WORKING DRAFT
TWG-98-31

W. E. Burr
3 June 1998

Public Key Infrastructure (PKI) Technical Specifications: Part A - Technical Concept of Operations

Non-Federal
Federal

Editor's Note: This is a draft of the CONOPS that incorporates extensive changes. In particular it introduces the "Bridge CA architecture" which was discussed in the May meeting, and this has resulted in extensive changes throughout the document. It also incoporates material on browser PKI clients, that was discussed in previous TWG meetings, and attempts to flesh out somewhat the use of OCSP responders in the Federal PKI. Significant issues remain about the plan for multi-algorithm interoperation in the Bridge CA architecture.

Please address comments to: william.burr@nist.gov

Table of Contents

Public Key Infrastructure (PKI) Technical Specifications (Version 3): Part C - Concept of Operations

1. Introduction

1.1 Purpose of the FPKI

The Federal Public Key Infrastructure (PKI) will support secure Federal Government use of information resources and the National Information Infrastructure (NII). The Federal PKI will establish the facilities, specifications and policies needed by Federal departments and agencies to use public key based certificates for information system security, electronic commerce, secure communications, and E-mail with each other as well as with entities of other branches of the Federal Government, state and local governments, business and the public. The Federal PKI will facilitate secure communications and information processing for unclassified but sensitive (UBS) applications.

PKI support for secure communications with business, the general public, other branches of the Federal Government and state and local governments is vital. The Federal PKI does not focus inward to secure communications and information systems only within a closed Government community, rather its purpose is to provide Federal users secure information access and communications with the entire nation and the rest of the world, as well as to provide secure internal Federal Government information access and communications.

The Federal PKI will not be a monolithic top down structure, rather it will be created largely from the bottom up. That is, Federal efforts to use public key cryptography generally begin with individual applications within agencies that provide immediate support for vital agency programs, and that yield an immediate return to the agency in terms of either improved effectiveness, or cost savings, or both. These implementations are paid for largely out of program funds, not funded as a centralized government PKI. Yet great benefits can result from melding together the separate systems that use certificates into a broader, coherent Federal PKI, that is itself a part of a larger national or global PKI. Just as communications networks become more useful as they encompass more stations, so also will the “trust network” of a PKI as it allows secure communications and networking with a broader, global community.

The great challenge of the Federal PKI, therefore, is to meld the individual agency projects that use PKI technology, from a variety of commercial vendors, into a broadly interoperable trust network, where the whole is, in fact greater, than the sum of the parts.

1.2 Basic Concepts

Digital signatures are based on the concept of a public-private key pair. A signatory, Alice, has a private key which she must keep a secret, and an associated public key, which is made public. Given the public key, it is infeasible to find the private key. A digital document (a message or file) is signed by Alice with her private key, and the signature may be verified, by a *relying party*, with Alice’s public key. Digital signatures can provide three important security services:

- *integrity*: any change to a signed document will cause the signature verification to fail;

- *source non-repudiation*: since only Alice knows her private key, only she can sign a document with it;

- *authentication*: Bob, the relying party, can authenticate Alice's identity, if he knows her public key, by having her sign a challenge. If the signature can be verified with Alice's public key, it must have been signed by Alice.

In addition, public key technology can provide confidentiality to Bob and Alice:

- Bob can encrypt a message with Alice's public key, which can only be decrypted her private key, or:

- Bob can encrypt a traffic key in Alice's encryption public key and send it to her. She can then decrypt the traffic key with her private key, and Bob and Alice can then use that traffic key with a symmetric key encryption algorithm to encrypt a message between them. Symmetric key cryptography is commonly used to encrypt bulk data because it is generally much faster than public key encryption.

Public key certificates are digital documents that, at a minimum, contain the name and public key of a user, and are digitally signed by a certification authority (CA). The purpose of a certificate is to reliably associate a users name and public key. Bob, trusting the CA and knowing its public key, may reliably authenticate Alice's public key from her certificate by verifying the signature on the certificate.

In addition there are registration authorities, certificate status responders, and management authorities in the federal PKI. A *registration authorities (RA)*, does not itself issue certificates,

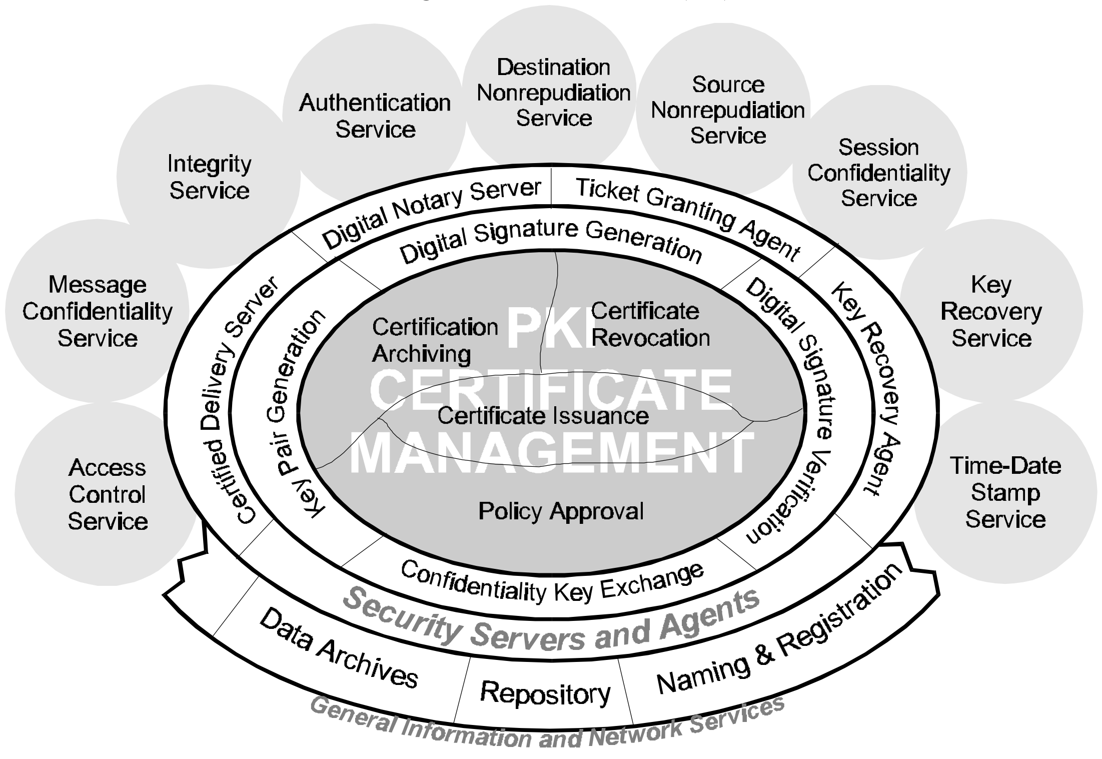
Integrity Service
Authentication Service
Destination Nonrepudiation Service
Source Nonrepudiation Service
Session Confidentiality Service
Message Confidentiality Service
Key Recovery Service
Access Control Service
Time-Date Stamp Service
Digital Notary Server
Ticket Granting Agent
Certified Delivery Server
Key Recovery Agent
Digital Signature Generation
Key Pair Generation
Digital Signature Verification
Confidentiality Key Exchange
PKI CERTIFICATE MANAGEMENT
Certification Archiving
Certificate Revocation
Certificate Issuance
Policy Approval
Security Servers and Agents
Data Archives
Repository
Naming & Registration
General Information and Network Services

Figure 1 - A PKI Centered View of Security Services

but registers or vouches for the identity of end entities (users) to a CA that then issues them a certificate. *Certificate status responders* are trusted online servers that act for the CA to provide authenticated certificate status information to relying parties. A *Policy Management Authority (PMA)* approves or coordinate the policies used to operate CAs and RAs and to issue certificates.

Figure 1 shows a "PKI-centric" view of certificate-base security services. The security services enabled by a PKI are shown as circles around the periphery of the PKI and the underlying information, security servers that support them. At the center is the core PKI, which is concerned with issuing and managing certificates. The first ring is the PKI clients that use the certificates, and the next ring shows various security specific servers and agents that provide or facilitate various security services. An incomplete outer ring of general information and communications services is shown at the bottom, to highlight some general services that, while not specifically security related, are important to the security services. The outermost ring of "bubbles" represent services that can be realized through the core PKI, the clients and the agents and servers.

The core Federal PKI consists of CAs, RAs, certificate status responders, and management authorities that manage public key certificates used by Federal departments and agencies for unclassified, sensitive applications. The core Federal PKI will:

1. issue public key certificates;
2. revoke public key certificates when required;
3. establish the policies that govern the issuance and revocation of certificates;
4. archive the information needed to validate the certificates at a later date;

PKI Clients will use the public key certificates that are issued and managed by the PKI to provide security services to Federal users. PKI clients perform four primitive functions with and for certificates:

1. public-private key pair generation;
2. digital signature generation;
3. digital signature verification;
4. confidentiality key management (i.e., agreement or distribution of a session or message key to be used with a symmetric key algorithm for confidentiality).

The first function, key pair generation, can also be performed by CAs or RAs, but generation of digital signature key pairs by the clients helps to maintain the integrity of the system and preserve nonrepudiation, since only the client then ever possesses the private key used for digital signatures. The last three functions enable a variety of public key based security protocols and services by direct client to client protocols, including authentication, access control based on an authenticated identity or role, nonrepudiation services, and confidentiality.

A number of servers and agents support the infrastructure and clients and clients may obtain security services from these servers and agents. The full range of servers and agents that may be needed is not fully understood, and is still evolving, but they may include:

- *digital notaries*: A digital notary provides a service analogous to a notary public; A notary may also provides a trusted date and time stamp for a document, that proves that it existed at a point in time and, may also verify the signatures on a signed document;

- *key recovery agents*: CAs may require that a copy of private keys used for confidentiality key management be turned over to a key recovery agent (KRA) as a condition for issuing a key management certificate. Alternatively, a client that sends an encrypted message, may include the encryption session key encrypted in the public key of the KRA, with the message. The purpose of the KRA is to allow decryption of encrypted data when keys are lost, or for management supervision or law enforcement purposes. It may be useful to separate the CA and key recovery functions because, while the only secret a CA inherently need protect or access is its own private key, a key recovery agent may need to store, protect, and provide carefully controlled access to a large number of user private keys. KRAs may provide for split control of the user private keys it holds, so that the cooperation of two or more agents is required to access the keys. This applies only to key management key pairs; private keys used for digital signatures should ordinarily never be divulged by certificate holders to any other party;

- *certified delivery agents*: These servers provide a destination non-repudiation service, analogous to certified mail or process servers, to prove that a message was received by a possibly uncooperative recipient, or that a good faith attempt was made to deliver the message;

- *ticket granting agents*: These agents provide cryptographic digital "tickets" that can be used for access to systems or data. They can use either public key or symmetric key cryptography, and provide a means of centralizing and managing access control in distributed systems.

Three general information and network services are of particular significance to the PKI:

- *repositories:* A repository is an on-line, publicly accessible system for storing and retrieving certificates and other information relevant to certificates, such as revocation information. In effect, a certificate is published by putting it in a repository. Repositories also contain Certification Practice Statements (see 7.2 below). In the Federal PKI the expected normal repository is a directory that uses the LDAP [RFC1777] "lightweight" directory access protocol, however other forms of repositories may be used including X.500 directories (that use the DAP directory access protocols) and World Wide Web or anonymous FTP servers;

- *data archives*: Archives provide a very long term repository for storing information. The life time of CAs may be relatively short. But it may be important to verify the validity of signatures on very old documents. CAs must make provision to store the information needed to verify the signatures of its users, in archives that will be able to make the data available at a much later date, perhaps centuries later;

- *naming and registration*: in a distributed environment, many objects must have unique names. This is true for security objects, for example certificate subjects and issuers, must have unique names.

2. PKI Services

The PKI will provide the services and facilities needed for UBS secure Federal information processing and use of the NII, including:

- digital signatures for:
  - ◊ authentication;
  - ◊ integrity;
  - ◊ nonrepudiation.
- management of symmetric keys for UBS level confidentiality for:
  - ◊ communications confidentiality session keys;
  - ◊ e-mail confidentiality message keys.

When extended by Key Recovery Agents, the PKI will also provide key recovery services for encrypted data.

The PKI will provide the services and facilities needed for secure information access, communication, messaging and electronic commerce with commercial and personal users employing common defacto and formal security standards and using mainstream commercial security products; The Federal PKI will be implemented primarily with ordinary COTS security products.

3. PKI Data Structures

Three basic data structures defined in the X.509 standard are used by the Federal PKI are the : *certificate*, *cross certificate pair* and *Certificate Revocation List (CRL)*.

3.1 Nomenclature and Typographical Conventions

Certificates, cross-certificates and CRLs are all directory attributes from the X.509 standard, and defined there in the ASN.1 syntax [X.509 97] . Formal names in ASN.1 are written without spaces and the separate words in the names are indicated by capitalizing the first letter of each word but the first. For example, the formal ASN.1 name of a cross-certificate is "crossCertificatePair," while the formal name of a certificate is "certificate." In this CONOPS, when it is useful to be very specific that a word means a particular formal directory attribute, as defined in X.509, they are shown in their ASN.1 form, set in boldface type, for example: **certificate**, **crossCertificatePair,** and **certificateRevocationList**. However, frequent use of this convention does not contribute to readability, and where it is not necessary to stress the formalism, "plain English" names are used in normal roman typeface.

3.2 Certificates

The Federal PKI will use X.509 v3 certificates. The structure of the X.509 v3 **certificate** is illustrated in Figure 2. In particular, Figure 2 shows how the certificate may be extended with optional extension fields. Table 1 gives more information about the use various certificate fields.

