACs, the PKIX working group had to
develop a profile of the AC.

The AC profile is description of the contents of the AC, the allowed
and required extensions, and applicable attributes. [AC] provides
such a profile of the X.509 v2 AC.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
     Certificate and CRL Profile (RFC2459) [FORMAT]

     DESCRIPTION: This document describes the profiles to be used for
     X.509 v3 PKCs and version 2 CRLs by Internet PKI participants. The
     profiles include the identification of ISO/IEC/ITU and ANSI
     extensions which may be useful in the Internet PKI. The profiles
     are presented in the 1988 Abstract Syntax Notation One (ASN.1)
     rather than the 1994 syntax used in the ISO/IEC/ITU standards.
     Would-be PKIX implementors and developers of certificate-using
     applications should start with the Internet PKI Profile [FORMAT] to

     ensure that their systems will be able to interoperate with other
     users of the PKI.

     The Internet PKI Profile [FORMAT] also includes path validation
     procedures. The procedures presented are based upon the ISO/IEC/ITU
     definition, but the presentation assumes one or more self-signed
     trusted CA PKCs. The procedures are provided as examples only.
     Implementations are not required to use the procedures provided;
     they may implement whichever procedures are efficient for their
     situation. However, implementations are required to derive the same
     results as the example procedures.

     STATUS: Proposed Standard.

   - DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
     Representation of Key Exchange Algorithm (KEA) Keys in Internet
     X.509 Public Key Infrastructure Certificates (RFC 2528) [KEA]

     DESCRIPTION: This document provides Object Identifiers (OIDs) and
     other guidance for IPKI users who use the Key Exchange Algorithm
     (KEA). It profiles the format and semantics of the
     subjectPublicKeyInfo field and the keyUsage extension in X.509 v3
     PKCs containing KEA keys. This document should be used by anyone
     wishing to support KEA; others who do not support ECDSA are not
     required to comply with it.

     STATUS: Informational RFC.

   - DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Qualified
     Certificates (RFC 3039) [QC]

     DESCRIPTION: This document profiles the format for and defines
     requirements on information content in a specific type of PKCs
     called Qualified Certificates. A "Qualified Certificate" is a PKC
     that is issued to a natural person (i.e., a living human being);

contains an unmistakable identity based on a real name or a
pseudonym of the subject; exclusively indicates non-repudiation as
the key usage for the certificate's public key; and meets a number
of requirements.

STATUS: Proposed Standard.

- DOCUMENT TITLE: An Internet Attribute Certificate Profile for
  Authorizations <draft-ietf-pkix-ac509prof-09.txt> [AC]

  DESCRIPTION: This document profiles the format for an defines
  requirements on X.509 v2 ACs to support authorization services
  required by various Internet protocols (TLS, CMS, and the consumers
  of CMS, etc.). Two profiles are defined in support of basic
  authorizations and in support of services that can operate via
  proxy.

STATUS: Approved as Proposed Standard; in RFC editor's Queue.
Issuance as an RFC blocked until the normative reference [2459bis]
progresses to Proposed Standard as well. (See below.)

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
  Certificate and CRL Profile <draft-ietf-pkix-new-part1-12.txt>
  [2459bis]

  DESCRIPTION: This document is an update of the Internet PKI Profile
  [2459bis]. The treatment of path validation is enhanced, and
  additional specificity is offered for various certificate and CRL
  extensions. This document omits the encoding and identification of
  public keys and digital signatures. (See [RPKDS] below.)

  STATUS: Tentatively approved by IESG.

- DOCUMENT TITLE: Algorithms and Identifiers for the Internet X.509
  Public Key Infrastructure Certificate and CRL Profile <draft-ietf-
  pkix-ipki-pkalgs-05.txt> [RPKDS]

  DESCRIPTION: This document specifies algorithm identifiers and
  encoding formats for the representation of cryptographic algorithms
  keys, associated parameters, and digital signatures in Internet PKI
  and X.509 certificates and certificate revocation lists. This draft
  does not attempt to define the cryptographic algorithms themselves.
  It instead references other appropriate standards. This draft
  incorporates information from Section 7 of RFC 2459 and the
  Internet-Draft "Representation of Elliptic Curve Digital Signature
  Algorithm (ECDSA) Keys in Internet X.509 Public Infrastructure

Certificates."

STATUS: Tentatively approved by IESG.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Permanent
  Identifier <draft-ietf-pkix-pi-03.txt> [PI]

  DESCRIPTION: This document defines a new form of name, the
  permanent identifier, which is a name assigned by an organization,
  unique within that organization, that singles out a particular
  entity from all other individuals. The permanent identifier is an
  optional feature that may be used by a CA to indicate that the
  certificate relates to the same individual even if the name or the
  affiliation of that entity has changed. The permanent identifier is
  important in the context of access control and of non-repudiation.

  STATUS: Under AD review.

- DOCUMENT TITLE: Supplemental Algorithms and Identifiers for the
  Internet X.509 Public Key Infrastructure Certificate and CRL
  Profile <draft-ietf-pkix-pkalgs-supp-01.txt> [SUPPALGS]

  DESCRIPTION: This document supplements [RPKDS], defining specifies
  algorithm identifiers and encoding formats for the representation

  of emerging cryptographic algorithms and associated keys. The
  document encompasses lattice-based public key algorithms as well as
  digital signatures using larger hash algorithms (e.g., SHA-256).

  STATUS: Under WG review.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Logotypes
  in X.509 Certificate <draft-ietf-pkix-logotypes-02.txt> [LOGO]

  DESCRIPTION: This document specifies a certificate extension for
  including logotypes in public key certificates and attribute
  certificates.

  STATUS: Under WG review.

- DOCUMENT TITLE: X.509 Extensions for IP Addresses and AS
  Identifiers <draft-ietf-pkix-x509-ipaddr-as-extn-00.txt> [IPEXT]

  DESCRIPTION: This document specifies a certificate extension for
  including logotypes in public key certificates and attribute
  certificates.

  STATUS: Under WG review.

- DOCUMENT TITLE: Warranty Certificate Extension <draft-ietf-pkix-warranty-extn-00.txt> [WARR]

     DESCRIPTION: This document describes a certificate extension to explicitly state the warranty offered by a Certificate Authority (CA) for the certificate containing the extension.

     STATUS: Under WG review.

4.2 Operational Protocols

   Operational protocols are required to deliver certificates and CRLs (or other certificate status information) to certificate using systems. Provision is needed for a variety of different means of certificate and CRL delivery, including distribution procedures based on DNS, LDAP, HTTP, FTP, and X.500. A limited protocol to support AC retrieval has also been documented.

   - DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2 (RFC 2559) [PKI-LDAPv2]

     DESCRIPTION: This document describes the use of LDAPv2 as a protocol for PKI elements to publish and retrieve certificates and CRLs from a repository. [LDAPv2] is a protocol that allows publishing and retrieving of information.

     STATUS: Proposed Standard.

   - DOCUMENT TITLE: Internet X.509 Public Key Infrastructure LDAPv2 Schema (RFC 2587) [SCHEMA]

     DESCRIPTION: This document defines a minimal schema necessary to support the use of LDAPv2 for PKC and CRL retrieval and related functions for PKIX. This document supplements [LDAPv2] by identifying the PKIX-related attributes that must be present.

     STATUS: Proposed Standard.

   - DOCUMENT TITLE: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP (RFC 2560) [OCSP]

     DESCRIPTION: This document specifies a protocol useful in determining the current status of a certificate without the use of CRLs. A major complaint about certificate-based systems is the need for a relying party to retrieve a current CRL as part of the

certificate validation process. Depending on the size of the CRL,
this can cause severe problems for bandwidth-challenged devices.
Depending on the frequency of CRL issuance, this can also cause
timeliness problems. (E.g., if CRLs are only published weekly, with
no interim releases, a certificate could actually have been revoked
for just short of one week without it being on the current CRL, and
thus improper use of that certificate could still be occurring.)

OCSP attempts to address those problems. It provides a mechanism
whereby a relying party identifies one or more certificates to an
approved OCSP "responder", and the responder sends back the current
status of the certificate(s) - e.g., "revoked", "notRevoked",
"unknown". This can dramatically reduce the bandwidth required to
transmit revocation status - a relying party does not have to
retrieve a CRL of many entries to check the status of one
certificate. It can (although it is not guaranteed to) improve the
timeliness of revocation notification, and thus reduce the window
of opportunity for someone trying to use a revoked certificate.

STATUS: Proposed Standard.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
  Operational Protocols: FTP and HTTP (RFC 2585) [FTPHTTP]

  DESCRIPTION: This document describes the use of the File Transfer
  Protocol (FTP) and the Hyper-text Transfer Protocol (HTTP) to
  obtain certificates and CRLs from PKI repositories.

  STATUS: Proposed Standard.

- DOCUMENT TITLE: Diffie-Hellman Proof-of-Possession Algorithms (RFC
  2875) [DHPOP]

  DESCRIPTION: It allows Diffie-Hellman, a key agreement algorithm,
  to be used instead of requiring that the public key being requested
  for certification correspond to an algorithm that is capable of

signing and encrypting. The first algorithm generates a signature
for a specific verifier where the signer and recipient have the
same public key parameters. The second algorithm generates a
signature for arbitrary verifiers where the signer and recipient do
not have the same public key parameters.

