                 Privacy Enhancement for Internet Electronic Mail:
                    Part II: Certificate-Based Key Management

Status of this Memo

   This RFC specifies an IAB standards track protocol for the Internet
   community, and requests discussion and suggestions for improvements.
   Please refer to the current edition of the "IAB Official Protocol
   Standards" for the standardization state and status of this protocol.
   Distribution of this memo is unlimited.

Acknowledgements

   This memo is the outgrowth of a series of meetings of the Privacy and
   Security Research Group of the Internet Research Task Force (IRTF)
   and the Privacy-Enhanced Electronic Mail Working Group of the
   Internet Engineering Task Force (IETF). I would like to thank the
   members of the PSRG and the PEM WG for their comments and
   contributions at the meetings which led to the preparation of this
   document. I also would like to thank contributors to the PEM-DEV
   mailing list who have provided valuable input which is reflected in
   this memo.

1. Executive Summary

   This is one of a series of documents defining privacy enhancement
   mechanisms for electronic mail transferred using Internet mail
   protocols. RFC 1421 [6] prescribes protocol extensions and
   processing procedures for RFC-822 mail messages, given that suitable
   cryptographic keys are held by originators and recipients as a
   necessary precondition. RFC 1423 [7] specifies algorithms, modes and
   associated identifiers for use in processing privacy-enhanced
   messages, as called for in RFC 1421 and this document. This document
   defines a supporting key management architecture and infrastructure,
   based on public-key certificate techniques, to provide keying
   information to message originators and recipients. RFC 1424 [8]
   provides additional specifications for services in conjunction with
   the key management infrastructure described herein.

   The key management architecture described in this document is
   compatible with the authentication framework described in CCITT 1988
   X.509 [2]. This document goes beyond X.509 by establishing

procedures and conventions for a key management infrastructure for
use with Privacy Enhanced Mail (PEM) and with other protocols, from
both the TCP/IP and OSI suites, in the future. There are several
motivations for establishing these procedures and conventions (as
opposed to relying only on the very general framework outlined in
X.509):

   -It is important that a certificate management infrastructure
       for use in the Internet community accommodate a range of
       clearly-articulated certification policies for both users
       and organizations in a well-architected fashion.
       Mechanisms must be provided to enable each user to be
       aware of the policies governing any certificate which the
       user may encounter. This requires the introduction
       and standardization of procedures and conventions that are
       outside the scope of X.509.

   -The procedures for authenticating originators and recipient in
       the course of message submission and delivery should be
       simple, automated and uniform despite the existence of
       differing certificate management policies. For example,
       users should not have to engage in careful examination of a
       complex set of certification relationships in order to
       evaluate the credibility of a claimed identity.

   -The authentication framework defined by X.509 is designed to
       operate in the X.500 directory server environment. However
       X.500 directory servers are not expected to be ubiquitous
       in the Internet in the near future, so some conventions
       are adopted to facilitate operation of the key management
       infrastructure in the near term.

   -Public key cryptosystems are central to the authentication
       technology of X.509 and those which enjoy the most
       widespread use are patented in the U.S. Although this
       certification management scheme is compatible with
       the use of different digital signature algorithms, it is
       anticipated that the RSA cryptosystem will be used as
       the primary signature algorithm in establishing the
       Internet certification hierarchy. Special license
       arrangements have been made to facilitate the
       use of this algorithm in the U.S. portion of Internet
       environment.

The infrastructure specified in this document establishes a single
root for all certification within the Internet, the Internet Policy
Registration Authority (IPRA). The IPRA establishes global policies,
described in this document, which apply to all certification effected

under this hierarchy. Beneath IPRA root are Policy Certification
Authorities (PCAs), each of which establishes and publishes (in the
form of an informational RFC) its policies for registration of users
or organizations. Each PCA is certified by the IPRA. (It is
desirable that there be a relatively small number of PCAs, each with
a substantively different policy, to facilitate user familiarity with
the set of PCA policies. However there is no explicit requirement
that the set of PCAs be limited in this fashion.) Below PCAs,
Certification Authorities (CAs) will be established to certify users
and subordinate organizational entities (e.g., departments, offices,
subsidiaries, etc.). Initially, we expect the majority of users will
be registered via organizational affiliation, consistent with current
practices for how most user mailboxes are provided. In this sense
the registration is analogous to the issuance of a university or
company ID card.

Some CAs are expected to provide certification for residential users
in support of users who wish to register independent of any
organizational affiliation. Over time, we anticipate that civil
government entities which already provide analogous identification
services in other contexts, e.g., driver's licenses, may provide
this service. For users who wish anonymity while taking advantage of
PEM privacy facilities, one or more PCAs will be established with
policies that allow for registration of users, under subordinate CAs,
who do not wish to disclose their identities.

2. Overview of Approach

This document defines a key management architecture based on the use
of public-key certificates, primarily in support of the message
encipherment and authentication procedures defined in RFC 1421. The
concept of public-key certificates is defined in X.509 and this
architecture is a compliant subset of that envisioned in X.509.

Briefly, a (public-key) certificate is a data structure which
contains the name of a user (the "subject"), the public component
(This document adopts the terms "private component" and "public
component" to refer to the quantities which are, respectively, kept
secret and made publicly available in asymmetric cryptosystems. This
convention is adopted to avoid possible confusion arising from use of
the term "secret key" to refer to either the former quantity or to a
key in a symmetric cryptosystem.) of that user, and the name of an
entity (the "issuer") which vouches that the public component is
bound to the named user. This data, along with a time interval over
which the binding is claimed to be valid, is cryptographically signed
by the issuer using the issuer's private component. The subject and
issuer names in certificates are Distinguished Names (DNs) as defined
in the directory system (X.500).

Once signed, certificates can be stored in directory servers,
transmitted via non-secure message exchanges, or distributed via any
other means that make certificates easily accessible to message
system users, without regard for the security of the transmission
medium. Certificates are used in PEM to provide the originator of a
message with the (authenticated) public component of each recipient
and to provide each recipient with the (authenticated) public
component of the originator. The following brief discussion
illustrates the procedures for both originator and recipients.

Prior to sending an encrypted message (using PEM), an originator must
acquire a certificate for each recipient and must validate these
certificates. Briefly, validation is performed by checking the
digital signature in the certificate, using the public component of
the issuer whose private component was used to sign the certificate.
The issuer's public component is made available via some out of band
means (for the IPRA) or is itself distributed in a certificate to
which this validation procedure is applied recursively. In the
latter case, the issuer of a user's certificate becomes the subject
in a certificate issued by another certifying authority (or a PCA),
thus giving rise to a certification hierarchy. The validity interval
for each certificate is checked and Certificate Revocation Lists
(CRLs) are checked to ensure that none of the certificates employed
in the validation process has been revoked by an issuer.

Once a certificate for a recipient is validated, the public component
contained in the certificate is extracted and used to encrypt the
data encryption key (DEK), which, in turn, is used to encrypt the
message itself. The resulting encrypted DEK is incorporated into the
Key-Info field of the message header. Upon receipt of an encrypted
message, a recipient employs his private component to decrypt this
field, extracting the DEK, and then uses this DEK to decrypt the
message.

In order to provide message integrity and data origin authentication,
the originator generates a message integrity code (MIC), signs
(encrypts) the MIC using the private component of his public-key
pair, and includes the resulting value in the message header in the
MIC-Info field. The certificate of the originator is (optionally)
included in the header in the Certificate field as described in RFC
1421. This is done in order to facilitate validation in the absence
of ubiquitous directory services. Upon receipt of a privacy enhanced
message, a recipient validates the originator's certificate (using
the IPRA public component as the root of a certification path),
checks to ensure that it has not been revoked, extracts the public
component from the certificate, and uses that value to recover
(decrypt) the MIC. The recovered MIC is compared against the locally
calculated MIC to verify the integrity and data origin authenticity

of the message.