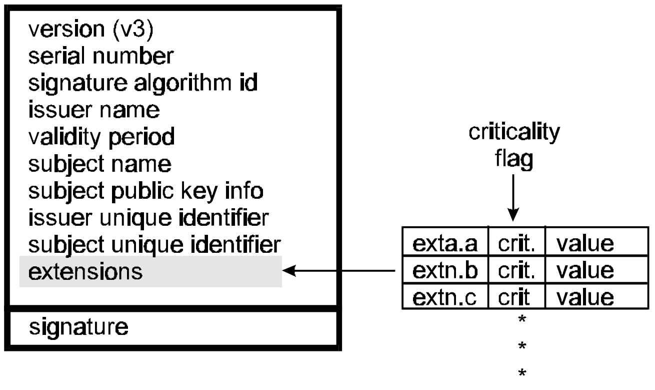

Figure 2 - Certificate Format

These certificates are digitally signed using the private key of the issuing CA. Certificates issued for Federal users will be signed using FIPS approved algorithms for digital signatures. A number of optional extension fields have been standardized [X.509 1997]. In addition, Federal specific extensions may be defined and incorporated in Federal certificates. Certificates may be stored in hardware based tokens or cryptographic modules. Certificates, however, in general, need not be protected and can be stored on any digital medium. The issuer's signature on the certificate itself is created from a hash (for Federal use the FIPS 180-1 Secure Hash Algorithm) of the body of the certificate, and the issuer's private key. In Table 1, the fields covered by the hash are shown in the shaded area.

Note that the FIPS approved Digital Signature Algorithm requires the establishment of three parameters. Syntactically there are three fields in the certificate that could contain parameters, how-

Table 1 - X.509 v3 Certificate

version	version number; an integer, value is "2" for version 3	
serial number	unique identifier for each certificate generated by issuer; integer	
signature	algorithm identifier	algorithm used to sign certificate
	parameters	should not be used
issuer	name of issuer (X.500 "distinguished name" that uniquely identifies a directory object),	
validity	notBefore	Time or
	notAfter	Time
subject	name of subject (X.500 "distinguished name")	
subject's public key info	algorithm identifier	subject's signature algorithm
	parameters	parameters applicable to subj. pub. key
	public key	subject's public key
issuer unique identifier	(optional) contains additional information about the subject; must be version 2 or higher - not used by the Federal PKI.	
subject unique identifier	(optional) contains additional information about the issuer; must be version 2 or higher - not used by the Federal PKI.	
extensions	(optional)	
issuer's signature	algorithm identifier	algorithm used for this signature
	parameters	should not be used
	ENCRYPTED (certificate hash)	

ever signatures should be validated using the parameters contained in the parameters value of the same subject public key info field as the public key used to validate the signature, or, if that parameters value is null, using the parameters "inherited" from the parameters used for the previous certificate validation in the certification path.

The optional extensions that have been defined for standardization include extensions that:

- identify the policies under which the certificate was issued;
- map equivalent policies in different domains;
- require subsequent certificates in a certification path to include specific policy identifiers, or inhibit policy mapping;
- limit the subject name space for subsequent certificates in the certification path;
- restrict key usage;
- limit the number of subsequent certificates in a certification path;
- distinguish between a CA certificate and an end-entity certificate.

Extensions may be labeled *critical*. A client must either be able to process all critical extensions contained in a certificate, or it must not validate that certificate. Most standardized extensions are “optionally critical,” and may be made critical or non-critical at the choice of a CA. A few are always non-critical or always critical, and some must be critical in particular situations. The Federal PKI will use these standardized extensions as needed and will require that all Federal certificates include certain extensions and that all clients be capable of processing certain extensions if they are present. Table 2 summarizes the standardized optional extensions to the X.509 certificate and their intended use in the FPKI. Additional material on certificate extensions is contained in [PROF 98].

The primary purpose of an X.509 certificate is to associate the subject’s public key and name. Public keys may be used to verify signatures, or to manage keys used with symmetric key algorithms for confidentiality. Separate certificates will be used in the Federal PKI for digital signatures and key management.

3.3 Cross-Certificates

CAs may cross-certify each other, that is each issue the other a certificate and combine the two certificates in a single directory attribute called a **crossCertificatePair**. The attribute **crossCertificatePair** supports chains of trust that run in both directions and are needed to implement trust models that begin at the CA which issued the users certificate, rather than trust models where trust originates from a common “root” CA.

A **crossCertificatePair**, includes two certificates, **forward** and **reverse**. The subject of the forward certificate is the issuer of the reverse certificate and vice-versa. When CA A cross certifies with CA B, in A’s crossCertificatePair attribute, A is the subject (and B is the issuer) of **forward**, and B is the subject (and A the issuer) of **reverse**. In B’s crossCertificatePair attribute, A is the issuer of **forward** and B is the issuer of **reverse**.

Table 2 - X.509 Standard Extensions and the FPKI

Extension	**Used By**	**Use**	**Critical** (see Note)
Key and Policy Information			
authorityKeyIdentifier	all	identifies the CA key used to sign this certificate	No
keyIdentifier	all	unique with respect to authority.	
authorityCertIssuer	all	identifies issuing authority of CA's certificate; alternative to key identifier	
authorityCertSerialNumber	all	used with authorityCertIssuer	
subjectKeyIdentifier	all	identifies different keys for same subject	No
keyUsage	all	defines allowed purposes for use of key (e.g., digital signature, key agreement...)	Yes*
privateKeyUsagePeriod	all	for digital signature keys only. Signatures on documents that purport to be dated outside the period are invalid.	Opt.
certificatePolicies	all	policy identifiers and qualifiers that identify and qualify the policies that apply to the certificate	Opt.
policyIdentifiers	all	the OID of a policy.	
policyQualifiers	all	more information about the policy	
policyMappings	CA	indicates equivalent policies	No
Certificate Subject and Issuer Attributes			
subjectAltName	all	used to list alternative names (e.g., rfc822 name, X.400 address, IP address...)	Opt.
issuerAltName	all	used to list alternative names	Opt.
subjectDirectoryAttributes	all	lists any desired attributes	Opt.
Certification Path Constraints			
basicConstraints	all	constraints on subject's role & path lengths	Yes*
cA	all	distinguish CA from end-entity cert.	
pathLenConstraint	CA	number of CAs that may follow in cert. path; 0 indicates that CA may only issue end-entity certs.	
nameConstraints	CA	limits subsequent CA cert. Name space.	Yes*
permittedSubtrees		names outside indicated subtrees are disallowed	
excludedSubtrees		indicates disallowed subtrees	
policyConstraints	all	constrains certs. Issued by subsequent CAs	Yes*
policySet	all	those policies to which constraints apply	
requireExplicitPolicy	all	All certs. Following in the cert. Path must contain an acceptable policy identifier	
inhibitPolicyMapping	all	prevent policy mapping in following certs.	
CRL Identification			
crlDistributionPoints	all	mechanism to divide long CRL into shorter lists	Opt.
distributionPoint	all	location from which CRL can be obtained	
reasons	all	reasons for cert. inclusion in CRL	
cRLIssuer	all	name of component that issues CRL.	

NOTE: " "No" means the standard requires the extension be noncritical if used, and "Opt." means that the issuing CA may choose to make that extension either critical or noncritical. "Yes*" means that the standard allows the field to be either critical or noncritical, but the recommendation for the Federal PKI is that it be set to critical. There are no v3 certificate extensions that are required by the standard to be critical.

Table 3 - X.509 v2 Certificate Revocation List

signature	algorithm identifier	algorithm used to sign CRL
	parameters	any parameters needed
issuer	name of CRL issuer (X.500 “distinguished name,” a sequence of Relative Distinguished Names that uniquely identify a directory object)	
this update	Time	update time stamp
next update	Time	optional time of next update
revoked certificates	list of revoked certificates	
CRL extensions (optional) zero or more extensions	criticality flag	if “true” extension must be processed
	extension parame- ters	
issuer’s signature		

serial number	serial number of revoked certificate (unique for the issuer)	
revocation date	Time	
crl entry extensions (optional) zero or more extensions	criticality flag	if “true” extension must be proc- essed
	extension parameter	

3.4 Certificate Revocation Lists

It is at times necessary to revoke certificates, for example when the certificate holder leaves the issuing organization or when the private key is compromised. The mechanism defined in X.509 for revoking certificates is the Certificate Revocation List (CRL). The X.509 v2 CRL is illustrated in Table 3. Alternatively, some CAs may use the emerging On Line Certificate Status Protocol (OCSP) [OCSP 98], with a certificate status responder, to provide relying parties with current certificate status information on individual certificates. OCSP is further discussed in 5.2.1 below.

3.4.1 CRL Data Structure

A CRL contains a list signed by a CA of unexpired certificates that have been revoked. Since the CRL contains the time it was issued, and the time the next CRL will be issued, a user can determine if a copy of the CRL is still current. Federal CAs will issue CRLs for the certificates they have issued at periodic intervals.

While it is anticipated that all Federal CAs will issue CRLs for historical purposes, if no other purpose, some Federal CAs may also provide an on-line certificate status service, via the emerging Online Certificate Status Protocol (OCSP) now being developed in the Internet Engineering Task

Table 4 - Summary of CRL Extensions

Extension	Use	Critical
authorityKeyIdentifier	identifies the CA key used to sign CRL.	No
keyIdentifier	unique key identifier; alternative to **certIssuer** & **authorityCertSerialNumber**	
certIssuer	name of CA's cert. issuer	
authorityCertSerialNumber	used with **certIssuer**; combination must be unique	
issuerAltName	alternate name of CRL issuer	No*
cRLNumber	sequence number for CRL	No
issuingDistributionPoint	name of CRL distribution point; also gives reasons for revocations contained in CRL.	Yes
deltaCRLIndicator	indicates delta CRL (lists certificates. revoked since last full CRL) & gives sequence number	Yes

NOTES:
* Standard allows either critical or noncritical. Indication is for use in FPKI.

Force [OCSP 98]. On-line certificate status checks provide a fresher check of certificate status than a CRL can, and may also provide a mechanism for entering into a relationship or agreement between the CA and a relying party.

Optional extensions are also defined for the X.509 CRL as a whole, and for each entry in the CRL. Table 4 summarizes the standardized extensions for the CRL, while Table 5 summarizes the standard extensions for the CRL entries. In addition, certain of the certificate extensions summarized in Table 2 (labeled " CRL Identification" in the table) allow information about CRLs, the issuer of the CRL, and where to get a copy of the CRL, to be included in the certificate itself.

3.4.2 CRL Distribution Points and Indirect CRLs

If a CA issues a large number of certificates, and if certificates are frequently revoked, perhaps because of personnel turnover, name changes, or reorganizations, then CRLs could become quite large. A relying party, then, needing to validate a single certificate, might be required to download a large CRL, with much information he does not need. One way to avoid this is with the **crlDistributionPoints** extension. This certificate extension specifies where the CRL for the certificate may be found, and allows the CA to partition the CRL space into smaller segments, reducing the

Table 5 - Summary of CRL Entry Extensions

Extension	Use	Critical
reasonCode	identifies the reason for the revocation of this certificate	No
instructionCode	used with **certificateHold reasonCode**; indicates action to be taken when encountering a held certificate	No
invalidityDate	date certificate became invalid	No
certificateIssuer	Issuer of revoked certificate in an indirect CRL	Yes

amount of data that a relying party needs to download to check the status of a particular certificate.

The default CRL issuer is the CA that issued the certificate. However, the certificate and CRL extensions allow a certificate to identify a second CA, other than the CA that issued the certificate, to be the CA that will issue a CRL applicable to that certificate. This indirect CRL may contain revocation information on certificates issued by many CAs. The data structures for distribution points and indirect CRLs are illustrated in Figure 3.

The FPKI will make use of these indirect CRLs to maintain an *Authority Revocation List (ARL)*, that is an indirect CRL that covers all the CA certificates in the Federal PKI, but not end entity certificates. It provides a single consolidated CRL for all CAs in the Federal PKI.

3.5 Attribute Certificates and Subject Directory Attributes

Attribute certificates are defined in [X9.45 97] and [X.509 1997]. Attribute certificates might, for example, indicate credit limits, authority to obligate the government, access privileges and so on. An attribute certificate is a signed by an attribute authority (AA) and is a digital document binding some attribute to either:

1. a name; or,

2. a specific X.509 digital signature certificate.