STATUS: Proposed Standard.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Additional
  Schema for PKIs and PMIs <draft-ietf-pkix-schema-02.txt>
  [ADDSCHEMA]

DESCRIPTION: This document describes the Lightweight Directory Access Protocol (LDAP) schema features that, in addition to RFC 2587, are needed to support a Privilege Management Infrastructure and a Public Key Infrastructure. It also describes the schema for the storage and matching of attribute certificates and revocation lists in an LDAP directory server. This Internet Draft supplements, rather than revokes, the contents of RFC 2587.

STATUS: Under WG review.

- DOCUMENT TITLE: Delegated Path Validation and Delegated Path Discovery Protocol Requirements (DPV&DPD-REQ) <draft-ietf-pkix-dpd-dpv-req-04.txt> [DPREQ]

DESCRIPTION: This document specifies requirements for two request/response pairs. The first, called Delegated Path Validation (DPV), can be used to fully delegate a path validation processing to an DPV server. The second, called Delegated Path Discovery (DPD), can be used to delegate development of a path, including certificate status information, to a DPD server.

STATUS: Under WG review.

- DOCUMENT TITLE: Simple Certificate Validation Protocol (SCVP) <draft-ietf-pkix-scvp-08.txt> [SCVP]

DESCRIPTION: The SCVP protocol allows a client to offload certificate handling to a server. The server can give a variety of valuable information about the certificate, such as whether or not the certificate is valid, a chain to a trusted root, and so on.

STATUS: Under WG review.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv3 <draft-ietf-pkix-ldap-v3-05.txt> [PKI-LDAPv3]

DESCRIPTION: This document describes the features of the Lightweight Directory Access Protocol (LDAP) v3 that are needed in order to support a public key infrastructure based on x.509 certificates and certificate revocation lists. Because LDAPv2 has a

number of deficiencies that may limit its usefulness in certain circumstances, the IETF has ceased its standardization and replaced it with LDAPv3. This document describes the features of LDAPv3 that are necessary, not required, or are optional for servers to support a PKI based on X.509.

STATUS: Under WG Review.

4.3 **Management Protocols**

   Management protocols are required to support on-line interactions
   between PKI user and management entities. For example, a management
   protocol might be used between a CA and a client system with which a
   key pair is associated, or between two CAs which cross-certify each
   other. A management protocol can be used to carry user or client
   system registration information, or a request for revocation of a
   certificate.

   There are two parts to a "management protocol." The first is the
   format of the messages that will be sent, and the second is the
   actual protocol that governs the transmission of those messages.
   Originally, the PKIX working group developed two documents, [CRMF]
   and certificate management message format (CMMF), that together
   described the necessary set of message formats, and two other
   documents, [CMP] and [CMC], that described protocols for exchanging
   those messages. However, the message formats defined in the CMMF
   draft were inserted into both [CMP] and [CMC], and thus the (CMMF)
   draft has been dropped as a PKIX document.

   - DOCUMENT TITLE: Certificate Management Messages over CMS (RFC 2797)
     [CMC]

     DESCRIPTION: This document defines the means by which PKI clients
     and servers may exchange PKI messages when using S/MIME's
     Cryptographic Message Syntax [CMS] as a transaction envelope. CMC
     supports the certificate request message body specified in the
     Certificate Request Message Format [CRMF] documents, as well as a
     variety of other certificate management messages. The primary
     purpose of this specification is to allow the use of an existing
     protocol (S/MIME) as a PKI management protocol, without requiring
     the development of an entirely new protocol such as CMP. A
     secondary purpose is to codify in IETF standards the current
     industry practice of using PKCS-10 messages [PKCS10] for
     certificate requests.

     STATUS: Proposed Standard.

   - DOCUMENT TITLE: Internet X.509 Certificate Request Message Format
     (RFC 2511) [CRMF]

     DESCRIPTION: CRMF specifies a format recommended for use whenever a
     relying party is requesting a certificate from a CA or requesting

that an RA help it get a certificate. The request message format was needed before many of the other message formats had to be finalized, and so it was put into a separate document. This document only specifies the format of a message. Specification of a protocol to transport that message is beyond the scope of CRMF.

STATUS: Proposed Standard.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Certificate Management Protocols ([RFC 2510](#)) [[CMP](#)]

DESCRIPTION: This document specifies a new protocol specifically developed for the purpose of transporting messages like those specified in CRMF among PKI elements. In general, CMP will be used in conjunction with CRMF, and will then be run over a transfer service (e.g., S/MIME, HTTP) to provide a complete PKI management service.

STATUS: Proposed Standard.

- DOCUMENT TITLE: Certificate Request Message Format <[draft-ietf-pkix](#)- rfc2511bis-04.txt> [[2511bis](#)]

DESCRIPTION: This document is an update of [[CRMF](#)] and reflects the results of interoperability testing.

STATUS: Awaiting documentation of Interoperability Testing results.

- DOCUMENT TITLE: Certificate Management Protocols <[draft-ietf-pkix-rfc2510bis-06.txt](#)> [[2510bis](#)]

DESCRIPTION: This document is an update of [[CMP](#)] and reflects the results of interoperability testing. The document omits the transport protocols found in [[CMP](#)] which are addressed in [CMPT]. (See below).

STATUS: Awaiting documentation of Interoperability Testing results.

- DOCUMENT TITLE: Transport Protocols for CMP <[draft-ietf-pkix-cmp-protocols-04.txt](#)> [[TPCMP](#)]

DESCRIPTION: This document describes how to layer Certificate Management Protocols (CMP) over various transport protocols. In [Section 5 of RFC 2510](#), the process of sending DER-encoded CMP messages directly over various protocols is specified. Implementers found that the protocol was lacking in several regards. This document is an effort to enhance the protocol now in order to avoid interoperability conflicts later and to make the transport section a separate draft.

STATUS: Under WG review.

   - DOCUMENT TITLE: Certificate Management Messages over CMS <draft-
     ietf-pkix-2797-bis-01.txt> [2797bis]

     DESCRIPTION: This document is an update to [CMC].

     STATUS: Under WG review.

   - DOCUMENT TITLE: CMC Transport <draft-ietf-pkix-cmc-trans-01.txt>
     [TPCMC]

     DESCRIPTION: This document defines a number of transport mechanisms
     that are used to move [CMC] messages. The transport mechanisms
     described in the document are: HTTP, file, mail and TCP.

     STATUS: Under WG review.

   - DOCUMENT TITLE: CMC Extensions: Server Side Key Generation and Key
     Archival <draft-ietf-pkix-cmc-archive-00.txt> [SSKGKA]

     DESCRIPTION: This document defines a set of extensions to [CMC]
     that address the desire for having two additional services:
     Server generation of keys, and server-side archival and subsequent
     recovery of key material by the server. These services are
     provided by the definition of additional control statements within
     the CMC architecture.

     STATUS: Under WG review.

   - DOCUMENT TITLE: Attribute Certificate Request Message Format
     <draft-ietf-pkix-acrmf-01.txt> [ACRMF]

     DESCRIPTION: The Certificate Request Message Format ([CRMF])
     specifies a format for requesting an X.509 public key certificate
     from a Certification Authority (CA), possibly with assistance from
     an Local Registration Authority (LRA). This specification, ACRMF,
     is modeled on CRMF, extending similar functionality to requests
     for X.509 attribute certificates from Attribute Authorities (AA),
     possibly via an Attribute Registration Authority (ARA).

     STATUS: Under WG review.

   - DOCUMENT TITLE: Attribute Certificate Management Messages over CMS
     <draft-ietf-pkix-acmc-01.txt> [ACMC]

     DESCRIPTION: This document specifies modifications to the

Certificate Management Messages over CMS specification ([CMCbis])
to permit the management of attribute certificates. This document
does not stand alone, but must be used in conjunction with
[CMCbis]. It is expected that the modifications proposed here
will also be used in conjunction with the Attribute Certificate
Request Message Format specification ([ACRMF]).

STATUS: Under WG review.

4.4 Policy Outline

As mentioned before, profiling certificates and specifying
operational and management protocols only addresses a part of the
problem of actually developing and implementing a secure PKI. What is
also needed is the development of a certificate policy (CP) and
certification practice statement (CPS), and then following those
documents. The CP and CPS should address physical and personnel
security, subject identification requirements, revocation policy, and
a number of other topics. [POLPROC] provides a framework for
certification practice statements.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
  Certificate Policy and Certification Practices Framework (RFC
  2527) [POLPRAC]

  DESCRIPTION: As noted before, the specification and implementation
  of certificate profiles, operational protocols, and management
  protocols is only part of building a PKI. Equally as important - if
  not more important - is the development and enforcement of a
  certificate security policy, and a Certification Practice Statement
  (CPS). The purpose of this document (PKIX-4) is to establish a
  clear relationship between certificate policies and CPSs, and to
  present a framework to assist the writers of certificate policies
  or CPSs with their tasks. In particular, the framework identifies
  the elements that may need to be considered in formulating a
  certificate policy or a CPS. The purpose is not to define
  particular certificate policies or CPSs, per se.

  STATUS: Informational RFC.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
  Certificate Policy and Certification Practices Framework <draft-
  ietf-pkix-ipki-new-rfc2527-01.txt>

  DESCRIPTION: This specification is an update to RFC 2527. As above,
  the purpose of this document is to establish a clear relationship

between certificate policies and CPSs, and to present a framework
to assist the writers of certificate policies or CPSs with their
tasks. The framework specified in this documents is basically a
superset of the framework specified in RFC 2527.