3. Architecture

   3.1 Scope and Restrictions

   The architecture described below is intended to provide a basis for
   managing public-key cryptosystem values in support of privacy
   enhanced electronic mail in the Internet environment.   The
   architecture describes procedures for registering certification
   authorities and users, for generating and distributing certificates,
   and for generating and distributing CRLs. RFC 1421 describes the
   syntax and semantics of header fields used to transfer certificates
   and to represent the DEK and MIC in this public-key context.
   Definitions of the algorithms, modes of use and associated
   identifiers are separated in RFC 1423 to facilitate the adoption of
   additional algorithms in the future. This document focuses on the
   management aspects of certificate-based, public-key cryptography for
   privacy enhanced mail.

   The proposed architecture imposes conventions for the certification
   hierarchy which are not strictly required by the X.509 recommendation
   nor by the technology itself. These conventions are motivated by
   several factors, primarily the need for authentication semantics
   compatible with automated validation and the automated determination
   of the policies under which certificates are issued.

   Specifically, the architecture proposes a system in which user (or
   mailing list) certificates represent the leaves in a certification
   hierarchy. This certification hierarchy is largely isomorphic to the
   X.500 directory naming hierarchy, with two exceptions: the IPRA forms
   the root of the tree (the root of the X.500 DIT is not instantiated
   as a node), and a number of Policy Certification Authorities (PCAs)
   form the "roots" of subtrees, each of which represents a different
   certification policy.

   Not every level in the directory hierarchy need correspond to a
   certification authority. For example, the appearance of geographic
   entities in a distinguished name (e.g., countries, states, provinces,
   localities) does not require that various governments become
   certifying authorities in order to instantiate this architecture.
   However, it is anticipated that, over time, a number of such points
   in the hierarchy will be instantiated as CAs in order to simplify
   later transition of management to appropriate governmental
   authorities.

   These conventions minimize the complexity of validating user
   certificates, e.g., by making explicit the relationship between a

certificate issuer and the user (via the naming hierarchy). Note that
in this architecture, only PCAs may be certified by the IPRA, and
every CA's certification path can be traced to a PCA, through zero or
more CAs. If a CA is certified by more than one PCA, each
certificate issued by a PCA for the CA must contain a distinct public
component. These conventions result in a certification hierarchy
which is a compatible subset of that permitted under X.509, with
respect to both syntax and semantics.

Although the key management architecture described in this document
has been designed primarily to support privacy enhanced mail, this
infrastructure also may, in principle, be used to support X.400 mail
security facilities (as per 1988 X.411) and X.500 directory
authentication facilities. Thus, establishment of this
infrastructure paves the way for use of these and other OSI protocols
in the Internet in the future. In the future, these certificates
also may be employed in the provision of security services in other
protocols in the TCP/IP and OSI suites as well.

3.2 Relation to X.509 Architecture

CCITT 1988 Recommendation X.509, "The Directory - Authentication
Framework", defines a framework for authentication of entities
involved in a distributed directory service. Strong authentication,
as defined in X.509, is accomplished with the use of public-key
cryptosystems. Unforgeable certificates are generated by
certification authorities; these authorities may be organized
hierarchically, though such organization is not required by X.509.
There is no implied mapping between a certification hierarchy and the
naming hierarchy imposed by directory system naming attributes.

This document interprets the X.509 certificate mechanism to serve the
needs of PEM in the Internet environment. The certification
hierarchy proposed in this document in support of privacy enhanced
mail is intentionally a subset of that allowed under X.509. This
certification hierarchy also embodies semantics which are not
explicitly addressed by X.509, but which are consistent with X.509
precepts. An overview of the rationale for these semantics is
provided in Section 1.

3.3 Certificate Definition

Certificates are central to the key management architecture for X.509
and PEM. This section provides an overview of the syntax and a
description of the semantics of certificates. Appendix A includes
the ASN.1 syntax for certificates.   A certificate includes the
following contents:

      1. version

      2. serial number

      3. signature (algorithm ID and parameters)

      4. issuer name

      5. validity period

      6. subject name

      7. subject public key (and associated algorithm ID)

   3.3.1 Version Number

   The version number field is intended to facilitate orderly changes in
   certificate formats over time. The initial version number for
   certificates used in PEM is the X.509 default which has a value of
   zero (0), indicating the 1988 version. PEM implementations are
   encouraged to accept later versions as they are endorsed by
   CCITT/ISO.

   3.3.2 Serial Number

   The serial number field provides a short form, unique identifier for
   each certificate generated by an issuer. An issuer must ensure that
   no two distinct certificates with the same issuer DN contain the same
   serial number. (This requirement must be met even when the
   certification function is effected on a distributed basis and/or when
   the same issuer DN is certified under two different PCAs. This is
   especially critical for residential CAs certified under different
   PCAs.) The serial number is used in CRLs to identify revoked
   certificates, as described in Section 3.4.3.4. Although this
   attribute is an integer, PEM UA processing of this attribute need not
   involve any arithmetic operations. All PEM UA implementations must
   be capable of processing serial numbers at least 128 bits in length,
   and size-independent support serial numbers is encouraged.

   3.3.3 Signature

   This field specifies the algorithm used by the issuer to sign the
   certificate, and any parameters associated with the algorithm. (The
   certificate signature is appended to the data structure, as defined
   by the signature macro in X.509. This algorithm identification
   information is replicated with the signature.) The signature is
   validated by the UA processing a certificate, in order to determine
   that the integrity of its contents have not been modified subsequent

to signing by a CA (IPRA, or PCA). In this context, a signature is
effected through the use of a Certificate Integrity Check (CIC)
algorithm and a public-key encryption algorithm. RFC 1423 contains
the definitions and algorithm IDs for signature algorithms employed
in this architecture.

3.3.4 Subject Name

A certificate provides a representation of its subject's identity in
the form of a Distinguished Name (DN). The fundamental binding
ensured by the key management architecture is that between the public
component and the user's identity in this form. A distinguished name
is an X.500 directory system concept and if a user is already
registered in an X.500 directory, his distinguished name is defined
via that registration. Users who are not registered in a directory
should keep in mind likely directory naming structure (schema) when
selecting a distinguished name for inclusion in a certificate.

3.3.5 Issuer Name

A certificate provides a representation of its issuer's identity, in
the form of a Distinguished Name. The issuer identification is used
to select the appropriate issuer public component to employ in
performing certificate validation. (If an issuer (CA) is certified
by multiple PCAs, then the issuer DN does not uniquely identify the
public component used to sign the certificate. In such circumstances
it may be necessary to attempt certificate validation using multiple
public components, from certificates held by the issuer under
different PCAs. If the 1992 version of a certificate is employed,
the issuer may employ distinct issuer UIDs in the certificates it
issues, to further facilitate selection of the right issuer public
component.) The issuer is the certifying authority (IPRA, PCA or CA)
who vouches for the binding between the subject identity and the
public key contained in the certificate.