The holder of the attribute certificate then establishes his right to use the attribute by:

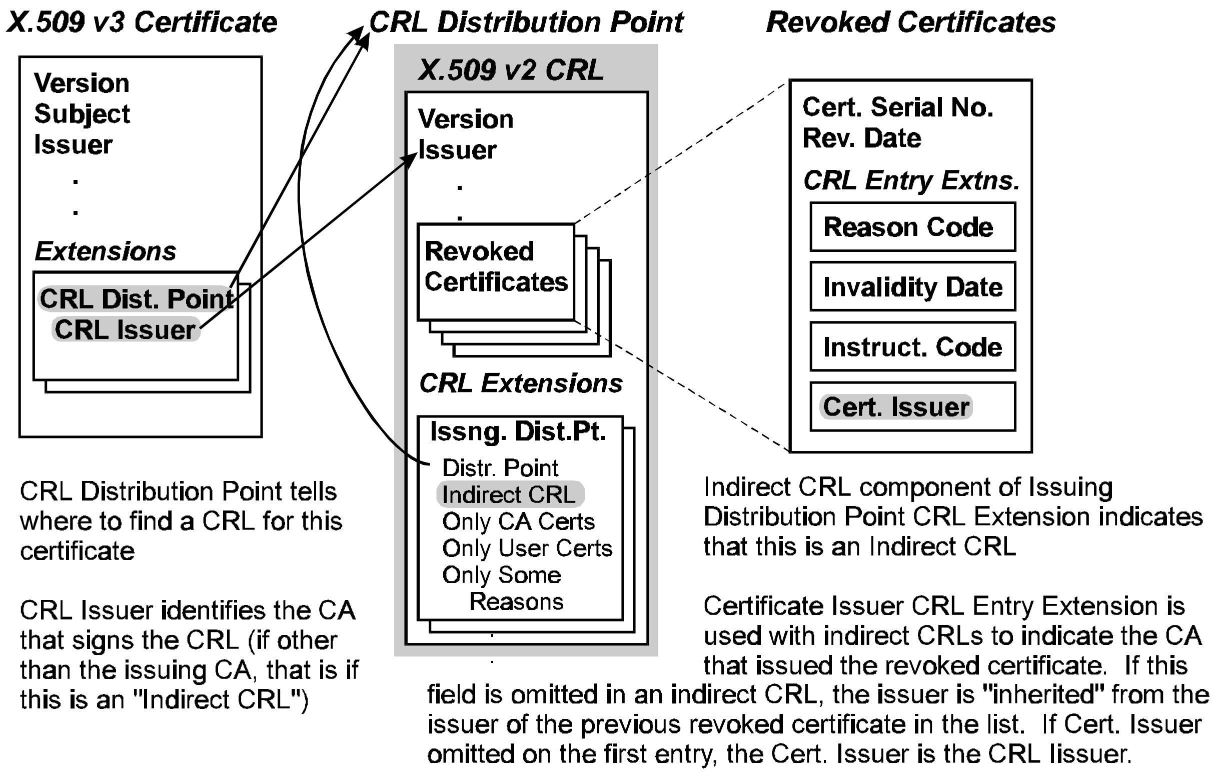
X.509 v3 Certificate
Version
Subject
Issuer
Extensions
CRL Dist. Point
CRL Issuer
CRL Distribution Point
X.509 v2 CRL
Version
Issuer
Revoked Certificates
CRL Extensions
Issng. Dist.Pt.
Distr. Point
Indirect CRL
Only CA Certs
Only User Certs
Only Some Reasons
Revoked Certificates
Cert. Serial No.
Rev. Date
CRL Entry Extns.
Reason Code
Invalidity Date
Instruct. Code
Cert. Issuer
CRL Distribution Point tells where to find a CRL for this certificate
CRL Issuer identifies the CA that signs the CRL (if other than the issuing CA, that is if this is an "Indirect CRL")
Indirect CRL component of Issuing Distribution Point CRL Extension indicates that this is an Indirect CRL
Certificate Issuer CRL Entry Extension is used with indirect CRLs to indicate the CA that issued the revoked certificate. If this field is omitted in an indirect CRL, the issuer is "inherited" from the issuer of the previous revoked certificate in the list. If Cert. Issuer omitted on the first entry, the Cert. Issuer is the CRL Iissuer.

Figure 3 - CRL Distribution Points and Indirect CRLs

1. the use of a public key certificate issued in the name specified in the attribute certificate; or,

2. possession of the key certified in the specified digital signature certificate.

Attribute certificates also allow the certification of attributes to be made by the entity that is responsible for the attribute, rather than a general CA. However, use of separate attribute certificates requires that the certification paths of both the attribute certificate and the subject's public key certificate be validated, which may significantly increase overhead.

Alternatively, the **subjectDiractoryAttributes** extension can be used to include in a certificate any desired directory attributes that have been defined for the subject. If attributes included in a digital signature certificate are changed, then it is necessary to revoke the certificate and issue a new certificate, to update the certificate. Therefore it is only appropriate to include subject attributes in a digital signature certificate when:

- the attribute is frequently needed when signatures are used; and,
- the attribute is not expected to change frequently.

Several attributes have been defined for this purpose for the Multilevel Information System Security Initiative (MISSI) [SDN.706]. They are summarized in Table 6. These attributes will be used in the Federal PKI wherever it is necessary to include hierarchical classification levels or citizenship information digital signature certificates and may be used as appropriate for their uses as well.

Table 6 - Federal Subject Directory Attributes

Attribute	Purpose
prbacInfo	conveys subjects security clearance, security categories and citizenship
prbacCAConstraints	constrains the authorizations that a CA may assign to a cert.
sigOrKMPrivileges	defines subject's signature & key management privileges
commPriviliges	defines communications precedence the subject may assign to messages

Use of attribute certificates in the Federal PKI is for further study.

4. PKI Certification Path Architecture

In a PKI trust is transferred by certification paths, that is along chains of certificates. A relying party verifies a signature by verifying the signatures on a chain of certificates from the public key of a CA that is trusted by the verifier.

4.1 Signatures and Certification Paths

Figure 4 depicts the relationship of a certificate and a signed document. The public key in the certificate is used to validate the signature of the signed document. Note that the signature itself does not identify the certificate for the public key used to sign the document. Rather the relying party must either know *a priori* which certificate to find the public key in, or must try all the keys of the certificates held by the signer until he finds one that validates. If no public key in any certificate validates the signature on the document, then the signature cannot be validated.

A certificate is itself a signed document. Assume that Alice wishes to validate the signature of a document signed by Bob. Alice has a CA which she trusts, and whose public key she knows. In the usual case, Alice knows this public key because she has been given a "self-signed" certificate containing the key by some authenticated, but "out-of band" process. Usually she gets it when she is given her own certificate. Bob holds a certificate issued by a different CA, whose public key Alice does not know. But that CA may itself have been issued a certificate by the CA Alice trusts, or there may exist a chain of certificates leading from the CA Alice trusts, to other CAs and ultimately to Bob's certificate. Such a chain, illustrated in Figure 5, is called a certification path, and Alice can start with the public key of CA1, that she knows, and successively validate the signatures on each of the certificates, until she reaches Bob's certificate.

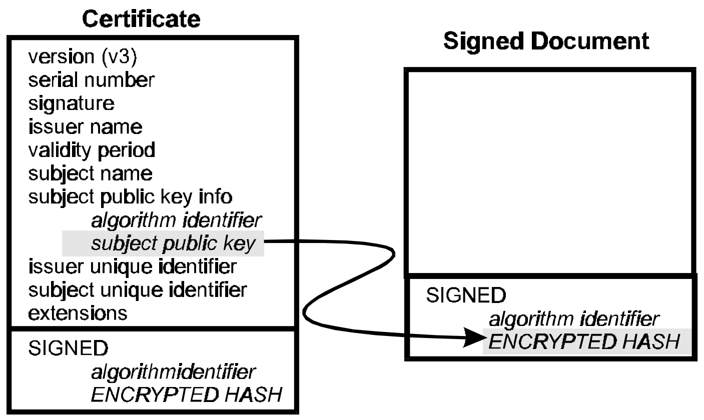

Figure 4 - Relationship of Certificate and Signature

Figure 5 shows a detailed view of the certification path, including the keys and the signatures, and a simplified view, which depicts CA keys as circles, end-entity keys as rectangles, and certificates as arrows from the signing key to the certified key. It is convenient to use this notation to draw a certification path topology for a PKI. But it is important to understand that what we are illustrating in such figures is really a trust topology by the certificates that CAs issued to other CAs, and to end-entities. It is not a communication path, and the fact that a node (really a key) is on many paths may indicate that it is an important trust node, but it does not indicate that the CA (which may not even be an on-line entity) is actively involved in the validation of the certification paths, or that it represents any kind of communications bottleneck.

4.2 Certification Path Architectures

CAs can issue certificates to each other in a systematic and ordered way or in a more flexible and less ordered way. In addition, current web browser products implement only a simplified "flat" PKI that does not make much use of certification path processing, in the general sense. The systematic, ordered topology of certification paths that is normally employed is a hierarchy, while the more general topology is a mesh of cross-certified CAs. The alternatives are illustrated in Figure 6 and described below:

- *Hierarchical*: Authorities are arranged hierarchically under a "root" CA that issues certificates to subordinate CAs. These CAs may issue certificates to CAs below them in the hierarchy, or to users. In a hierarchical PKI, the public key of the root CA is known to every user, and any user's certificate may be verified by verifying the certification path of certificates from the root CA. Alice verifies Bob's certificate, issued by CA 4, then CA 4's certificate, issued by CA 2, and then CA 2's certificate issued by CA 1, the root, whose public key she knows;

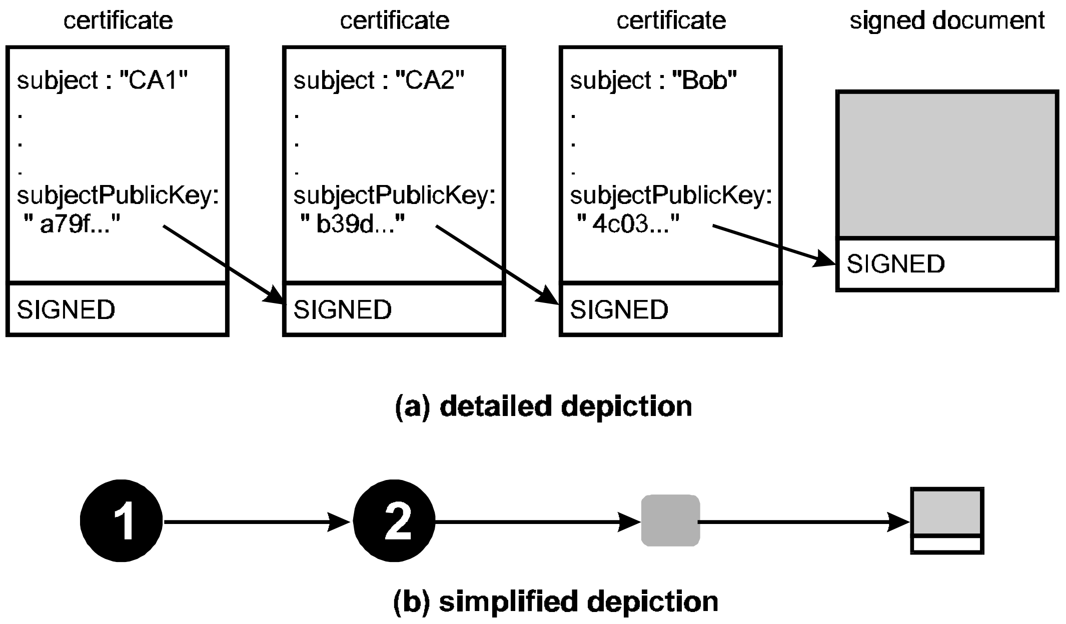

Figure 5 - Certification Path

- *Mesh*: Independent CA's cross certify each other (that is issue certificates to each other), resulting in a general mesh of trust relationships between peer CAs. Figure 6 (b) illustrates a mesh of authorities. A user knows the public key of a CA near himself, generally the one that issued his certificate, and verifies the certificates of other users by verifying a certification path of certificates that leads from that trusted CA. CAs cross certify with each other, that is they issue certificates to each other, and combine the two in a **crossCertificatePair**. So, for example, Alice knows the public key of CA 3, while Bob knows the public key of CA 4. There are several certification paths that lead from Bob to Alice, but the shortest requires Alice to verify Bob's certificate, issued by CA 4, then CA 4's certificate issued by CA 5 and finally CA 5's certificate, issued by CA 3. CA 3 is Alice's CA and she trusts CA 3 and knows its public key.

- *Browser Oriented*: Current web browser and server products are the most widely used PKI clients, however they lack a well developed certification path processing capability. Browsers contain a file of trusted self-signed certificates. In most cases, a certificate can be validated only if it is directly signed by one of the keys contained in the file of trusted CA certificates. This is illustrated in Figure 6 (c). While this approach can be effective for local applications, and might be workable on a broader scale if all other certificates were issued by one of a handful of "national" CAs, it does not appear to be sufficient to implement a broad Federal or national PKI.

Table 7 summarizes the characteristics of the three architectures.

4.2.1 Hierarchical Architecture

In the hierarchical architecture all relying parties base their trust on the key of a singe root CA, whose public key must be distributed in some authenticated fashion to all relying parties, to "bootstrap" trust in the PKI. Trust paths descend from the root through subordinate CAs. The hierarchical certification path architecture has some advantages:

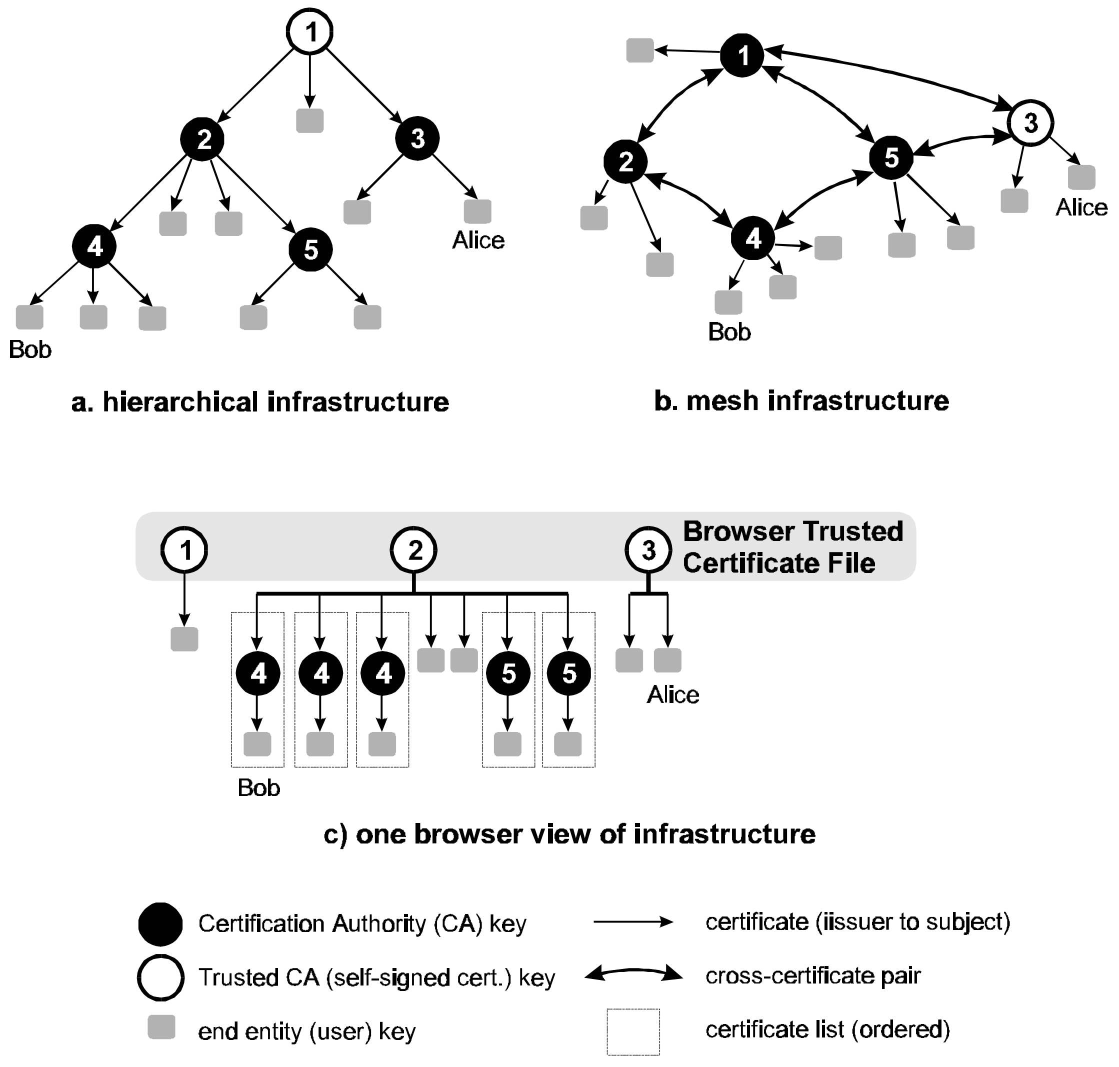
1
2
3
4
5
Bob
Alice
a. hierarchical infrastructure
b. mesh infrastructure
Browser Trusted Certificate File
c) one browser view of infrastructure
Certification Authority (CA) key
Trusted CA (self-signed cert.) key
end entity (user) key
certificate (iissuer to subject)
cross-certificate pair
certificate list (ordered)

Figure 6 -Different PKI Infrastructure Topologies

- The organizational management structure of many organizations such as the government is largely hierarchical. Trust relationships are frequently aligned with organizational structure, so it is natural to align the certification path with the organizational structure;
- The hierarchy may be aligned with hierarchical directory names;
- The strategy for searching for a certification path is straightforward;
- Important existing Federal PKI components are designed hierarchically;
- Each user has a certification path back to the root. The user can provide his path to any other user and any user can verify the path, since all users know the root's public key.

A strictly hierarchical certification path architecture also has some disadvantages:

- It is improbable that there will be a single root CA for the world PKI;
- Commercial and business trust relationships are not necessarily hierarchical;
- Compromise of the root private key is catastrophic and recovery requires the secure distribution of the new public key to every user.

Table 7 - A comparison of PKI Architecture Characteristics

Characteristics	*Network*	*Hierarchical*	*Browser*
Trusted key(s)	CA that issued user's certificate	"root" CA	file of (usually) many trusted CA certificates in each browser
Trust paths	mesh of bi-directional cross certificate pairs	chain of parent-child certificates	pre-ordered certificate-list
Trust path finding	directory based, complex	directory based, comparatively simple	minimal; find individual certificates in LDAP directory
Cross-certification	basis of PKI	may be supported	no direct capability
Extensions			
key usage	yes	yes	yes
basic constraints	yes	yes	yes
authority key id	yes	yes	yes
subject key id	yes	yes	yes
certificate policy	yes	usually	no
policy mapping	probably needed	less need	no
name constraint	?	?	no
Certificate Status	Certificate Revocation List	Certificate Revocation List	none (may add support for On-line Certificate Status Protocol in future)

Early attempts to design PKIs, such as the Internet Privacy Enhanced Mail (PEM) standard [RFC 1421, RFC 1422, RFC 1423], generally featured a hierarchical structure. A principle reason for this was to facilitate the management of security policies and trust relationships, that is, branches of the tree were aligned with security policies. The X.509 v3 certificate structure, however, introduces several extensions that allow the management of policies and trust relationships in a non-hierarchical PKI, and this rationale for a hierarchical PKI is no longer compelling.

4.2.2 Mesh Certification Path Architecture

In the mesh architecture, each relying party relies on the public key of one of the CAs in the PKI, generally the one which issued the relying party's certificate. Rather than superior-subordinate relationships between CAs, trust relationships are peer-to-peer: CAs exchange certificates to form cross certificate pairs. The mesh certification path architecture has some advantages:

- It is flexible, facilitates ad hoc associations and trusted relationships, and reflects the bilateral trust relationships of business;
- A user must trust at least the CA that issued its own certificate in any PKI, and it is reasonable to make this the foundation of all trust relationships;
- CAs that are organizationally remote, but whose users work together with a high degree of trust, can be directly cross-certified under a high trust policy that is not extended to other CAs and is higher than would be practical through a long, hierarchical chain of certificates;

- It allows direct cross-certification of CAs whose users communicate frequently, reducing certification path processing load;
- Recovery from the compromise of any CA's private key requires only that the new public key (and certificates signed with the corresponding new private key) be securely distributed to the holders of certificates from that CA.

The mesh PKI also has at least two disadvantages:

- Certification path search strategies can be more complex;
- A user cannot provide a single certification path that is guaranteed to enable verification of his signatures by all other users of the PKI.

4.3 Browser-oriented PKI

This is, in terms of the number users with clients that support it, undoubtedly the most widely implemented architecture. It relies on the fairly rudimentary client capabilities of the two nearly ubiquitous leading browser products. Each browser user has a local file of trusted self-signed "root" certificates. The browsers normally are distributed with an initial set of root certificates. Control of that file may be managed by the individual user, or organizations may have provision for loading or managing it from a central network management serve. Users can add to this set or delete certificates from it.

The certificate path processing capability of the web browsers is currently quite limited. For some applications, browsers can process an ordered certificate list leading from the sender of a message back to a certificate signed under one of the self-signed certificates in the trusted certificate file. The browsers have a limited capability to retrieve a certificate from an LDAP directory, as requested by the user, but no capability to automatically find an ordered, complete multi-step certification path. A few standardized extensions, including Key Usage and Basic Constraints are supported. Some private extensions for code signing and SSL may be supported by browser products. The X.509 name constraints, certificate policy and policy constraints extensions are not supported. The concept of cross-certificates is not directly supported.

The browsers can use certificates to sign, verify, encrypt and decrypt S/MIME email messages. They can use certificates to establish Secure Socket Layer (SSL) sessions. SSL is a transport level session encryption and authentication protocol that is being adopted by the IETF as the Transport Security Protocol (TSP). In an SSL session, users can, for example, submit forms to a server or receive information from a server in an encrypted, authenticated transport session. Browsers can also verify the signatures on signed code.

SSL is a client-server transport level protocol. Web server products from the browser vendors and other server vendors, then implement the SSL server functionality. A number of Certification Authority (that is certificate server) products are available that can issue client certificates to browsers and SSL server certificates to web servers. Several commercial CAs also offer certificates to browsers and servers over the Internet. LDAP products and some browser configuration management products complete the suite of currently available PKI tools for these systems.

Since certificate policies are not supported by browsers, each CA must issue certificates under a single policy. If an CA wants to distinguish between 'high," "medium" and "low" assurance certificates, then three separate CAs with separate CA keys are needed, and the appropriate self-signed CA certificates must be loaded into the browser trusted certificate files. This, however, is at best an awkward solution, since no particular tools are provided in the browsers to manage separate trusted certificate files, and select one, depending on the type of transaction being evaluated. Multiple users are supported for each browser, so, as a sort of kludge, one might create several browser users, each with a different set of trusted certificates. Perhaps, then, some rough equivelent to the X.509 initial policy set might be achieved.

These tools, however, suffice to build useful single-enterprise scale applications, where all the users have certificates from a single CA, and where all certificates subjects can be equally trusted. They can also allow some limited hierarchical structure, if all S/MIME users provide a certificate list back to a root CA, and all browsers are initialized to accept that CA. It is far from clear, however, that they are adequate for an "open" national or international PKI. The absence of direct support for cross certification and certificate policies limits the use of browsers for large-scale open PKI applications, as does their inability to automatically find certification paths.

The initialization and management of a file of possibly numerous "trusted CA certificates" in every workstation in a sizable agency or organization is a somewhat questionable proposition, as a way to manage enterprise trust relationships. It is not clear that there is any practical way to prevent either users, or others with access to their workstations, from making unauthorized alterations to this file. Certainly, the process of distributing an initial file of such self-signed certificates with browser products that are routinely downloaded over the Internet, is not a secure way to initialize the trusted keys. Moreover, with only two browser products of commercial consequence, the initial distribution certificate choices by browser vendors are bound to distort the market for open PKI services. In effect, the browser vendors choice of which self-signed certificates to include with their "out of the box" product may establish a kind of defacto CA accreditation.

The browser-oriented PKI architecture has several advantages:

- It is simple, straightforward and comparatively easy to implement;
- The end user has total control of the contents of the file of CAs he trusts;
- It would work well with a simple On-line Certificate Status Protocol (OCSP), since there is not a complex certification path to validate, rather the status request is addressed only to the CA in the trusted certificate file. (note that browsers do not now implement such a protocol);
- Certification paths are generally simple, which should speed processing.

The browser-oriented PKI architecture has several disadvantages:

- The browser user has total control of the contents of the file of CAs he trusts. It is hard to see how organizations can reliably manage this as organizations and prevent individual users from altering these files;
- The inclusion of a default list of trusted CA's in browsers as they come, out of the box, is hardly an authenticated process for initializing the known public keys upon which all else rests;
- The careful management of the potentially large file of trusted CA certificates may be too tedious and difficult for most end users;
- Little support is provided for finding certification paths;
- There is no direct support for cross-certificate pairs, limiting the role of CAs in managing trust;

- Certificate Policies are not supported, requiring that a Certification Authority entity operate several CAs, if it is to support different certificate policies and assurance levels. This in turn ads more Certificates to the browser's trusted certificate file;
- Browsers now support no automated method of verifying the status of certificates, nor of revoking them.

The most important advantage of the browser-oriented PKI is not inherent in the architecture: support for S/MIME and SSL is available out of the box, in current browser products. These browser clients are now ubiquitous. The World Wide Web and browsers are the dominant network client technology today, and the main platform for implementing distributed applications. Web servers also support the browser-oriented PKI model. Any PKI application, that hopes to reach the public at large, must find this existing ubiquitous capability compelling. Moreover, web technology is now the platform of choice for many intranet applications as well. Whatever the abstract merits of the PKI architecture implemented by browsers, the reality of their pervasive availability, and their key position, makes this an important architecture.

5. Federal PKI Architecture

There are now many more or less independent efforts in Federal agencies to set up independent CAs to support individual applications. In most cases, the cost of setting up and operating the CA is born by some application that supports the agency mission, such as purchasing, grants, or travel, etc. and the use of public key technology must be justified in terms of its direct benefit to that agency application. In other cases the government will use commercial CA service providers to issue certificates to the public, to facilitate delivering services to the public, and the cost will be born by the various agency projects that rely upon those certificates. Relatively little thought has generally been given to broader government-wide PKI needs, and the systems that are set up do not generally facilitate interagency operation, or the creation of a broader national PKI.

But "Metcalf's Law," that a network becomes more valuable as it reaches more users, must surely also apply to a PKI. It is apparent that there are great benefits to a system that propagates trust not just in the local environment, but throughout the entire Federal Government, the nation, and the world. Trust in a PKI propagates through certification paths. The main issue for the Federal PKI is: Given that many, often quite different, systems that use certificates are now being implemented by agencies, how do we create certification paths between them, in a sufficiently consistent and coherent fashion, to allow reasonably reliable and broad propagation of trust?

This CONOPS is uses a *Bridge CA (BCA)* that provides systematic certification paths between CAs in agencies, and outside the government. Federal CAs that meet certain standards and requirements will be eligible to cross-certify with the BCA, thereby gaining the certification paths needed for broad trust interoperation in the larger Federal and national PKI. While the certification path processing limitations of some less functional clients may confound interoperability at times, the existence of these certification paths is a necessary precondition for broad trust interoperation.