STATUS: Under WG Review.

4.4 Time Stamping and Data Certification

In late 1998, the PKIX working group began two efforts that were not
in the original working group charter, but were deemed to be
appropriate because they described infrastructure services that could
be used to provide desired security services. The first of these is

time stamping, described in [TSP]. Time stamping is a service in
which a trusted third party - a Time Stamp Authority, or TSA - signs
a message, in order to provide evidence that it existed prior to a
given time. Time stamping provides some support for non- repudiation,
in that a user cannot claim that a transaction was later forged after
compromise of a private key, because the existence of the signed time
stamp indicates that the transaction in question could not have been
created after the indicated time.

[TSP] also defines the role of a Temporal Data Authority, or TDA. A
TDA is a Trusted Third Party (TTP) that creates a temporal data
token. This temporal data token associates a message with a
particular event and provides supplementary evidence for the time
included in the time stamp token. For example, a TDA could associate
the message with the most recent closing value of the Dow Jones
Average. The temporal data with which the message is associated
should be unpredictable in order to prevent forward dating of tokens.
The third iteration of the draft removed support for TDAs as no one
in the WG expressed a requirement for the role.

At the Minneapolis IETF meeting (IETF 44), it was disclosed that the
materials covered in [TSP] draft may be covered by patent(s). Use of
the material covered by the patent(s) in question has not be granted
by the patent holder. Thus, anyone interested in implementing the
PKIX [TSP] draft must be aware of this intellectual property issue.

The second new effort is the definition of a Data Validation and
Certification Server, or DVCS, protocol [DVCS]. A DVCS is a Trusted
Third Party that verifies the correctness of specific data submitted
to it. It also allows the delegation of trustworthy servers and
allows for chaining of verifications.

This services offered by DVCS are different from the TSP service in

that a TSA will not attempt to parse or verify a message sent to it
for certification; instead, it will merely append a reliable
indication of the current time, and sign the resulting string-of-
bits. This offers an indication that the given string-of-bits existed
at a specified time; it does not offer any indication of the
correctness or relevance of that string of bits. By contrast, the
DVCS certifies possession of data or the validity of another entity's
signature. As part of this, the DVCS verifies the mathematical
correctness of the actual signature value contained in the request
and also checks the full certification path from the signing entity
to a trusted point (e.g., the DVCS's CA, or the Root CA in a
hierarchy).

The DVCS supports non-repudiation in two ways. First, it provides
evidence that a signature or PKC was valid at the time indicated in
the token. The token can be used even after the corresponding PKC
expires and its revocation information is no longer available on CRLs
(for example). Second, the production of a data certification token
in response to a signed request for certification of another

signature or PKC also provides evidence that due diligence was
performed by the requester in validating the signature or PKC.

The concept of a delegated signature validation server was introduced
in [DSV] as an analog to the delegated path validation server. A DSV
services permits the relying party to prove they validated a
digitally signed object, including the certification path, at a
particular time.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Time Stamp
  Protocols (RFC 3161) [TSP]

  DESCRIPTION: This document defines the specification for a time
  stamp service. It defines a Time Stamp Authority, or TSA, a trusted
  third party who maintains a reliable time service. When the TSA
  receives a time stamp request, it appends the current time to the
  request and signs it into a token to certify that the original
  request existed prior to the indicated time. This helps provide
  non- repudiation by preventing someone (either a legitimate user or
  an attacker who has successfully compromised a key) from "back-
  dating" a transaction. It also makes it more difficult to challenge
  a transaction by asserting that it has been back-dated. Note that
  the TSA does not provide any data parsing service; that is, the TSA
  does not attempt to validate that which it signs; it merely regards
  it as a string of bits whose meaning is unimportant, but existence
  is vital.

STATUS: Proposed Standard.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Data
  Certification Server Protocols (RFC 3029) [DVCS]

  DESCRIPTION: This document describes a general Data Validation and
  Certification Server (DVCS) and the protocols to be used when
  communicating with it. The Data Validation and Certification
  Server is a Trusted Third Party (TTP) that can be used as one
  component in building reliable non-repudiation services.

  Useful Data Validation and Certification Server responsibilities
  in a PKI are to assert the validity of signed documents, public
  key certificates, and the possession or existence of data.

  As a result of the validation, a DVCS generates a Data Validation
  Certificate (DVC). The data validation certificate can be used
  for constructing evidence of non-repudiation relating to the
  validity and correctness of an entity's claim to possess data, the
  validity and revocation status of an entity's public key
  certificate and the validity and correctness of a digitally signed
  document.

  The presence of a data validation certificate supports non-
  repudiation by providing evidence that a digitally signed document

  or public key certificate was valid at the time indicated in the
  DVC.

  A DVC validating a public key certificate can for example be used
  even after the public key certificate expires and its revocation
  information is no longer or not easily available. Determining the
  validity of a DVC is assumed to be a simpler task, for example, if
  the population of DVCS is significantly smaller than the
  population of public key certificate owners.

  The production of a data validation certificate in response to a
  signed request for validation of a signed document or public key
  certificate also provides evidence that due diligence was
  performed by the requester in validating a digital signature or
  public key certificate.

  STATUS: Experimental RFC.

- DOCUMENT TITLE: Delegated Signature Validation Protocol
  Requirements (DSV-REQ) <draft-ietf-pkix-dsv-req-00.txt>

DESCRIPTION: This document specifies requirements to fully delegate
the validation of a digital signature to a DSV (Delegated Signature
Validation) server. The validation is performed using a set of
rules, called a signature policy.

It also defines the requirements for two optional request/response
pairs, either for allowing to indicate to a signature validation
server a signature policy, or giving the reference of a signature
policy to obtain the details of an already defined signature
policy.

STATUS: Under WG review.

- DOCUMENT TITLE: Policy Requirements for Time-Stamp Authorities
  <draft-ietf-pkix-pr-tsa-00.txt>

DESCRIPTION: This document specifies policy requirements relating
to the operation of Time-stamping Authorities (TSAs). It defines
policy requirements on the operation and management practices of
TSAs such that subscribers and relying parties may have confidence
in the operation of the time-stamping services.

The contents of this Informational RFC is technically equivalent
to ETSI TS 102 023 V1.1.1 (2002-01) [TS 102023]. The ETSI TS is
under the ETSI Copyright (C). Individual copies of this ETSI
deliverable can be downloaded from http://www.etsi.org.

STATUS: Under WG review.

4.5 Expired Drafts

There have been numerous drafts that have been produced by the
working group that for some reason or another did not make it to RFC
status. The following is a list of these drafts.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
  Certificate Management Message Formats

DESCRIPTION: This document contained the formats for a series of
messages important in certificate and PKI management. These
messages let CA's, RA's, and relying parties communicate with each
other. Note that this document only specified message formats; it
did not specify a protocol for transferring messages. That protocol

could have be either CMP or CMC, or perhaps another custom
protocol.

STATUS: Work has been discontinued. All useful information from it
has been moved into [CMP] and [CMC].

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Enhanced
  CRL Distribution Options (OpenCDP)

  DESCRIPTION: This document proposed an alternative to the CRL
  Distribution Point (CDP) approach documented in the Internet PKI
  Profile [FORMAT]. OCDP separates the CRL location function from the
  process of certificate and CRL validation, and thus claimed some
  benefits over the CDP approach.

  STATUS: Work has been discontinued, as all useful information has
  been incorporated into [X.509]. An updated the Internet PKI Profile
  [2459bis] RFC should profile the use of the CDP approach.

- DOCUMENT TITLE: Internet Public Key Infrastructure: Caching the
  Online Certificate Status Protocol

  DESCRIPTION: To improve the degree to which it can scale, OCSP
  allows caching of responses - e.g., at intermediary servers, or
  even at the relying party's end system. This document described how
  to support OCSP caching at intermediary servers.

  STATUS: Work has been discontinued.

- DOCUMENT TITLE: WEB based Certificate Access Protocol-- WebCAP/1.0

  DESCRIPTION: This document specified a set of methods, headers, and
  content-types ancillary to HTTP/1.1 to publish, retrieve X.509 PKCs
  and Certificate Revocation Lists. This protocol also facilitated
  determining current status of a digital certificate without the use
  of CRLs. This protocol defined new methods, request and response
  bodies, error codes to HTTP/1.1 protocol for securely publishing,
  retrieving, and validating certificates across a firewall.

STATUS: Expired.

- DOCUMENT TITLE: Basic Event Representation Token

  DESCRIPTION: This document defined a finite method of representing
  a discrete instant in time as a referable event. The Basic Event
  Representation Token (BERT) was a lightweight binary token designed
  for use in large numbers over short periods of time. The tokens

contained only a single instance of an event stamp and the trust
process is referenced against an external reference.

STATUS: Expired.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Extending
  trust in non repudiation tokens in time

  DESCRIPTION: This document described a method to maintain the trust
  in a token issued by a non-repudiation Trusted Third Party (NR TTP)
  (DVCS/TSA/TDA) after the key initially used to establish trust in
  the token expires. The document described a general format for
  storage of DVCS/TS/TDA tokens for this purpose, which establishes a
  chain of custody for the data.