3.3.6 Validity Period

A certificate carries a pair of date and time indications, indicating
the start and end of the time period over which a certificate is
intended to be used. The duration of the interval may be constant
for all user certificates issued by a given CA or it might differ
based on the nature of the user's affiliation. For example, an
organization might issue certificates with shorter intervals to
temporary employees versus permanent employees. It is recommended
that the UTCT (Coordinated Universal Time) values recorded here
specify granularity to no more than the minute, even though finer
granularity can be expressed in the format. (Implementors are warned
that no DER is defined for UTCT in X.509, thus transformation between

local and transfer syntax must be performed carefully, e.g., when
computing the hash value for a certificate. For example, a UTCT
value which includes explicit, zero values for seconds would not
produce the same hash value as one in which the seconds were
omitted.) It also recommended that all times be expressed as
Greenwich Mean Time (Zulu), to simplify comparisons and avoid
confusion relating to daylight savings time. Note that UTCT
expresses the value of a year modulo 100 (with no indication of
century), hence comparisons involving dates in different centuries
must be performed with care.

The longer the interval, the greater the likelihood that compromise
of a private component or name change will render it invalid and thus
require that the certificate be revoked. Once revoked, the
certificate must remain on the issuer's CRL (see Section 3.4.3.4)
until the validity interval expires. PCAs may impose restrictions on
the maximum validity interval that may be elected by CAs operating in
their certification domain (see Appendix B).

3.3.7 Subject Public Key

A certificate carries the public component of its associated subject,
as well as an indication of the algorithm, and any algorithm
parameters, with which the public component is to be used. This
algorithm identifier is independent of that which is specified in the
signature field described above. RFC 1423 specifies the algorithm
identifiers which may be used in this context.

3.4 Roles and Responsibilities

One way to explain the architecture proposed by this document is to
examine the roles which are defined for various entities in the
architecture and to describe what is required of each entity in order
for the proposed system to work properly. The following sections
identify four types of entities within this architecture: users and
user agents, the Internet Policy Registration Authority, Policy
Certification Authorities, and other Certification Authorities. For
each type of entity, this document specifies the procedures which the
entity must execute as part of the architecture and the
responsibilities the entity assumes as a function of its role in the
architecture.

3.4.1 Users and User Agents

The term User Agent (UA) is taken from CCITT X.400 Message Handling
Systems (MHS) Recommendations, which define it as follows: "In the
context of message handling, the functional object, a component of
MHS, by means of which a single direct user engages in message

handling."   In the Internet environment, programs such as rand mh
and Gnu emacs rmail are UAs. UAs exchange messages by calling on a
supporting Message Transfer Service (MTS), e.g., the SMTP mail relays
used in the Internet.

3.4.1.1 Generating and Protecting Component Pairs

A UA process supporting PEM must protect the private component of its
associated entity (e.g., a human user or a mailing list) from
disclosure, though the means by which this is effected is a local
matter. It is essential that the user take all available precautions
to protect his private component as the secrecy of this value is
central to the security offered by PEM to that user.   For example,
the private component might be stored in encrypted form, protected
with a locally managed symmetric encryption key (e.g., using DES).
The user would supply a password or passphrase which would be
employed as a symmetric key to decrypt the private component when
required for PEM processing (either on a per message or per session
basis). Alternatively, the private component might be stored on a
diskette which would be inserted by the user whenever he originated
or received PEM messages. Explicit zeroing of memory locations where
this component transiently resides could provide further protection.
Other precautions, based on local operating system security
facilities, also should be employed.

It is recommended that each user employ ancillary software (not
otherwise associated with normal UA operation) or hardware to
generate his personal public-key component pair. Software for
generating user component pairs will be available as part of the
reference implementation of PEM distributed freely in the U.S.
portion of the Internet. It is critically important that the
component pair generation procedure be effected in as secure a
fashion as possible, to ensure that the resulting private component
is unpredictable. Introduction of adequate randomness into the
component pair generation procedure is potentially the most difficult
aspect of this process and the user is advised to pay particular
attention to this aspect. (Component pairs employed in public-key
cryptosystems tend to be large integers which must be "randomly"
selected subject to mathematical constraints imposed by the
cryptosystem. Input(s) used to seed the component pair generation
process must be as unpredictable as possible. An example of a poor
random number selection technique is one in which a pseudo-random
number generator is seeded solely with the current date and time. An
attacker who could determine approximately when a component pair was
generated could easily regenerate candidate component pairs and
compare the public component to the user's public component to detect
when the corresponding private component had been found.)

There is no requirement imposed by this architecture that anyone
other than the user, including any certification authority, have
access to the user's private component. Thus a user may retain his
component pair even if his certificate changes, e.g., due to rollover
in the validity interval or because of a change of certifying
authority. Even if a user is issued a certificate in the context of
his employment, there is generally no requirement that the employer
have access to the user's private component. The rationale is that
any messages signed by the user are verifiable using his public
component.   In the event that the corresponding private component
becomes unavailable, any ENCRYPTED messages directed to the user
would be indecipherable and would require retransmission.

Note that if the user stores messages in ENCRYPTED form, these
messages also would become indecipherable in the event that the
private component is lost or changed. To minimize the potential for
loss of data in such circumstances messages can be transformed into
MIC-ONLY or MIC-CLEAR form if cryptographically-enforced
confidentiality is not required for the messages stored within the
user's computer. Alternatively, these transformed messages might be
forwarded in ENCRYPTED form to a (trivial) distribution list which
serves in a backup capacity and for which the user's employer holds
the private component.

A user may possess multiple certificates which may embody the same or
different public components. For example, these certificates might
represent a current and a former organizational user identity and a
residential user identity. It is recommended that a PEM UA be
capable of supporting a user who possess multiple certificates,
irrespective of whether the certificates associated with the user
contain the same or different DNs or public components.

3.4.1.2 User Registration

Most details of user registration are a local matter, subject to
policies established by the user's CA and the PCA under which that CA
has been certified. In general a user must provide, at a minimum,
his public component and distinguished name to a CA, or a
representative thereof, for inclusion in the user's certificate.
(The user also might provide a complete certificate, minus the
signature, as described in RFC 1424.) The CA will employ some means,
specified by the CA in accordance with the policy of its PCA, to
validate the user's claimed identity and to ensure that the public
component provided is associated with the user whose distinguished
name is to be bound into the certificate. (In the case of PERSONA
certificates, described below, the procedure is a bit different.) The
certifying authority generates a certificate containing the user's
distinguished name and public component, the authority's

distinguished name and other information (see Section 3.3) and signs
the result using the private component of the authority.

3.4.1.3 CRL Management

Mechanisms for managing a UA certificate cache are, in typical
standards parlance, a local matter. However, proper maintenance of
such a cache is critical to the correct, secure operation of a PEM UA
and provides a basis for improved performance. Moreover, use of a
cache permits a PEM UA to operate in the absence of directories (and
in circumstances where directories are inaccessible). The following
discussion provides a paradigm for one aspect of cache management,
namely the processing of CRLs, the functional equivalent of which
must be embodied in any PEM UA implementation compliant with this
document. The specifications for CRLs used with PEM are provided in
Section 3.5.

X.500 makes provision for the storage of CRLs as directory attributes
associated with CA entries. Thus, when X.500 directories become
widely available, UAs can retrieve CRLs from directories as required.
In the interim, the IPRA will coordinate with PCAs to provide a
robust database facility which will contain CRLs issued by the IPRA,
by PCAs, and by all CAs. Access to this database will be provided
through mailboxes maintained by each PCA. Every PEM UA must provide
a facility for requesting CRLs from this database using the
mechanisms defined in RFC 1424. Thus the UA must include a
configuration parameter which specifies one or more mailbox addresses
from which CRLs may be retrieved. Access to the CRL database may be
automated, e.g., as part of the certificate validation process (see
Section 3.6) or may be user directed. Responses to CRL requests will
employ the PEM header format specified in RFC 1421 for CRL
propagation. As noted in RFC 1421, every PEM UA must be capable of
processing CRLs distributed via such messages. This message format
also may be employed to support a "push" (versus a "pull") model of
CRL distribution, i.e., to support unsolicited distribution of CRLs.