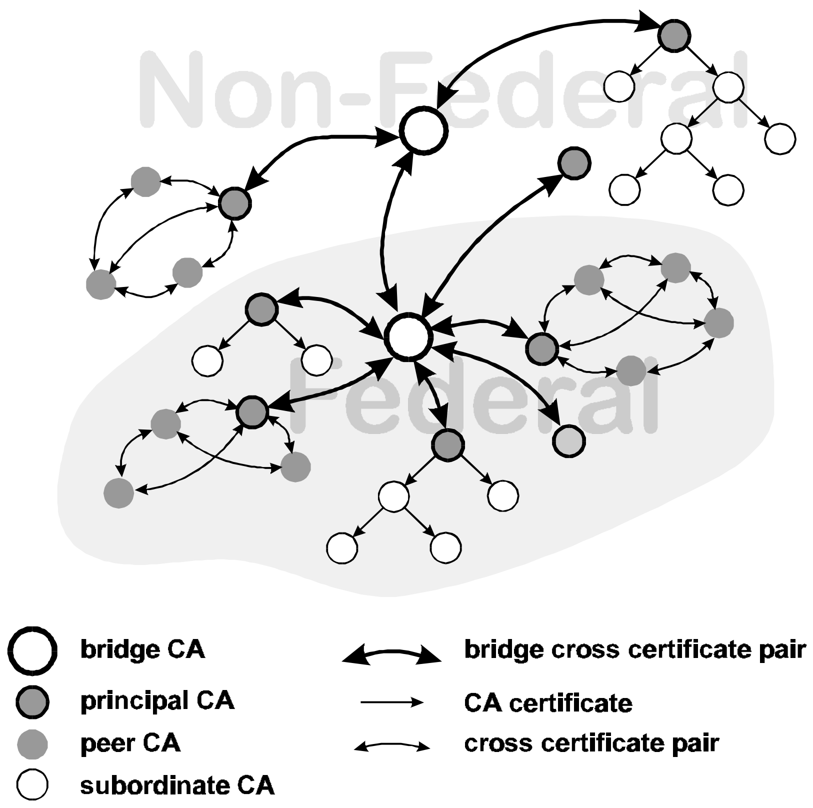

Figure 7 - Proposed Federal PKI Certification Path Architecture

5.1 *Architecture Components*

The certification path elements of the proposed architecture are illustrated in Figure 7. The complete architecture is composed of the following components:

- ***Federal Policy Management Authority (FPMA)***: this management authority sets the overall policies of the Federal PKI, and approves the policies and procedures of trust domains within the federal PKI. It operates a Federal Bridge CA, and repository;
- ***Trust Domains***: In the Federal context a trust domain is a portion of the Federal PKI that operates under the management of a single *policy management authority*. One or more Certification Authorities exist within the trust domain. Each trust domain has a single *principal CA*, but may have many other CAs. Each trust domain has a domain repository. In the non-Federal Context, trust domains, may be more loosely organized, but consist at a minimum of a group of CAs that share trust and operate under consistent policies.
- ***Domain Policy Management Authorities (DPMA)***: a policy management authority approves the certification practice statements of the CAs within a trust domain, and monitors their operation. The DPMAs operate or supervise a domain repository. In the non-federal context, a DPMA may be an association of CAs that share trust and use consistent or comparable CA policies;
- ***Certification Authorities (CA)***:
  - ◊ ***Bridge CA (BCA)***: the Federal Bridge CA is operated by the Federal Policy Management Authority. Its purpose is to be a bridge of trust that provide trust paths between the various trust domains of the Federal PKI, as well as between the Federal PKI and non-federal

trust domains. Trust domains that operate with policies and practices that are approved by the FPMA, designate a principal CA that is eligible to cross-certify with the Federal BCA. Note that the BCA is not a *root CA*, since it does not ordinarily begin certification paths. When the BCA cross certifies with CAs it may include **nameConstraints**, **pathLength-Constraints** or **policyConstraints** that limit the propagation of trust to other, cross-certified domains. The BCA also issues a consolidated Federal ARL;

◊ ***Principal CA***: A CA within a trust domain that cross-certifies with the Federal BCA. Each trust domain has one principal CA. In the case of a domain with hierarchical certification paths it will be the root CA of the domain. In the case of a domain with mesh certification paths, the principal CA may be any CA in the domain, however it will normally be one operated by, or associated with, the domain policy management authority;

◊ ***Peer CA***: A CA in a mesh domain, that has a self-signed certificate which is distributed to its certificate holders, and which is used by them to start certification paths. Peer CAs cross-certify with other CAs in their trust domain;

◊ ***Root CA***: In a hierarchical trust domain, the CA that starts all trust paths. In the hierarchical domain, certificate holders and relying parties are given the self-signed root CA certificate, by some authenticated, out-of-band means, and start all trust paths from that point. For hierarchical trust domains the root CA is also the principle CA for that domain;

◊ ***Subordinate CA***: A CA in a hierarchical domain that does not begin trust paths; rather trust starts from some root CA. In a hierarchical trust domain, a subordinate CA receives a certificate from it's superior CA, and may also have subordinate CAs of its own, to which it issues certificates;

- ***Repositories*:** Repositories are on-line facilities that provide certificates and certificate status information. Repositories in the Federal PKI will provide information via the LDAP protocol and they may also provide information in other ways. The FPMA will maintain an open LDAP repository for CA certificates and revocations. Repositories that contain end-entity certificates and CRLs for end-entity certificates, or other certificate status responders, are a policy matter for individual trust domains. Some domains may choose to make end entity certificates available in open repositories, and other domains may restrict access to end-entity certificates. Similarly some domains may implement CRL based certificate revocation while others may choose to implement OCSP responders. Some domains may elect to make certificates and certificate status information available only to relying parties that have entered into an agreement with the CA or domain management authority;
- ***BCA Repository***: The BCA repository will be open to Internet access by anyone, and will make available:
  - All certificates issued by the BCA;
  - All certificates held by the BCA;
  - All cross certificate pairs containing certificates held or issued by the BCA;
  - All CA certificates issued by CAs within the overall Federal PKI;
  - All cross certificate pairs between CAs in the Federal PKI;
  - A consolidated Federal ARL, that covers all CAs in the Federal PKI. This implies a requirement to include appropriate CRL Issuer and CRL Distribution Point extensions in all CA Certificates issued by CAs within the Federal PKI;
  - Other certificates and CRLs as determined by the FPMA;
- ***Certificate Status Responders (CSR)***: CSRs will use the emerging Internet Online Certificate Status protocol [OCSP 98] to give relying parties an online, real time response to the simaple question: "Has this end entity certificate been revoked or suspended?" CAs that use OCSP re-

sponders rather than (or in addition to CRLs) will specify the address of the OCSP responder in those certificates, and issue a certificate to the responder. CSRs will only be used for end-entity certificates, which will be the vast number of certificates in the Federal PKI and which will be changed and revoked much more often than CA certificates.

5.2 Operational concept

The *Federal Bridge CA (BCA)* will be the unifying element to link otherwise unconnected agency CAs into a systematic overall Federal PKI. The BCA is not a root CA, in that it does not start certification paths, it simply connects *trust domains* through cross certificate pairs to a designated "principal CA" in that trust domain. It is a "bridge of trust." The BCA will be operated by a *Federal Policy Management Authority (FPMA)*, which would establish the requirements for cross certifying with the BCA. These trust domains may be within the government or outside the government.

Those Federal (or non-Federal) CAs that operate in trust domains which meet the requirements established by the FPMA will be eligible to cross certify with the BCA. This will then connect them to the overall trust network of the Federal PKI, and provide relying parties and certificate holders in their trust domains with connectivity to the larger Federal PKI. this will be simpler and more effective than trying to manage an ad hoc collection of possibly many cross certifications with CAs in other trust domains.

As a further aid to efficient and effective trust propagation (i.e., certification path creation and validation), the BCA will maintain a repository that focuses on CA certificates, and the BCA will issue an *Authority Revocation List (ARL)*, an indirect CRL that covers all the CAs in the Federal PKI (the Federal PKI being defined simply as all the CAs within the Federal trust domains that cross certify with the BCA). The ARL will also cover the certificates of the certificate status responders associated with any Federal CAs. The total number of certificates issued by the BCA will be modest, since the total number of CA certificates not large, and the repository database will not be large.

The intention of this CONOPS is to embrace as many PKI approaches as can be made to work together (hierarchical, mesh, and large scale web oriented commercial CA oriented approaches), as well as can be arrangeed. Given the limitations of current PKI clients, the existence of certification paths does not guarantee trust interoperability, but it is a necessary precondition.

5.2.1 Certificate Status

An important part of certification path processing is confirming that certificates have not been revoked or suspended. Certificates may be revoked for a number of reasons, including changes in the name of individuals, reorganizations that change organizational names, because the subject has left the organization or changed his job, because any attributes bound to the subject in the certificate has changed, or because of a known or suspected key compromise. Two standardized mechanisms are available for determining current status, CRLs and OCSP responders.

The CRL is a signed list of certificates that have been revoked or suspended. A revoked certificate is removed from a CRL after the expiration date of the certificate. The CRL is normally signed by the CA that issued the certificates, however there is a mechanism that includes a pointer in a certificate to an applicable "indirect CRL" issued to another CA. The indirect CRL is illustrated in Figure 3.

CRLs have several useful properties. Since they are validated by the signature on the CRL, it does not matter where CRLs are obtained, and they can be made available at many locations and cached. They contain an issuance date and an expected date when the next CRL will be issued. These can be used as an aid to managing a cache of CRLs, and it is reasonable to adopt a CRL cache management scheme that continues to use a cached CRL until the date of the next CRL. Provided that revocation rates are not very high, CRLs make an efficient, scaleable mechanism. Moreover, they can be partitioned and certificate lifetimes managed to keep CRL sizes reasonable even in the face of large CAs with significant revocation rates.

But there are several limitations to CRLs:

- CRLs are periodic, and therefore may not have the freshest possible information about certificate status. But if the CRL period is very short, and therefore the information always very fresh, then caching doesn't work well, and the CRLs will be inefficient: a relying party will have to download a new CRL nearly every time, and will get a bunch of information he won't ever need;
- If revocations are frequent, then CRLs get big. This can be ameliorated by adjusting certificate validity periods (so that expired certificates are removed from CRLs) and partitioning CRLs into pieces with distribution points (see Figure 3);
- Since it doesn't matter where a CRL is obtained, and since they are designed to be cached, mirrored and shadowed, it will be very hard to charge relying parties for access to CRLs, limiting this possible revenue source to fund PKI operation;
- Similarly, it will be difficult to use the CRL as a mechanism to bind relying parties to CAs for the purpose of contractual privity;
- CRLs (particularly indirect CRLs) are complex structures that add a good deal of complexity to clients for relying parties, including the ability to access the CRLs from repositories.

In consequence a proposal for an Online Certificate Status Protocol (OCSP) has been introduced to the IETF, originally as a fairly simple HTML oriented proposal, which has recently converted to an ASN.1 encoding of requests and responses. With OCSP a relying party will send a (possibly signed) request identifying a certificate to an on-line certificate status responder. The responder will reply with a signed message stating something about the status of the certificate.

This on-line approach has several advantages:

- the information can potentially be very fresh, that is each response can potentially be generated from the most current information a CA has for each certificate;
- for any one certificate check, much less response information is required than would be with CRLs;
- the subject certificate and the signed OCSP response provide a relatively compact (compared to a certificate and a CRL) set of information to store with a signed document to facilitate later revalidation of the signature;
- the code needed in clients to implement OCSP is probably less than the code needed to process CRLs;
- it is comparatively straightforward to bill relying parties on a per status request basis;
- it provides an opportunity to enter into a "click-wrap" contract of some sort between the CA, (or at least the responder as the agent of the CA), and the relying party, which could, for example, define the CA's potential liability.

Despite these advantages there are also some issues and uncertainties with the OCSP approach. These include:

- The responder is a trusted on-line server with high availability requirements, which is replacing a repository that didn't need to be trusted. In addition, the repository is still needed (to find certificates in the first place);
- The responder is not easily mirrored or shadowed and it potentially becomes a centralized performance bottleneck or point of failure, in a system that hitherto had none, moreover:
  - ◊ on-line transaction systems will have sharp load peaks that are much greater then average loads;
  - ◊ it is a bottleneck that must generate a high peak volumes of signatures, and, possibly (if requests must be validated) validate a high volume of signatures as well;
- some of the fixes for the performance issues (precomputing responses) also reduce freshness;

The actual model, the trust relationships supported or assumed and the actual question answered by the OCSP responder are not yet entirely clear.

Several considerations apply to the use of CRLs in the Federal PKI:

- CRLs are periodically "pulled" from repositories by verifiers. This results in some potential delay in notifications of suspected key compromises. "Push" models have been used for distributing a Compromised Key List (CKL), but in the broad context of a large Federal PKI a CKL might become unworkably large for push type distribution, and it is unclear to whom should it be pushed. A possible solution is an indirect compromised key CRL, that consolidated key compromise information and is frequently updated.
- For a large Federal PKI ,the communications needed to access CRLs could become quite large and the cost of distributing CRLs is also potentially large [MITRE 94]. Several approaches are available to minimize this:
  - ◊ delta CRLs (which include only new revocations since some base CRL) and CRL distribution points (which include only revocations for a particular arc of the name-space) may be used to minimize ;
  - ◊ certificate validity periods can be adjusted to keep CRLs short. When revoked certificates expire they are removed from the CRL;
  - ◊ relatively coarse names will minimize revocations due to name changes;
  - ◊ applications where certificate "churn" is high are good candidates for OCSP status validation rather than CRLs.

In the Federal PKI CAs may use OCSP responders, or CRLs to make end entity certificate status available. Since end entity certificates may exhibit significant volatility, and since the total number of end entity certificates will be large, use of OCSP certificate status validation may both be more efficient and provide more timely revocation information than would be possible with the CRL status mechanism. It may also facilitate cost recovery mechanisms where relying parties (e.g. government web servers) are charged for each status validation. However CRL publication of end-entity revocations places a considerably lesser burden on the PKI, since it avoids the expense of a trusted real-time responder, and can be both effective and efficient.

Although CA certificates should be validated frequently during certification path processing, the database is not large and does not change rapidly. CA certificates and ARLs are ideal data structures for local cache memory: get them once and use them many times. Therefore the Federal PKI will rely on the ARL as a "one stop" mechanism for validating the current status of all CAs in the Federal PKI. The BCA will issue an ARL, an indirect CRL that covers all CA and OSCP responder certificates in the Federal PKI. CA certificates will not be nearly as volatile as end-entity certificates (although their validity periods will be larger), so the ARL should remain fairly small. The BCA repository will then be a key resource for creating and validating certification paths, and will have high availability requirements, medium bandwidth requirements, and low storage requirements. The CA certificates and the ARL will be available on the BCA repository, and may readily be shadowed or replicated in other repositories.