  STATUS: Expired.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure
  Representation of Elliptic Curve Digital Signature Algorithm
  (ECDSA) Keys and Signatures in Internet X.509 Public Key
  Infrastructure Certificates

  DESCRIPTION: This document provided Object Identifiers (OIDs) and
  other guidance for IPKI users who use the Elliptic Curve Digital
  Signature Algorithm (ECDSA). It profiled the format and semantics
  of the subjectPublicKeyInfo field and the keyUsage extension in
  X.509 v3 PKCs containing ECDSA keys. This document should have been
  used by anyone wishing to support ECDSA; others who do not support
  ECDSA are not required to comply with it.

  STATUS: Finished WG Last Call. Merged into Representation of Public
  Keys and Digital Signatures in Internet X.509 Public Key
  Infrastructure Certificates.

- DOCUMENT TITLE: A String Representation of General Name

  DESCRIPTION: This document specified a string format for the ASN.1
  construct GeneralName.

  STATUS: Expired.

- DOCUMENT TITLE: OCSP Extensions

  DESCRIPTION: This document defined Internet-standard extensions to
  OCSP that enable a client to delegate processing of certificate

acceptance functions to a trusted server. The client could control
the degree to which delegation takes place. In addition limited

support was provided for delegating authorization decisions.

STATUS: The work has been incorporated into [DPV] and [DPD].

- DOCUMENT TITLE: Using HTTP as a Transport Protocol for CMP

  DESCRIPTION: This document described how to layer [CMP] over
  [HTTP]. A simple method for doing so was described in [CMP], but
  that method does not accommodate a polling mechanism, which may be
  required in some environments. This document specified an
  alternative method that used the polling protocol defined in [CMP].
  A new Content-Type for messages was also defined.

  STATUS: The work has been merged into [TPCMP].

- DOCUMENT TITLE: Using TCP as a Transport Protocol for CMP

  DESCRIPTION: This document described how to layer Certificate
  Management Protocols [CMP] over [TCP]. A method for doing so is
  described in [CMP], but that method did not solve problems
  encountered by implementors. This document specified an enhanced
  method which extends the protocol.

  STATUS: The work has been merged into [TPCMP].

- DOCUMENT TITLE: Delegated Path Validation

  DESCRIPTION: This specification builds on the Online Certificate
  Status Protocol (OCSP) framework's extensibility by defining an
  Internet-standard extension to OCSP that can be used to fully
  delegate all path validation processing to an OCSP server. The
  Delegated Path Validation (DVP) extension to OCSP described in this
  document accomplishes the task of locating the certificate
  validation process within a trusted server. This in turn reduces
  the technical footprint of certificate-using applications and may
  ease the integration of certificate path processing with other
  authorized data.

  STATUS: Expired.

- DOCUMENT TITLE: Delegated Path Discovery with OCSP

  DESCRIPTION: This document establishes an Internet-standard
  extension that enables relying-party software to acquire
  certification path data from an OCSP server rather than replicate
  the same functionality. This Delegated Path Discovery (DPD)
  extension delegates the acquisition process to a separate server,
  thereby greatly simplifying and reducing the size of public key
  based credential validation systems or other relying party
  software. The DPD extension also enables such software to select

from among various trust paths in the event of the existence of
multiple paths.

STATUS: Expired.

- DOCUMENT TITLE: Online Certificate Status Protocol, Version 2

DESCRIPTION: This document is an update to RFC 2560.

STATUS: Expired.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Repository
Locator Service

DESCRIPTION: This document defines a PKI repository locator
service, which enable certificate-using systems to locate PKI
repositories based on a domain name, to identify the protocols that
can be used to access the repository, and obtain addresses for the
servers that host the repository service. The Internet Draft
defines SRV records for a PKI repository locator service to enable
PKI clients to obtain necessary information to connect to a
domain's repository. It also includes the definition of a SRV RR
format for this service.

STATUS: Expired.

- DOCUMENT TITLE: Internet X.509 Public Key Infrastructure Technical
Requirements for a non-Repudiation Service

DESCRIPTION: This document describes those features of a service
which processes signed documents which must be present in order for
that service to constitute a "technical non-repudiation" service. A
technical non-repudiation service must permit an independent
verifier to determine whether a given signature was applied to a
given data object by the private key associated with a given valid
certificate, at a time later than the signature. The features of a
technical non-repudiation service are expected to be necessary for
a full non-repudiation service, although they may not be
sufficient.

This document is intended to clarify the definition of the "non-
repudiation" service in RFC 2459. It should thus serve as a guide
to when the nonRepudiation bit of the keyUsage extension should be
set and to when a Certificate Authority is required to archive
CRL's.

STATUS: Expired.

- DOCUMENT TITLE: Limited Attribute Certificate Acquisition Protocol
  <draft-ietf-pkix-laap-01.txt>

  DESCRIPTION: This document specifies a deliberately limited
  protocol for requesting ACs from a server. It is intended to be

  complementary to the use of LDAP for AC retrieval, covering those
  cases where use of an LDAP server is not suitable due to the type
  of authorization model being employed.

  STATUS: Expired.

5 Implementation Advice

This section provides guidance to those who would implement various
parts of the PKIX suite of documents. The topics discussed in this
section engendered significant discussion in the working group, and
there, was at times, either widespread disagreement or widespread
misunderstanding of them. Thus, this discussion is provided to help
readers of the PKIX document set understand these issues, in the hope
of fostering greater interoperability among eventual PKIX
implementations.

5.1 Names

PKIX has been referred to as a "name-centric" PKI because the PKCs
associate public keys with names of entities. Each PKC contains at
least one name for the owner of a particular public key. The name can
be an X.500 distinguished name, contained in the subjectDN field of
the PKC. There can also be names such as RFC822 e-mail addresses, DNS
domain names, and uniform resource identifiers (URIs) associated with
the key; these attributes are kept in the subjectAltName extension of
the PKC. A PKC must contain at least one of these name forms, it may
contain multiple forms if deemed appropriate by the CA based on the
intended usage of the PKC.

5.1.1 Name Forms

There are two possible places to put a name in an X.509 v3 PKC. One
is the subject field in the base PKC (often called the "Distinguished
Name" or "DN" field), and the other is in the subjectAltName
extension.

5.1.1.1 Distinguished Names

According to the Internet PKI Profile [2459bis], a CA's PKC must have
a non-null value in the subject field, while EE's PKCs are permitted
to have an empty subject field. If a PKC has a non-null subject
field, it must contain an X.500 Distinguished Name.

5.1.1.2 SubjectAltName Forms

In addition to the DN, a PKIX PKC may have one or more values in the
subjectAltName extension.

The subjectAltName extension allows additional identities to be bound
to the subject of the PKC (e.g., it allows "umbc.edu" and
"130.85.1.3" to be associated with a particular subject, as well as
"C=US, O=University of Maryland, L=Baltimore, CN=UMBC"). X.509-
defined options for this extension include: Internet electronic mail
addresses; DNS names; IP addresses; and URIs. Other options can
exist, including locally-defined name forms.

A single subjectAltName extension can include multiple name forms,
and multiple instances of each name form.

Whenever such alternate name forms are to be bound into a PKC, the
subjectAltName (or issuerAltName) extension must be used. It is
technically possible to embed an alternate name form in the subject
field. For example, one could make a DN contain an IP address via a
kludge such as "C=US, L=Baltimore, CN=130.85.1.3". However, this
usage is deprecated; the alternative name extension is the preferred
location for finding such information. As another example, some
legacy implementations exist where an RFC822 name is embedded in the
subject distinguished name as an EmailAddress attribute. Per Internet
Profile [2459bis], PKIX-compliant implementations generating new PKCs
with electronic mail addresses must use the rfc822Name in the
subjectAltName extension to describe such EEs. Simultaneous inclusion
of the EmailAddress attribute in the subject distinguished name to
support legacy implementation is deprecated but permitted.

In line with this, if the only subject identity included in a PKC is
an alternative name form, then the subject distinguished name must be
empty (technically, an empty sequence), and the subjectAltName
extension must be present. If the subject field contains an empty
sequence, the subjectAltName extension must be marked critical.

If the subjectAltName extension is present, the sequence must contain
at least one entry. Unlike the subject field, conforming CAs shall
not issue PKCs with subjectAltNames containing empty GeneralName
fields. For example, an rfc822Name is represented as an IA5String.
While an empty string is a valid IA5String, such an rfc822Name is not
permitted by PKIX. The behavior of clients that encounter such a PKC
when processing a certification path is not defined by this working
group. Because the subject's alternative name is considered to be
definitively bound to the public key, all parts of the subject's
alternative name must be verified by the CA.

5.1.1.2.1 Internet e-mail addresses

When the subjectAltName extension contains an Internet mail address,
the address is included as an rfc822Name. The format of an rfc822Name

is an "addr-spec" as defined in [RFC-822]. An addr-spec has the form
local-part@domain; it does not have a phrase (such as a common name)
before it, or a comment (text surrounded in parentheses) after it,
and it is not surrounded by "<" and ">".