CRLs received by a PEM UA must be validated (A CRL is validated in
much the same manner as a certificate, i.e., the CIC (see RFC 1113)
is calculated and compared against the decrypted signature value
obtained from the CRL. See Section 3.6 for additional details
related to validation of certificates.) prior to being processed
against any cached certificate information. Any cache entries which
match CRL entries should be marked as revoked, but it is not
necessary to delete cache entries marked as revoked nor to delete
subordinate entries. In processing a CRL against the cache it is
important to recall that certificate serial numbers are unique only
for each issuer and that multiple, distinct CRLs may be issued under
the same CA DN (signed using different private components), so care

must be exercised in effecting this cache search. (This situation
may arise either because an organizational CA is certified by
multiple PCAs, or because multiple residential CAs are certified
under different PCAs.)

This procedure applies to cache entries associated with PCAs and CAs,
as well as user entries. The UA also must retain each CRL to screen
incoming messages to detect use of revoked certificates carried in
PEM message headers. Thus a UA must be capable of processing and
retaining CRLs issued by the IPRA (which will list revoked PCA
certificates), by any PCA (which will list revoked CA certificate
issued by that PCA), and by any CA (which will list revoked user or
subordinate CA certificates issued by that CA).

3.4.1.4 Facilitating Interoperation

In the absence of ubiquitous directory services or knowledge
(acquired through out-of-band means) that a recipient already
possesses the necessary issuer certificates, it is recommended that
an originating (PEM) UA include sufficient certificates to permit
validation of the user's public key. To this end every PEM UA must
be capable of including a full (originator) certification path, i.e.,
including the user's certificate (using the "Originator-Certificate"
field) and every superior (CA/PCA) certificate (using "Issuer-
Certificate" fields) back to the IPRA, in a PEM message. A PEM UA
may send less than a full certification path, e.g., based on analysis
of a recipient list, but a UA which provides this sort of
optimization must also provide the user with a capability to force
transmission of a full certification path.

Optimization for the transmitted originator certification path may be
effected by a UA as a side effect of the processing performed during
message submission. When an originator submits an ENCRYPTED message
(as per RFC 1421, his UA must validate the certificates of the
recipients (see Section 3.6). In the course of performing this
validation the UA can determine the minimum set of certificates which
must be included to ensure that all recipients can process the
received message. Submission of a MIC-ONLY or MIC-CLEAR message (as
per RFC 1421) does not entail validation of recipient certificates
and thus it may not be possible for the originator's UA to determine
the minimum certificate set as above.

3.4.2 The Internet Policy Registration Authority (IPRA)

The IPRA acts as the root of the certification hierarchy for the
Internet community. The public component of the IPRA forms the
foundation for all certificate validation within this hierarchy. The
IPRA will be operated under the auspices of the Internet Society, an

international, non-profit organization. The IPRA certifies all PCAs,
ensuring that they agree to abide by the Internet-wide policy
established by the IPRA. This policy, and the services provided by
the IPRA, are detailed below.

3.4.2.1 PCA Registration

The IPRA certifies only PCAs, not CAs or users. Each PCA must file
with the IPRA a description of its proposed policy. This document
will be published as an informational RFC. A copy of the document,
signed by the IPRA (in the form of a PEM MIC-ONLY message) will be
made available via electronic mail access by the IPRA. This
convention is adopted so that every Internet user has a reference
point for determining the policies associated with the issuance of
any certificate which he may encounter. The existence of a digitally
signed copy of the document ensures the immutability of the document.
Authorization of a PCA to operate in the Internet hierarchy is
signified by the publication of the policy document, and the issuance
of a certificate to the PCA, signed by the IPRA. An outline for PCA
policy statements is contained in Section 3.4.3 of this document.

As part of registration, each PCA will be required to execute a legal
agreement with the IPRA, and to pay a fee to defray the costs of
operating the IPRA. Each a PCA must specify its distinguished name.
The IPRA will take reasonable precautions to ensure that the
distinguished name claimed by a PCA is legitimate, e.g., requiring
the PCA to provide documentation supporting its claim to a DN.
However, the certification of a PCA by the IPRA does not constitute a
endorsement of the PCA's claim to this DN outside of the context of
this certification system.

3.4.2.2 Ensuring the Uniqueness of Distinguished Names

A fundamental requirement of this certification scheme is that
certificates are not issued to distinct entities under the same
distinguished name. This requirement is important to the success of
distributed management for the certification hierarchy. The IPRA
will not certify two PCAs with the same distinguished name and no PCA
may certify two CAs with the same DN. However, since PCAs are
expected to certify organizational CAs in widely disjoint portions of
the directory namespace, and since X.500 directories are not
ubiquitous, a facility is required for coordination among PCAs to
ensure the uniqueness of CA DNs. (This architecture allows multiple
PCAs to certify residential CAs and thus multiple, distinct
residential CAs with identical DNs may come into existence, at least
until such time as civil authorities assume responsibilities for such
certification. Thus, on an interim basis, the architecture
explicitly accommodates the potential for duplicate residential CA

DNs.)

In support of the uniqueness requirement, the IPRA will establish and
maintain a database to detect potential, unintended duplicate
certification of CA distinguished names. This database will be made
accessible to all PCAs via an email interface. Each entry in this
database will consist of a 4-tuple. The first element in each entry
is a hash value, computed on a canonical, ASN.1 encoded
representation of a CA distinguished name. The second element
contains the subjectPublicKey that appears in the CA's certificate.
The third element is the distinguished name of the PCA which
registered the entry. The fourth element consists of the date and
time at which the entry was made, as established by the IPRA. This
database structure provides a degree of privacy for CAs registered by
PCAs, while providing a facility for ensuring global uniqueness of CA
DNs certified in this scheme.

In order to avoid conflicts, a PCA should query the database using a
CA DN hash value as a search key, prior to certifying a CA. The
database will return any entries which match the query, i.e., which
have the same CA DN. The PCA can use the information contained in
any returned entries to determine if any PCAs should be contacted to
resolve possible DN conflicts. If no potential conflicts appear, a
PCA can then submit a candidate entry, consisting of the first three
element values, plus any entries returned by the query. The database
will register this entry, supplying the time and date stamp, only if
two conditions are met: (1) the first two elements (the CA DN hash
and the CA subjectPublicKey) of the candidate entry together must be
unique and, (2) any other entries included in the submission must
match what the current database would return if the query
corresponding to the candidate entry were submitted.

If the database detects a conflicting entry (failure of case 1
above), or if the submission indicates that the PCA's perception of
possible conflicting entries is not current (failure of case 2), the
submission is rejected and the database will return the potential
conflicting entry (entries). If the submission is successful, the
database will return the timestamped new entry. The database does
not, in itself, guarantee uniqueness of CA DNs as it allows for two
DNs associated with different public components to be registered.
Rather, it is the responsibility of PCAs to coordinate with one
another whenever the database indicates a potential DN conflict and
to resolve such conflicts prior to certification of CAs. Details of
the protocol used to access the database will be provided in another
document.

As noted earlier, a CA may be certified under more than one PCA,
e.g., because the CA wants to issue certificates under two different

policies. If a CA is certified by multiple different PCAs, the CA
must employ a different public key pair for each PCA. In such
circumstances the certificate issued to the CA by each PCA will
contain a different subjectPublicKey and thus will represent a
different entry in this database. The same situation may arise if
multiple, equivalent residential CAs are certified by different PCAs.