The Federal PKI will normally follow the "pull" distribution model for CRLs, that is verifiers will request CRLs from repositories on an as needed basis. Since local hard disk storage is relatively inexpensive, and since CRLs need not be stored in trusted memory, a CRL, once retrieved by a client verifier, will normally be retained in a local cache memory until the date of the next update (which is stated in the CRL), to reduce communications costs. The period for CRLs will be a compromise, balancing the delay in distributing revocation notices to verifiers against the communications costs of frequent updates.

6. Cryptographic Algorithms and Interoperability

[Editor's Note: In section for this revision, all I have done so far is to simply replace the term "inconsistent certificate" with "hybrid certificate." I believe that we have more work to do to fit this with the Bridge CA concept: I can see two approaches:

1. *There is only 1 BCA that always signs with a single algorithm, so when principal CAs that use other algorithms cross-certify with the bridge they do so through hybrid certificates. This would mean that even though 2 CAs in different domains use the same algorithm, the cert. path between then would involve 2 hybrid certificates. No additional certificates would be introduced into such mixed metaphor cert. paths.*

2. *There are 3 (or I suppose n) BCAs, one for each algorithm. The 3 are cross certified with hybrid certificates, and these are the only hybrid certificates in the FPKI.. A principal CA cross certifies only with the BCA that uses the same algorithm. Now we effectively have 3 parallel consistent algorithm federal PKIs, and as long as the signatory and the relying party's CA are consistent, no hybrid certificates are ever in the certification path. If the signatory and the relying party have CAs with different algorithms, then the hybrid certificates between the BCAs will complete that path, but add one additional certificate in the path, as compared to alternative 1 above. There probably would also have to be 3 ALRs, one signed by each algorithm, although each ARL could cover the entire Federal CA space.*

Note that a client that would work "fully" with one approach would work fully with the other. But a client that could not validate multiple algorithms, would work better with case 2]

Only the FIPS 186-1 Digital Signature Algorithm (DSA) is now approved for Federal use. However, NIST expects to extend Federal Information Processing Standard (FIPS) 186, the Digital Signature Standard [FIPS 186], to specify additional public-key based digital signature algorithms. The Rivest-Shamir-Adelman (RSA) algorithm [X9.31] and the Elliptic Curve Digital Signature Algorithm (ECDSA) [X9.62] are strong candidates for inclusion in an expanded FIPS 186. This would allow Federal users to choose the present Digital Signature Algorithm, the RSA algorithm or the ECDSA algorithm.

6.1 Definitions

The X.509 standard specifically allows a key for one algorithm to be certified in a certificate signed by another algorithm. For the discussion in this section the following terms are defined:

- *consistent certificate*: a certificate is considered to be consistent when the same algorithm is used for the public key certified in the certificate and to sign the certificate;

- *hybrid certificate*: A certificate where the subject's algorithm for the certified key is different than the algorithm used by the issuing CA to sign the certificate.

- *self-signed certificate*: A certificate signed with the key it certifies. It is used by a CA to state (but not authenticate) its public key, and any associated parameters. Self-signed certificates are transmitted to relying parties by some out-or-band, authenticated mechanism;

6.2 Parameter Inheritance

Some algorithms (DSA and ECDSA) require that parameters be specified. Parameters are public constants, needed for the algorithms, that can be common to all the certificates issued by a CA, to a group of certificates, or to a large domain of the PKI. The algorithm identifier field can (optionally) state the parameters used. The only secure place to find either the public key used to sign a message (including a certificate) and the associated parameters, is in the subject public key field of the signer's certificate [CHOK]. Parameters are often large numbers. In the case of the DSA, two of the parameters, p and g, are the same size as the public key, between 512 and 1024 bits. In this case, it is desirable to omit parameters, wherever it is possible to reduce the size of the certificates. Therefore, a set of "parameter inheritance" rules that allows parameters to be inherited. Those rules have been incorporated in some draft standards [WD 15782], and are summarized as follows:

- parameters should be obtained from the same authenticated source as the public key, the subject public key field of the signer's certificate;

- if the parameters in the subject key field of the signer's certificate are null (for those algorithms requiring parameters), then the parameters are "inherited" from the preceding step in the certification path;

- parameter inheritance does not apply to inconsistent certificates, that is a inconsistent certificate must contain the parameters in the subject public key field, if parameters are used for the subject algorithm.

Parameter inheritance is illustrated in Figure 8. In this case CA2 inherits its parameters from the certificate of CA1, but Bob has different parameters which must be stated explicitly in his certificate.

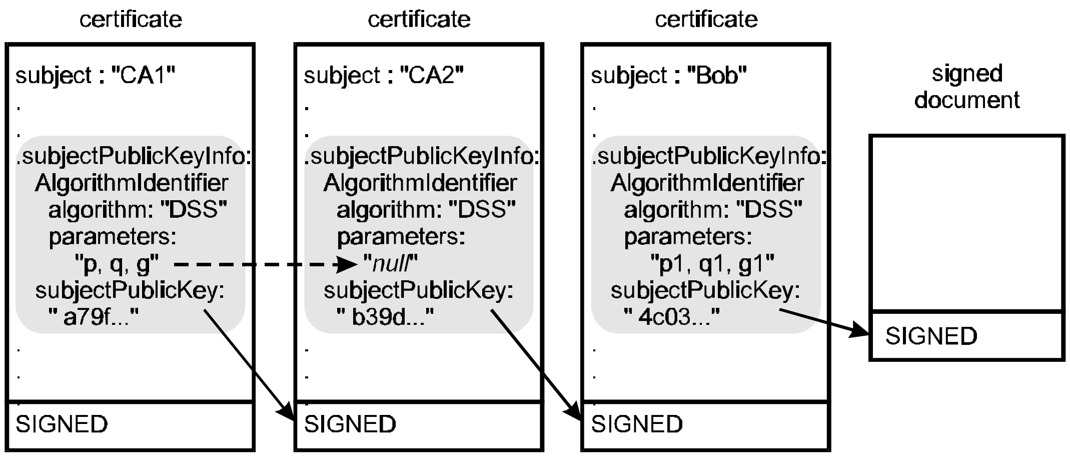

Figure 8 - Parameter Inheritance in Certification Path

6.3 Interoperability rules

It is relatively simple to implement signature validation for several algorithms, but much more burdensome to sign with different algorithms, since this implies more keys and certificates for users to manage. Moreover, the secrecy of private keys must be strictly maintained. So most end-entities will prefer to use as few signature keys as possible and sign with a single algorithm. It seems evident that end-entities with certificates that use the same algorithm, should not ordinarily have to use any other algorithm to validate each other's certificates.

If there is to be interoperability in a Federal PKI where different digital signature algorithms are used, then it is necessary, at a minimum, to use clients that are capable of validating all the digital signature algorithms that are used. Given that clients can validate the different signature algorithms, there is no need for end-entities to hold signature keys for more than one algorithm, unless it is for the purpose of interoperating with other clients that can validate only one particular algorithm.

The following rules will simplify interoperability in a Federal PKI that uses several digital signature algorithms:

- End-entity certificates will be consistent;
- A CA will sign certificates with only one algorithm;
- Where an organization needs to issue certificates with different algorithms to its certificate holders, it will use different CAs, with separate names, to issue those certificates. This will allow Certificate Revocation Lists to be confined to certificates with a single algorithm;
- Where one management entity operates several CAs for different algorithms, it will cross-certify those CAs with hybrid certificates, as appropriate;
- Independent CA's will cross certify each other with hybrid certificates, as required to meet the needs of their certificate holders.
- All self-signed certificates for algorithms that use parameters will include the parameters in the subject public key field;
- Other Certificates will only include parameters if, and only if:

- ◊ the certificate is a hybrid certificate, or;
- ◊ the parameters are different from the parameters of the issuing CA.

- Federal users will be encouraged to use client systems that can validate all Federally approved digital signature algorithms.

A Federal PKI which follows these rules will have certain desirable properties. Two end-entities certified by the same CA who use the same signature algorithm will not require the use of additional algorithms to validate certification paths. In most cases, two end entities certified by different CAs who use the same signature algorithm will not require the use of additional algorithms to validate certification paths.[1] Each certificate will be consistent with the CRL required to determine its status. In most cases, inconsistent certificates, if they occur in a certification path will occur only once, and the number of hybrid certificates, which may require that parameters be included in the certificates, will be minimized.

7. FPKI Management

The management of the Federal PKI will be through Policy Management Authorities, which set the polices for the CAs within individual trust domains.

7.1 Policy Management Authorities

Policy management authorities have purview over the operation of Federal CAs within a trust domain. Every CA operates under the control of a PMA. Management authorities approve CA certification practice statements and certificate policies. Efforts are under way in the American Bar Association and elsewhere to develop criteria for accrediting CAs, and it is expected that, as an accepted framework for CA accreditation evolves, the CAs in the Federal PKI will become accredited.

7.1.1 The Federal Policy Management Authority (FPMA)

The Federal Policy Management Authority (FPMA) is charged with the supervision of the bridge CA, establishing the policies used in BCA certificates, developing and promoting model policies for general use, and approving cross certification between the BCA and the principal CAs in other trust domains. When a PMA wishes to cross certify it's PCA with the Federal BCA, the FPMA will examine the certification practice statements and certificate policies used in that trust domain and make a decision about cross certification. In particular, the FPMA will decide which certificate policies and policy mappings to include in the certificate issued by the BCA to the PCA.

The FPMA will develop and maintain model certificate policies with different assurance levels, for use within the Federal PKI. The concept of assurance levels is described in 7.2.1 below.

[1] There is a special case where certification paths involving CA1 and CA3 "go through" CA2, and CA2 does not support the algorithm used by CA1 and CA3. In this case, validation of the certification path will require use of two algorithms.

7.1.2 Domain Policy Management Authorities

The CAs in each trust domain will operate under the supervision of a domain PMA. The domain PMA will approve the certification practice statements of the CAs within the domain, and the certificate policies used to issue certificates.

7.2 Certification Practice Statements and Certificate Policies

A Certification Practice Statement (CPS) [ABA 96] is a statement of the practices which a CA employs in issuing certificates. A CPS describes the details of the system used and the practices employed by a CA to issue certificates. A CPS details the procedures used to implement the policies identified in the certificates issued by a CA, including the means used to identify certificate subjects. It also states the means used to protect the public key of the CA, and the other operational practices followed by the CA to ensure security. The CPSs for all Federal CAs will be posted in the BCA Repository, and should also be posted in any repositories associated with the CA. There is a standardized outline and list of factors to be considered in writing a CPS or certificate policy [PKIX4 98]. The outline addresses the following major subject areas:

1. *Introduction:* Identifies who operates and manages the CA (or the certificate policy), the users it serves and how to contact it.

2. *General Provisions:* covers liability, fiscal responsibility, governing law, fees and similar considerations

3. *Identification and Authentication*: deals with how names are assigned and identities proofed.

4. **Operational Requirements**: describes the process for issuing and revoking certificates, the records that are kept the audits that are performed, and CA compromise, disaster recovery and termination provisions;

5. *Physical, Procedural and Personnel Security Controls*: describes the physical facilities, the trusted roles in operating the CA and issuing certificates, and the personnel controls on CA/RA personnel;

6. *Technical Security Controls*: covers cryptographic issues, key pair generation, the algorithms used, how private keys are activated, protected, and managed, and the technical security considerations for CA, RA and end entity systems;

7. *Certificate and CRRL Profile*: states the rules for the use of certificate and CRL extensions;

8. *Specification Administration*: states how the CP or CPS is administered and maintained.

X.509 defines a certificate policy as "a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements"[X.509 97]. An X.509 v3 certificate may state one or more certificate policies in the **certificatePolicy** extension, which may be used by a relying party to decide whether or not to trust a certificate for a particular purpose.

A certificate policy, which needs to be recognized by both the issuer and relying party of a certificate, is represented in a certificate by a **certificatePolicy** extension that contains a unique, registered Object Identifier. The registration process follows the procedures specified in ISO/IEC and ITU standards. The party that registers the Object Identifier also publishes a textual specification of the certificate policy, for examination by certificate users and other parties.

7.2.1 Assurance Levels

The X.509 standard does not associate certificate policies with particular assurance levels, however a Government of Canada (GOC) proposal [GOC 97] proposes to define four certificate policies ordered by their assurance level, for use by the Canadian government. This proposal has been circulated to bodies working on certificate polices as a strawman for standardized policies In the Federal PKI each certificate will contain at least one certificate policy . Four ordered levels of assurance are proposed by the GOC:

- *Rudimentary Assurance*
- *Basic Assurance*
- *Medium-level Assurance*
- *High Assurance*

In order to promote interoperability, and the appropriate use of certificate policies, the Federal PKI will adopt the principle of four levels of assurance, and cooperate to establish consistent standards for these levels so that trustworthy cross-certification with non-Federal CAs can be achieved. In the end, a broad system of accreditation is envisioned for CAs, probably thorough some industry association, to enable a national trust framework.

Certificates issued within the Federal PKI will contain at least one of the standard assurance level policy Ids in the certificate policies extension. Certificates issued by federal CAs may also contain other, application specific, policy identifiers.

7.2.2 Certificate Policy Field contents

The **certificatePolicies** extension in Federal PKI certificates will be used to identify the policies that apply to a certificate. For signature certificates issued to CAs the **certificatePolicies** field will contain the identifier of an assurance level policy, that states the highest level of trust that can be supported through this certification path, as determined by the issuing CA and its policy management authority. The **certificatePolicies** extension may also contain other specific policy identifiers that also apply to the certificate. In general, with the assurance level concept, a certificate issued under a high assurance policy would also contain the policy identifiers for the low, medium and rudimentary assurance policies, because the high assurance certificate should adequately meet the requirements of lower assurance policies.

7.2.3 Use of assurance level policies by relying parties

The X.509 standard specifies that the inputs to the certification path processing include a set of initial-policy identifiers, which identifies one or more certificate policies, any one of which would be acceptable for the certification path processing. At each stage of the certification path processing the verifier requires that there be at least one member of the acceptable policy set appear in the **certificatePolicies** field.