5.1.1.2.2 DNS Names

When the subjectAltName extension contains a domain name service
label, the domain name is stored in the dNSName attribute(an
IA5String). The string shall be in the "preferred name syntax," as
specified by [DNS]. Note that while upper and lower case letters are
allowed in domain names, no significance is attached to the case. In
addition, while the string " " is a legal domain name, subjectAltName
extensions with a dNSName " " are not permitted. Finally, the use of
the DNS representation for Internet mail addresses (wpolk.nist.gov
instead of wpolk@nist.gov) is not permitted; such identities are to
be encoded as rfc822Name.

5.1.1.2.3 IP addresses

When the subjectAltName extension contains an iPAddress, the address
shall be stored in the octet string in "network byte order," as
specified in [IP]. The least significant bit (LSB) of each octet is
the LSB of the corresponding byte in the network address. For IP
Version 4, as specified in [IP], the octet string must contain
exactly four octets. For IP Version 6, as specified in [IPv6], the
octet string must contain exactly sixteen octets.

URIs

   The Internet PKI Profile [2459bis] notes "When the subjectAltName
   extension contains a URI, the name must be stored in the
   uniformResourceIdentifier (an IA5String). The name must be a non-
   relative URL, and must follow the URL syntax and encoding rules
   specified in [RFC 1738]. The name must include both a scheme (e.g.,
   "http" or "ftp") and a scheme-specific- part. The scheme-specific-
   part must include a fully qualified domain name or IP address as the
   host. As specified in [RFC 1738], the scheme name is not case-
   sensitive (e.g., "http" is equivalent to "HTTP"). The host part is
   also not case-sensitive, but other components of the scheme-specific-
   part may be case-sensitive. When comparing URIs, conforming
   implementations must compare the scheme and host without regard to
   case, but assume the remainder of the scheme-specific-part is case
   sensitive."

5.1.2 **Scope of Names**

   The original X.500 work presumed that every subject in the world
   would have a globally unique distinguished name. Thus, every subject

Arsenault, Turner   38

Internet-Draft   PKIX Roadmap   July 2002

   could be easily located, and there would never be a conflict. All
   that would be needed is a sufficiently large name space, and rules
   for constructing names based on subordination and location.

   Obviously, that is not practical in the real world, for a variety of
   reasons. There is no single entity in the world trusted by everybody
   to reside at the top of the name space, and there is no way to
   enforce uniqueness on names for all entities. (These were among the
   reasons for the failure of PEM to be widely implemented.)

   This does not mean, however, that a name-based PKI cannot work. It is
   important to recognize that the scope of names in PKIX is local; they
   need to be defined and unique only within their own domain.

   Suppose for example that a Top CA is established with DN "O=IETF,
   OU=PKIX, CN=PKIX_CA". That CA will then issue PKCs for subjects
   subordinate to it. The only requirement, which can be enforced
   procedurally, is that no two distinct entities beneath this Top CA
   have the same name. We can't prevent somebody else in the world from
   creating her own CA, called "O=IETF, OU=PKIX, CN=PKIX_CA", and it is
   not necessary to do so. Within the domain of the original Top CA,
   there will be no conflict, and the fact that there is another CA of
   the same name in some other domain is irrelevant.

This is analogous to the current DNS or IP address situations. On the Internet, there is only one node called www.ietf.org. The fact that there might be 10 different intranets, each with a host given the DNS name www.ietf.org, is irrelevant and causes no interoperability problems - those are different domains. However, if there were to be another node on the Internet with domain name www.ietf.org, then there would be a problem due to name duplication.

The same applies for IP addresses. As long as only one node on the Internet responds to the IP address 130.85.1.3, there is no problem, despite the fact that there are 100 different corporate Intranets, each using that same IP address. However, if two different nodes on the Internet each responded to the IP address 130.85.1.3, there would be a problem.

5.1.3 Certificate Path Construction

Certificate path construction has been the topic of many discussions in the WG. The issue centered on how best to get a certificate when the connection between the subject's name and the name of their CA, as well as the CA's repository (or directory), may not be obvious. Many proposals were put forth, including implementing a global X.500 Directory Service, using DNS SRV records, and using an extension to point to the directory provider. At the end of the discussion the group decided to use the authority information access extension defined in the Internet PKI Profile [2459bis], which allows the CA to indicate the access method and location of CA information and services. The extension would be included in subject's certificates

and could be used to associate an Internet style identity for the location of repository to retrieve the issuer's certificate in cases where such a location is not related to the issuer's name.

Another discussion related to certificate path construction was where to store the CA and EE PKCs in the directory (specifically LDAPv2 directories). Two camps emerged with different views on where to store CA and cross-certificates. In the CA's directory entry, one camp wanted self-issued PKCs stored in the cACertificate attribute, PKCs issued to this CA stored in the forward element of the crossCertificatePair, and PKCs issued from this CA for other CAs in the reverse element of the crossCertificatePair attribute. The other camp wanted all CA PKCs stored in the cACertificate attribute, and PKCs issued to and from another domain stored in the crossCertificatePair attribute. There was a short discussion that the second was more efficient than first and that one solution or the other was more widely deployed. The end result was to indicate that

self-issued PKCs and PKCs issued to the CA by CAs in the same domain
as the CA are stored in the cACertificate attribute. The
crossCertificatePair attribute's forward element will include all but
self-issued PKCs issued to the CA. The reverse element may include a
subset of PKCs issued by the CA to other CAs. With this resolution
both camp's implementations are supported and are free to choose the
location of CA PKCs to best support their implementation.

5.1.4 Name Constraints

A question that has arisen a number of times is "how does one enforce
Name constraints when there is more than one name form in a PKC?"
That is, the Internet PKI Profile [2459bis] states that:

Subject's alternative names may be constrained in the same manner as
subject distinguished names using the name constraints extension as
described in section 4.2.1.11.

What does this mean? Suppose that there is a CA with DN "O=IETF,
OU=PKIX, CN=PKIX_CA", with the subjectAltName extension having
dNSName "PKIX_CA.IETF.ORG". Suppose that that CA has issued a PKC
with an empty DN, with subjectAltName extension having dNSName set to
"PKIX_CA.IETF.ORG", and rfc822Name set to Steve@PKIX_CA.IETF.ORG. The
question is: are name constraints enforced on these two PKCs - is the
name of the EE PKC considered to be properly subordinate to the name
of the CA?

The answer is "yes". The general rules for deciding whether a PKC
meets name constraints are:

- If a PKC complies with name constraints in any one of its name
  forms, then the PKC is deemed to comply with name constraints.

- If a PKC contains a name form that its issuer does not, the PKC is
  deemed to comply with name constraints for that name form.

In deciding whether a name form meets name constraints, the following
rules apply (in all cases Name B is the name in the name constraints
extension):

5.1.4.1 rfc822Names

Three variations are allowed:

- If the name constraint is specified as "larry@mail.mit.edu", then

Name A is subordinate to Name B (in B's subtree) if Name A
contains all of Name B's name (specifies a particular mailbox).
For example, larry@mail.mit.edu is subordinate, but
larry_sanders@mail.mit.edu is not.

- If the name constraint is specified as "mail.mit.edu", then Name A
  is subordinate to Name B (in B's subtree) if Name A contains all
  of Name B's name (specified all mailboxes on host mail.mit.edu).
  For example, curly@mail.mit.edu and mo@mail.mit.edu are
  subordinate, but mo@mail6.mit.edu and curly@smtp.mail.mit.edu are
  not.

- If the name constraint is specified as ".mit.edu", then Name A is
  subordinate to Name B (in B's subtree) if Name A contains all of
  Name B's name, with the addition of zero or more additional host
  or domain names to the left of Name B's name. That is,
  smtp.mit.edu is subordinate to .mit.edu, as is pop.mit.edu.
  However, mit.edu is not subordinate to .mit.edu. When the
  constraint begins with a "." it specifies any address within a
  domain. In the previous example, "mit.edu" is not in the domain of
  ".mit.edu".

Note: If rfc822 names are constrained, but the PKC does not contain a
subjectAltName extension, the EmailAddress attribute will be used to
constrain the name in the subject distinguished name. For example (c
is country, o is organization, ou is organizational unit, and em is
EmailAddress), Name A ("c=US, o=MIT, ou=CS, em=curly@mail.mit.edu")
is subordinate to Name B ("c=US, o=MIT, ou=CS") (in B's subtree) if
Name A contains all of Name B's names.

5.1.4.2 dNSNames

Name A is subordinate to Name B (in B's subtree) if Name A contains
all of Name B's name, with the addition of zero or more additional
domain names to the left of Name B's name. That is, www.mit.edu is
subordinate to mit.edu, as is larry.cs.mit.edu. However, www.mit.edu
is not subordinate to web.mit.edu.

5.1.4.3 x.400 addresses

Name A is subordinate to Name B (in B's subtree) if Name A contains
all of Name B's name. For example (c is country-name, admd is
administrative-domain-name, and o is organization-name, ou is

organizational-unit-name, pn is personal-name, sn=surname, and gn is
given-name in both Name A and B), the mnemonic X.400 address (using
PrintableString choices for c and admd) "c=US, admd=AT&T, o=MIT,
ou=cs, pn='sn=Doe,gn=John'" is subordinate to "c=US, admd=AT&T,
o=MIT, ou=cs" and "c=US, admd=AT&T, o=MIT, pn='sn=DOE,gn=JOHN'" (pn
is a SET that includes, among other things, sn and gn).