To complete the strategy for ensuring uniqueness of DNs, there is a
DN subordination requirement levied on CAs. In general, CAs are
expected to sign certificates only if the subject DN in the
certificate is subordinate to the issuer (CA) DN. This ensures that
certificates issued by a CA are syntactically constrained to refer to
subordinate entities in the X.500 directory information tree (DIT),
and this further limits the possibility of duplicate DN registration.
CAs may sign certificates which do not comply with this requirement
if the certificates are "cross-certificates" or "reverse
certificates" (see X.509) used with applications other than PEM.

The IPRA also will establish and maintain a separate database to
detect potential duplicate certification of (residential) user
distinguished names. Each entry in this database will consist of 4-
tuple as above, but the first components is the hash of a residential
user DN and the third component is the DN of the residential CA DN
which registered the user. This structure provides a degree of
privacy for users registered by CAs which service residential users
while providing a facility for ensuring global uniqueness of user DNs
certified under this scheme. The same database access facilities are
provided as described above for the CA database. Here it is the
responsibility of the CAs to coordinate whenever the database
indicates a potential conflict and to resolve the conflict prior to
(residential) user certification.

3.4.2.3 Accuracy of Distinguished Names

As noted above, the IPRA will make a reasonable effort to ensure that
PCA DNs are accurate. The procedures employed to ensure the accuracy
of a CA distinguished name, i.e., the confidence attached to the
DN/public component binding implied by a certificate, will vary
according to PCA policy. However, it is expected that every PCA will
make a good faith effort to ensure the legitimacy of each CA DN
certified by the PCA. Part of this effort should include a check
that the purported CA DN is consistent with any applicable national
standards for DN assignment, e.g., NADF recommendations within North
America [5,9].

3.4.2.4 Distinguished Name Conventions

A few basic DN conventions are included in the IPRA policy. The IPRA
will certify PCAs, but not CAs nor users. PCAs will certify CAs, but
not users. These conventions are required to allow simple
certificate validation within PEM, as described later. Certificates
issued by CAs (for use with PEM) will be for users or for other CAs,
either of which must have DNs subordinate to that of the issuing CA.

The attributes employed in constructing DNs will be specified in a
list maintained by the IANA, to provide a coordinated basis for
attribute identification for all applications employing DNs. This
list will initially be populated with attributes taken from X.520.
This document does not impose detailed restrictions on the attributes
used to identify different entities to which certificates are issued,
but PCAs may impose such restrictions as part of their policies.
PCAs, CAs and users are urged to employ only those DN attributes
which have printable representations, to facilitate display and
entry.

3.4.2.5 CRL Management

Among the procedures articulated by each PCA in its policy statement
are procedures for the maintenance and distribution of CRLs by the
PCA itself and by its subordinate CAs. The frequency of issue of
CRLs may vary according to PCA-specific policy, but every PCA and CA
must issue a CRL upon inception to provide a basis for uniform
certificate validation procedures throughout the Internet hierarchy.
The IPRA will maintain a CRL for all the PCAs it certifies and this
CRL will be updated monthly. Each PCA will maintain a CRL for all of
the CAs which it certifies and these CRLs will be updated in
accordance with each PCA's policy.   The format for these CRLs is
that specified in Section 3.5.2 of the document.

In the absence of ubiquitous X.500 directory services, the IPRA will
require each PCA to provide, for its users, robust database access to
CRLs for the Internet hierarchy, i.e., the IPRA CRL, PCA CRLs, and
CRLs from all CAs. The means by which this database is implemented
is to be coordinated between the IPRA and PCAs. This database will
be accessible via email as specified in RFC 1424, both for retrieval
of (current) CRLs by any user, and for submission of new CRLs by CAs,
PCAs and the IPRA. Individual PCAs also may elect to maintain CRL
archives for their CAs, but this is not required by this policy.

3.4.2.6 Public Key Algorithm Licensing Issues

This certification hierarchy is architecturally independent of any
specific digital signature (public key) algorithm. Some algorithms,

employed for signing certificates and validating certificate
signatures, are patented in some countries. The IPRA will not grant
a license to any PCA for the use of any signature algorithm in
conjunction with the management of this certification hierarchy. The
IPRA will acquire, for itself, any licenses needed for it to sign
certificates and CRLs for PCAs, for all algorithms which the IPRA
supports. Every PCA will be required to represent to the IPRA that
the PCA has obtained any licenses required to issue (sign)
certificates and CRLs in the environment(s) which the PCA will serve.

For example, the RSA cryptosystem is patented in the United States
and thus any PCA operating in the U.S. and using RSA to sign
certificates and CRLs must represent that it has a valid license to
employ the RSA algorithm in this fashion. In contrast, a PCA
employing RSA and operating outside of the U.S. would represent that
it is exempt from these licensing constraints.

3.4.3 Policy Certification Authorities

The policy statement submitted by a prospective PCA must address the
topics in the following outline. Additional policy information may
be contained in the statement, but PCAs are requested not to use
these statements as advertising vehicles.

1. PCA Identity- The DN of the PCA must be specified. A postal
address, an Internet mail address, and telephone (and optional fax)
numbers must be provided for (human) contact with the PCA. The date
on which this statement is effective, and its scheduled duration must
be specified.

2. PCA Scope- Each PCA must describe the community which the PCA
plans to serve. A PCA should indicate if it will certify
organizational, residential, and/or PERSONA CAs.   There is not a
requirement that a single PCA serve only one type of CA, but if a PCA
serves multiple types of CAs, the policy statement must specify
clearly how a user can distinguish among these classes. If the PCA
will operate CAs to directly serve residential or PERSONA users, it
must so state.

3. PCA Security & Privacy- Each PCA must specify the technical and
procedural security measures it will employ in the generation and
protection of its component pair. If any security requirements are
imposed on CAs certified by the PCA these must be specified as well.
A PCA also must specify what measures it will take to protect the
privacy of any information collected in the course of certifying CAs.
If the PCA operates one or more CAs directly, to serve residential or
PERSONA users, then this statement on privacy measures applies to
these CAs as well.

4. Certification Policy- Each PCA must specify the policy and procedures which govern its certification of CAs and how this policy applies transitively to entities (users or subordinate CAs) certified by these CAs. For example, a PCA must state what procedure is employed to verify the claimed identity of a CA, and the CA's right to use a DN. Similarly, if any requirements are imposed on CAs to validate the identity of users, these requirements must be specified. Since all PCAs are required to cooperate in the resolution of potential DN conflicts, each PCA is required to specify the procedure it will employ to resolve such conflicts. If the PCA imposes a maximum validity interval for the CA certificates it issues, and/or for user (or subordinate CA) certificates issued by the CAs it certifies, then these restrictions must be specified.

5. CRL Management- Each PCA must specify the frequency with which it will issue scheduled CRLs. It also must specify any constraints it imposes on the frequency of scheduled issue of CRLs by the CAs it certifies, and by subordinate CAs. Both maximum and minimum constraints should be specified. Since the IPRA policy calls for each CRL issued by a CA to be forwarded to the cognizant PCA, each PCA must specify a mailbox address to which CRLs are to be transmitted. The PCA also must specify a mailbox address for CRL queries. If the PCA offers any additional CRL management services, e.g., archiving of old CRLs, then procedures for invoking these services must be specified. If the PCA requires CAs to provide any additional CRL management services, such services must be specified here.

6. Naming Conventions- If the PCA imposes any conventions on DNs used by the CAs it certifies, or by entities certified by these CAs, these conventions must be specified. If any semantics are associated with such conventions, these semantics must be specified.

7. Business Issues- If a legal agreement must be executed between a PCA and the CAs it certifies, reference to that agreement must be noted, but the agreement itself ought not be a part of the policy statement. Similarly, if any fees are charged by the PCA this should be noted, but the fee structure per se ought not be part of this policy statement.