Assume, for the sake of illustration, that the assurance level policies *rudimentary, low, medium* and *high,* as proposed by the Government of Canada, have been generally accepted. A relying party who wished to require an assurance level of at least *medium*, would include both *medium* and *high,* but not *rudimentary* or *low,* in the set of initial-policy identifiers.

8. Transition to an Integrated Federal FPKI

Many Federal CAs will initially be created as stand alone certificate management systems that support a particular application. It will be desirable to use the existing CA for other purposes that require trust paths to other CAs and their certificate holders. It will not be cost effective for agencies to operate dozens or hundreds or CAs, each dedicated to a particular application. Moreover such isolated CAs can not create a framework for broad secure interoperation between agencies, or with state and local governments, business and the public.

When a PMA that supervises a trust domain of one or more CAs wishes to join the overall Federal PKI, it submits the certificate policies used in the trust domain (or the policies which it wishes to be accepted as equivelent to one of the Federal assurance levels) and the certification practice statements of its CAs to the FPMA. The FPMA reviews the material, and decides if the applicant trust domain qualifies to join the FPKI. It determines the assurance level certificate policies, and policy mappings that may be included in the certificate to be issued by the Federal BCA to the applicant PCA. The Federal BCA then cross certifies with the PCA.

However there are many possible obstacles to incorporating an existing, isolated CA and its certificate holders into the FPKI. If the isolated CA does not use FIPS approved algorithms, then it cannot be incorporated into the FPKI. It may be necessary for the formerly isolated CA to issue new certificates to its certificate holders that contain the extensions required in FPKI certificates (in particular, a **certificatePolicy** extension containing a certificate policy approved by the PMA). Certificates do not use proper distinguished names as the subject name will not fit well into the FPKI, and it may be necessary to reissue certificates with distinguished names before joining the FPKI. If CAs issue certificates with critical extensions that are not supported by most federal clients, then this may create a barrier to effective participation in the Federal PKI. A CA that expects to participate in the FPKI should support or use a repository that provides LDAP access. It is also expected that the FPMA will require that Federal CAs use cryptographic modules that have been validated under FIPS 140-1, and there may also require other administrative CA operational policy requirements, such as split knowledge dual control activation of the CA private key.

The PMA will develope model assurance level certificate policies. The incorporation of new trust domains into the FPKI will be facilitated if those domains adopt the model certificate policies of the PMA, since this will, at least simplify the evaluation and mapping of certificate policies.

Perhaps more fundamental and difficult to deal with, however, may be inconsistencies in clients. Clients that cannot process certain extensions identified in the Certificate Profile [PROF 97] will not be interoperable, since these extensions will be labeled critical in the FPKI. FPKI clients should implement the rules for parameter inheritance discussed briefly in section 3.2 above. Clients must adequately protect private keys, in accordance with rules to be established for each Federal Assurance Level, and clients should use FIPS 140-1 cryptographic modules.

9. Attribute Authorities

Attribute authorities issue attribute certificates. While attribute authorities may be operated in conjunction with a CA, they may be entirely separate. In general, attribute authorities are associated with privileges, roles and access controls. An attribute authority is issued a signature certificate by a CA. The role of attribute authorities and attribute certificates in the FPKI is a subject for further study.

10. Compromise Recovery

Compromise recovery is a significant concern in the PKI. Provisions are needed to recover from compromises of private keys. The detailed policies for recovery from key compromise are stated in [POL 95]. This section summarizes the principles of compromise recovery.

10.1 Compromise of User Keys

Recovery from compromised user keys is the responsibility of the CA that issued the certificate that has been compromised. The compromised certificate is added to the next CRL and to the consolidated Federal CRL for compromised keys. The client may be issued a new certificate by the authority if appropriate.

A user may sign a message to the CA that issued his certificate, stating that the private key has been compromised and requesting that the certificate be revoked, using the compromised private key. RAs may also request CAs to revoke certificates issued through their registration process, if a key compromise is detected.

10.2 Compromise of CA Keys

Compromise of the private key of a CA invalidates all the certificates issued by that authority, since it results in the possibility of certificates forged with the compromised private key. In general, recovery from the compromise of a CA private key requires:

- the generation of a generation of a new CA public-private key pair;
- the revocation of all CA certificates for the compromised key;
- the issuance of new CA certificates for the new key from those CAs that have issued certificates to the compromised CA;
- the issuance of new certificates, signed with the new CA private key, to replace the certificates issued with the compromised key;
- secure distribution of the new public key to all relying parties who use this key to start certification paths. Note that this must be a "out-of-band" process, since the old, compromised key cannot be used to validate the new key.

Since compromise of the CA private key does not itself compromise the private keys of its subordinates (although it may allow the forgery of new certificates), it is not, in general necessary to replace the subordinate public-private key pairs, and the new certificates may retain the old user public key. However, it is necessary to provide all subjects of certificates signed with the compromised key with new certificates and the public key of the CA, by a secure process.

10.2.1 Root CA Key Compromise

Whenever a CA's public key is used to start certification paths, a compromised public key requires that all relying parties which begin their certification paths with that key, must be given the new public key by some authenticated, but out-of-band process. Therefore, when a root CA private key in a hierarchical domain is compromised it is necessary to:

- revoke all CA certificates for the compromised key;
- generate a new key;
- reissue all the certificates issued by the CA under the compromised key;
- obtain new CA certificates to replace the revoked CA certificates;
- notify all relying parties within the hierarchical domain, including certificate holders of certificates from subordinate CAs whose keys have not been compromised, of the compromise and distribute the new public key in an authenticated manner..

10.2.2 Subordinate CA Key Compromise

Since the keys of subordinate CAs in hierarchical domains are not used to start certification paths, recovery from key compromise is comparatively straightforward. To recover it is necessary to:

- revoke all CA certificates for the compromised key;
- generate a new key;
- reissue all the certificates issued by the CA under the compromised key;
- obtain new CA certificates to replace the revoked CA certificates.

10.2.3 Peer CA Key Compromise

In a mesh domain, relying parties will generally start certification paths with the public key of the CA that issued them their certificate. If the key of a peer CA in a mesh domain is compromised, then it is necessary to:

- revoke all CA certificates for the compromised key;
- generate a new key;
- reissue all the certificates issued by the CA under the compromised key;
- obtain new CA certificates to replace the revoked CA certificates;
- notify all certificatholders of the key compromise and securely distribute the new CA public key to all certificateholders.

10.2.4 Bridge CA Key Compromise

Although the BCA key is particularly important to the overall operation of the FPKI, in some ways the compromise of its key is easier to recover from than other CAs, since the BCA key is not used to start certification paths, and does not issue end-entity certificates. Therefore there is no need for authenticated out-of-band distribution the new key to a large community of relying parties. However, since the BCA will hold a certificate from every PCA in the FPKI, there will be a comparatively large number of CA certificates issued by the various PCAs to the BCA that must be revoked and reissued To recover from BCA key compromise the BCA must:

- notify all PCAs of the compromise and they must revoke all certificates they have issued for the compromised key;
- generate a new key;
- reissue all the PCA certificates issued by the BCA under the compromised key;
- obtain new CA certificates from the PCAs for the new BCA key.

10.3 RA Compromise

If an RA key is compromised, it is possible that user certificates may have been issued to fictitious users, or with incorrect attributes. Recovery requires revocation of all certificates issued through the compromised RA using the compromised key. If the compromise date is known, then it is only necessary to revoke certificates issued after the compromise occurred. If the RA has retained complete, uncompromised records of the users who applied for certificates through the RA, then new certificates may be automatically issued to users whose certificates have been revoked. Otherwise, users whose certificates are revoked, must reapply to the RA to be issued a new certificate.

11. Implementation Technologies

11.1 Clients

Availability of PKI client functionality to every Federal information system and every Federal user is needed. Some users require the assurance that is best provided by personal hardware cryptographic tokens. Other users and applications cannot justify the additional expense of cryptographic hardware and tokens for every client (i.e., every PC, every data terminal, every workstation). Entirely software based solutions are appropriate for many clients and their applications. Hardware and software based clients must be capable of secure interoperation.

11.1.1 Software Implementation

To provide government users with the availability of an affordable suite of security services (i.e., signature based authentication, integrity, and confidentiality key exchange, as well as confidentiality for communications sessions or messaging), the PKI, in combination with clients that use its certificates, will:

- provide the following security services:
  - ◊ public key digital signatures for:
    - – authentication;
    - – integrity.
  - ◊ certificate or public key based symmetric key exchange/agreement for:
    - – messaging confidentiality;
    - – communications session confidentiality;
  - ◊ provide confidentiality using symmetric key encryption.
- allow clients to be implemented entirely in software;
- use publicly disclosed algorithms that need not be protected;

- be able to operate on single user computer systems that have not been validated to conform to any trust criteria.

Technical policies for implementing PKI client software cryptographic modules and storing private keys are stated in ???. The implementation of client cryptographic modules affects the level of assurance accorded to certificates and signatures. Accordingly some high assurance policies for issuing certificates will require hardware cryptographic tokens.

11.1.2 Hardware Implementation

Alternatively, user keying material may be retained in trusted hardware cryptographic modules and signatures or session keys generated in that trusted module. Such modules will not export user signature private keys. Access to the cryptographic functionality of the module may be protected by passwords, or possibly by other means such as biometrics.

The PKI may also support algorithms that are not publicly disclosed and for which software implementations are not authorized. Such algorithms will be implemented in trusted cryptographic modules.

Some high assurance policies for issuing certificates will require use of hardware cryptographic tokens.

11.2 Authorities

The systems used by CAs to sign certificates will require a high level of assurance. Technical policies governing the assurance required in the implementation of CAs will be stated in ????.

12. Interoperation

The PKI will support secure communications with business, other branches of the Federal Government, the public, state and local governments, as well as between Federal departments and agencies. Interoperation may also be required with Federal users of Type 1 cryptography (cryptography used to protect classified National Security information). The Federal PKI will therefore support and interoperate with a broad range of technologies, as appropriate, including commonly used technologies that are not approved for use to protect UBS communications between Federal users. [INT 95] contains a more detailed treatment of interoperation issues.

12.1 Interoperation with the Type 1 Infrastructure

The primary burden for interoperation between Type 1 users and the Federal PKI rests with the Type 1 users:

- Type 1 Certificate Authorities may also issue PKI certificates;
- Type 1 tokens and modules may also support PKI certificates and algorithms;
- the impact of the PKI design on the ability of the existing Type 1 infrastructure to support the PKI will be considered in the design of the PKI.

12.2 Interoperation with non-Federal Users

12.2.1 Algorithms

It is expected that the digital signature algorithms that are in wide commercial use will also be approved for use by Federal agencies. The basic principle for interoperation between users of aifferent algorithms is called the "end system" approach. It is explained in more detail in section 5 above. In this approach, the Federal PKI issues only certificates that use FIPS approved algorithms.. The basic principles of interoperation with non-Federal users are:

- The Federal PKI supports only signature certificates that use FIPS approved algorithms. Federal users who require signature capability using a non-approved algorithm must waive the FIPS;
- Non-Federal users are expected to be able to validate signatures using FIPS approved algorithms.
- Federal PKI users may validate signatures of non-Federal parties, in accordance with their security policy. If the algorithms used are not FIPS approved, a FIPS waiver may be required;
- In general, it is the algorithm used in the key management certificate of the destination of an encrypted message that determines the key management algorithm used to establish the key the message encryption key.

12.2.2 Cross-Certification

The Federal PMA will approve BCAcross-certification with non-Federal CAs. The optional policy mapping, path length constraint, and the subtrees constraint extension fields of the X.509 v3 certificates may be used by CAs to constrain the use of cross certification links with non-federal CAs and infrastructures.

12.2.3 Certificates Issued to non-Federal Users

Federal CAs may issue end-entity certificates to non-Federal users in accordance with their CPS. Name space constraints in the BCA certificates may, however, limit propagation of trust to certificates issued to subjects with non-Federal names.

13. Records and Archives

While the validation of digital signatures, and the certification paths that support them is comparatively straightforward at the time the signatures are executed, later validation becomes increasingly problematic. Validation of digital signatures can be broken down into three distinct problems, or regimes, depending upon the time that has elapsed since the signatures were executed:

1. *near term*: validation while all the certificates and CRLs required to validate the signatures are current and generally available;
2. *intermediate term*: validation after expiration of one or more of the certificates required to validate the signature, but while the cryptography used in the signatures is still secure;
3. *archival:* validation indefinitely, and in particular after the time when the cryptography used is no longer secure, that is validation after the time when it would be practical to derive private keys from public keys, or to generate hash collisions.

13.1 Near term

It is the responsibility of CAs to make available in repositories all current certificates and the current CRL. This will provide the information needed to validate any current signature, before any of the certificates in its certification path have been revoked or expired.

13.2 Intermediate term

The intermediate term begins when one or more of the certificates needed to validate a signature expires, and lasts until the cryptography used for the signatures is no longer secure. Since the cryptography is still considered secure, if the signatures are valid and the certification paths of the signatories were valid at the time the signature was executed, then a strong validation of the signature is still possible if the needed certificates and CRLs can be obtained.

Digital signatures, however, do not reliably date a signed document. Therefore it is easy for signatories to predate and sign a document, just as it is with handwritten signatures. If the actual date of a digitally document is at issue, then this must be established by some other means. One such means would be a dated signature by a trusted third party, (a trusted time stamp). The trusted time stamp may be applied to a hash of the signed document; it is not necessary for the time stamp service to see the document itself, nor, in principle, is it necessary for the service to record the time stamp transaction. Other approaches to the proof of the age of documents would be to publish the hash of the document in some publication of record (e.g., the Commerce Business Daily, the New York Times, etc.). or to enter a copy of the document in some archive that would take custody of the document, and be able to attest to the time at which the document was entered into the archive.