5.1.4.5 DNs

Name A is subordinate to Name B (in B's subtree), if Name A contains
all of Name B's name, treating attribute values encoded in different
types as different strings, ignoring case in PrintableString (in all
other types case is not ignored), removing leading and trailing white
space in PrintableString, and converting internal substrings of one
or more consecutive white space characters to a single space. For
example, (c is country, o is organization, ou is organizational unit,
and cn is common name):

- Assuming PrintableString types for all attribute values in Name A
  and B, "c=US, o=MIT, ou=CS" is subordinate to "c=us, o=MIT,
  ou=cs", as is "c=US, o=MIT, ou=CS, ou=administration". Another
  example, "c=US, o=MIT, ou=CS, cn= John Doe" (note the white
  spaces) is subordinate to "c=US, o=MIT, ou=CS, cn=John Doe".

- Assuming UTF8String types for all attribute values in Name A and B,
  "c=US, o=MIT, ou=CS, ou=administration" is subordinate to "c=US,
  o=MIT, ou=CS", but "c=US, o=MIT, ou=cs, ou=Administration". "c=US,
  o=MIT, ou=CS, cn= John Smith" (note the white spaces) is not
  subordinate to "c=US, o=MIT, ou=CS, cn= John Smith".

- Assuming UTF8String types for all attribute values in Name A and
  PrintableString types for all attribute values in Name B, "c=US,
  o=MIT, ou=cs" is subordinate to "c=US, o=MIT, ou=CS" if the
  verification software supports the full comparison algorithm in
  the X.500 series. "c=US, o=MIT, ou=cs" is NOT subordinate to
  "c=US, o=MIT, ou=CS" if the verification software supports the
  comparison algorithm in the Internet PKI Profile [2459bis].

5.1.4.6 URIs

The constraints apply only to the host part of the name. Two
variations are allowed:

- If the name constraint is specified as ".mit.edu", then Name A is
  subordinate to Name B (in B's subtree) if Name A contains all of
  Name B's name, with the addition of zero or more additional host

or domain names to the left of Name B's name. That is, www.mit.edu
is subordinate to .mit.edu, as is curly.cs.mit.edu. However,
mit.edu is not subordinate to .mit.edu. When the constraint begins
with a "." it specifies a host.

- If the name constraint is specified as "abc.mit.edu", then Name A
  is subordinate to Name B (in B's subtree) if Name A contains all
  of Name B's name. No leftward expansion of the host or domain name
  is allowed.

5.1.4.7 iPaddresses

Two variations are allowed depending on the IP version:

- For IPv4 addresses: Name A (128.32.1.1 encoded as 80 20 01 01) is
  subordinate to Name B (128.32.1.0/255.255.255.0 encoded as 80 20
  00 00 FF FF FF 00) (in B's subtree) if Name A falls within the net
  denoted in Name B's CIDR notation.

- For IPv6 addresses: Name A (4224.0.0.0.8.2048.8204.16762 encoded as
  10 80 00 00 00 00 00 00 00 08 08 00 20 0C 41 7A) is subordinate to
  Name B (4224.0.0.0.8.2048.8204.0/
  65535.65535.65535.65535.65535.65535.65535.0 encoded as 10 80 00 00
  00 00 00 00 00 08 08 00 20 0C 00 00 FF FF FF FF FF FF FF FF FF FF
  FF FF FF FF 00 00) (in B's subtree) if Name A falls within the net
  denoted in Name B's CIDR notation.

5.1.4.8 Others

As the Internet PKI Profile [2459bis] does not define requirements
for the name forms otherName, ediPartyName, or registeredID there are
no corresponding name constraints requirements.

5.1.5 Wildcards in Name Forms

A "wildcard" in a name form is a placeholder for a set of names
(e.g., "*.mit.edu" meaning "any domain name ending in .mit.edu", and
*@aol.com meaning "email address that uses aol.com"). There are many
people who believe that allowing wildcards in name forms in PKIX PKCs
would be a useful thing to do, because it would allow a single PKC to
be used by all members of a group. These members would presumably
have attributes in common; e.g., access rights to some set of
resources, and so issuance of a PKC with a wildcard for the group
would simplify management.

After much discussion, the PKIX working group decided to permit the
use of wildcards in PKCs. That is, it is permissible for a PKIX-
conformant CA to issue a PKC with a wildcard. However, the semantics

of subjectAltName extension that include wildcard characters are not

addressed by PKIX. Applications with specific requirements may use
such names but must define the semantics.

5.1.6 Name Encoding

A very important topic that consumed much of the WG's time was the
implementation of the directory string choices. While the long term
goal of the IETF was clear, use UTF8String, the short term goals were
not so clear. Many implementations only use PrintableString, others
use BMPString, and still others use Latin1String (ISO 8859-1) and tag
it as TeletexString (there are others still). To ensure that there is
consistency with encodings the Internet PKI Profile [2459bis] defines
a set of rules for the string choices. PrintableString was kept as
the first choice because of it's widespread support by vendors.
BMPString was the second choice, also for it's widespread vendor
support. Failing support by PrintableString and BMPString, UTF8String
must be used. In keeping with the IETF mandate, UTF8String can be
used at any time if the CA supports it. Also, processing
implementations that wish to support TeletexString should handle the
entire ISO 8859-1 character set and not just the Latin1String subset.

5.2 POP

Proof of Possession, or POP, or also CA POP, means that the CA is
adequately convinced that the entity requesting a PKC containing a
public key Y has access to the private key X corresponding to that
public key.

There has been some debate whether POP was or not needed.

This question needs to be addressed separately considering the
context of use of the key, in particular whether a key is used for an
authentication or non repudiation service, or in a confidentiality
service. Note that this does not map to the key usage bit directly,
since it is possible to use a confidentiality key to perform an
authentication service, e.g. by asking to decrypt an encrypted
challenge.

5.2.1 POP for Signing Keys

It is important to provide POP for keys used to sign material, in
order to provide non-repudiation of transactions. For example,

suppose Alice legitimately has private key X and its corresponding
public key Y. Alice has a PKC from Charlie, a CA, containing Y. Alice
uses X to sign a transaction T. Without POP, Mal could also get a PKC
from Charlie containing the same public key, Y. Now without POP,
there are two possible threats: Mal could claim to have been the real
signer of T; or Alice can falsely deny signing T, claiming that it
was instead Mal. Since no one can reliably prove that Mal did or did
not ever possess X, neither of these claims can be refuted, and thus

the service provided by and the confidence in the PKI has been
defeated. (Of course, if Mal really did possess X, Alice's private
key, then no POP mechanism in the world will help, but that is a
different problem.)

Protection can be gained by having Alice, as the true signer of the
transaction, include in the signed information her PKC or an
identifier of her PKC (e.g., a hash of her PKC). This makes
impossible for Mal to claim authorship because he does not know the
private key from Alice and thus is unable to include his certificate
under the signature.

The adequate protection may be obtained in the general case, by
mandating the inclusion of a reference of the certificate every time
a signature (or non repudiation) key is being used in a protocol.

However, because all the protocols were not doing so, a conservative
measure has been taken by requesting POP to be performed by CAs. It
should thus be understood, that this measure is not strictly
necessary and is only a temporary measure to make old protocols
secure. New protocols or data formats are being developed. If the key
being used is always used in a context where the identifier of the
certificate is included in the protocol, then POP by the CA is not
necessary. The inclusion of the identifier of the certificate in the
protocol or data format may be understood as POP being performed at
the transaction time rather than by the CA, at the registration time
of the user in the PKI.

5.2.2 POP for Key Management Keys

Suppose that Al is a new instructor in the Computer Science
Department of a local University. Al has created a draft final exam
for the Introduction to Networking course he is teaching. He wants to
send a copy of the draft final to Dorothy, the Department Head, for
her review prior to giving the exam. This exam will of course be
encrypted, as several students have access to the computer system.
However, a quick search of the PKC repository (e.g., search the

repository for all records with subjectPublicKey=Dorothy's-value)
turns up the fact that several students have PKCs containing the same
public key management key as Dorothy. At this point, if no POP has
been done by the CA, Al has no way of knowing whether all of the
students have simply created these PKCs without knowing the
corresponding private key (and thus it is safe to send the encrypted
exam to Dorothy), or whether the students have somehow acquired
Dorothy's private key (and thus it is certainly not safe to send the
exam).

The later case may seem unsafe. However, if the other students have
acquired the key, they do not even need to publish their certificates
and can simply decrypt in parallel.

The end story is that, if the key only known to Dorothy, there is no
problem. Thus POP by the CA is not needed.

If the key, like a Diffie-Hellman key, is used for an authentication
service the answer depends from the protocol being used. In the
former example, the decryption was done locally and no data was sent
back on the wire. In an authentication protocol, the story is
different because either some encrypted or decrypted data is sent
back. If the data sent back contains the identifier of the
certificate in a way that it cannot be modified without that
modification being detected, then there is no need for POP. On the
contrary, POP by the CA is needed.

As a conservative measure, POP for encryption keys is recommended,
but it should be realized that it is not always needed.

In general it should be noticed that POP at the time of the
transaction is much superior than POP made by the CA, since it is
possible in real time to be sure that everything is correct, rather
than rely on that verification to be done at the time of registration
by the CA. Should the CA fail in that verification, then there is a
security breach. On the contrary, doing POP at the time of the
transaction, eliminates that problem.