8. Other- Any other topics the PCA deems relevant to a statement of its policy can be included. However, the PCA should be aware that a policy statement is considered to be an immutable, long lived document and thus considerable care should be exercised in deciding what material is to be included in the statement.

3.4.4 Certification Authorities

In X.509 the term "certification authority" is defined as "an
authority trusted by one or more users to create and assign
certificates". X.509 imposes few constraints on CAs, but practical
implementation of a worldwide certification system requires
establishment of technical and procedural conventions by which all
CAs are expected to abide. Such conventions are established
throughout this document. All CAs are required to maintain a
database of the DNs which they have certified and to take measures to
ensure that they do not certify duplicate DNs, either for users or
for subordinate CAs.

It is critical that the private component of a CA be afforded a high
level of security, otherwise the authenticity guarantee implied by
certificates signed by the CA is voided. Some PCAs may impose
stringent requirements on CAs within their purview to ensure that a
high level of security is afforded the certificate signing process,
but not all PCAs are expected to impose such constraints.

3.4.4.1 Organizational CAs

Many of the CAs certified by PCAs are expected to represent
organizations. A wide range of organizations are encompassed by this
model: commercial, governmental, educational, non-profit,
professional societies, etc. The common thread is that the entities
certified by these CAs have some form of affiliation with the
organization. The object classes for organizations, organizational
units, organizational persons, organizational roles, etc., as defined
in X.521, form the models for entities certified by such CAs. The
affiliation implied by organizational certification motivates the DN
subordination requirement cited in Section 3.4.2.4.

As an example, an organizational user certificate might contain a
subject DN of the form: C = "US" SP = "Massachusetts" L = "Cambridge"
O = "Bolt Beranek and Newman" OU = "Communications Division" CN =
"Steve Kent". The issuer of this certificate might have a DN of the
form: C = "US" SP = "Massachusetts" L = "Cambridge" O= "Bolt Beranek
and Newman". Note that the organizational unit attribute is omitted
from the issuer DN, implying that there is no CA dedicated to the
"Communications Division".

3.4.4.2 Residential CAs

Users may wish to obtain certificates which do not imply any
organizational affiliation but which do purport to accurately and
uniquely identify them. Such users can be registered as residential
persons and the DN of such a user should be consistent with the

attributes of the corresponding X.521 object class. Over time we
anticipate that such users will be accommodated by civil government
entities who will assume electronic certification responsibility at
geographically designated points in the naming hierarchy. Until
civil authorities are prepared to issue certificates of this form,
residential user CAs will accommodate such users.

Because residential CAs may be operated under the auspices of
multiple PCAs, there is a potential for the same residential CA DN to
be assumed by several distinct entities. This represents the one
exception to the rule articulated throughout this document that no
two entities may have the same DN. This conflict is tolerated so as
to allow residential CAs to be established offering different
policies. Two requirements are levied upon residential CAs as a
result: (1) residential CAs must employ the residential DN conflict
detection database maintained by the IPRA, and (2) residential CAs
must coordinate to ensure that they do not assign duplicate
certificate serial numbers.

As an example, a residential user certificate might include a subject
name of the form: C = "US" SP = "Massachusetts" L = "Boston" PA = "19
North Square" CN = "Paul Revere." The issuer of that certificate
might have a DN of the form: C = "US" SP = "Massachusetts" L =
"Boston". Note that the issuer DN is superior to the subject DN, as
required by the IPRA policy described earlier.

3.4.4.3 PERSONA CAs

One or more CAs will be established to accommodate users who wish to
conceal their identities while making use of PEM security features,
e.g., to preserve the anonymity offered by "arbitrary" mailbox names
in the current mail environment. In this case the certifying
authority is explicitly NOT vouching for the identity of the user.
All such certificates are issued under a PERSONA CA, subordinate to a
PCA with a PERSONA policy, to warn users explicitly that the subject
DN is NOT a validated user identity. To minimize the possibility of
syntactic confusion with certificates which do purport to specify an
authenticated user identity, a PERSONA certificate is issued as a
form of organizational user certificate, not a residential user
certificate. There are no explicit, reserved words used to identify
PERSONA user certificates.

A CA issuing PERSONA certificates must institute procedures to ensure
that it does not issue the same subject DN to multiple users (a
constraint required for all certificates of any type issued by any
CA). There are no requirements on an issuer of PERSONA certificates
to maintain any other records that might bind the true identity of
the subject to his certificate. However, a CA issuing such

certificates must establish procedures (not specified in this
document) in order to allow the holder of a PERSONA certificate to
request that his certificate be revoked (i.e., listed on a CRL).

As an example, a PERSONA user certificate might include a subject DN
of the form: C = "US" SP = "Massachusetts" L = "Boston" O =
"Pseudonyms R US" CN = "Paul Revere." The issuer of this certificate
might have a DN of the form: C = "US" SP = "Massachusetts" L =
"Boston" O = "Pseudonyms R US". Note the differences between this
PERSONA user certificate for "Paul Revere" and the corresponding
residential user certificate for the same common name.

3.4.4.4 CA Responsibilities for CRL Management

As X.500 directory servers become available, CRLs should be
maintained and accessed via these servers. However, prior to
widespread deployment of X.500 directories, this document adopts some
additional requirements for CRL management by CAs and PCAs. As per
X.509, each CA is required to maintain a CRL (in the format specified
by this document in Appendix A) which contains entries for all
certificates issued and later revoked by the CA. Once a certificate
is entered on a CRL it remains there until the validity interval
expires. Each PCA is required to maintain a CRL for revoked CA
certificates within its domain. The interval at which a CA issues a
CRL is not fixed by this document, but the PCAs may establish minimum
and maximum intervals for such issuance.

As noted earlier, each PCA will provide access to a database
containing CRLs issued by the IPRA, PCAs, and all CAs. In support of
this requirement, each CA must supply its current CRL to its PCA in a
fashion consistent with CRL issuance rules imposed by the PCA and
with the next scheduled issue date specified by the CA (see Section
3.5.1). CAs may distribute CRLs to subordinate UAs using the CRL
processing type available in PEM messages (see RFC 1421). CAs also
may provide access to CRLs via the database mechanism described in
RFC 1424 and alluded to immediately above.

3.5 Certificate Revocation

3.5.1 X.509 CRLs

X.509 states that it is a CA's responsibility to maintain: "a time-
stamped list of the certificates it issued which have been revoked."
There are two primary reasons for a CA to revoke a certificate, i.e.,
suspected compromise of a private component (invalidating the
corresponding public component) or change of user affiliation
(invalidating the DN). The use of Certificate Revocation Lists
(CRLs) as defined in X.509 is one means of propagating information

relative to certificate revocation, though it is not a perfect
mechanism. In particular, an X.509 CRL indicates only the age of the
information contained in it; it does not provide any basis for
determining if the list is the most current CRL available from a
given CA.

The proposed architecture establishes a format for a CRL in which not
only the date of issue, but also the next scheduled date of issue is
specified. Adopting this convention, when the next scheduled issue
date arrives a CA (Throughout this section, when the term "CA" is
employed, it should be interpreted broadly, to include the IPRA and
PCAs as well as organizational, residential, and PERSONA CAs.) will
issue a new CRL, even if there are no changes in the list of entries.
In this fashion each CA can independently establish and advertise the
frequency with which CRLs are issued by that CA. Note that this does
not preclude CRL issuance on a more frequent basis, e.g., in case of
some emergency, but no system-wide mechanisms are architected for
alerting users that such an unscheduled issuance has taken place.
This scheduled CRL issuance convention allows users (UAs) to
determine whether a given CRL is "out of date," a facility not
available from the (1988) X.509 CRL format.