However, in the intermediate term, a straightforward automatic validation of the certification paths will not occur, since one or more of the needed certificates has expired. Human intervention may be needed to determine the date of the signature. The support to be provided by certification path processing software for retrospective validations is unclear, but it may be necessary, for example, for the human user to override the path processor to tell it to accept expired certificates. Ideally, certification path processing logic would include a way to set a "signature date" parameter, and evaluate the validity of signatures and the certification path supplied as of that date, to support retrospective signature validations.

A trusted digital notary might be used to simplify the process of later validating signatures. Such a notary would examine a signed document (it is not necessary for the notary to see the document text itself, but the notary must see the document hash and all the signatures) and validate all the signatures, then "notarize" the document by signing the document hash and signatures, plus a time stamp and a notice attesting that the signatures had been validated. This might simplify the records that need be retained to allow later validation (particularly documents with multiple signatures) if the notary undertook to maintain and make available a trustworthy long term record of its own certificates.

13.3 Archived Signatures

Archival storage is indefinite, and may extend long past the time where the cryptography used to sign documents is still secure. All the issues about validation that apply to intermediate term signature validation apply to the archival case. Moreover, future advances in computing are likely to make it practical to generate private keys from public keys or hash collisions for current digital signature and hashing algorithms, where this is infeasible today. Therefore validation of archived signatures cannot rely solely on the cryptography, even when complete certification paths are available.

The approaches to archival validation require that the signed document be turned over to a trusted archive that can be trusted to keep a true copy of signed document[2] while the cryptography remains secure. There are several variations:

- The document is entered into the archive with all the necessary CRLs and certificates needed to validate the signatures. Since the archive is trusted to maintain a true copy, any forgery must have occurred before entry into the archive. But if a forgery were impractical at the time of entry into the archive, and the signatures are valid, then the signatures can be judged valid.
- The document is first notarized as described above, then entered into the archive. Records to substantiate the public key of the notary must also be archived;
- The archive itself validates the signatures and acts as a notary at the time the document is entered into the archive; In this case a simple unsigned notation that the signatures had been validated by the archive would suffice, since the archive is already a trusted custodian for the document.

Notary and suitable validated archival services for digitally signed documents yet are not available.

13.4 Responsibilites

Federal CAs will make their certificates and CRLs available through repositories while they are current. They will maintain publicly available records of the certificates issued and CRLs for a period of TBD years after their expiration. The Bridge CA will make authentic copies of the public keys they use to issue certificates available indefinitely However, CAs are not necessarily permanent institutions and the long term responsibility for preserving the records needed to validate signatures indefinitely rests with signatories or relying parties, who may retain or archive the certification paths needed to substantiate signatures, or to have signed documents timestamped or notarized as required.

14. Servers and Agents

Some security related servers and agents may be operated by the Federal PKI, or be used by the Federal PKI. The important services that have been identified are described in some detail below. The following list is not exhaustive, and a need for additional kinds of security servers and agents may be identified.

14.1 Repositories

The PKI will use repositories to make certificates and CRLs available. It is expected that repositories will normally be directories that provide access through the Lightweight Directory Access Protocol (LDAP) [RFC1777], however other World Wide Web oriented alternatives may also be supported. Repositories may be operated by a CA, or may be a part of a separate directory that serves broader purposes than just supporting the PKI. Repositories are not, in general, trusted en-

[2] While the briefly TWG discussed periodic resigning of documents as a technique for ensuring that a true copy had been kept, it came to the conclusion that the procedures that ensure that a true copy is kept could also be administrative and procedural, and that archives have long been trusted to maintain true copies of documents without any cryptographic means.

tities; that is it is the signatures on CRLs and certificates that authenticate them, not the repository from which they were obtained.

14.2 Servers and Agents

The infrastructure will be augmented by a variety of security-related servers and agents. These servers or agents will generally be trusted users with certificates issued by some appropriate authority. They will support the needs of users, or, in some cases, of the Federal PKI hierarchy itself. In this section these servers are described in terms, such as digital notary, and document recorder that approximate services now available for paper documents. The sections below describe some of the services may be provided. This aspect of the infrastructure is not well understood, and the actual digital servers that emerge may be combined in other ways, and given other names.

14.2.1 Digital Notary Servers

The digital notary server provides a service roughly equivalent to a notary public. Digital notaries operate on the message digests of digital and the digital signatures that may be applied to those documents, rather than the documents themselves. The digital notary may offer at least two services:

- to add a time-date stamp to a message digest of a document and sign the message digest plus the time-date stamp. This notarization provides proof that the document existed at a particular point in time and is the date-time stamp service of Figure 1;
- to verify the signatures applied to a signed document and generate a signed, dated notarization statement that includes the message digest of the subject document and the distinguished names of each of the signatories.

The certificates of notaries should be registered in a permanent archive. The notary keeps a log of the notarizations but not copies of the documents.

14.2.2 Digital Recorder Servers and Archives

The digital recorder server provides the service of a document recorder, that is a document archive where documents may be registered for safekeeping and to make a permanent record. The digital recorder permanently retains, under its sole control, a dated copy of digital documents registered with it. The digital recorder will, upon request, issue a signed copy of recorded documents, stating the date of registration.

Among the uses of digital registries are the key (or certificate) archiving service of Figure 1. They can provide a repository for expired certificates, to allow the verification of signatures on old documents. If digital documents are to replace paper documents, then it must be possible to verify digital signatures long after the certificates used to sign the documents have expired, and after the authorities that issued the certificates have ceased to exist. To be useful for this purpose, a digital registry must be a permanent archive, independent of particular authorities.

The verification of the authenticity of signed digital documents, long after they were signed, presents particular problems. The advantage of digital documents in archives is that they can be copied as often as needed without any alteration or degradation, which is generally impossible with analog records. This allows a perfect copy to be maintained through many generations of copies. The disadvantage of digital documents for archival purposes is that, as a practical matter digital storage media have a short useful life, and documents must be copied frequently to maintain them [Rothen 95]. Therefore, the physical forensic evidence that may help to authenticate original old paper documents is not available for digital documents. Moreover, we must assume that any presently secure digital signature technology may eventually fall to advances in computer power and cryptography.

Since any present digital signature technology may ultimately be compromised by future progress in computing power and cryptography, the validity of recorded documents does not primarily depend on any digital signature made by the recorder at the time of document registration. Rather the validity of recorded documents depends on the physical control of the registered copy of the document by the trusted recorder. If a recorder has retained an accurate dated copy of a signed digital document, if a true copy of the signer's certificate exists, and if it were impractical to forge the signature at the time the document was registered, then verification of the signature is meaningful even after it becomes possible to forge the signature.

For truly long term verification of certificates and signatures, it is not sufficient to register only the certificates of the authorities themselves, because we must assume that it will eventually be possible to derive the authority's private key for any current public key and forge a certificate. Registering all certificates issued by an authority would allow detection of certificates forged at a later date. Similarly, signed digital documents whose authenticity may be important at a much later date, must registered contemporaneously with their signing. Then, knowing from the registry:

1. the contents and signature of a document that was registered on a given date, and,
2. the complete certificate path supporting the signatures;

It is possible to verify the authenticity of the document at a later time, even after it is practical to derive the private key from the public key.

However, very long term (i.e., for centuries) digital archive services are problematic, due to the rapid rate of change of digital media, applications and file formats, and the uncertain storage lifetimes of digital media [Rothen 95]. Any presently practical digital archive must periodically copy data onto new media, for even where a storage medium has a long lifetime, data readers for the storage medium will become unavailable after newer media come into use. This means that a requirement to maintain long term archives of large numbers of public key certificates is potentially burdensome. Unless registries are funded from public tax moneys, they will need to charge a registration fee sufficient to maintain copies of the documents in perpetuity.

14.2.3 Digital Certified Delivery Servers

Proper maintenance and design of the PKI makes it difficult for users to repudiate their own signatures. While digital nonrepudiation can be accomplished by cooperating parties without a trusted server, cooperation is not always assured. Certified delivery servers provide services analogous to those of process servers and certified or registered mail. A digital nonrepudiation server provides a trusted third party that can verify that a digital document was in fact delivered to its intended recipient, or that a good faith, best effort was made to deliver the document, even without the cooperation of the intended recipient. The Federal Government may need nonrepudiation servers, but they are not needed to support the operation of the PKI itself.

14.2.4 Ticket Granting Agents

Ticket granting agents issue tickets that grant limited-time access to information resources. Although now widely used with symmetric key rather than public key authentication systems, they can also be used with public key technology to centralize management of access control. In effect, a Registration Authority (RA) that does not issue certificates itself, but vouches for the identity of prospective certificate holders to a certification authority, is a ticket granting agent.

14.2.5 Key Recovery Agents

[To be Supplied]

List of Acronyms

ARL	Authority Revocation List
BCA	Bridge CA
CA	Certification Authority
CKL	Compromises Key List
CRL	Certificate Revocation List
CSOR	Computer Security Objects Register
FPKI	Federal PKI
KMI	Key Management Infrastructure
KRA	Key Recovery Agent
LDAP	Lightweight Directory Access Protocol
MISSI	Multilevel Information Systems Security Initiative
OCSP	Online Certificate Status Protocol
OID	Object IDentifier
PCA	Principal CA
RA	Registration Authority
PMA	Policy Management Authority
PKI	Public Key Infrastructure
UBS	Unclassified But Sensitive

References

[ABA 96]	Information Security Committee Electronic Commerce and Information Technology Division Section of Science and Technology American Bar Association, *Digital Signature Guidelines,* August 1, 1996.
[CHOK]	Chokhani, Santosh, *A Security Flaw in the X.509 Standard*, in the Proceedings of the 19th National Information Systems Security Conference, Volume 2. October 22-25, 1996.
[FIPS 46]	FIPS PUB 46-2, Data Encryption Standard (DES), NIST, 1993.
[FIPS 180]	FIPS PUB 180, Secure Hash Standard, NIST, 1993.
[FIPS 140]	FIPS 140-1, Security Requirements for Cryptographic Modules, NIST, 1994.
[FIPS 185]	FIPS PUB 185, Escrowed Encryption Standard (EES), NIST, 1994.
[FIPS 186]	FIPS PUB 186, Digital Signature Standard (DSS), NIST, 1994.
[FR 97]	NIST, "Announcing Plans to Revise Federal Information Processing Standard 186, Digital Signature Standard," *Federal Register*, May 13, 1997 pp. 26293-26294.
[Garf 95]	Simon Garfinkle, PGP: Pretty Good Privacy, O'Reilly & Associates, Inc., 1995.
[GOC 97]	Working Draft Certificate Policies for the Government of Canada Public Key Infrastructre (GOK PKI), 10/14/97.
[INT 95]	TWG-95-89, Federal Public Key Infrastructure (PKI) Technical Specification: Part D -Interoperability Profiles, 09/27/95. Available at: http://csrc.ncsl.nist.gov/pki/
[McCon 96]	Bruce W. McConnell and Edward J. Appel, Draft paper, "Enabling Privacy, Commerce, Security and Public Safety on the Global Information Infrastructure," Office of Management and Budget Memorandum for Interested Parties, May 20, 1996. Available at: http://www.cdt.org/crypto/clipper_III/clipper_III_draft.html
[PKIX1 98]	Internet Draft, *Internet Public Key Infrastructure: X.509 Certificate and CRL Profile*, <draft-ietf-pkix-ipki-part1-07.txt>, R Housley, W. Ford, W. Polk and D. Solo, march 25, 1997. Available at: http://www.ietf.org/html.charters/pkix-charter.html
[PKIX4 98]	Internet Draft, *Internet Public Key Infrastructure Part IV: Certificate Policy and Certification Practices Framework*, <draft-ietf-pkix-ipki-part4-03.txt>, S. Chokhani and W. Ford, April , 1998. Available at: http://www.ietf.org/html.charters/pkix-charter.html
[OCSP 98]	Internet Public key Infrastructure Online Certificate Status Protocol - OSCP, OCSP <draft-ietf-pkix-ocsp-03/txt>, March 1998, available at:: http://www.ietf.org/html.charters/pkix-charter.html
[POL 95]	TWG-95-81, *Technical Security Policy for the Federal PKI*, 25 Aug. 1995. Available at: http://csrc.ncsl.nist.gov/pki/

[PROF 98]	TWG-98-07, *Public Key Infrastructure (PKI) Federal X.509 Certificate Profile,* March 9, 1998, available at: http://csrc.ncsl.nist.gov/pki/
[RFC 1421]	Linn, J., Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures, RFC 1421, February 1993.
[RFC 1422]	Kent, S., Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management, RFC 1422, BBN, February 1993.
[RFC 1423]	Baleson, D., Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes and Identifiers, RFC 1423, TIS, February 1993.
[RFC1777]	RFC 1777, *Lightweight Directory Access Protocol,* Ed Yeoung, Howes, and Killie. March 1995.
[RIV 78]	Ronald L. Rivest, Adi Shamir and Leonard L. Adleman, "A Method for Obtaining Public Key Signatures and Public Key Cryptosystems,*" Communications of the ACM,* 21 (1978).
[Rothen 95*]*	Rothenberg, Jeff, *"Ensuring the Longevity of Digital Documents,"* Scientific American, Vol. 72, no. 1, Jan 1995, pp. 42-47.
[X.500 93]	CCITT Recommendation X.500*, The Directory, 1993.*
[X.509 97]	CCITT Recommendation 509*, The Directory: Authentication Framework, 1997.*
[X9.45 97]	Working draft, *X9.45-199x: Enhanced Management Controls Using Digital Signatures and Attribute Certificates,* June 24, 1997.
[X9.31]	Working Draft American National Standard X9.31-199x, Public Key Cryptography for the Financial Services Industry: The Reversable Digital Signature Algorithm,
[X9.62]	Working Draft American National Standard X9.62-199x, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm, June 21, 1996