CMP requires that POP be provided for all keys, either through on-
line or out-of-band means. There are any number of ways to provide
POP, and the choice of which to use is a local matter. Additionally,
a PKC requester can provide POP to either a CA or to an RA, if the RA
can adequately assure the CA that POP has been performed. Some of the
acceptable ways to provide POP include:

- Out-of-band means:

- For keys generated by the CA or an RA (e.g., on a token such as
  a smart card or PCMCIA card), possession of the token can
  provide adequate proof of possession of the private key.

- For user-generated keys, another approach can be used in
  environments where "key recovery" requirements force the
  requester to provide a copy of the private key to the CA or an
  RA. In this case, the CA will not issue the requested PKC until
  such time as the requester has provided the private key. This
  approach is in general not recommended, as it is extremely risky
  (e.g., the system designers need to figure out a way to protect
  the private keys from compromise while they are being sent to
  the CA/RA/other authority, and how to protect them there), but
  it can be used.

- On-line means:

  - For signature keys that are generated by an EE, the requester of
    a PKC can be required to sign some piece of data (typically, the
    PKC request itself) using the private key. The CA will then use

  the requested public key to verify the signature. If the
  signature verification works, the CA can safely conclude that
  the requester had access to the private key. If the signature
  verification process fails, the CA can conclude that the
  requester did not have access to the correct private key, and
  reject the request.

  - For key management keys that are generated by the requester, the
    CA can send the user the requested public key, along with the
    CA's own public key, to encrypt some piece of data, and send it
    to the requester to be decrypted. For example, the CA can
    generate some random challenge, and require some action to be
    taken after decryption (e.g., "decrypt this encrypted random
    number and send it back to me"). If the requester does not take
    the requested action, the CA concludes that the requester did
    not possess the private key, and the PKC is not issued.

Another method of providing POP for key management keys is for the CA
to generate the requested PKC, and then send it to the requester in
encrypted form. If the requester does not have access to the
appropriate private key, the requester cannot decrypt the PKC, and
thus cannot use it. After some period of time in which the PKC is not
used, the CA will revoke the PKC. (This only works if the PKC is not
made available to any untrusted entities until after the requester
has successfully decrypted it.)

   The key usage extension defines the purpose (e.g., encipherment,
   signature, certificate signing) of the key contained in the PKC. This
   extension is used when a key that could be used for more than one
   operation is to be restricted. For example, if a CA's RSA key should
   be used only for signing CRLS, the cRLSign bit would be asserted.
   Likewise, when an RSA key should be used only for key management, the
   keyEncipherment bit would be asserted. When used, this extension
   should be marked critical.

   The Internet PKI Profile [2459bis] includes some text for how the
   bits in the KeyUsage type are used. Developing the text for some of
   the bits was easy; however, many discussions were needed to arrive at
   a common agreement on the meaning of the digitalSignature (DS bit)
   and nonRepudiation (NR bit) bits and when they should be set. The
   group quickly realized that key usage extension mixes services and
   mechanisms. The DS bit indicates a mechanism - a public key used to
   verify a digital signature. The NR bit indicates a service - a public
   key used to verify a digital signature and to provide a non-
   repudiation service. When trying to indicate when each bit should be
   indicated arguments were based on:

   The lifetime of the object being signed. Some felt that the DS bit
   should be set when the certificate is used to sign ephemeral objects
   (e.g., bind tokens) while the NR bit should be set for things that

Arsenault, Turner   47

   are survive longer (e.g., documents). Of course, the problem with
   this distinction to determine how long is the time period for
   ephemeral?

   A conscious act taken by the signer. Many felt that the NR bit should
   be set only when the subject has expressly acknowledged that they
   want to use the private key to sign an object. Signing a document say
   where there is a conscious decision by the subject would be
   appropriate for the key usage extension to contain NR, but when the
   key is used for authentication purposes, which can occur
   automatically and more frequently, the DS bit is more appropriate.
   The discussion also concluded that since some authentication schemes
   occur automatically, that the DS bit and NR bit should never be set
   together in the same certificate. Some agreed to the differentiation
   of the bits based on the time, but did not agree that the two could
   not be in the same key usage extension.

   The procedures followed by the CA. Some felt that NR bit was kind of
   'quality mark' indicating to the verifier that the verifier could be

assured that the CA is implementing appropriate procedures for
checking the subject's identity, performing certificate archival,
etc. Other felt that it was not entirely the CAs job and that some
other entity must be involved.

In the end the WG agreed to a few things:

- Provision of the service of non-repudiation requires more than a
  single bit set in a PKC. It requires an entire infrastructure of
  components to preserve for some period of time the keys, PKCs,
  revocation status, signed material, etc., as well as a trusted
  source of time. However, the nonRepudiation key usage bit is
  provided as an indicator that such keys could be used as a
  component of a system providing a non-repudiation service.

- The certificate policy is the appropriate place to indicate the
  permitted combinations of key usages. That is, a policy may
  indicate that the DS and NR bits can not be set in the same
  certificate while another may say that the DS and NR bits can be
  set in the same certificate.

[2459bis] includes new text indicating the above agreements.

5.4 Non-Repudiation

The major benefit of the whole DS bit vs NR bit discussion is
development of the Technical Requirements for Non-Repudiation
[TECHNR] draft. To fill this void [TECHNR] was developed to "describe
those features of a service which processes signed documents which
must be present in order for that service to constitute a 'technical
non-repudiation' service." The basic understanding of non-repudiation
is that it requires that a digital signature be preserved in such a
manner that it can convince a neutral third party that it was

actually created by someone with access to the private key of a
certified key pair. Whether this definition of non-repudiation is
enough to form a legally bind agreement is still being debated.

5.5 Trust Models

An important design decision is, for a given application, where the
particular EE's trust points are located (i.e. what are the Top
CAs). There are a number of models that have been developed and
depending on the environment some models may be more suited than
others. The following provides some background on the models.

5.5.1 Hierarchical

One of the initial trust models proposed was the hierarchical model.
In this model the trust point or root CA for an entire domain is the
top most CA. The root CA in turn issues certificates to subordinate
CAs, and the subordinate CAs issue certificates to EEs. When
verifying a PKC, the RP must verify ever certificate in the path from
the EE's PKC to the root CA.

The main benefit of the hierarchical model is the fact that controls
imposed from the top down. For example, name constraints can be
included in the subordinate CAs to limit the name space in which they
are allowed to issue certificates. Further, the root CA ensure domain
wide policies on cross-certification (though there are no controls to
prevent another PKI from issuing PKCs to members of the domain, but
then those members could be thought of as members of two distinct
PKIs).

Interoperability is achieved through the use of cross-certificates.
Cross-certificates can be issued by the root CA or if allowed by
subordinate CAs.

5.5.2 Local/Federation

Another model that has been around a long time is the local trust
model. In this model, the RPs trust the CA that issued their
certificate to them. The idea is that since the CA is local and
probably known to the RP, that there is more trust rather than with
some distant unknown CA.

In order for EEs under different CAs to communicate the CAs issue
each other certificates thereby creating a certification path from
one EE to another. The process of the CAs issuing one another PKCs
forms a kind of federation

The main benefit of the local model is its flexibility. Many believe
that the local CA knows best how to support its user community and

should be given cart blanche to generate certificates as it sees fit
to allow the user community to perform their functions.

5.5.3 Root Repository

A model made famous in the web browser community is the root

repository. This model uses a file to store the PKCs of many CAs. The RP then trusts any PKC included in the file. The PKC included in the root repository may be a root CA for some other domain or subordinate CA, but when included in the trust file whatever type of PKC it is in the other domain, it becomes a root CA for the RP. Obviously, the main advantage is the fact that cross-certification is not required. If the RP does not have the root CA's certificate and it is included in with the object, the RP can just add it to the file to "trust" it (this should only be done if the RP truly trusts the root CA).

5.5.4 RP's Perspective

Another model recently getting attention is the model where instead of the CA imposing restraints on the RP (in the PKC), the RP instead makes the determination as to which certificates to trust. The RP determines which domain it will accept certificates from, which key usages it will accept, etc. Cross-certification is also not required because the RP can just chose to trust a particular PKC or domain of PKCs. This obviously turns the first three models on their heads. Special care must be taken to ensure that the RP is properly configured.

5.5.5 Validation Policies

Another model considers a set of rules that apply to an application context. Every application context may have a different set of rules. When choosing to use certificates in the context of that application, the EE selects the set of rules for that context. In a set of rules, one or more Top CAs may be trusted, each one may be associated with different constraints, like the certificate policies that are trusted or the naming constraints that apply. These constrains may be specified either in self-signed certificates or in addition to self-signed certificates when they do not incorporate these constraints. This set of rules is called a validation policy (when validating a certificate) or a signature policy (when validating a digital signature).

6 References

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2459bis] Housley, R., Ford, W., Polk, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," RFC 3280, April 2002.

[2510bis] Adams, C., Farrell, S., "Internet X.509 Public Key Infrastructure Certificate Management Protocols," <draft-ietf-pkix-rfc2510bis-06.txt>, December 2001.

[2511bis] Myers, M., Adams, C., Solo, D., and Kemp D. "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)," <draft-ietf-pkix-rfc2511bis-04.txt>, December 2001.

[2527bis] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and Wu, S., "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," <draft-ietf-pkix-ipki-new-rfc2527-01.txt>, January 2002.