The description of CRL management in the text and the format for CRLs
specified in X.509 (1988) are inconsistent. For example, the latter
associates an issuer distinguished name with each revoked certificate
even though the text states that a CRL contains entries for only a
single issuer (which is separately specified in the CRL format). The
CRL format adopted for PEM is a (simplified) format consistent with
the text of X.509, but not identical to the accompanying format. The
ASN.1 format for CRLs used with PEM is provided in Appendix A.

X.509 also defines a syntax for the "time-stamped list of revoked
certificates representing other CAs." This syntax, the
"AuthorityRevocationList" (ARL) allows the list to include references
to certificates issued by CAs other than the list maintainer. There
is no syntactic difference between these two lists except as they are
stored in directories. Since PEM is expected to be used prior to
widespread directory deployment, this distinction between ARLs and
CRLs is not syntactically significant. As a simplification, this
document specifies the use the CRL format defined below for
revocation both of user and of CA certificates.

3.5.2 PEM CRL Format

Appendix A contains the ASN.1 description of CRLs specified by this
document. This section provides an informal description of CRL
components analogous to that provided for certificates in Section
3.3.

    1. signature (signature algorithm ID and parameters)

    2. issuer

    3. last update

    4. next update

    5. revoked certificates

The "signature" is a data item completely analogous to the signature
data item in a certificate. Similarly, the "issuer" is the DN of the
CA which signed the CRL. The "last update" and "next update" fields
contain time and date values (UTCT format) which specify,
respectively, when this CRL was issued and when the next CRL is
scheduled to be issued. Finally, "revoked certificates" is a
sequence of ordered pairs, in which the first element is the serial
number of the revoked certificate and the second element is the time
and date of the revocation for that certificate.

The semantics for this second element are not made clear in X.509.
For example, the time and date specified might indicate when a
private component was thought to have been compromised or it may
reflect when the report of such compromise was reported to the CA.

For uniformity, this document adopts the latter convention, i.e., the
revocation date specifies the time and date at which a CA formally
acknowledges a report of a compromise or a change or DN attributes.
As with certificates, it is recommended that the UTCT values be of no
finer granularity than minutes and that all values be stated in terms
of Zulu.

3.6 Certificate Validation

3.6.1 Validation Basics

Every UA must contain the public component of the IPRA as the root
for its certificate validation database. Public components
associated with PCAs must be identified as such, so that the
certificate validation process described below can operate correctly.
Whenever a certificate for a PCA is entered into a UA cache, e.g., if
encountered in a PEM message encapsulated header, the certificate
must NOT be entered into the cache automatically. Rather, the user
must be notified and must explicitly direct the UA to enter any PCA
certificate data into the cache. This precaution is essential
because introduction of a PCA certificate into the cache implies user
recognition of the policy associated with the PCA.

Validating a certificate begins with verifying that the signature
affixed to the certificate is valid, i.e., that the hash value
computed on the certificate contents matches the value that results
from decrypting the signature field using the public component of the
issuer. In order to perform this operation the user must possess the
public component of the issuer, either via some integrity-assured
channel, or by extracting it from another (validated) certificate.
In order to rapidly terminate this recursive validation process, we
recommend each PCA sign certificates for all CAs within its domain,
even CAs which are certified by other, superior CAs in the
certification hierarchy.

The public component needed to validate certificates signed by the
IPRA is made available to each user as part of the registration or
via the PEM installation process. Thus a user will be able to
validate any PCA certificate immediately. CAs are certified by PCAs,
so validation of a CA certificate requires processing a validation
path of length two. User certificates are issued by CAs (either
immediately subordinate to PCAs or subordinate to other CAs), thus
validation of a user certificate may require three or more steps.
Local caching of validated certificates by a UA can be used to speed
up this process significantly.

Consider the situation in which a user receives a privacy enhanced
message from an originator with whom the recipient has never
previously corresponded, and assume that the message originator
includes a full certification path in the PEM message header. First
the recipient can use the IPRA's public component to validate a PCA
certificate contained in an Issuer-Certificate field. Using the
PCA's public component extracted from this certificate, the CA
certificate in an Issuer-Certificate field also can be validated.
This process cam be repeated until the certificate for the
originator, from the Originator-Certificate field, is validated.

Having performed this certificate validation process, the recipient
can extract the originator's public component and use it to decrypt
the content of the MIC-Info field. By comparing the decrypted
contents of this field against the MIC computed locally on the
message the user verifies the data origin authenticity and integrity
of the message. It is recommended that implementations of privacy
enhanced mail cache validated public components (acquired from
incoming mail) to speed up this process. If a message arrives from
an originator whose public component is held in the recipient's cache
(and if the cache is maintained in a fashion that ensures timely
incorporation of received CRLs), the recipient can immediately employ
that public component without the need for the certificate validation
process described here. (For some digital signature algorithms, the
processing required for certificate validation is considerably faster

than that involved in signing a certificate. Use of such algorithms
serves to minimize the computational burden on UAs.)

3.6.2 Display of Certificate Validation Data

PEM provides authenticated identities for message recipients and
originators expressed in the form of distinguished names. Mail
systems in which PEM is employed may employ identifiers other than
DNs as the primary means of identifying recipients or originators.
Thus, in order to benefit from these authentication facilities, each
PEM implementation must employ some means of binding native mail
system identifiers to distinguished names in a fashion which does not
undermine this basic PEM functionality.

For example, if a human user interacts directly with PEM, then the
full DN of the originator of any message received using PEM should be
displayed for the user. Merely displaying the PEM-protected message
content, containing an originator name from the native mail system,
does not provide equivalent security functionality and could allow
spoofing. If the recipient of a message is a forwarding agent such
as a list exploder or mail relay, display of the originator's DN is
not a relevant requirement. In all cases the essential requirement
is that the ultimate recipient of a PEM message be able to ascertain
the identity of the originator based on the PEM certification system,
not on unauthenticated identification information, e.g., extracted
from the native message system.

Conversely, for the originator of an ENCRYPTED message, it is
important that recipient identities be linked to the DNs as expressed
in PEM certificates. This can be effected in a variety of ways by
the PEM implementation, e.g., by display of recipient DNs upon
message submission or by a tightly controlled binding between local
aliases and the DNs. Here too, if the originator is a forwarding
process this linkage might be effected via various mechanisms not
applicable to direct human interaction. Again, the essential
requirement is to avoid procedures which might undermine the
authentication services provided by PEM.

As described above, it is a local matter how and what certification
information is displayed for a human user in the course of submission
or delivery of a PEM message. Nonetheless all PEM implementations
must provide a user with the ability to display a full certification
path for any certificate employed in PEM upon demand. Implementors
are urged to not overwhelm the user with certification path
information which might confuse him or distract him from the critical
information cited above.

3.6.3 Validation Procedure Details

Every PEM implementation is required to perform the following
validation steps for every public component employed in the
submission of an ENCRYPTED PEM message or the delivery of an
ENCRYPTED, MIC-ONLY, or MIC-CLEAR PEM message. Each public component
may be acquired from an internal source, e.g., from a (secure) cache
at the originator/recipient or it may be obtained from an external
source, e.g., the PEM header of an incoming message or a directory.
The following procedures applies to the validation of certificates
from either type of source.

Validation of a public component involves constructing a
certification path between the component and the public component of
the IPRA. The validity interval for every certificate in this path
must be checked. PEM software must, at a minimum, warn the user if
any certificate in the path fails the validity interval check, though
the form of this warning is a local matter. For example, the warning
might indicate which certificate in the path had expired. Local
security policy may prohibit use of expired certificates.

Each certificate also must be checked against the current CRL from
the certificate's issuer to ensure that revoked certificates are not
employed. If the UA does not have access to the current CRL for any
certificate in the path, the user must be warned. Again, the form of
the warning is a local matter. For example, the warning might
indicate whether the CRL is unavailable or, if available but not
current, the CRL issue date should be displayed. Local policy may
prohibit use of a public component which cannot be checked against a
current CRL, and in such cases the user should receive the same
information provided by the warning indications described above.

If any revoked certificates are encountered in the construction of a
certification path, the user must be warned. The form of the warning
is a local matter, but it is recommended that this warning be more
stringent than those previously alluded to above. For example, this
warning might display the issuer and subject DNs from the revoked
certificate and the date of revocation, and then require the user to
provide a positive response before the submission or delivery process
may proceed. In the case of message submission, the warning might
display the identity of the recipient affected by this validation
failure and the user might be provided with the option to specify
that this recipient be dropped from recipient list processing without
affecting PEM processing for the remaining recipients. Local policy
may prohibit PEM processing if a revoked certificate is encountered
in the course of constructing a certification path.

Note that in order to comply with these validation procedures, a

certificate cache must maintain all of the information contained in a
certificate, not just the DNs and the public component. For example
the serial number and validity interval must be associated with the
cache entry to comply with the checks described above. Also note
that these procedures apply to human interaction in message
submission and delivery and are not directly applicable to forwarding
processes. When non human interaction is involved, a compliant PEM
implementation must provide parameters to enable a process to specify
whether certificate validation will succeed or fail if any of the
conditions arise which would result in warnings to a human user.

Finally, in the course of validating certificates as described above,
one additional check must be performed: the subject DN of every
certificate must be subordinate to the certificate issuer DN, except
if the issuer is the IPRA or a PCA (hence another reason to
distinguish the IPRA and PCA entries in a certificate cache). This
requirement is levied upon all PEM implementations as part of
maintaining the certification hierarchy constraints defined in this
document. Any certificate which does not comply with these
requirements is considered invalid and must be rejected in PEM
submission or delivery processing. The user must be notified of the
nature of this fatal error.

A.   Appendix A: ASN.1 Syntax for Certificates and CRLs

A.1 Certificate Syntax

   The X.509 certificate format is defined by the following ASN.1
   syntax:

   Certificate ::= SIGNED SEQUENCE{
           version [0]   Version DEFAULT v1988,
           serialNumber   CertificateSerialNumber,
           signature   AlgorithmIdentifier,
           issuer   Name,
           validity   Validity,
           subject   Name,
           subjectPublicKeyInfo   SubjectPublicKeyInfo}

   Version ::=   INTEGER {v1988(0)}

   CertificateSerialNumber ::=   INTEGER

   Validity ::=   SEQUENCE{
           notBefore   UTCTime,
           notAfter   UTCTime}

   SubjectPublicKeyInfo ::=   SEQUENCE{
           algorithm   AlgorithmIdentifier,
           subjectPublicKey   BIT STRING}

   AlgorithmIdentifier ::= SEQUENCE{
           algorithm   OBJECT IDENTIFIER,
           parameters   ANY DEFINED BY algorithm OPTIONAL}

   The components of this structure are defined by ASN.1 syntax defined
   in the X.500 Series Recommendations. RFC 1423 provides references
   for and the values of AlgorithmIdentifiers used by PEM in the
   subjectPublicKeyInfo and the signature data items. It also describes
   how a signature is generated and the results represented. Because
   the certificate is a signed data object, the distinguished encoding
   rules (see X.509, section 8.7) must be applied prior to signing.

A.2 Certificate Revocation List Syntax

   The following ASN.1 syntax, derived from X.509 and aligned with the
   suggested format in recently submitted defect reports, defines the
   format of CRLs for use in the PEM environment.

   CertificateRevocationList ::= SIGNED SEQUENCE{
           signature   AlgorithmIdentifier,
           issuer   Name,
           lastUpdate   UTCTime,
           nextUpdate   UTCTime,
           revokedCertificates
                            SEQUENCE OF CRLEntry OPTIONAL}

   CRLEntry ::= SEQUENCE{
           userCertificate SerialNumber,
           revocationDate UTCTime}

References

   [1] CCITT Recommendation X.411 (1988), "Message Handling Systems:
       Message Transfer System: Abstract Service Definition and
       Procedures".

   [2] CCITT Recommendation X.509 (1988), "The Directory -
       Authentication Framework".

   [3] CCITT Recommendation X.520 (1988), "The Directory - Selected
       Attribute Types".

   [4] NIST Special Publication 500-183, "Stable Agreements for Open
       Systems Interconnection Protocols," Version 4, Edition 1,
       December 1990.

   [5] North American Directory Forum, "A Naming Scheme for c=US", RFC
       1255, NADF, September 1991.

   [6] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part
       I: Message Encryption and Authentication Procedures", RFC 1421,
       DEC, February 1993.

   [7] Balenson, D., "Privacy Enhancement for Internet Electronic Mail:
       Part III: Algorithms, Modes, and Identifiers", RFC 1423, TIS,
       February 1993.

   [8] Balaski, B., "Privacy Enhancement for Internet Electronic Mail:
       Part IV: Notary, Co-Issuer, CRL-Storing and CRL-Retrieving
       Services", RFC 1424, RSA Laboratories, February 1993.

   [9] North American Directory Forum, "NADF Standing Documents: A Brief
       Overview", RFC 1417, NADF, February 1993.

Patent Statement

   This version of Privacy Enhanced Mail (PEM) relies on the use of
   patented public key encryption technology for authentication and
   encryption. The Internet Standards Process as defined in RFC 1310
   requires a written statement from the Patent holder that a license
   will be made available to applicants under reasonable terms and
   conditions prior to approving a specification as a Proposed, Draft or
   Internet Standard.

   The Massachusetts Institute of Technology and the Board of Trustees
   of the Leland Stanford Junior University have granted Public Key
   Partners (PKP) exclusive sub-licensing rights to the following
   patents issued in the United States, and all of their corresponding
   foreign patents:

      Cryptographic Apparatus and Method
      ("Diffie-Hellman")............................... No. 4,200,770

      Public Key Cryptographic Apparatus
      and Method ("Hellman-Merkle").................... No. 4,218,582

      Cryptographic Communications System and
      Method ("RSA")................................... No. 4,405,829

      Exponential Cryptographic Apparatus
      and Method ("Hellman-Pohlig").................... No. 4,424,414

   These patents are stated by PKP to cover all known methods of
   practicing the art of Public Key encryption, including the variations
   collectively known as El Gamal.

   Public Key Partners has provided written assurance to the Internet
   Society that parties will be able to obtain, under reasonable,
   nondiscriminatory terms, the right to use the technology covered by
   these patents. This assurance is documented in RFC 1170 titled
   "Public Key Standards and Licenses". A copy of the written assurance
   dated April 20, 1990, may be obtained from the Internet Assigned
   Number Authority (IANA).

   The Internet Society, Internet Architecture Board, Internet
   Engineering Steering Group and the Corporation for National Research
   Initiatives take no position on the validity or scope of the patents
   and patent applications, nor on the appropriateness of the terms of
   the assurance. The Internet Society and other groups mentioned above

have not made any determination as to any other intellectual property
rights which may apply to the practice of this standard. Any further
consideration of these matters is the user's own responsibility.

Security Considerations

This entire document is about security.

Author's Address

Steve Kent
BBN Communications
50 Moulton Street
Cambridge, MA 02138

Phone: (617) 873-3988
EMail: kent@BBN.COM