[2797bis] Myers, M., Liu, X., Fox, B., and Weinstein, J., "Certificate Management Messages over CMS," <draft-ietf-pkix-2797-bis-01.txt>, February 2002.

[AC] Farrell, S., and Housley, R., "An Internet Attribute Certificate Profile for Authorization," RFC 3281, April 2002.

[ACRMF] Yee, P., "Attribute Certificate Request Message Format," <draft-ietf-pkix-acrmf-01.txt>, March 2002.

[ACMC] Yee, P., "Attribute Certificate Management Messages over CMS," <draft-ietf-pkix-acmc-01.txt>, March 2002.

[ADDSCHEMA] Chadwick, D., Legg, S., "Internet X.509 Public Key Infrastructure Additional LDAP Schema for PKIs and PMIs," <draft-ietf-pkix-ldap-schema-02.txt>, November 2001.

[CMC] Myers, M., Liu, X., Schaad, J., and Weinstein, J., "Certificate Management Messages over CMS," (RFC 2797), April 2000.

[CMP] Adams, C., Farrell, S., "Internet X.509 Public Key Infrastructure Certificate Management Protocols," RFC 2510, March 1999.

[CMS] R. Housley, "Cryptographic Message Syntax," RFC 2630, July 1999.

[CRMF] Myers, M., Adams, C., Solo, D., and Kemp, D., "Internet X.509 Certificate Request Message Format," RFC 2511, March 1999.

[DNS] Mockapetris, P.V., "Domain names - concepts and facilities," RFC 1034, November 1987.

[DHPOP] Prafullchandra, H., and Schaad, J., "Diffie-Hellman Proof-of-Possession Algorithms," RFC 2875, July 2000 1999.

[DPD] Myers, M., Adams, C., Farrell, S., "Delegated Path Discovery
with OCSP".

[DPV] Myers, M., Adams, C., Farrell, S., "Delegated Path Validation".

[DPREQ] Pinaks, D., Housley, R., "Delegated Path Validation and
Delegated Path Discovery Protocol Requirements (DPV&DPD-REQ),"
<draft-ietf-pkix-dpv-dpd-req-04.txt>, April 2002.

[DVCS] Adams, C., Sylvester, P., Zolotarev, M., Zuccherato, R.,
"Internet X.509 Public Key Infrastructure Data Certification Server
Protocols", RFC 3029, February 2001.

[FTPHTTP] Housley, R., and Hoffman, P., "Internet X.509 Public Key
Infrastructure Operational Protocols: FTP and HTTP," RFC 2585, July
1998.

[IP] Postel, J., "Internet Protocol," RFC 791, September 1981.

[IPEXT] Lynn, C., Kent, S., Seo, K., "X.509 Extensions for IP
Addresses and AS Identifiers," <draft-ietf-pkix-x509-ipaddr-as-extn-
00.txt>, February 2002.

[IPv6] Deering, S., and Hinden, R., "Internet Protocol, Version 6
[IPv6] Specification," RFC 1883, December 1995.

[KEA] Housley, R., and Polk, W., "Internet X.509 Public Key
Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in
Internet X.509 Public Key Infrastructure Certificates," RFC 2528,
March 1999.

[LAAP] Farrell, S., Chadwick, C.W., "Limited Attribute Certificate
Acquisition Protocol".

[LDAPv2] Yeong, Y., Howes, T., and Kille, S., "Lightweight Directory
Access Protocol", RFC 1777, March 1995.

[LOGO] Santesson, S. Housley, R., Freeman, T., "X.509 Internet Public
Key Infrastructure Logotypes in X.509 Certificates," <draft-ietf-
pkix-logotypes-02.txt>, April 2002.

[OCSP] Myers, M., Ankney, R., Malpani, A., Galperin, S., and Adams,
C., "X.509 Internet Public Key Infrastructure Online Certificate
Status Protocol - OCSP," RFC 2560, June 1999.

[OCSPv2] Myers, M., Ankney, R., Adams, C., "Online Certificate Status

Protocol, version 2," <draft-ietf-pkix-ocspv2-02.txt>, March 2001.

[MISPC] Burr, W., Dodson, D., Nazario, N., and Polk, W., "MISPC Minimum Interoperability Specification for PKI Components, Version 1", <http://csrc.nist.gov/pki/mispc/welcome.html>, 3 September 1997.

[PEM] Kent, S., "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management," RFC 1422, February 1993.

[PI] Pinka, D., Gindin, T., "Internet X.509 Public Key Infrastructure Permanent Identifier," <draft-ietf-pkix-pi-03.txt>, February 2002.

[PKI-LDAPv2] Boeyen, S., Howes, T., and Richard, P., "Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2," RFC 2559, April 1999.

[PKI-LDAPv3] Chadwick, D.W., "Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv3," <draft-ietf-pkix-ldap-v3-05.txt>, January 2002.

[POLPRAC] Chokhani, S., and Ford, W., "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," RFC 2527, March 1999.

[QC] Santesson, S., Polk, W., Barzin, P., and Nystrom, M., "Internet X.509 Public Key Infrastructure Qualified Certificates," RFC 3039, January 2001.

[RLS] Boeyen, S., Hallam-Baker, P., "Internet X.509 Public Key Infrastructure Repository Locator Service," <draft-ietf-pkix-pkixrep-00.txt>, July 2000.

[RPKDS] Bassham, L., Housley, R., Polk, W., "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile," RFC 3279, April 2002.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages," RFC 822, August 1982.

[SCHEMA] Boeyen, S., Howes, T., and Richard, P., "Internet X.509 Public Key Infrastructure LDAPv2 Schema," RFC 2587, June 1999.

[SCVP] Malpani, A., Hoffman, P., Housley, R., and Freeman, T., "Simple Certificate Validation Protocol (SCVP)," <draft-ietf-pkix-scvp-08.txt>, March 2002.

[SIMONETTI] Simonetti, D., "Re: German Key Usage", posting to ietf-

pkix@imc.org mailing list, 12 August 1998.

[SSKGKA] Schaad, J., " CMC Extensions: Server Side Key Generation and Key Archival," <draft-ietf-pkix-cmc-archive-00.txt>, July 2001.

[SUPPALGS] Singer, A., and Whyte, W., "Supplemental Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile," <draft-ietf-pkix-pkalgs-supp-01.txt>, March 2002.

[TECHNR] Gindin, T., "Internet X.509 Public Key Infrastructure Technical Requirements for a non-Repudiation Service," December 2000.

[TPCMC] Schaad, J. Myers, M., Liu, X., Weinstein, J., "CMC Transport," <draft-ietf-pkix-cmc-trans-01.txt>, March 2002.

[TPCMP] Kapoor , A., Tschalaer, R., "Transport Protocols for CMP," <draft-ietf-pkix-cmp-transport-protocols-04.txt>, November 2000.

[TSP] Adams, C., Cain, P., Pinkas, D., and Zuccherato, R., "Internet X.509 Public Key Infrastructure Time Stamp Protocols", RFC 3161, August 2001.

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.

[X9.42] ANSI X9.42-199x, Public Key Cryptography for The Financial Services Industry: Agreement of Symmetric Algorithm Keys Using Diffie-Hellman (Working Draft), December 1997.

[X9.55] ANSI X9.55-1995, Public Key Cryptography For The Financial Services Industry: Extensions To Public Key Certificates And Certificate Revocation Lists, 8 December, 1995.

[X9.57] ANSI X9.57-199x, Public Key Cryptography For The Financial Services Industry: Certificate Management (Working Draft), 21 June, 1996.

[WARR] Linsenbardt, D., Pontius, S., "Warranty Certificate Extension," <draft-ietf-pkix-warranty-extn-00.txt>, April 2002.

[PKCS10] RSA Laboratories, "The Public-Key Cryptography Standards(PKCS)," RSA Data Security Inc., Redwood City, California, November 1993 Release.

7 Security Considerations

This document records the history and design philosophy of the of
standards to support in a particular infrastructure or application.
Individual PKIX documents specify a variety of message formats and
protocols; a selection of these formats and protocols will be
necessary to construct a complete infrastructure.

This specification does not specify security considerations, since
they are determined by selection of formats and protocols. However,
each PKIX document specifies security considerations that are
associated with the message formats or protocols defined in that
particular standard. These security considerations must be
considered in aggregate when deploying or designing a particular
infrastructure or PKI-enabled application protocol.

8 Acknowledgements

A lot of the information in this document was taken from the PKIX
source documents; the authors of those deserve the credit for their
own words. Other good material was taken from mail posted to the PKIX
working group mail list (ietf-pkix@imc.org). Among those with good
things to say were (with apologies to anybody we've missed): Sharon
Boeyen, Santosh Chokhani, Warwick Ford, Russ Housley, Steve Kent,
Ambarish Malpani, Matt Fite, Michael Myers, Tim Polk, Stefan
Santesson, Dave Simonetti, Paul Hoffman, Denis Pinkas, Ed Gerck, Tom
Gindin, Parag Namjoshi, Peter Sylvester, and Michael Zolotarev.

9 Author's Addresses

Alfred W. Arsenault
Diversinet Corp.
P.O. Box 6530
Ellicott City, MD 21042-0530
aarsenault@dvnet.com

Sean Turner
IECA, Inc.
9010 Edgepark Road Vienna, VA 22182
(703) 628-3180
turners@ieca.com

Expires January 2003